BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
Facultad de Ciencias de la Electrónica

Desarrollo de un sistema de control y guiado automático de drones civiles basado en FPSoC

Tesis que para obtener el grado de Doctor en investigación aplicada a la industria

Presenta: Juan Díaz Téllez
Director de tesis: José Fermi Guerrero Castellanos

Puebla, Pue. México

Diciembre 2022

Resumen

En este trabajo de investigación se proporcionan las bases para diseñar, desarrollar e implementar algoritmos robustos basados en la metodología anti-perturbaciones. Una de las principales contribuciones de este trabajo es el diseño de un control de rechazo activo de perturbaciones mejorado que integra en el ADRC clásico una función sobre la dinámica del error y un coeficiente de suficiencia de control, que ayuda a la convergencia del error de estimación y su posible saturación. La nueva metodología estimula futuras interacciones y la fertilización cruzada de ideas entre control automático y disciplinas de aprendizaje por refuerzo. Se realizan pruebas de seguimiento de trayectorias, la aeronave se encuentra sujeta a múltiples perturbaciones como efecto suelo, ráfagas de viento, fuerzas y pares generados por una carga suspendida acoplada al multirotor. Otra contribución significativa es el desarrollo de una arena de vuelo para sistemas heterogéneos enfocada al estudio, desarrollo e implementación de algoritmos de control, navegación autónoma y aprendizaje por refuerzo. Este manual sirve para prototipar algoritmos complejos en un enfoque de arquitectura abierta. Finalmente, la metodología de control desarrollada se utilizará para crear un piloto automático robusto con capacidades de rechazo de activos perturbadores. Se aplicará a un futuro proyecto de agricultura inteligente para rescatar el campo poblano. El proyecto se basa en la Agenda para el Desarrollo Sostenible 2030 de la ONU, el Plan Nacional de Desarrollo 2019-2024, el Plan Estatal de Desarrollo 2019-24 y la normativa vigente.

Abstract

This research provides the bases to design, develop and implement robust algorithms based on the anti-perturbation methodology. One of the main contributions of this work is the design of an improved active disturbance rejection control that integrates into the classic ADRC a function on the dynamics of the error and a control adequacy coefficient, which helps the convergence of the estimation error. And its possible saturation. The new methodology stimulates future interactions and cross-fertilisation of ideas between the machine control and reinforcement learning disciplines. Trajectory monitoring tests are carried out; the aircraft is subject to multiple disturbances such as ground effect, gusts of wind, forces and torques generated by a suspended load attached to the multirotor. Another significant contribution is the development of a flight arena for heterogeneous systems focused on the study, development and implementation of control algorithms, autonomous navigation and reinforcement learning. This manual is for prototyping complex algorithms in an open architecture approach. Finally, the developed control methodology will create a robust autopilot with disruptive asset rejection capabilities. It will be applied to a future intelligent agriculture project to rescue the countryside of Puebla. The project is based on the UN 2030 Agenda for Sustainable Development, the 2019-2024 National Development Plan, the 2019-24 State Development Plan and current regulations.

Acknowledgements

Thanks to my wife, Monserrat Alejandra Herrera Pérez and my daughter Alessandra Díaz Herrera who were in the whole process of this thesis; Thank you for your patience, your love and for motivating me to keep going.

Thanks to my mother, Maura Téllez Mercado, none of this would have been possible without your support, thank you for believing in me. Thanks to my father, Eliseo Bolívar Díaz Pérez who has been an example to follow in life. Thanks to my brother Emmanuel Nieve Téllez for always being with me.

I am deeply grateful to my thesis director, PhD José Fermi Guerrero Castellanos, I have always admired him, and I thank him for the trust placed in these four years. Thank you for all your knowledge, teachings and professionalism of him.

I thank the committee members for their valuable contribution to the thesis work; their comments and observations served to improve this work. Thank you, PhD Richard Torrealba Meléndez; thank you, PhD Gerardo Mino Aguilar; thank you, PhD Olga Guadalupe Félix Beltrán, and thank you, PhD Jacob Javier Vásquez Sanjuan. I thank BUAP for allowing me to get ahead with my preparation professional, and to CONACyT for the maintenance scholarship in the last year.

Thank GOD for this achievement.

Dedication

To my mother Maura Téllez Mercado.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Life is and will ever remain an equation incapable of solution, but it contains certain known factors.

– **Nikola Tesla**

1.1 Motivation

Today, multirotor have established themselves in the unmanned aerial vehicle (UAV) field. This dominance in the military and civil areas has expanded rapidly and displaced conventional UAVs. Their high manoeuvrability for navigating confined spaces, hovering, the promotion of open source autopilots, payload capacity, the participation of the academy and the continuous capital of research have contributed to the research development of these types of aircraft, particularly multirotor vehicles. Air pollution monitoring [1], aerial photography [2], package delivery services [3], precision agriculture [4], and search and rescue operation [5]. Many of these applications require a navigation system, a decision-making module, a robust trajectory control system, sensors, and additional instrumentation. Currently, the trend in UAVs is towards aircraft with the ability to adapt to dynamic environments, intelligent aircraft capable of making decisions in real-time, cooperative and robust in the face of different disturbances. Since the multirotor experiences multiple aerodynamic effects, environmental disturbances, unmodeled dynamics, parametric uncertainties, noise, and even actuator failures, it is essential to design robust control strategies to attenuate or eliminate these unpredictable effects. From a theoretical perspective, in this thesis research, the trajectory tracking

problem subject to multiple aerodynamic effects such as ground effect, wind gusts, noise, failures in the rotors and non-modelled dynamics is addressed through the design of new novel robust control in position and attitude. From a practical perspective, a new laboratory-level architecture for cooperative control of heterogeneous multi agent systems is proposed. The flying arena consists of a motion capture system (optitrack), a ground station module with a local server, ROS middleware for interfacing, a wireless UDP command interface via a single-chip computer, and robots capable of Communication.

1.2 Justification

In this thesis research, a new extended state nonlinear observer (NESO) is designed for accurate trajectory tracking and with an improvement in its physical implementation. A non-linear extended state observer is designed in the three axes of the inertial frame, feeding back the attitude variables and the calculated thrust. The observer performs on the linear velocity error dynamics, avoiding the numerical calculation of the feedforward term $\dot{v}_d$ and being able to estimate it together with the global disturbance.

For altitude control, an improved ADRC design with the in-ground-effect model is proposed. An IGE model is used as a multiplicative factor in the control gain to compensate for the insufficient disturbance compensation ability of ADRC. Motivated by the aforementioned discussion, in this research, a new extended state nonlinear observer (NESO) is designed for accurate trajectory tracking and with an improvement in its physical implementation. A non-linear extended state observer is designed in the three axes of the inertial frame, feeding back the attitude variables and the calculated thrust. The observer performs on the linear velocity error dynamics, avoiding the numerical calculation of the feedforward term $\dot{v}_d$ and being able to estimate it together with the global disturbance.

The contribution of this work focuses on the proposal of a position control robust to wind gusts, ground effects, parametric uncertainties and coupling effects. A new extended-state nonlinear observer under error dynamics and a disturbance error function is proposed. This function does not allow the actuators to be saturated when the control action cannot reject the estimated disturbance.

This effect is widespread in the physical.

1. An unsimplified model is established in the presence of endogenous and exogenous disturbances.

2. A novel robust trajectory tracking control scheme is designed to stabilise the position of the multirotor subject to parametric uncertainties, ground effect and wind gusts.

3. A non-linear extended state observer is designed for the multirotor position, taking measurements of the Euler angles (pitch and roll) and the altitude.

4. A nonlinear extended state observer (ESO) is designed on the vertical axis to estimate the ground effect at low altitudes.

5. The algorithm is physically implemented and is shown to be lightweight and efficient.

6. A formal closed-loop control-observer stability test is performed in the ISS sense.

7. An integral square error performance index compares the classic ADRC and the proposed control.

8. Experimental tests on a physical UAV platform validate the proposal.

9. This paper's main contribution is adding a function on the dynamics of the error and the coefficient of the sufficiency of control, which help the convergence of the estimation error and its possible saturation. This new methodology stimulates future interactions and cross-fertilisation of ideas between the various disciplines of automatic control.

1. An unsimplified model is established in the presence of endogenous and exogenous disturbances.

2. A novel robust trajectory tracking control scheme is designed to stabilise the position of the multirotor subject to noise, ground effect, parametric uncertainties, coupling effects and wind gusts.

3. A non-linear extended state observer is designed for the multirotor position, taking measurements of the Euler angles (pitch and roll) and the altitude.

4. A nonlinear extended state observer (ESO) is designed on the vertical axis to estimate the ground effect at low altitudes in hovering flights.
5. The algorithm is physically implemented and is shown to be lightweight and efficient.
6. A formal closed-loop control-observer stability test is performed in the ISS sense.
7. An integral square error performance index compares the classic ADRC and the proposed control.
8. Experimental tests on a physical UAV platform validate the proposal.

1.3 Objectives

1.3.1 General Objective

Develop a autopilot for the control and guidance of Drones with communication capabilities for the implementation of Drone networks.

1.3.2 Specific Objectives

1. Propose an ecosystem for the development and life cycle stage of the autopilot system.
2. Define and justify the system architecture and requirements.
3. Define and justify the security levels that the system must contain.
4. Develop control and navigation algorithms for the multicopter indoors.
    - Modeling and control of unmanned aerial vehicles.
    - Navigation systems, sensors, vision and actuators.
    - Operating systems that can be used RTOS, NuttX and ROS.
    - High performance wireless communications for the implementation of Drone networks.
    - Approach to the development cycle from the implementation of algorithms in high-level language, up to implementation in low-level language.

- Suppliers and supply chain for the development of the system.
- Identification of interest groups: public or private sector (companies).
- Control of attitude and posture using the control technique active disturbance rejection control (ADRC).
- Position control, where the position will be obtained using an Optitrack motion capture system.
- Develop and implement the communication system for data transmission to the land base through point-to-point communications.

5. Develop control and navigation algorithms for an outdoor multicopter. Unlike the previous objective, the position of the system must be obtained from a GPS system.
6. Development of comprehensive verification tests of the development cycle.

Chapter 2

State of the Art and Theoretical Framework

The true sign of intelligence is not knowledge but imagination.

– **Albert Einstein**

This chapter presents the state of the art of UAVs, their generalities, classification, operating principles and mathematical model. It begins with a detailed and exhaustive study of state of the art, focusing on the most outstanding research, the flagship products that influenced its development and the most relevant facts. Next, UAVs are classified into four categories based on their aerodynamic configuration; fixed-wing aircraft, rotary-wing aircraft, flapping-wing aircraft, and hybrid aircraft. Focused on multirotor UAVs, its operating principle is described. For a better understanding, the multirotor has been divided into three systems; propulsion and energy, the communication system and the control and decision-making system. Finally, the dynamics of the multirotor are derived with a detailed analysis of multiple disturbances, namely ground effect, suspended load, and wind gusts.

2.1 State of the Art

The development of multirotors has been divided into four stages, the beginning stage (1990-2002), the growth stage (2003-2010), the consolidation stage (2011-2018) and the present (2019-2022).

1. **Beginning Stage (1990-2002):** The first reported multirotor helicopter was Gyroplane No.1 in 1907 by brothers Louis and Jacques Bréguet. The first experiment in Douai, France, was a hover flight with a pilot on board at the height of 1.5 meters [6]. Years later, in 1923, Etienne Oemichen designed the Oemichen No. 2, an improved multirotor of his first model, with small vertically mounted rotors rotating in the opposite direction; it flew for fourteen minutes hovering [7]. Around the same year, in 1921, the United States Army Air Service hired George Bothezat to build a multirotor. Although the flight tests were surprising, they did not meet expectations, with a payload of three people, one pilot, and a height of one hundred meters [8]. These three aircraft were the founders of this period. However, many problems were evident, including impractical designs, poor engine performance, high vibration, and a lack of accurate measurement systems. Due to the results obtained, the projects cancelled by the US Army and the lack of large-scale development caused people to show little interest and opt to develop conventional aircraft, which were superior in speed, payload and time. Of flight. Years later, in the early 2000s, the development of microelectronics and microelectromechanical systems (MEMS) made it possible to incorporate a variety of computing and communications functions into small VTOL-UAVs; this opened the doors for the development of mini multirotor aircraft [9]. In addition, the first commercial mini multirotors designed for indoor use emerged. These contained a simple semi-automatic attitude stabiliser and a vehicle autonomy of three to five minutes and were piloted by radio control. This first generation includes the Keyence Gyrosaucer, the Roswell Flyer, the mesicopter, the X4-flyer, and the Silverlit X-UFO. In addition to the mentioned products, some multirotors projects at the academy started, such as Mesicopter, HoverBot and the Starman [10]. Some investigations of the most outstanding contributions of this period are P. Castillo's thesis at University Compiégne [11], P.Pounds's thesis at Australian National University ANU [12] and N. Guenard's thesis at CEA [13].

2. **Developing Stage (2003-2010).** The development stage is where research by universities is concentrated. Endless articles were published [14–19]. Regarding automatic control techniques, different approaches were presented, for example, PID proportional-integral-derivative in [20–22], linear feedback in [23, 24], proportional plus second-order differentiator PD^2 in

[25], nested saturation functions in [26, 27], among others. However, these early works are characterised by making some assumptions to facilitate the control algorithms, for example, small angles, absence of external disturbances, knowledge of the inertia/mass of the system, and disregard for gyroscopic effects or aerodynamic friction couples generated by the aircraft and rotors. Despite proposing linear control algorithms invariant in LTI time, they managed to stabilise these small aircraft. Most researchers used commercial multirotors to test their control algorithms since electronic commerce needed more maturity. The first commercial multirotors focused on sports entertainment, such as aerial photography, were commercialised at this stage. Microdrones GmbH was one of the first companies to design multirotors; it was founded in October 2005. The company sold thousands of drones, the MD4-200 and MD4-1000 being the company's first generation. Another big company was Draganfly, which launched the first multi-rotor UAV with an integrated camera system in 2002. In 2004 Parrot, a french company dedicated to wireless products manufacturing and signal processing for embedded systems initiated the AR. Drone project.

In 2006 several groups started using the optical motion-capture system, such as "Indoor multi-vehicle flight test-bed for fault detection, isolation and recovery" at Massachusetts Institute of Technology [28], "The Flying Machine Arena" at ETH Zurich [29], "the Grasp Multiple micro-UAV test bed" at the University of Pennsylvania [30]. In 2008 the Draganflyer X6 six-rotor helicopter was recognised by the popular Science's "Best of What's New" Award in the Aviation Space category. The AR. Drone was very successful in the toy market. It had a 32-bit ARM processor, Linux-based operating system, front and back camera to measure optical flow and detection capabilities, and two ultrasonic range finders to measure the altitude.

3. **Consolidation Stage (2011-2018)** At this stage, drones tend to be more autonomous, cooperative, and human-friendly (interacting with people). According to the International Organization for Standardization definition, robot autonomy is the ability to perform intended tasks based on the current state and sensing without human intervention. In [31] identify three levels of increasing autonomy:

(a) **Sensory-motor autonomy**. Translates high-level human commands such as reaching an altitude, following GPS global positioning system coordinates, holding a position, and trajectory tracking.

(b) **Reactive autonomy**. Avoid obstacles and maintain current position or trajectory in external perturbations, such as wind or electro-mechanical failure. It can keep a safe or predefined distance from the ground, coordinate with moving objects, including other drones, and take off and land automatically.

(c) **Cognitive autonomy**. Perform simultaneous localisation and mapping, decision-making, integrating learning algorithms, and resolving conflicting information.

Based on these levels of autonomy, the development of multirotors by companies were focused. One of the most influential events in this stage was in 2012. Vijay Kumar presented the conference entitled "Robots that fly and cooperate", where he demonstrated the potential applications of these aircraft, such as cargo transportation, assembly of structures and construction of buildings. A year later, DJI released its DJI Phantom drone in 2013. The Phantom ready-to-fly drone system was one of the products that popularised these aircraft. It already included an internal GPS and could hold its position, plus a high specific payload capacity and flight modes to resist wind. Users could mount a GoPro or other lightweight camera for aerial photography. In 2013 the PX4 team headed by Lorenz Meier at ETH Zurich announced Pixhawk, an advanced open-hardware autopilot design for multicopters, fixed-wing aircraft, ground rovers, and amphibious vehicles. In 2013, Raffaello D' Andrea at ETH Zurich made a speech at TED Global titled "Machine Athleticism". This presentation shows the ability of machines to perform dynamic feats that fully exploit their physical capabilities. In 2014 Amazon released a video hoping to deliver small packages home in just thirty minutes over "Prime Air", a future delivery system consisting of multirotors. Regarding automatic control, different non-linear control strategies have been published. Mainly to deal with noise in actuators, unmodeled dynamics, external disturbances, coupled systems like suspended loads and robotic arms. Different robust control techniques were designed and implemented. For example, the backstepping technique is robust to parametric uncertainties,

and does not need detailed model information. It is a recursive procedure that combines the choice of a Lyapunov function with the design of feedback control. Some works devoted to backstepping control can be found at [32]. In [33–35] a geometric control is proposed for trajectory tracking on the in special Euclidean group $SE(3)$. The sliding mode control (SMC) technique is robust to certain external disturbances and unpredictable parameter variations [32, 36, 37]. However, the chattering problem in SMC is one of the points to consider in real-time applications due to imperfections in switching devices, the fatigue caused by switching and delays. Another standard method to deal with parameter uncertainties and disturbances is the adaptive control approach [38, 39].

4. **Present (2019-2022).**

   Currently, the trend in UAVs is towards arcraft with greater autonomy capacity, capable f making decisions, cooperative and adaptive to different situations, for example, robust control [40], aggressive manoeuvring [41], [42], aerobatic flips [43],[44] and aerial manipulation [45],[46], [47]. Focusing on the central theme of this thesis, control, at this stage, different robust riot methodologies have been proposed. Anti-disturbance control methodology or observer-based control method has effectively dealt with multiple disturbances. Generally, this methodology relies on disturbance estimation to cancel the global disturbance's effect through feedback action [48]. It comprises two parts; an observer to estimate the disturbance and a controller based on feedback [49, 50]. In the literature, a range of different observers can be found that allow estimation of disturbances, sliding mode disturbance observer [51], disturbance observer (DO) [52], extended state observer ESO and frequency-domain disturbance observer [53].

2.1.1 Unmanned Aerial Vehicle (UAVs) and Classification

An unmanned aerial vehicle (UAV) is defined as an unmanned aircraft that a pilot from a ground station can operate or can fly autonomously according to preprogrammed flight plans or more complex dynamic automation systems [54].

The UAVs can be classified, taking into account different criteria. For example, in

1. **Fixed Wing.** Its wings are fixed to the fuselage of the aircraft. Its structure is simple and can carry a heavier payload over a longer distance while consuming less energy. Its most important applications are environmental monitoring, surveillance missions, crop spraying, etc. The downside is that it cannot take off and land vertically, in addition to needing a runway or launcher for takeoff and landing, see Fig. 2.1(a).

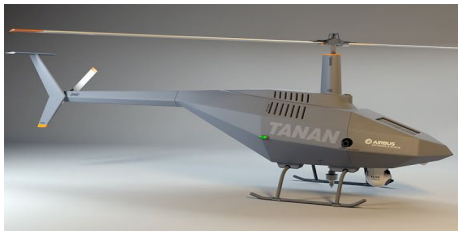
2. **Rotary Wing.** This aircraft airlift is provided directly by rotors. A typical example is the helicopters in their different configurations. This aircraft can take off and land (VTOL) vertically and does not need a runway or launcher. Some of its most important applications are in the urban environment, aerial manipulation, aerial photography, 3d reconstruction, search, surveillance, etc. One of the disadvantages is higher energy consumption, and high maintenance costs, see Fig. 2.1(b).

3. **Flapping Wing.** This type of aircraft attempts to reproduce the flying of insects or birds. These systems have low power consumption and the capacity to land and take off vertically. However, they have low resistance and minimum load capacity. Some of its most important applications are to fly indoors and in places where other autonomous aircraft cannot reach. The technology is not yet too mature to exploit these micro aircraft, see Fig. 2.1(c).

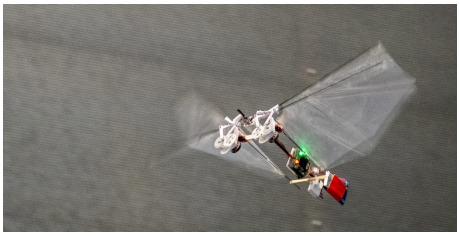
4. **Hybrid Aircraft.** These aircraft combine the advantages of fixed and rotary-wing aircraft. These aircraft can both take off and land vertically in addition to flying at high speed. Its carrying capacity is also increased. However, the design of these aircraft is complex, and their flight dynamics control challenging tasks, see Fig. 2.1(d).

The multirotor is a vertical takeoff and landing aircraft that uses multiple rotors to generate lift. Multirotors have three or more symmetrically distributed rotors attached to the end of an arm; typical configurations are trirotor, quadrotor, hexarotor, and octorotor. The number of n rotors and the angle φ between two arms satisfy the following relation $\varphi = 360°/n$. The multirotor contains several parts to function, and these components interact in complex ways. The good synchronisation of these devices influences the performance of the aircraft. To understand how this hardware relates to each other, we have divided the multirotor into three systems; the propulsion and energy system, the communication system, and the control and decision-making system, see Fig. 2.2.

(a) Fixed wing UAV.
SITARIA UAVOS-Company.



(b) Rotary wing UAV.
TANAN AIRBUS-Company.



(c) Flapping wing UAV.
DelFly Nimble Delft University .

(d) Hybrid UAV.
HQ L3HARRIS-Company.

Fig. 2.1: Classification of Unmanned Aerial Vehicles (UAVs).

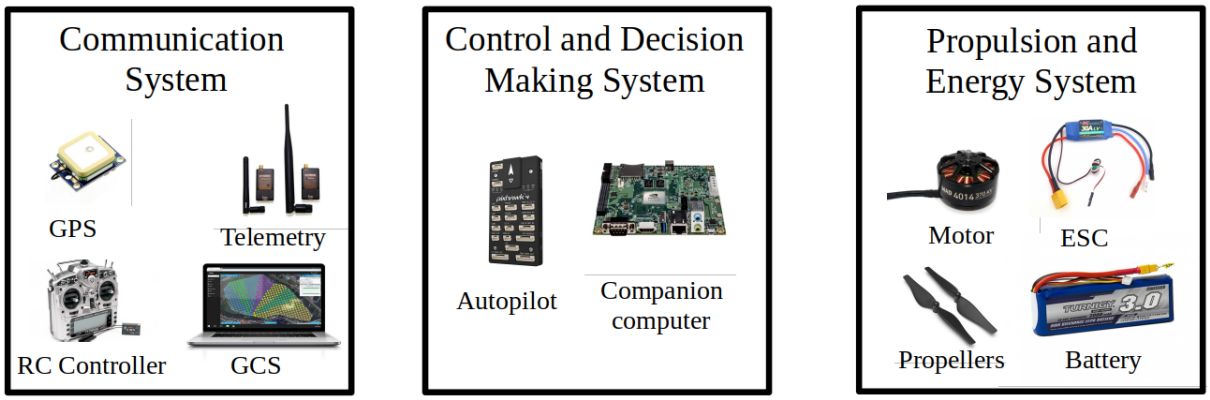


Fig. 2.2: Systems that make up a multirotor.

1. **Propulsion and Energy System.** The power system supplies power to the multirotor to function correctly. Different voltage levels are managed. The propulsion system includes motors, variable-speed drives and propellers. This system determines payload, flight speed, and range.

    - Lipo Battery: is used to provide energy to the motors. The battery cells can be in series, indicated by the letter s, parallel, indicated by the letter p or a combination of both; the nominal voltage cell is (3.7-4.2)V. The capacity measure in milliAmpere-hour

(mAh) indicates how much charge the battery has. The discharge rate helps to stay within the maximum limit of the current demanded from the battery.

- Brushless DC Motor: is used to create movement on the multirotor. The size of the motor is represented by four digit number; it represents the diameter and height stator measured in mm. Generally, the valor for engines is in VK, meaning the number of RPM that the motor will revolve when 1V is applied. Other parameters to consider are the maximum current that the motor can undertake and the efficiency.
- Electronic Speed Controller: is used to control motor speed by PWM signal. Transform the DC power into a three-phase Alternating Current power. An important parameter is an ampere and voltage they can bear.
- Propeller: is used to produce the lift and torque to control the multirotor. The most critical parameters are the diameter, pitch and number of blades propeller. The propeller is described by a four-digit, the first two represent the diameter, and the last two represent the pitch measure in inches.

2. **Communication System.** This system includes external communication devices with which the aircraft can communicate depending on the degree of autonomy. As shown in Fig. 2.2, the main components are: telemetry, GPS, Radio Control and the ground control station (GCS). The APIs used to communicate with the aircraft are omitted but are implicit.

   - RC transmitter: The RC transmitter's function is transmitting the desired command from the inertial frame to the multirotor, where the radio receiver RX must be connected to autopilot. Most RC transmitters and receivers communicate by 2.4 GHz radio frequency to avoid interfering with other pilots.
   - Ground Station: It is an application programming interface (API) which can be installed on a laptop, cell phone or tablet. The functions are the real-time monitoring of flight status (Telemetry, GPS, heading, airspeed, flight mode), creation of autonomous missions, streaming video, and configuring and tuning the UAVs. Some GCS software is QGroundControl, MissionPlanner, Openpilot, Multiwii and Crazyflie.

- Radio Telemetry: Radio telemetry sends information on variables such as temperature, voltage, heading, position, velocity and network management from the UAV to the GCS.

3. **Control and Decision Making System** This system contains two components: the automatic pilot and the decision module. The autopilot controls the UAV's attitude, position and trajectory; this depends on the level of autonomy required by the mission or programmed task. This system contains two components: the automatic pilot and the decision module. For most applications, the autopilot is sufficient. However, applications that require analysis of a large amount of information and onboard processing require an external module. This module integrates functions such as perception, navigation and decision-making.

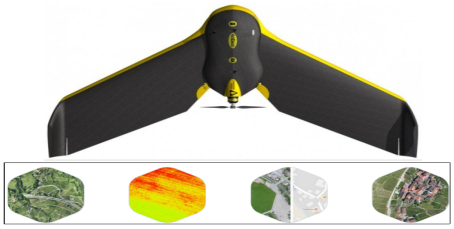
   - Autopilot: It is a flight control system that calculates the rotors' angular speed to achieve the desired attitude or position. It can be semi-automatic or fully automatic.
   - Decision Marking module: They are co-processors which help the autopilot to make intelligent decisions during the flight, in addition to performing specialised tasks such as mission planning, path following, obstacle avoidance, 3D reconstruction, SLAM and deep learning.

2.1.2 Civil Applications

Currently, multirotors are used for military and civil applications. These small aircraft have particular characteristics, such as high manoeuvrability for navigating confined spaces, hovering, open-source autopilots, and payload capacity. Many applications use UAVs. We will focus on civil applications that have had a great push in the last ten years. Air pollution monitoring [1], aerial photography [2], package delivery services [3], precision agriculture [4], and search and rescue operation [5]. Fig. 2.3 shows some applications that have had a more extraordinary boom in recent times. Many of these applications require a navigation system, decision-making, precise trajectory control, sensors and extra instrumentation.

1. **Package Delivery Services.** Aerial Manipulation is the load transport using different mechanisms (delivery-retrieving, claws and robotics arms). Purchased goods, food, medicines,

(a) Aerial Manipulation.



(b) Precision agriculture.

(c) Pandemic relief (Covid-19) .

(d) Weather prediction and energy.

Fig. 2.3: Applications of the Multirotors UAV.

vaccine and medical samples in disaster areas, into and out of remote or inaccessible regions are some applications in this field. In particular, a delivery system with VTOL-UAVs is taking a big boost; it is expected that revolutionise how customers receive purchased goods. Some advantages of using UAVs as a delivery system are the reduction of shipping costs, less manual supervision and faster delivery times since they do not depend on roadwork or traffic jams [55].

2. **Precision Agriculture.** Precision agriculture (PA) is a system that seeks to maximise long-term production and efficiency while optimising resource use and sustainability [56]. As field size has grown and agricultural practices intensified, it has become increasingly challenging to address this variability without increasing reliance on technology [57]. The multirotors can assist farmers in various tasks such as irrigation, fertilisation, pesticide usage, weed management, plant growth and disease monitoring. This can maximise productivity, quality, and yield.

3. **Pandemic Relief (Covid-19).** UAVs are being used for endemic controls. These UAVs integrate thermal cameras to detect the temperature of a particular group of people; detection occurs in a matter of seconds and favours isolation in addition to adding other sensors to see the individual's heart and respiratory rates. Disinfecting large areas is one of the tasks being

carried out.

4. **Air Pollution Monitoring.** UAVs are increasingly used to assess, prevent and even combat environmental disasters. The UAVs are used for pollution monitoring, detection, and reduction at altitudes above ground level in a specific geographic region [58]. The environmental drones produce Air Quality Health Index (AQHI) maps of covered areas for ecological data monitoring, long-term analysis, finding the hazardous zone, and future predictions.

5. **Search and Rescue Operation.** The search and rescue operations assist, detect, and rescue people who have had accidents in hazardous environments such as mountains, lowlands, cities, disaster scenarios and marine rescue. The multirotor used is to minimise the cost and time spent.

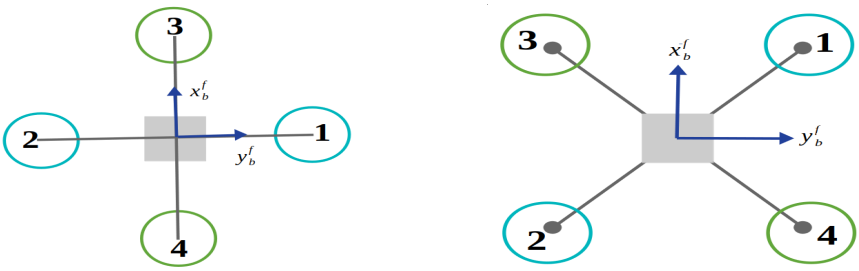
2.2 Theoretical Framework

The principle of operation is described in this section. Mainly we focus on the description of the quadrotor that is the aircraft under study in this thesis project. Next, we set the bases for modelling the quadrotor, considering different aerodynamic effects and external disturbances.

2.2.1 Operating Principle

The quadrotor is a subactuated system which can only control four inputs simultaneously. However, it has six degrees of freedom. Thrust force, pitch, roll and yaw moments are the control input; these four inputs can control the attitude and position of the quadrotor. There are two airframe configurations, plus-configuration and cross-configuration; see Fig. 2.4. The rotors are symmetrically distributed at the end of the four arms. The cross-configuration involves more rotors in the roll and pitch movements, which generates greater maneuverability. Maneuverability is the ability to change its own state, which is closely related to the maximum acceleration of a multirotor [59], the cross-configuration cancelled because a faster response can be obtained.
The main commands of the multirotor are described below,

1. **Thrust command.** This command is accomplished by increasing all propellers by the same



(a) Plus-Configuration. (b) Cross-configuration.

Fig. 2.4: Quadrotor Configurations.

amount. As a result, the reactive moments will be cancelled, and the lift will be increased. Once the weight is overcome, the quadrotor will fly steadily. Otherwise, the quadrotor will move down. The Fig. 2.5(a) shows the hover command.

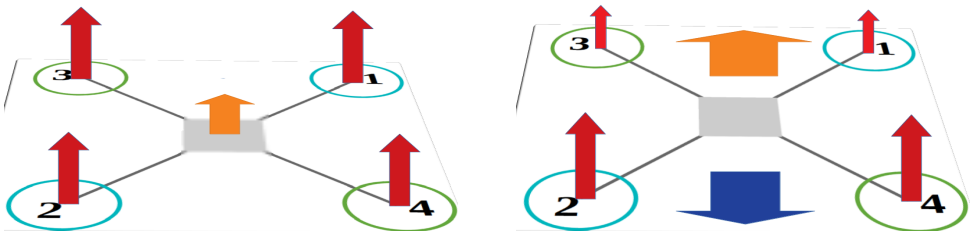
2. **Pitching command.** This command is accomplished by decreasing the propellers' angular speed (1,3) and increasing the propellers' angular speed (2,4) by the same amount. This command generates forward and backward motion, see Fig. 2.5(b).

3. **Rolling command.** This command is accomplished by decreasing the propellers' angular speed (1,4) and increasing the propellers' angular speed (2,3) by the same amount. This command generates rightward and leftward motion. The Fig. 2.5(c) shows the rolling base.

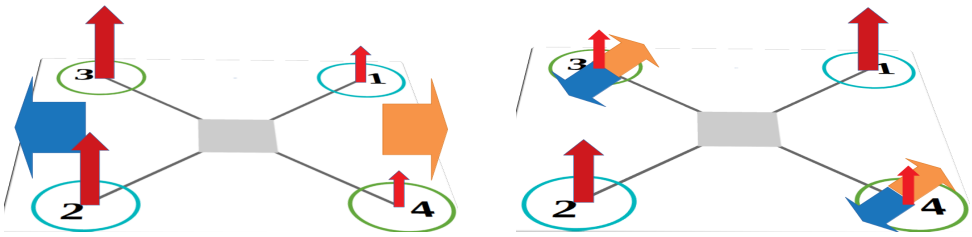
4. **Yawing command.** This command is accomplished by decreasing the angular speed of propellers (1,2) and increasing the angular momentum of propellers (3,4) by the same amount; see Fig. 2.5(d).

2.2.2 Mathematical Modelling

This subsection establishes the mathematical model of the multirotor by analysing multiple aerodynamic effects and external disturbances. The mathematical model of multirotor consists of four blocks, the kinematic model, the dynamic model, the effectiveness model and the propeller model. The kinematics and dynamics follow from the assumption that the aircraft is a rigid body; In addition, the effectiveness model differentiates the multicopter from other aircraft, which calculates the moments and forces from the angular speeds of the rotors. Finally, the propeller model includes a



(a) Hover command. (b) Pitching command.

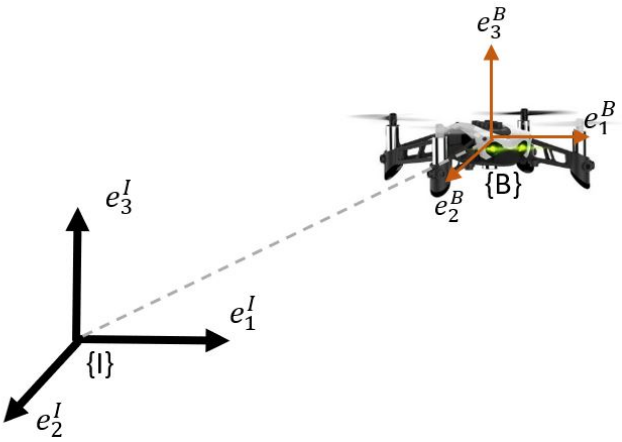


(c) Rolling command. (d) Yawing command.

Fig. 2.5: Commands Quadrotor.

brushless DC motor, ESC variable speed drives and propellers. In the kinematic analysis, we obtain the positioning in the three-dimensional space and its orientation without considering the forces or moments that produce it. The dynamic analysis explains the physical phenomena found in the aircraft, such as the centripetal and Coriolis forces, the gravitational torque, gyroscopic effects due to the rotors, in general, all the details and moments that influence the linear and angular velocity of the aircraft. One of the characteristics of the multirotor is the thrust force is perpendicular to the plane of the fuselage at all times. An accurate model of the system to be studied is the basis for analysing and designing control algorithms that globally stabilise the robot asymptotically. In addition, the influences of multiple disturbances, such as the ground effect, oscillation of a coupled suspended load, and wind gusts, are analysed. The multirotor can be modelled as a rigid body with six degrees of freedom sub-actuated with only four independent control inputs: three moments in their respective axes and one thrust force perpendicular to the aircraft. For a deep understanding of the path that the aircraft takes due to the forces and moments acting on it, it is necessary to introduce two reference systems, the inertial frame $\{I\} = \left[e_1^I, e_2^I, e_3^I\right]$ and the body frame $\{B\} = \left[e_1^B, e_2^B, e_3^B\right]$ is fixed to the multirotor. The Fig. 2.6 shows the relationship between the $\{I\}$ and $\{B\}$, both obeying the right-hand law.

Below we describe the main blocks that represent the complete model of the multirotor; see Fig. 2.7.



centring

Fig. 2.6: Relationship between the $\{I\}$ and $\{B\}$ Coordinate Frames.

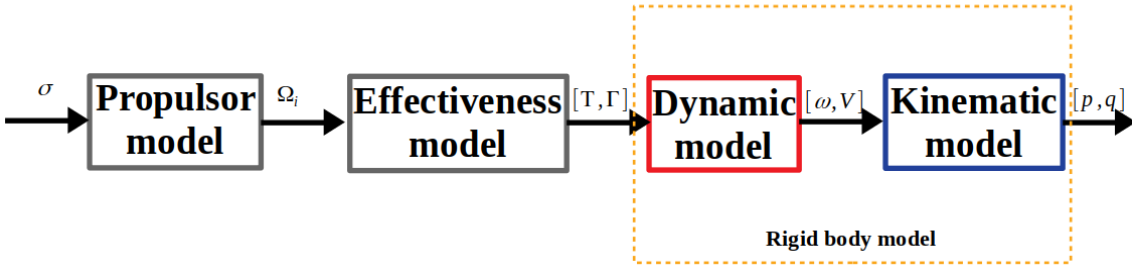


Fig. 2.7: Main blocks of the multirotors model mathematical.

1. **Kinematic model.** There are many methods to represent the attitude; each attitude representation has its advantages and disadvantages; the most used are Euler angles, rotation matrix and quaternions. Based on Euler's theorem, the rotation of the multirotor can be achieved by three elementary rotations about a fixed point of the inertial frame. The rotation of the coordinates of a point from body frame $\{B\}$ concerning inertial frame $\{I\}$ is represented by the orthogonal rotation matrix $R \in SO(3) = \{R \in^{3\times3}: R^T R = I_3, \det(R) = 1\}$, where $I_3 \in^{3\times3}$ is the identity matrix. Euler angles (roll, pitch, yaw) are a representation of attitude that can be obtained using the vector $\Theta = \begin{bmatrix}\phi & \theta & \psi\end{bmatrix}^T$. The corresponding rotation matrix is then given through the map $R :^3 \rightarrow SO(3)$ by

$$R = \begin{bmatrix} \cos\psi\cos\theta & \cos\phi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \cos\theta\cos\psi & \sin\phi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \tag{2.1}$$

The (C.1) rotation matrix converts most of the variables from the $\{B\}$ to the $\{I\}$ coordinate frame, variables such as force, magnetic force, position, speed and linear acceleration.

Remark 2.2.1 *In this investigation, $R_B^I = R$ is the matrix that rotates the coordinates of a point from $\{B\}$ to $\{I\}$.*

The velocities $\omega = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$ measurement in the body coordinate frame $\{B\}$, relative to the inertial coordinate frame $\{I\}$. The attitude matrix (C.2) relationship the attitude rate $\dot{\Theta}$ and the angular velocity of the multirotor ω.

$$W = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & -\sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \tag{2.2}$$

The kinematics of multirotor expressed in $\{I\}$ is denoted by the following equation

$$\dot{\Theta} = W\omega \tag{2.3}$$

Secondly, the quaternions are not affected by the gimbal lock phenomenon and further have significant advantages like nonlinear large angle attitude manoeuvres and high computer performance. Let the attitude quaternion by $\mathbf{q}_b^e$,

$$\mathbf{q}_b^e = \begin{bmatrix} \cos\frac{\theta}{2} \\ \mathbf{v}\sin\frac{\theta}{2} \end{bmatrix} \tag{2.4}$$

where $\mathbf{v} \in \Re^3$ is the vector part and represents the rotation axis and $\theta \in \Re$ is the scalar part and represents the amount of rotation around of axis $\mathbf{v}$. The attitude quaternion $\mathbf{q}_b^e = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T$ is used to rotate any vector $\mathbf{r}_e \in \Re^3$ from $\{B\}$ frame to the $\{I\}$ frame by the following operation

$$\begin{bmatrix} 0 \\ \mathbf{r}^e \end{bmatrix} = \mathbf{q}_b^e \otimes \begin{bmatrix} 0 \\ \mathbf{r}^b \end{bmatrix} \otimes (\mathbf{q}_b^e)^{-1} \tag{2.5}$$

The result of this rotation will be the new vector $\mathbf{r}_e \in \Re^3$ which belongs to $\{I\}$. To achieve this rotation, the product of the unit attitude quaternion $\mathbf{q}$ and its inverse; furthermore, because the vector is unitary, its inverse is equivalent to its conjugate. The 2.5 uses quaternion multiplication, which is previously defined. According to (A.5) the 2.5 can be expressed as follows:

$$\begin{bmatrix} 0 \\ \mathbf{r}^e \end{bmatrix} = (\mathbf{q}_b^e)^{-1} \otimes \mathbf{q}_b^e \otimes \begin{bmatrix} 0 \\ \mathbf{r}^b \end{bmatrix} \tag{2.6}$$

The attitude quaternion can be used to construct a 3×3 rotation matrix to perform the rotation in a single operation. The rotation matrix is defined as

$$\mathbf{R}_b^e(\mathbf{q}^b) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^3 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{2.7}$$

$$\mathbf{r}_e = \mathbf{R}_b^e(\mathbf{q}_e^b) r_b \tag{2.8}$$

Let $\mathbf{p}_e$ *in* $\Re^3$ a vector that represents the position of the multirotor from the origin to the CoG of the multirotor. The following equation denotes the kinematics of this rigid body

$$\dot{\mathbf{p}}_e = \mathbf{v}_e \tag{2.9}$$

where $\mathbf{v} \in \Re^3$ represent the linear velocity of the multirotor. The attitude representation using quaternion is expressed as

$$\dot{\mathbf{q}}_b^e = \frac{1}{2} \begin{bmatrix} 0 & -\omega_b^T \\ \omega_b & -[\omega_b]_\times \end{bmatrix} \mathbf{q}_b^e \tag{2.10}$$

With $\mathbf{q}_b^e = \begin{bmatrix} q_0 & \mathbf{q}_v^T \end{bmatrix}^T$ the equation (2.10) is rewritten as

$$\begin{aligned}\dot{q}_0 &= -\frac{1}{2}\mathbf{q}_v^T \cdot \omega_b \\ \dot{\mathbf{q}}_v &= \frac{1}{2}(q_0\mathbf{I}_3 + [\mathbf{q}_v]_\times)\omega_b\end{aligned} \tag{2.11}$$

2. **Dynamic model.** There are different methods to model the physics of a system; Newton, Hamilton and Euler-Lagrange are the most used methods. In this research work, Newton is used because it facilitates the analysis and design of control algorithms. The following assumptions are assumed to obtain the dynamic model of the multirotor.

   - The multirotor is a rigid body.
   - The structure is supposed to be symmetrical.
   - The CoG and the origin are assumed to coincide.

   The procedure to obtain the dynamic model is described below; analysing the forces on the multirotor in the inertial frame $\{I\}$, and deducing the following equation

$$\dot{\mathbf{v}}_e = g\mathbf{e}_3 - \frac{f}{m}\mathbf{R}_b^e\mathbf{e}_3 \tag{2.12}$$

   where g $\in \Re_+$ represent the acceleration if gravity and f $\in \Re_+$ is the total thrust generated by rotors. The (2.12) represent the position dynamic model expressed in $\{I\}$.
   The position dynamic model expressed in $\{B\}$ is described by the following equation

$$\begin{aligned}\mathbf{v}_e &= \mathbf{R}_e^b\mathbf{v}_b \\ \dot{\mathbf{v}}_e &= \dot{\mathbf{R}}_b^e\mathbf{v}_b + \mathbf{R}_b^e\dot{\mathbf{v}}_b \\ \dot{\mathbf{v}}_b &= g\mathbf{R}_e^b\mathbf{e}_3 - [\omega_b]_\times\mathbf{v}_b - \frac{f}{m}\mathbf{e}_3\end{aligned} \tag{2.13}$$

   the equation (A.12) was used to derive the (2.13) equation.

Analysing the moments on the multirotor in $\{B\}$, obtain the attitude dynamic equation as

$$\mathbf{J} \cdot \omega_b = -\omega_b \times (\mathbf{J} \cdot \omega_b) + \mathbf{G}_a + \tau \tag{2.14}$$

where $\mathbf{J} \in \Re^{3\times3}$ is the multirotor inertia matrix constant and symmetric. $\mathbf{G}_a \in \Re^3$ represents the gyroscopic torques caused by the rotation of the propellers and $\tau \in \Re^3$ represent the moments generated by the rotors,therefore its the control inputs. Using the above equations, we obtain the mathematical model of the multirotor combining both reference frames, the $\{I\}$ and $\{B\}$. The multirotor rigid model is expressed as

$$\begin{aligned}
\dot{\mathbf{p}}_e &= \mathbf{v}_e \\
\dot{\mathbf{v}}_e &= g\mathbf{e}_3 - \frac{f}{m}\mathbf{R}_b^e\mathbf{e}_3 \\
\dot{q}_0 &= -\frac{1}{2}\mathbf{q}_v^T \cdot \omega_b \\
\dot{\mathbf{q}}_v &= \frac{1}{2}(q_0\mathbf{I}_3 + [\mathbf{q}_v]_\times) \cdot \omega_b \\
J \cdot \omega_b &= -\omega_b \times (\mathbf{J} \cdot \omega_b) + \mathbf{G}_a + \tau
\end{aligned} \tag{2.15}$$

The linear kinematics and dynamics are described in the $\{I\}$ variables, such as the position and the velocity of the multirotor measured with GPS. While the angular kinematics and dynamics are described in the $\{I\}$, variables such as attitude and angular velocity are measured with gyros, accelerometers and magnetometers.

3. **Effectiveness Model**

The relationship between the rotor angular speeds $\mathbf{\Omega_i}$ and the total thrust f and moments τ is deduced. In particular, the X configuration of the quadrotor structure will be used. A rotor generates a lift force that points horizontally to the plane, in addition to developing a reactive torque in the opposite direction to the speed direction of the rotor. When the rotor is energised with a PWM signal, its speed will start to increase; this, in turn, will generate a thrust force proportional to the square of the speed. The thrust force is expressed as,

$$f_i = c_T\Omega_i^2 \tag{2.16}$$

where c_T is a constant that depends on air density, the diameter of the propellers, and propeller speed, among other parameters. The reaction torque is expressed as

$$M_i = c_M \Omega_i^2 \tag{2.17}$$

where c_M is a constant which depends on air density, the diameter of the propellers, and propeller speed, among other parameters.

The total thrust is expressed as

$$f = \sum_{i=1}^{4} f_i = c_T(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \tag{2.18}$$

the moments produced by the rotors are

$$\begin{aligned} \tau_x &= \frac{\sqrt{2}}{2} \cdot d \cdot c_T(-\Omega_1^2 + \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \\ \tau_y &= \frac{\sqrt{2}}{2} \cdot d \cdot c_T(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\ \tau_z &= c_M(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{aligned} \tag{2.19}$$

where d is the distance between the quadrotor centre and the motor shaft. In a matrix form rewrite the (2.19) as

$$\begin{bmatrix} f \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} c_T & c_T & c_T & c_T \\ -\frac{\sqrt{2}}{2}dc_T & \frac{\sqrt{2}}{2}dc_T & \frac{\sqrt{2}}{2}dc_T & -\frac{\sqrt{2}}{2}dc_T \\ -\frac{\sqrt{2}}{2}dc_T & \frac{\sqrt{2}}{2}dc_T & -\frac{\sqrt{2}}{2}dc_T & \frac{\sqrt{2}}{2}dc_T \\ c_M & -c_M & c_M & -c_M \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{2.20}$$

where $\mathbf{M}_{CE} \in \Re^{4\times4}$ represents the control effectiveness matrix.

4. **Propulsor Model.** This block consists of modelling the propulsion system. The system includes a brushless DC motor, ESC and propellers. First, given a constant voltage by the LIPO battery, the ESC generate an average voltage signal which is a function of the throttle command or a pulse width modulation PWM signal, which will make the motor achieve steady

state speed. The above can be modeled as

$$\Omega_{ss} = C_R\sigma + \Omega_b \tag{2.21}$$

where C_R is a constant that depends on the battery voltage and the ESC, σ is the throttle command which has a value between 0 and 1 or can be parameterised by a PWM signal with a value between 1000 and 2000 indicate the duty cycles and Ω_b is the angular speed which the motor reaches once the engine initialises. Because the motor needs some time to achieve the steady state speed Ω_{ss}, a second equation is generated to model the dynamic response. Generally, the dynamics of a brushless DC motor can be simplified as a first-order equation.

$$\Omega = \frac{1}{T_m s + s}\Omega_{ss} \tag{2.22}$$

Combining the equations (2.22) and (2.21) get the complete propulsor model

$$\Omega = \frac{1}{T_m s + s}(C_R\sigma + \Omega_b) \tag{2.23}$$

2.3 Active Disturbance Rejection Control (ADRC)

In this section, the control by active disturbance rejection is described. A study of the most recent publications of this methodology is carried out, focusing on three disturbances: ground effect, wind gusts and suspended load. Since the multirotor experiences multiple aerodynamic effects, environmental disturbances, unmodeled dynamics, parametric uncertainties, noise, and failures in the rotors (loss of efficiency, wear, bias, friction or deformation of the propeller) [60]. These faults primarily affect the propulsion system, degrade the tracking performance and thus deteriorate the multirotor's stability [61–63]. Therefore it is essential to design robust control strategies to attenuate or eliminate these unpredictable effects.

2.3.1 Generalities of ADRC

Active Disturbance Rejection Control (ADRC) has become a central element in the design of riot control algorithms. Based on the so-called [64] invariance principle, the main objective of the ADRC is to simplify the system under study by estimating all endogenous and exogenous disturbances by an extended state observer (ESO) and the respective cancellation of the disturbance via a feedback control action [49]. The methodology has been successfully applied in the automatic control of UAV-multirotors recently [65–67].

Nevertheless, few contributions and practical implementations have existed in academia and industry. Some ADRC-based work has been done on multirotor attitude control, showing the ability to cancel disturbances [68–71]. Some of the essential works reported in the scientific literature focused on studying the ground effect, disturbances due to wind gusts and the impact generated by a suspended load coupled to the multirotor are described below.

2.3.2 Ground Effect Disturbance

In most applications, quadrotors interact with hard surfaces, obstacles, their environment, and other robots. In this interaction, an aerodynamic effect called *ground effect* is produced. The ground effect reduces the aerodynamic drag generated by an aircraft's wings or propellers when it is close to a fixed surface. This effect results from airflow distortion under such covers attributable to the proximity of the ground or hard surfaces [72]. Mathematical models have been proposed describing the ground effect in hover and forward flight. These models are essential when developing robust controls for applications such as landing, hover, ground altitude flight, and motion planners for energy-saving and energy-efficient applications. The ground effect has been studied in helicopters for years. A practical method describing the helicopter ground effect was proposed in [73]. This model represents the ground thrust effect (IGE) and off-ground effect (OGE) as a function of rotor radius, vertical distance, and forward speed. Similarly, the work in [74] obtains a ground effect model for a single rotor in hover by experimentation. Unfortunately, it has been shown that these models do not accurately describe the multirotor ground effect [75–77]. In [78], the authors use visual feedback to model the ground effect forces experienced by a multirotor. The work on [77] proposes a ground

Table 2.1: IGE Ground Effect Models.

Autors	Model	Multirotor	Forward F.
Cheeseman [73].	$\frac{T_{IGE}}{T_{OGE}} = \frac{1}{1-\frac{(r/4z_r)^2}{1+(\lvert V/v_i\rvert)^2}}$	No	Yes
Hayden [74]	$\frac{T_{IGE}}{T_{OGE}} = \left(0.9926 + \frac{0.03794}{(z/2r)^2}\right)^{2/3}$	No	No
Danjun [75]	$\frac{T_{output}}{T_{input}} = \frac{1}{1-\rho\left(\frac{r}{4z_r}\right)^2}$	Yes	No
Kan [76]	$\frac{T_{IGE}}{T_h} = \frac{1-\frac{3r}{25z}}{1+\frac{3}{50}\left(\frac{V}{v_h}\right)^3}$	Yes	Yes
Kan [76]	$\frac{T_{IGE}}{T_h} = \left(0.104\frac{r}{z} - 0.0952\right)\left(\frac{V}{v_h}\right)^2 - 0.171\frac{r}{z}$	Yes	Yes

effect model using the Draganflyer X8, powered by eight propellers arranged in four coaxial pairs in a quadrotor configuration; in this analysis, the ground effect is shown to be more substantial than that predicted in [73]. In [75], a ground effect compensator with a correction coefficient is proposed based on the Cheeseman model [73] with a correction coefficient. More recently, in [79], the influence of the ground effect on the lateral movement of the multirotor is analysed. The force generated by the ground effect, modelled as a spring, is integrated into the rotors' thrust. In [76], the authors present two effective data-driven models describing ground effect in hover and forward flight. Extensive tests under different conditions validate their proposed models. Most ground effect models depend on the propeller's radius r and the altitude z_r measured to the ground in the hover condition. In forwarding flight conditions, a dependency on the multirotor's longitudinal speed is added. Table 2.1 summarises the properties of the literature's most widely used ground effect models. Despite the significant development of IGE ground effect models, there was little progress in designing rotors' thrust that uses these models as part of the control law [80].

Low-altitude flights with multirotor vehicles are a challenge from an automatic control perspective. The ground effect, forces, and moments due to the crosswind generated by the flat surfaces cause degradation in the position and attitude control systems. Consequently, they are prone to fatal accidents. Recently, various control strategies have been proposed to deal explicitly with different aerodynamic effects and external disturbances [81–83]. In [84] safety control is proposed based on proportional-derivative (PD) to stabilise the position and control in sliding modes (SMC) to stabilise the attitude; a disturbance observer is integrated to deal with ground effect and blade damage. The

control law can follow a circular path close to a surface, simultaneously experiencing ground effect and blade damage. In [85], a non-linear disturbance control and observer is designed to compensate for the impact of the ground effect. Adaptive control based on ground-effect models is designed to operate at low altitudes where the ground effect is high and is addressed in [80]. Tests show that the algorithm can tune the ground effect at a low altitude and in the autonomous landing task.

2.3.3 Disturbance Due to Gusts of Wind

The wind is one of the most influential meteorological parameters to consider in designing robust control algorithms. A variation in direction or intensity can cause loss of lift, deviation from the programmed route and loss of stability. Gusts of wind also affect takeoff and landing manoeuvres, causing abrupt changes in their speed and direction. Some mathematical models describe approximately the effect of the wind. For example, the crosswind, which is perpendicular to the inertial frame and affects the direction of the wind. Constant currents are characterised by keeping their direction stable and non-periodic. Turbulence, characterised by a variation in wind speed, has a short duration and rapidly decreases rate. Recently, some works using anti-disturbance methodology have been done to attenuate the wind's effect on the multirotors. For example, in [51], a sliding mode control-based is designed for altitude and attitude in a low-altitude flight regimen, compensating for wind disturbance with a maximum speed of 3.3 m/s. Automatic take-off and landing manoeuvres are carried out practically. In [70], an ADRC control is performed to stabilise the attitude subject to wind gusts; a linear extended state observer LESO is developed. An advantage of using a LESO is its straightforward structure with reduced parameters to be tuned and little plant knowledge. However, the total capacity of the observer is not exploited, and the observer's estimation accuracy primarily determines the closed-loop system's performance [86]. An improved ADRC for attitude control is proposed in [66]. This attitude control adds a filter using the $fal(t)$ function to the ESO extended state observer to attenuate noise. The controller is analysed through numerical simulations under two scenarios, constant gusts and loss of an actuator at 25 % of its total thrust, both in hover flight. In [86] an ADRC for trajectory tracking control is proposed to deal with wind gusts and ground effects. The controller combines the technique of backstepping and generalised extended state observer (GESO) for attitude and position control. Experimental results exhibit a high degree

of accuracy in following a circular path. In [53], a nonlinear frequency-based perturbation observer (NDOB) is proposed to estimate wind gusts. This observer considers a limited frequency range of $0.005 - 0.1389Hz$ to isolate other disturbances and focuses on evaluating wind gusts.

2.3.4 Disturbance Due to Suspended Load

Recently UAVs have been used to transport cargo through some delivery and retrieval mechanism. These applications require reducing load swings as they can be fragile, vulnerable, or dangerous. For example, purchased goods, food, medicines, medical samples, vaccines, pesticides, and fertilisers. In particular, a delivery system with UAVs is taking a big boost; it is expected that revolutionise how customers receive purchased goods. Some advantages of using UAVs as a delivery system are reduced shipping costs, less manual supervision and faster delivery times since they do not depend on the roadwork. Aerial manipulation is a complicated task and a challenge in the design of the control algorithm. A quadrotor with cable-suspended load is a multi-input, multiple-output (MIMO) nonlinear system, underacted with eight degrees of freedom and four control inputs. Thus it is necessary to design robust control algorithms capable of adapting to changes in the system's dynamics and minimising the effects caused by them. Recently, the problem of stabilising or minimising the suspended load oscillation by trajectory generation has been addressed in [87, 88]. In [89], the dynamics of the suspended load are modelled. For the design of the control algorithm, the suspended load is taken as an external disturbance that is compensated with a nonlinear control based on the combination of back-stepping and sliding mode with an iterative learning algorithm. In [90], a robust altitude controller based on one-step-ahead predictive is proposed. Simulations are carried out against low-frequency disturbances and payload variations. However, no physical tests are performed to check the controller's robustness. Further to our work on [91], an DO and ESO scheme is proposed to reject cable-suspended payload oscillating disturbance and wind disturbance respectively. Experimental results show that both observers complement each other to estimate and mitigate both disturbances.

Chapter 3

Attitude Control

Our first endeavours are purely instinctive prompting of an imagination vivid and undisciplined. As we grow older reason asserts itself and we become more and more systematic and designing. But those early impulses, though not immediately productive, are of the greatest moment and may shape our very destinies.

– **Nikola Tesla**

In this section, we provide the basis to address the problem of attitude tracking control in VTOL-UAV under the ADRC approach. The objective is to design a robust control capable of following a trajectory, typically generated by a position controller, also called outer-loop, which considers external disturbances and unmodeled dynamics. In addition, an angular rate control algorithm for aerobatic flight is designed to test the ADRC control approach under aggressive manoeuvres such as somersaults, full inversion, rolling and turning. The control algorithms are implemented in an embedded system under the Hardware in the loop approach.

The proposed control scheme contains the following contributions:

1. Robust control to uncertainty, unknown and nonlinear dynamics.
2. Robust control to the aerodynamic effect: vortex ring state, noise, sensor biases and blade flapping.

3. Actuator limitations are ensured by a saturation function.

4. The control algorithm is compatible with embedded systems with low computing power.

The objective of the control law is to design a controller capable of following a varying trajectory in time $\mathbf{q}_d(t)$ while ensuring that the global disturbance $\xi_r(t)$ is limited to a small region of attraction. Mathematically, the above can be expressed as,

$$\mathbf{q}(t) \to \mathbf{q}_d, \omega(t) \to \omega_d(t), \|\xi_r(t)\| \to \varepsilon, \text{according} \quad t \to \infty \tag{3.1}$$

The purpose of the proposed attitude controller is to stabilize $\mathbf{q}(t) \in S^3$ and $\omega(t) \in \mathbb{R}^3$ at the desired reference $\mathbf{q}_d(t) \in S^3$, $\omega_d(t) \in \mathbb{R}^3$ respectively, while the global disturbance $\xi_r(t) \in \mathbb{R}^3$ approaches zero or a very small ε value. $\xi_r(t)$ is a global disturbance; it will be defined later.

3.1 Attitude Modeling with External Disturbance

In this section, a more in-depth analysis is carried out on the non-linear dynamics of the aircraft, taking into account endogenous (dependent on state variables) and exogenous disturbances (environment dependent).

$$\Sigma_R : \begin{cases} \dot{\mathbf{q}} = \frac{1}{2}\Xi(\mathbf{q})\omega \\ \mathbf{J}\dot{\omega} = -[\omega^{\times}]\mathbf{J}\omega + \tau - G_a + \mathbf{R}fe_{3b} + \zeta_r(t) \end{cases} \tag{3.2}$$

The inertial matrix $\mathbf{J}$ is square and not symmetrical. The vector $\mathbf{R}Fe_{3b}$ is the effect produced by the VTOL-UAV centre of mass, and the origin O_B is not coinciding. So the thrust vector f creates a torque because it is not applied precisely at the aircraft's centre of mass. $\zeta_r(t)$ is a time-dependent disturbance, possibly due to noise, sensor bias, aerodynamic effects, wind, friction and uncertainty dynamics. The vector $G_b = [G_1,\ G_2,\ G_3]^T \in \mathbb{R}^3$ represents the gyroscopic torques caused by the rotation of the propellers, which is given by

$$G_a = \sum_{i=1}^{4} J_r(\omega \times e_{3b})(-1)^{i+1}\Omega_i \tag{3.3}$$

where J_r denotes the moment of inertia of the rotor. For the purposes outlined in this section, it is convenient to deduce the dynamics of the error. Defining the angular velocity error as $\omega_e = \omega - \omega_c$, where $\omega_c = \mathbf{R}^T(\mathbf{q}_e)\omega_d$ is the commanded angular velocity obtained from the outer loop.
Obtaining the derivative with respect to time of ω_e,

$$\dot{\omega}_e = \dot{\omega} - \mathbf{R}^T(\mathbf{q}_e)[\omega_e^{\times}]\omega_d - \mathbf{R}^T(\mathbf{q}_e)\dot{\omega}_d \tag{3.4}$$

Substituting (3.4) in (3.2), the system error dynamics are given by

$$\Sigma_{R_e} : \begin{cases} \dot{\mathbf{q}}_e = \frac{1}{2}\Xi(\mathbf{q}_e)\omega_e \\ \mathbf{J}\dot{\omega}_e = -[(\omega_e + \omega_r)^{\times}]\mathbf{J}(\omega_e + \omega_r) + \mathbf{J}[\omega_e^{\times}]\omega_c - \mathbf{J}\mathbf{R}^T(\mathbf{q}_e)\dot{\omega}_d - G_a + \mathbf{R}fe_{3b} + \zeta_r(t) + \tau \end{cases} \tag{3.5}$$

Attitude control will be designed for the use of commercial autopilots. Since they use Euler angles to represent their attitude, it is necessary to have a mechanism that allows them to transform into quaternions Let us consider the attitude and angular error variables as

$$\begin{aligned} \Theta_e &= \Theta - \Theta_c \\ \omega_e &= \omega - \omega_c \end{aligned} \tag{3.6}$$

where $\Theta_c = [\phi_c,\, \theta_c,\, \psi_c]^T \in \mathbb{R}^3$ and are $\omega_c = [\omega_{c1},\, \omega_{c2},\, \omega_{c2}]^T \in \mathbb{R}^3$ the desired attitude and angular velocity varying in time, obtained from the outer loop.
Let us propose the following desired angular velocity,

$$\omega_d = \mathbf{W}^{-1}\left(\dot{\Theta}_d - \kappa_1\left(\Theta - \Theta_d\right)\right) \tag{3.7}$$

where κ_1 is a positive constant. The proposed angular velocity ω_d can be seen as a virtual control, which achieves that the system reaches the desired reference signal $\Theta(t)$ in a smooth and fast way. To demonstrate the above, consider the following candidate function of Lyapunov,

$$V = \frac{1}{2}\boldsymbol{\Theta}_\mathbf{e}{}^T\boldsymbol{\Theta}_\mathbf{e} \tag{3.8}$$

The derivative of the Lyapunov function is given by

$$\dot{V} = \mathbf{\Theta}_e^T \left(\mathbf{W}\omega - \dot{\Theta}_d \right) \tag{3.9}$$

If we make ω converge to 3.21, then obtain $\dot{V} < 0$.

$$\dot{V} < -\mathbf{\Theta}_e^T \kappa_1 \mathbf{\Theta}_e \tag{3.10}$$

Then, the following equation describes the error's dynamics, which will be used in the control algorithm.

$$\Sigma_{R_e} : \begin{cases} \dot{\Theta}_e & = \mathbf{W}\omega_e \\ \mathbf{J}\dot{\omega}_e & = -[(\omega_e + \omega_d)^\times]\mathbf{J}(\omega_e + \omega_d) - \mathbf{W}^{-1}[\omega_e^\times]\dot{\Theta}_d + \mathbf{W}^{-1}\ddot{\Theta}_d + \\ & = +\kappa_1 \mathbf{W}^{-1}[\omega_e^\times]\Theta_e - \kappa_1 \mathbf{W}^{-1}\dot{\Theta}_e - G_a + \mathbf{R}fe_{3b} + \zeta_r(t) + \tau \end{cases} \tag{3.11}$$

3.2 ADRC Controller for Attitude Tracking

Efficient attitude control is crucial to maintaining a desired orientation for the VTOL-UAV to achieve the desired position despite external disturbances and unmodeled dynamics. The proposed control scheme is based on the ADRC methodology. As mentioned, an ESO is designed to estimate endogenous and exogenous shocks online and subsequently cancel the term through feedback action. The control law considers the physical constraints of the VANT-UAV and the rotors. The control law is limited using a saturation function, avoiding damage to the rotors when they become saturated and maximising its effectiveness; in other words, the control action is bounded in the range $\tau \in [-\bar{\tau}, \bar{\tau}]$.

To address the ADRC control design, consider the error dynamics given by (3.11). Let's define a global disturbance that encompasses endogenous and exogenous disturbances as

$$\begin{aligned} \xi_r(t) &= -[(\omega_e + \omega_d)^\times]\mathbf{J}(\omega_e + \omega_d) - \mathbf{W}^{-1}[\omega_e^\times]\dot{\Theta}_d + \mathbf{W}^{-1}\ddot{\Theta}_d + \\ &= \kappa_1 \mathbf{W}^{-1}[\omega_e^\times]\Theta_e - \kappa_1 \mathbf{W}^{-1}\dot{\Theta}_e - \zeta_r(t) + \mathbf{R}fe_{3b} \end{aligned}$$

The term $\xi_r(t)$ will be estimated through ESO; then the following assumptions will be made,

- The total disturbance estimation, angular velocity estimation and its time derivative will be denoted by $\hat{\xi}(t)$, $\hat{\omega}_d(t)$ and $\hat{\dot{\omega}}_d(t)$ respectively;
- The perturbation function $\xi(\cdot)$ is a uniformly absolutely bounded disturbance, *i.e.*, $\sup_t \|\xi(\cdot)\| = \|\xi(\cdot)\|_\infty \leq K_0$.
- $\hat{\omega}_d(t)$ is uniformly absolutely bounded first-order time derivative, $\hat{\dot{\omega}}_d(t)$, *i.e.*, $\sup_t \|\hat{\omega}_d(t)\| \leq K_0$,$\sup_t \|\hat{\dot{\omega}}_d(t)\| \leq K_1$;

3.2.1 State Observer Extended Design for Attitude

Consider the attitude error system of the quadrotor equation (3.11), then define the following state variables $\hat{\omega}_e$, $\hat{\xi}_r(t)$ and $\dot{\hat{\xi}}_r(t)$, as the estimated angular velocity, estimated disturbance and the time derivative of the estimated disturbance respectively. Then we propose the following extended state observer of the ESO to estimate the global disturbance.

$$\Sigma_{ESO_e} : \begin{cases} \dot{\hat{\omega}}_e = \mathbf{J}^{-1}\tau + \hat{\xi}_r(t) + l_{r2}\left(\omega_e - \hat{\omega}_e\right) \\ \dot{\hat{\xi}}_r = \dot{\hat{\xi}}_r(t) + l_{r1}\left(\omega_e - \hat{\omega}_e\right) \\ \ddot{\hat{\xi}}_r = l_{r0}\left(\omega_e - \hat{\omega}_e\right) \end{cases} \tag{3.12}$$

where l_{r2}, l_{r1} and $l_{r0} \in \mathbb{R}_+$ The terms l_{r2}, l_{r1} and l_{r0} are selected to be a Hurwitz polynomial of the third order,

$$\begin{aligned} l_{r2} = & \quad -\tfrac{1}{\epsilon}\left(p_0 + 2\varsigma_0\omega_{n0}\right), \\ l_{r1} = & \quad -\tfrac{1}{\epsilon^2}\left(2p_0\varsigma_0\omega_{n0} + \omega_{n0}^2\right), \\ l_{r0} = & \quad -\tfrac{1}{\epsilon^3}\left(\omega_{n0}^2 p_0\right), \end{aligned}$$

with p_0, ς_0 and ω_{n0} are positive values, ϵ is a very small value for the fast convergence of ESO estimated variables.

3.2.2 Robust Attitude Tracking Controller Design

In this subsection, a robust attitude-tracking controller is designed based on the ADRC. From the previous analysis, the proposed desired angular velocity ω_d in the equation (3.21), can be seen as a

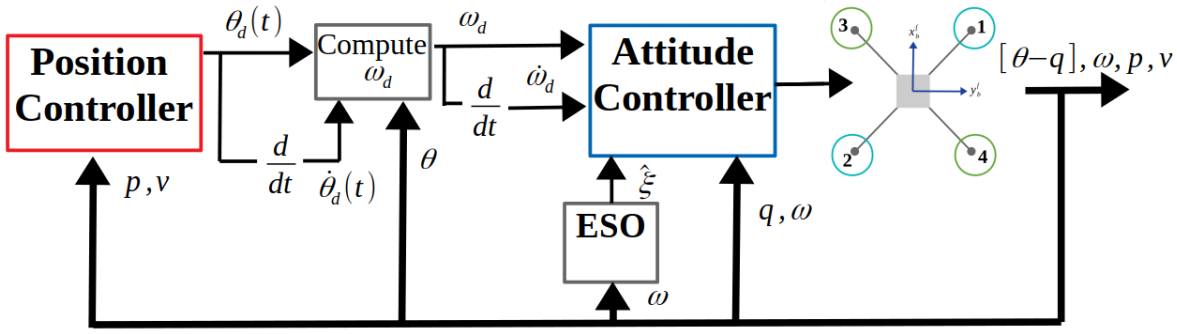


Fig. 3.1: Robust Attitude Controller quadrotor.

virtual control which achieves that the system reaches the desired reference signal $\Theta(t)$ is a smooth and fast way. The proposed control scheme is made up of three elements. An attitude path tracking controller, an extended state observer that estimates global disturbance, and a term calculates the desired angular velocity and acceleration see Fig. 3.1.

Definition 3.2.1 *Given a positive constant a, continuous, nondecreasing function* $\sigma_a(s) : \mathbb{R} \to \mathbb{R}$ *is defined by*

$$\sigma_a(s) = \begin{cases} s, & |s| \leq a \\ a \cdot sign(s), & |s| > a \end{cases} \tag{3.13}$$

Assuming a sufficiently long time, the ESO estimates the term $\xi_r(t)$, then the following control law is designed by feeding-back the estimated dynamics,

Proposition 3.2.2 *Consider the error dynamics described by* (3.11) *with the following bounded control input vector* τ *such that*

$$\tau_i = -\sigma_{M_2}\left(-J\dot{\omega}_d + \hat{\xi} + \sigma_{M_1}(\bar{\lambda}[\omega_e + \rho q_e])\right) \tag{3.14}$$

with $i \in \{1, 2, 3\}$ *and where* $\sigma_{M_{i1}}$ *and* $\sigma_{M_{i2}}$ *are saturation functions such that* $K_{0_i} < M_{i2} - M_{i1}$ *and* $M_{1i} \geq 3\bar{\lambda}_i\rho_i$. *Then the inputs* (3.14) *stabilize robustly* (2.14) *to the origin of the error space* $(1\ 0^T\ 0^T)^T$, *with a domain of attraction equal to* $\mathbb{S}^3 \times \mathbb{R}^3 \setminus (-1\ 0^T\ 0^T)^T$.

Nevertheless, the control algorithm proposed above requires the numerical calculation of the derivative of $\dot{\Theta}_d$ and $\dot{\omega}_d$, which is a costly operation in processor time, subject to inaccuracy and noise.

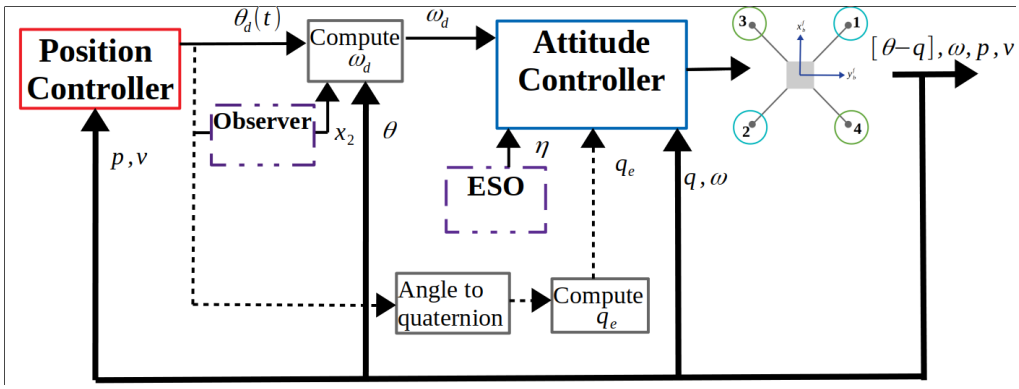


Fig. 3.2: Control Algorithm Optimized Scheme for the Attitude of Quadrotor.

To solve this problem, we propose the following high-gain observer ESO.

$$\Sigma_{ESO_{wd}} : \begin{cases} \dot{x}_1 = x_2 + \lambda_3 \left(\Theta_d - x_1\right) \\ \dot{x}_2 = \lambda_2 \left(\Theta_d - x_1\right) \end{cases} \tag{3.15}$$

Where $x_1 = \Theta_d$ and $x_2 = \dot{\Theta}_d$ are the attitude desired, and the derivative of the attitude desired estimated respectively. When using the dynamics of the error, it is no longer necessary to feedback the term feedforward $-J\dot{\omega}_d$ since the ESO estimates it, see Fig. 3.2.

Then the proposed control algorithm is as follows,

$$\tau_i = -\sigma_{M_2} \left(\hat{\xi} + \sigma_{M_1} (\bar{\lambda}[\omega_e + \rho q_e]) \right) \tag{3.16}$$

like the previous one, it robustly stabilizes the origin of the error space 3.2.2.

3.2.3 Simulations

The results obtained from the simulation of the proposed algorithm are presented. The table 3.1 shows the values used in the simulation. The simulations were carried out under a parametric uncertainty of 35 per cent of the aircraft's parameters: mass and inertia. The global disturbance $\zeta_r(t)$ has been modelled as a uniformly bounded function varying in time,

$$\zeta(t) = 1.5 \cos 50t \sin 5t \sin 0.2t \tag{3.17}$$

The function $\zeta_r(t)$ can represent noise, unmodeled dynamics, and the moment generated by the suspended load. The Fig. 3.3 shows the evolution of the modeled disturbance.

Table 3.1: Physical parameters of the quadrotor.

Parameters	Description	Units
g	Gravity	9.81 m/s^2
m	Mass	1.033 kg
J_{xx}	Inertia in the x axis	0.00653 $kg\ m^2$
J_{yy}	Inertia in the y axis	0.00653 $kg\ m^2$
J_{zz}	Inertia in the z axis	0.00978 $kg\ m^2$

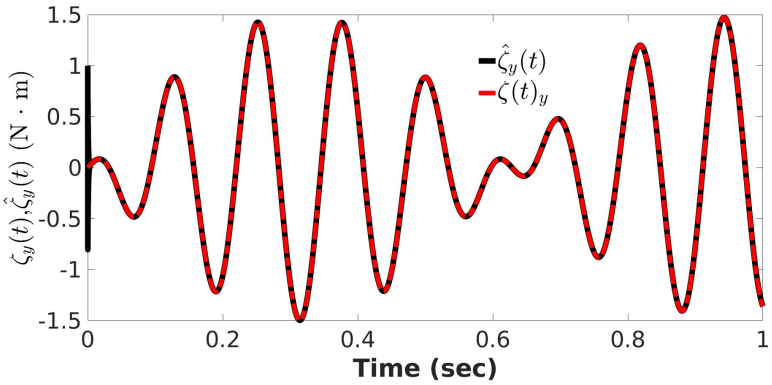


Fig. 3.3: Evolution of the external disturbance $\xi_r(t)$ and its estimates $\hat{\xi}_r(t)$.

The value of the integral squared error ISE(t) is used to demonstrate numerically the degree of estimation of the disturbance added to the mathematical model. A value close to zero indicates rapid convergence and accuracy in the estimate. The ISE is defined by the sum of the errors to consider,

$$ISE(t) = \int_0^t \left[(\omega_y(\sigma) - \hat{\omega}_y(\sigma))^2 + (\xi y(\sigma) - \hat{\xi}_y(\sigma))^2\right] d\sigma \tag{3.18}$$

As can be seen in Fig. 3.4 the value of the integral square error ISE(t) of the total disturbance estimate (3.18) indicates a constant value very close to zero, which indicates a rapid estimate of the global disturbance $\hat{\xi}_y$ and redundant estimate angular velocity $\hat{\omega}_y$.

A parameterised path is proposed using desired reference points. Fig. 3.5 shows the desired way parameterised by the reference points measured in meters; the time is implicit in each reference point. The position control algorithm is neglected and taking it up again in the next chapter.

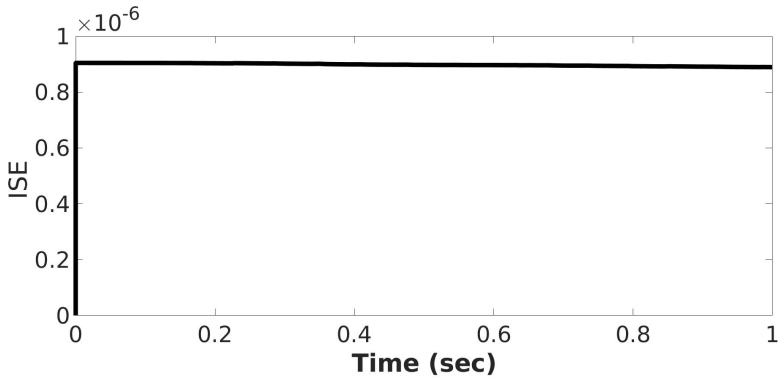


Fig. 3.4: Evolution of ESO via integral square error ISE.

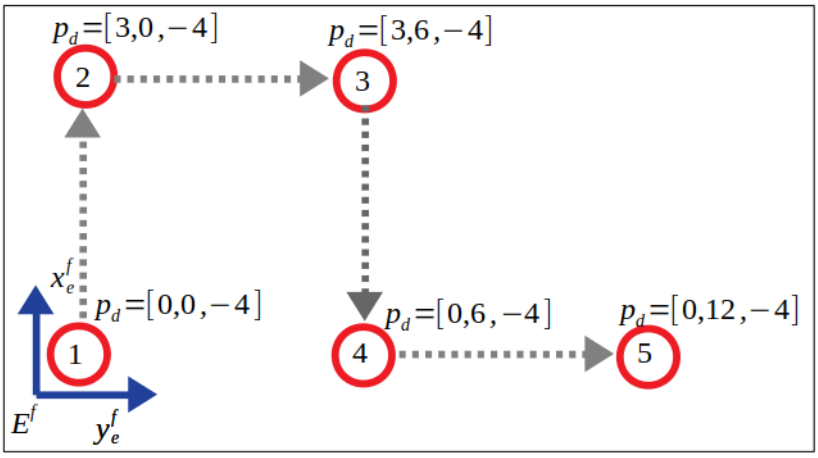


Fig. 3.5: The desired trajectory of the quadrotor, parameterised by waypoints.

We compare the proposed control with a saturated derivative proportional control. This comparison allows us to evaluate ESO's performance and effectiveness when incorporated into feedback. The Fig. 3.6 shows the evolution of the attitude in both controllers. Control A is the proposed control, and control B is the standard PD control. As shown in Fig. 3.6 and Fig. 3.9 the evolution of the angle in the proposed control is smooth and without so much oscillation. The proposed algorithm observes an attenuation of approximately 0.2 rad.

In the same way, Fig. 3.7 and 3.8 show the θ_d and ψ_d in both controllers. The proposed control shows a better response; it can also reject unmodeled dynamics, noise and external disturbances. The Figs. 3.9,3.10, and 3.11 show the evolution of the angular velocity in both controllers. An attenuation of approximately 1 $\frac{rad}{s}$ is observed in the proposed algorithm.

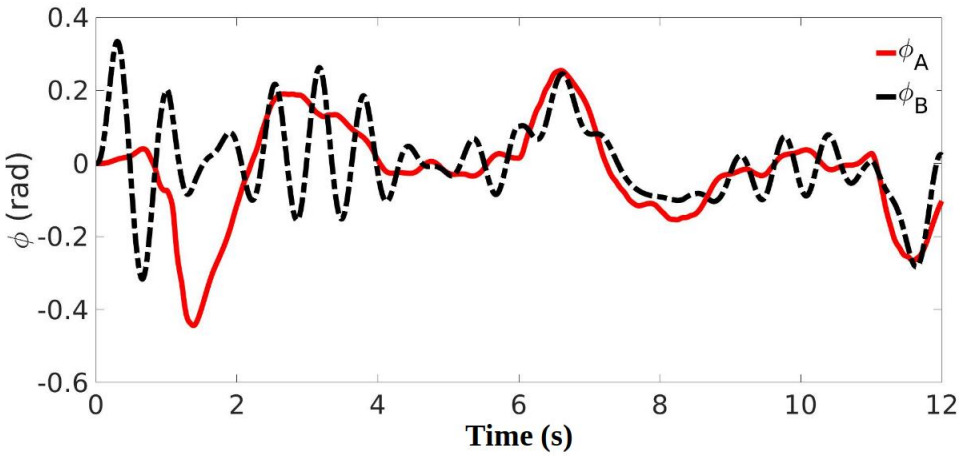


Fig. 3.6: Evolution of the orientation of the UAV parameterised in Euler angles, $\phi_d(t)$.

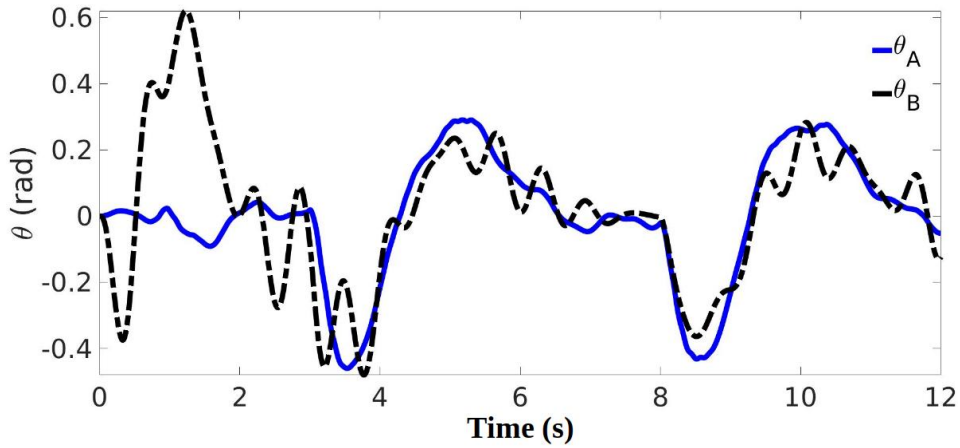


Fig. 3.7: Evolution of the orientation of the UAV parameterized in Euler angles, $\theta_d(t)$.

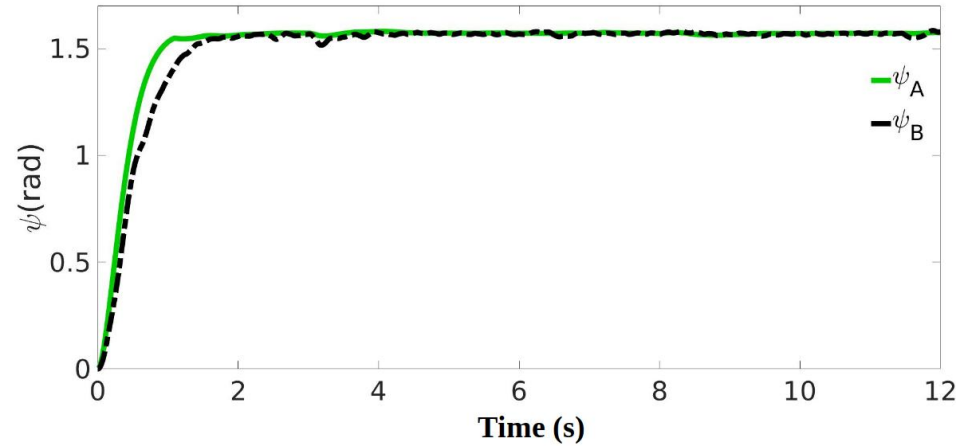


Fig. 3.8: Evolution of the orientation of the UAV parameterized in Euler angles, $\psi_d(t)$.

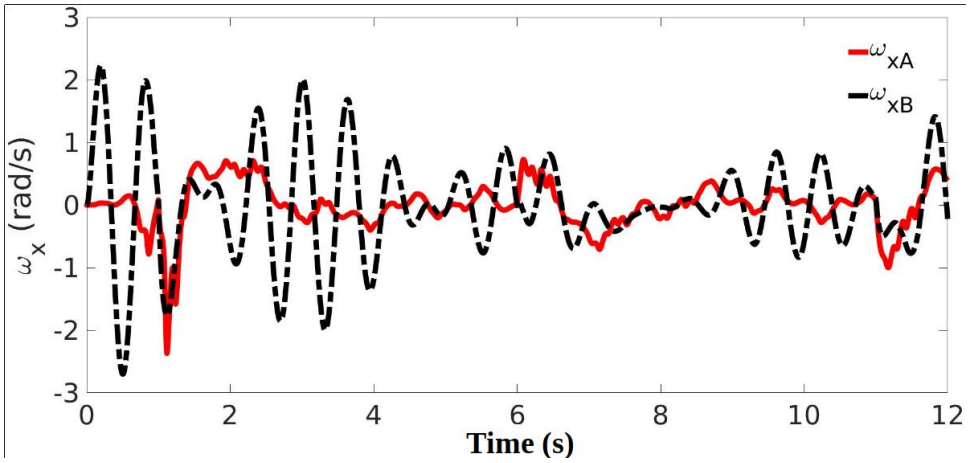


Fig. 3.9: Evolution of the angular velocity of the UAV , $\omega_1(t)$.

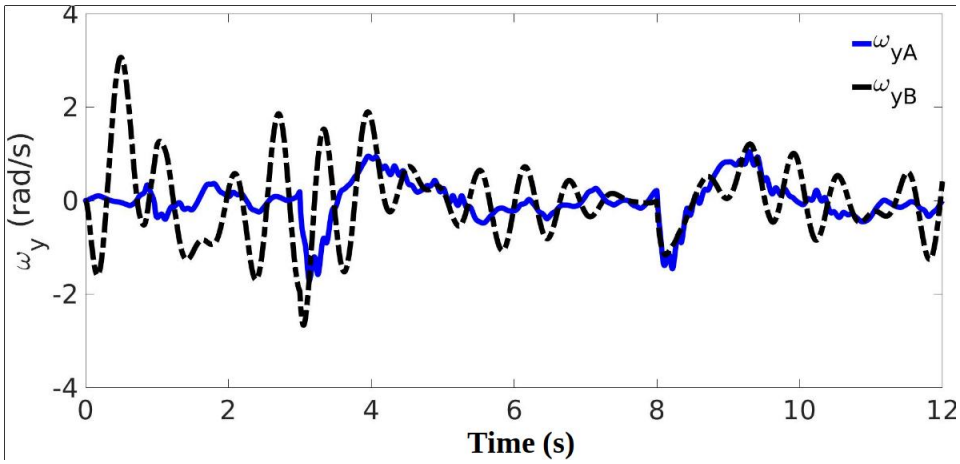


Fig. 3.10: Evolution of the angular velocity of the UAV , $\omega_2(t)$.

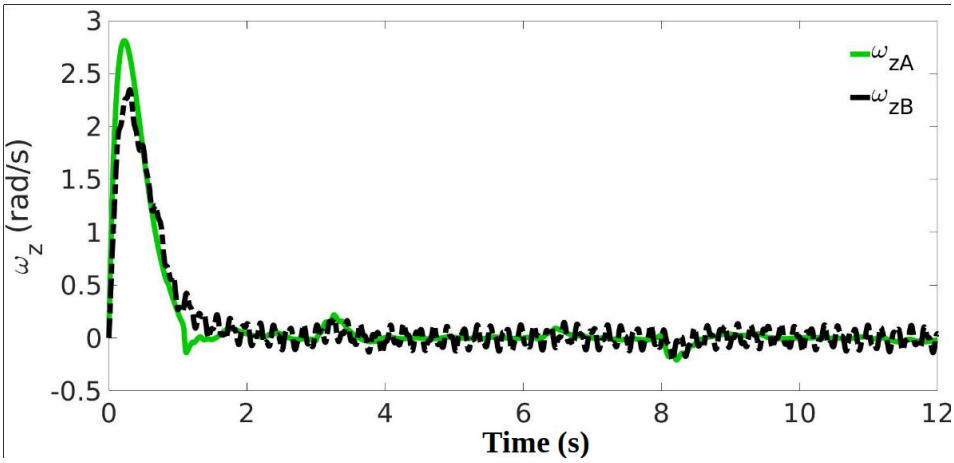


Fig. 3.11: Evolution of the angular velocity of the UAV , $\omega_3(t)$.

3.3 Attitude Control for Aggressive Maneuvers

This section features angular velocity control robust to unmodeled dynamics and external disturbances with the ability to follow aggressive references to generate aerobatic flight. Aggressive manoeuvres include cartwheels, full inversions, rolls, and loops. In particular, the generation of N flight over the pitch angle is formulated. In this (aerobatic) flight mode, the angular rotation rate around the axes is controlled instead of the attitude. The algorithm is implemented in an embedded system under the SITL software simulation approach. The structure of the proposed control algorithm is divided into four blocks, Attitude Model Kinematic, ESO, Attitude ADRC Controller and Control Allocator, as shown in Fig. 3.12.

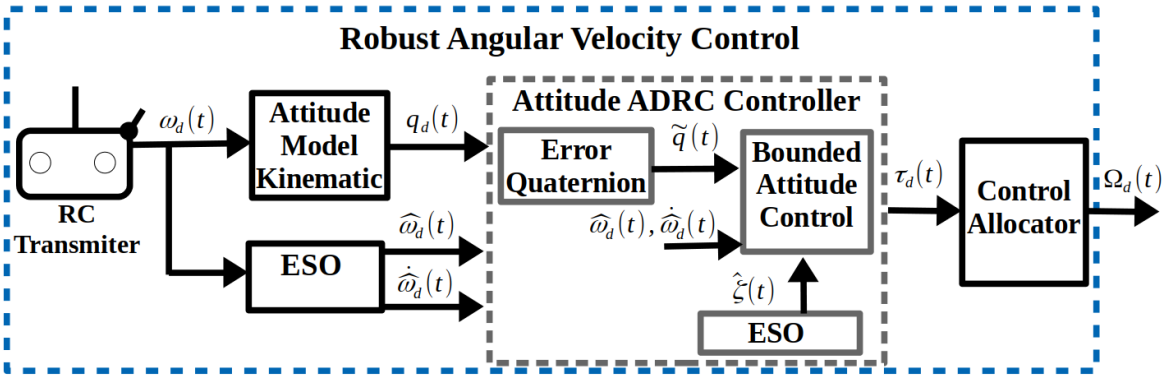


Fig. 3.12: Robust Angular Velocity Controller.

The desired angular velocity $\omega_d(t)$ is provided by an RC transmitter, then the desired quaternion $q_d(t)$ is obtained using the kinematic equation (2.11), concurrently an Extended State Observer ESO estimate the angular acceleration $\dot{\omega}_d(t)$ and angular velocity $omega_d(t)$. The $q_d(t)$, $\dot{\hat{\omega}}_d(t)$ and $\hat{\omega}_d(t)$ variables are used to calculate the control torques $\tau_d(t)$ by an internal control block that stabilises the orientation trajectory tracking; in this block, the ADRC is used in conjunction with bounded quaternion-based feedback. Finally, an allocator controller calculates the speeds of the rotors $\Omega_d(t)$ to generate the desired control torques $\tau_d(t)$.

For aggressive manoeuvres, the approach to small angles is not justifiable; on the contrary, there is a need for sudden changes in orientation, velocity and angular acceleration. Therefore, it is necessary to calculate and consider the feed-forward terms on the desired acceleration in the law control.

Since the desired angular velocity $\omega_d(t)$ commanded by the RC transmitter may not be smooth, we

use an extended state observer to estimate $\omega_d(t)$ and the angular acceleration $\dot{\omega}_d(t)$.

Consider the following ESO observer:

$$\Sigma_{ESO} := \begin{cases} \dot{\hat{\omega}}_d = \dot{\hat{\omega}}_d + \Lambda_1 \left(\omega_d - \hat{\omega}_d\right) \\ \ddot{\hat{\omega}}_d = \Lambda_0 \left(\omega_d - \hat{\omega}_d\right) \end{cases} \tag{3.19}$$

where $\Lambda_0 = \omega_n^2$ and $\Lambda_1 = 2\xi\omega_n$, also ξ and ω_n are the damping ratio and the natural frequency respectively.

The attitude ADRC controller is structured by three blocks, the error quaternion, the bounded attitude control and the disturbance observer. To calculate the angular velocities of the rotors $\Omega_i \in \{1, 2, 3, 4\}$, the inverse matrix (3.20) is used. However, the following normalised matrix is proposed, which has advantages in calculating pulse width modulation signals from rotors.

$$\begin{pmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{pmatrix} = \begin{pmatrix} 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} f_d/k_F \\ \tau_x/k_F \cdot L \\ \tau_x/k_F \cdot L \\ \tau_x/k_M \end{pmatrix} \tag{3.20}$$

The k_F, k_M and L are set unknown; the control law will compensate them.

3.3.1 Trajectory Generation and Control of Multiple-Flips

In this subsection, we design a module for the trajectory generation of aggressive manoeuvres. In particular, we address the problem of multiple flips around an axis of the VTOL-UAV. The issue of generating consecutive turns on an axis of the VANT-UAV is broken down as a different generation of trajectory and robust trajectory tracking controller. The trajectory requires rapid angular velocity stabilisation that evolves dramatically with acceleration and deceleration in a short time. The robustness of the proposed ADRC is demonstrated by faithfully ensuring the following of the desired reference against external disturbances and unpredictable dynamics during high-speed manoeuvring. Both the trajectory generation module and the proposed controller are programmed into the Pixhak autopilot. Therefore, all processing is carried out on board using the IMU inertial measurement system built into the Pixhawk. This avoids the need to use motion capture systems

like Optitrack or Vicon and the high instrumentation that entails.

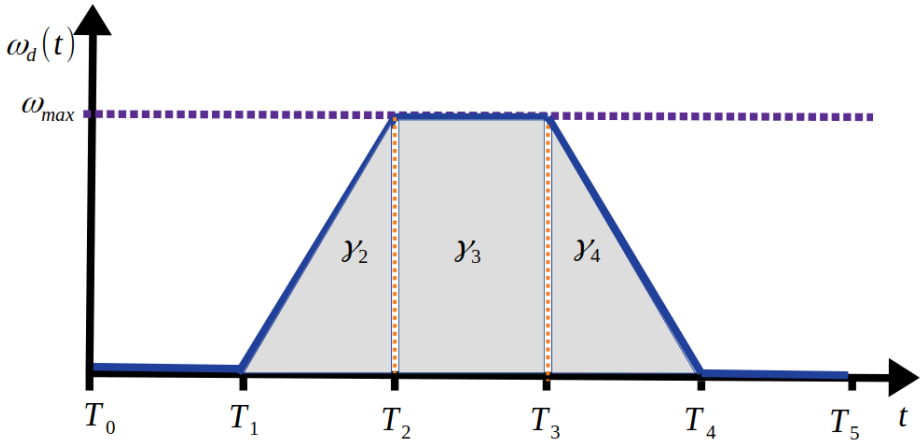


Fig. 3.13: Desired trajectory to generate multiple spins ω_d(t).

The Pixhawk mini contains an MPU9250 (3-Axis gyroscope and the 3-Axis accelerometer); the angular rate sensor can supply an angular velocity of up to: 35 rad/sec. We have proposed a bounded angular momentum ω_{max} 70 per cent of the capacity of the IMU sensor. The proposed method generates a margin of control for possible parametric uncertainties, sensor noise and external disturbance. As shown in Fig. 3.13, the trajectory is divided into five states: elevation, increase, top, decrease and hover. We use the increment-type throttle to control the total thrust. The stick is set at 90 per cent in the elevation state and 60 per cent in the other manoeuvres.

1. **Elevation state.** The quadrotor accelerate up to gain enough height with respect to the inertial frame to avoid hitting the ground while performing the entire maneuver.

2. **Increase state.** The quadrotor increases its angular velocity linearly until it reaches ω_{max} in T_1 seconds.

3. Stop state: The angular velocity is set at zero.

4. **Decrease state.** The quadrotor begins to fall with an angular velocity $-\omega_{max}$, it decrease linearly until it becomes zero in T_4 seconds.

5. **Hover state.** The quadrotor remains static with an angular velocity equal to zero.

The desired angular velocity $\omega_d(t)$ that makes the drone rotate on the axis and is defined by the following equation:

$$\omega_d(t) = \begin{cases} 0 & T_0 \leq t < T_1 \\ \frac{\omega_{max}}{\Delta_2}(t - T_0) & T_1 \leq t < T_2 \\ \omega_{max} & T_2 \leq t < T_3 \\ -\frac{\omega_{max}}{elta_4}(t - T_3) + \omega_{max} & T_3 \leq t < T_4 \\ 0 & T_4 \leq t < T_5 \end{cases} \tag{3.21}$$

where $\Delta T_i = T_{i-1} - T_i$.

The variable γ_i represents the area limited by the graph of the $\omega_d(t)$, is represented by the definite integral (3.22).

$$\gamma_i = \int_{T_{i+1}}^{T_i} \omega_{d,i}(t) \cdot dt \tag{3.22}$$

The sum of the total area of the desired angular velocity denotes the amount of rotation accumulated in the maneuver. Mathematically the above can be written as

$$\sum_{i=1}^{5} \gamma_i = 2\eta\pi \tag{3.23}$$

Setting a priori $n, \omega_{max}, \Delta_1, \Delta_5$ using the equations (3.21) (3.22), (3.23), and $\Delta_2 = \Delta_4$ solve for T_2, T_3 and T_4. Consequently the maneuver is completely defined.

Considering the case of a double flip, the following parameters are obtained: $\omega_{max} = 26 rad/sec$, $T_0 = 0$, $T_1 = 0.5$, $T_2 = 0.7$,$T_3 = 0.0983$,$T_4 = 1.183$, $T_5 = 1.6$.Considering the case of a double flip, the following parameters are obtained: $\omega_{max} = 26 rad/sec$, $T_0 = 0$, $T_1 = 0.5$, $T_2 = 0.7$,$T_3 = 0.0983$,$T_4 = 1.183$, $T_5 = 1.6$.

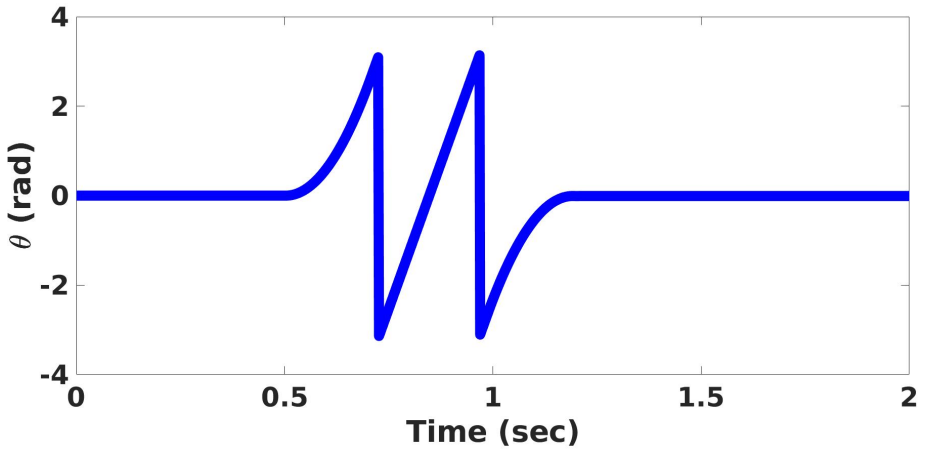


Fig. 3.14: Attitude a conversion is made to Euler's angles for a straightforward interpretation.

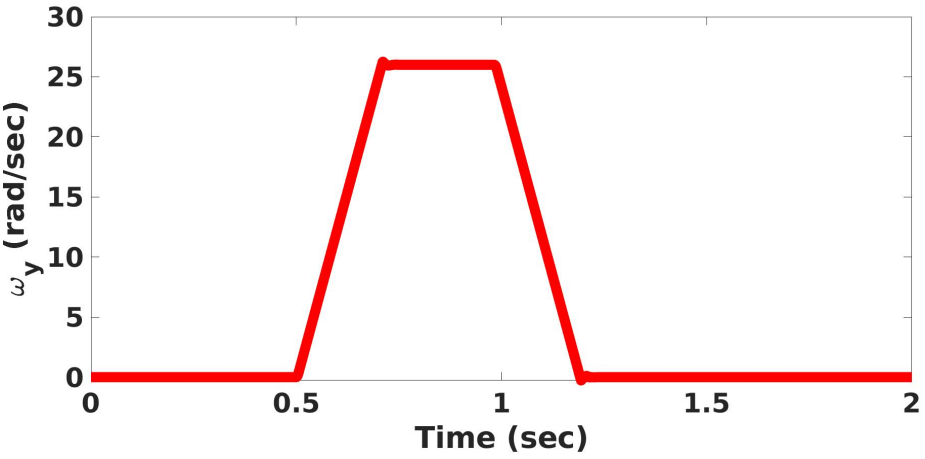


Fig. 3.15: Angular velocity evolution.

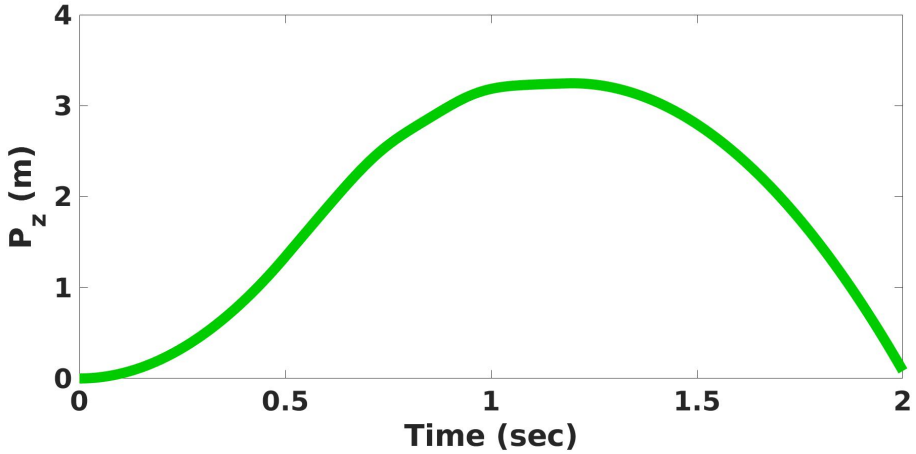


Fig. 3.16: Altitude position evolution.

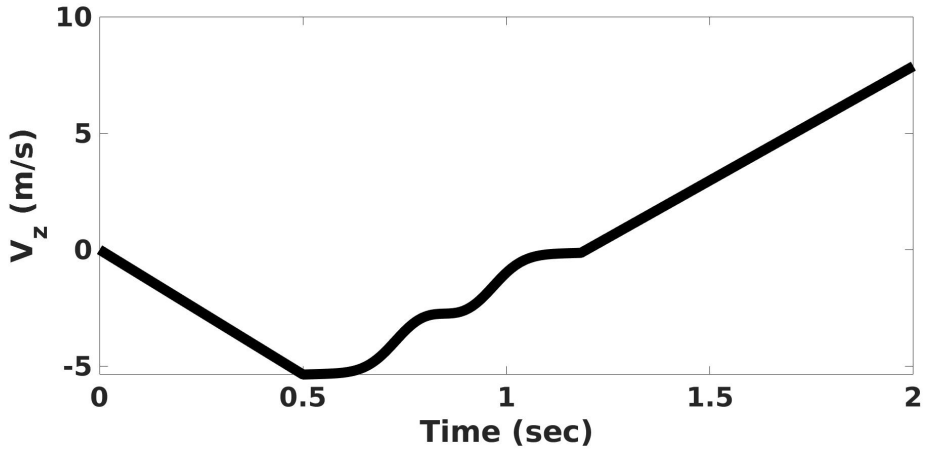


Fig. 3.17: Linear velocity in the z-axis.

The Figs. 3.14, 3.15, 3.16 and 3.17 show the evolution of the UAV states attitude, angular velocity, altitude and linear velocity respectively. The control algorithm follows the reference despite external disturbances and uncertain parameters. There is no oscillation and the response is smooth and bounded. A video of the operation of the orientation control implemented in the Pixhawk autopilot is shown in, `https://www.youtube.com/watch?v=95VFD91TFrw`.

Chapter 4

Robust Position Tracking with Suspended Load

It is not knowledge, but the act of learning, not possession but the act of getting there, which grants the greatest enjoyment.

– **Carl Friedrich Gauss**

This section addresses the problem of controlling the transport of a load from one point to another via cable through a quadrotor, eliminating the oscillation generated by the suspended load. Robust non-linear control is featured, which asymptotically stabilises the desired quadrotor position while simultaneously limiting load swing and aligning it to the vertical position of the inertial reference frame. The proposed control scheme considers the dynamics generated by the suspended load to generate the control law. Unlike the approaches used in the scientific literature, no feedback is needed on the dynamics of the suspended load; an extended state observer estimates these dynamics. The simulation results show that the proposed control can be helpful for the safe transport of the payload.

4.1 Introduction

Recently UAVs have been used to transport cargo through some delivery and retrieval mechanism. These applications require reducing load swings as they can be fragile, vulnerable, or dangerous. Aerial manipulation is a complicated task and a challenge in the design of the control algorithm. The controller can be divided into two categories: the first focuses on eliminating load swing, and the second is the generation of trajectories considering the system's dynamics. We focus on the first category, a robust control capable of stabilising the position of the load and, at the same time minimising its oscillation.

The main points of our contribution are:

1. Robust control stabilises the UAV position while simultaneously limiting load swing and aligning it to the vertical position of the inertial reference frame.
2. The controller proposed guarantees asymptotical convergence of the error variables in the presence of uncertainties.
3. The control design takes into account the dynamics of the suspended load.
4. The system's dynamics are not linearised around its equilibrium point.
5. You don't need to give feedback on the dynamics of the suspended load attached to the UAV.
6. An ESO is added to the position control.
7. The control algorithm takes into account the limits of thrust, angle and maximum torque allowed by the UAV using saturation functions.
8. The algorithm is relatively easy to implement in an embedded system.

4.2 ADRC Controller for a Quadrotor with Cable-Suspended Load.

This section, we describe the effect a suspended load has on the quadrotor. For their analysis, we consider only the planar dynamics shown in Fig. 4.1, and the remaining dynamics are stabilised

separately. The author believes that a complete model does not add a contribution. Next, propose a control algorithm to minimise the effect of the suspended load. Let us consider two orthogonal coordinate frames; the inertial frame $E^i = [O, i, k]$ where vectors i and k point north and up, respectively. The body frame $E^b = [O, x, z]$ is attached to the UAV's centre of mass. The angle θ is the rotation of the inertial angle to the body frame. An additional frame is entered, denoted as load frame $E^l = [O, x_l, z_l]$; it is rotated concerning the body frame by an angle α. L represents the cable of length connecting M to the suspend load. The UAV is actuated by two motors generating forces $f_1 \in \mathbb{R}_{\geq}$ and $f_2 \in \mathbb{R}_{\geq}$ where $\mathbb{R}_{\geq} = \{\mathbb{R} \geq 0\}$.

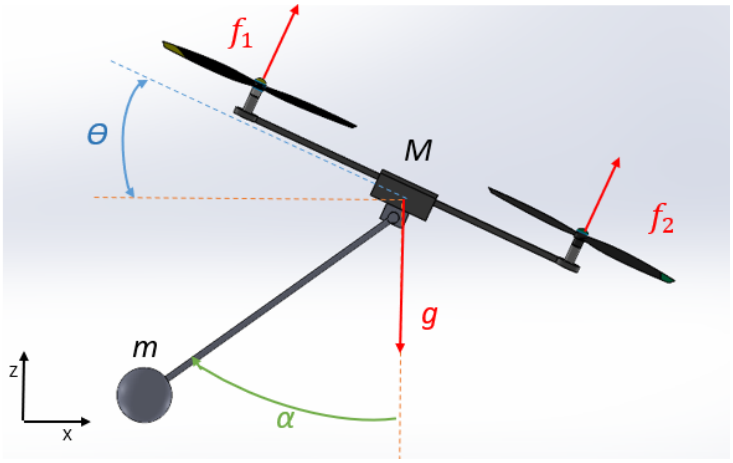


Fig. 4.1: Two-dimensional model of a UAV transporting a suspended load.

The system inputs are $u_1 = f_1 + f_2$ and $u_2 = (f_1 - f_2)b$, where b is the distance between the motors and the centre of the UAV. The work inspires the modeling of the helicopter system with the suspended load carried out in [92, 93].

$$\Sigma_Q \quad : \quad \begin{cases} (M+m)\ddot{y} = u_1 \cos\theta - mL(\cos\alpha\dot{\alpha}^2 + \sin\alpha\ddot{\alpha}) - (M+m)g \\ (M+m)\ddot{x} = u_1 \sin\theta - mL(\sin\alpha\dot{\alpha}^2 - \cos\alpha\ddot{\alpha}) \\ J\ddot{\theta} = u_2 \end{cases} \tag{4.1}$$

$$\Sigma_L \quad : \quad \left\{ mL^2\ddot{\alpha} = -mL\sin\alpha\ddot{z} + mL\cos\alpha\ddot{x} - mgL\sin\alpha \right. \tag{4.2}$$

The objective of the control law is to design a controller capable of following a varying trajectory in time $p_d(t)$ while ensuring that the global disturbance $\xi_t(t)$ is limited to a small region of attraction. Mathematically, the above can be expressed as,

$$p(t) \to p_d, \check{(t)} \to v_d(t), \|\xi_t(t)\| \to \varepsilon, \text{according} \quad t \to \infty \tag{4.3}$$

The purpose of the proposed position controller is to stabilize $p(t) \in R^3$ and $\mathbf{v}(t) \in \mathbb{R}^3$ at the desired reference $p_d(t) \in S^3$, $v_d(t) \in \mathbb{R}^3$ respectively, while the global disturbance $\xi_t(t) \in \mathbb{R}^3$ approaches zero or a very small ε value. $\xi_{,t}(t)$ is a global disturbance, it will be defined later. This work aims to stabilise the quadrotor's desired position by minimising the suspended load's oscillation; that is, the part of the load is stabilised to the vertical axis of the inertial frame while the quadrotor moves to the desired position. In previous works cited, the different sensors allowed to estimate the position and speed of the suspended load; in this work, an extended state observer assesses the dynamics of the suspended load.

(5.19).

$$\mathbf{q}(t) \to \mathbf{q}_d, \ \dot{q}(t) \to 0, \|\alpha(t)\| \to \varepsilon, \text{according} \quad t \to \infty. \tag{4.4}$$

where $q = \begin{bmatrix} z & x & \theta \end{bmatrix}^T \in \mathbb{R}^3$, ε is a number approximately equal to zero.

The control algorithm for the UAV position has been divided into the control for the vertical dynamics and the control for the horizontal dynamics.

4.2.1 Altitude Control for Quadrotor with Cable-Suspended Load

Consider the vertical dynamics of the UAV, then defining $\xi_z(\mathrm{t}) = -mL\left(\cos\alpha\ddot{\alpha}^2 + \sin\alpha\ddot{\alpha}\right) - (M + m)g$ as a disturbance with uniformly absolutely bounded first $\dot{\xi}_z(t)$ and second $\ddot{\xi}_z(t)$ derivatives. Therefore can be overwritten as,

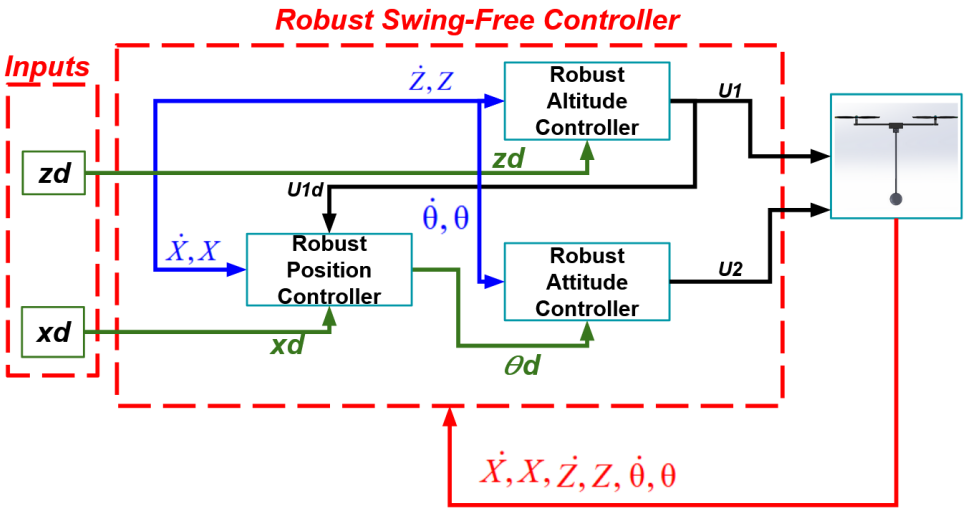


Fig. 4.2: Scheme of the proposed control algorithm.

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= \frac{1}{\bar{m}} \left(u_1 \cos\theta + \xi_z(t) \right) \end{aligned} \tag{4.5}$$

where $\bar{m}$ is defined as the sum of both masses, z_1 and z_2 are the vertical position and velocity respectively. The following observer is proposed to estimate $\xi_z(t)$, where l_2, l_1 y $l_0 \in \mathbb{R}_+$, the parameters are chosen such that the polynomial is Hurwitz.

$$\Sigma_{ESO_z} : \begin{cases} \dot{\hat{z}}_1 = \mathbf{z}_2 + l_2 \left(z_1 - \hat{z}_1 \right) \\ \dot{\hat{z}}_2 = \dfrac{1}{\bar{m}} \left(u_1 \cos\theta + \hat{z}_3 \right) + l_1 \left(z_1 - \hat{z}_1 \right) \\ \dot{\mathbf{z}}_3 = l_0 \left(z_1 - \hat{z}_1 \right) \end{cases} \tag{4.6}$$

The proposed controller generates a smooth response to the UAV's bounded permitted maximum angle θ_{max}.

Theorem 4.2.1 *Consider the vertical dynamics described by (4.1) with the following control input:*

$$u_1 = sat \left(\frac{\bar{m} f_{sat}(z_1, z_2) - \hat{z}_3}{\cos\theta}, \Gamma_{max} \right) \tag{4.7}$$

Therefore (4.10) *stabilizes asymptotically to the desired position* $x(t) = x_d, \dot{x}(t) = 0$ *and* $\|\xi_x(t)\| \leq \varepsilon$.

4.2.2 Horizontal Control for Quadrotor with Cable-Suspended Load

The control algorithm for the horizontal UAV position is addressed. Considering the horizontal dynamics of the UAV and defining $\xi_x(\mathrm{t}) = -mL\left(\sin\alpha\dot{\alpha}^2 - \cos\alpha\ddot{\alpha}\right)$ as a disturbance with uniformly absolutely bounded first $\dot{\xi}_x(t)$ and second $\ddot{\xi}_x(t)$ derivatives,

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{1}{\bar{m}}\left(u_1 \sin\theta + \xi_x(t)\right) \end{aligned} \tag{4.8}$$

where x_1 and x_2 are the horizontal position and velocity respectively. The following observer is proposed to estimate $\xi_x(t)$,

$$\Sigma_{ESO_x} : \begin{cases} \dot{\hat{x}}_1 = \mathbf{x}_2 + l_{x,2}\left(x_1 - \hat{x}_1\right) \\ \dot{\hat{x}}_2 = \dfrac{1}{\bar{m}}\left(u_1 \cos\theta + \hat{x}_3\right) + l_{x,1}\left(x_1 - \hat{x}_1\right) \\ \dot{\mathbf{x}}_3 = l_{x,0}\left(x_1 - \hat{x}_1\right) \end{cases} \tag{4.9}$$

where $l_{x,2}$, $l_{x,1}$ y $l_{x,0} \in \mathbb{R}_+$, the parameters are chosen such that the polynomial is Hurwitz.

Theorem 4.2.2 *Consider the horizontal dynamics with the following control input:*

$$\theta_c(t) = sat\left(\arcsin\frac{\bar{m}f_{sat}(x_1, x_2) - \hat{x}_3}{u_1}, \theta_{max}\right) \tag{4.10}$$

Therefore (4.10) *stabilizes asymptotically to the desired position* $x(t) = x_d$, $\dot{x}(t) = 0$ *and* $\|\xi_x(t)\| \leq \varepsilon$.

4.3 Simulation results

This section presents the results obtained in the simulation. Two scenarios have been established; the first scenario is the ability to stabilise the robot to the desired position (x_d, z_d) while limiting the oscillation of the load $(\alpha(t), \dot{\alpha})$. It is compared with the results obtained in [92, 94]. In the second scenario, the UVA tries compensating the load to the vertical axis, while the UAV hovers, conditions away from the equilibrium position are proposed for the cargo. Results are compared

using the extended state observer and without the extended state observer. The physical parameters

Table 4.1: Parameters

Symbol	Description	Value	Units
M	Quadrotor mass	2	Kg
m	Load mass	0.5	Kg
l	Cable length	0.1	m
J	Moment of inertia of the Quadrotor	0.0043	kg m^2
g	Gravity	9.81	m/s^2

used for the simulation are found in Table 4.1.

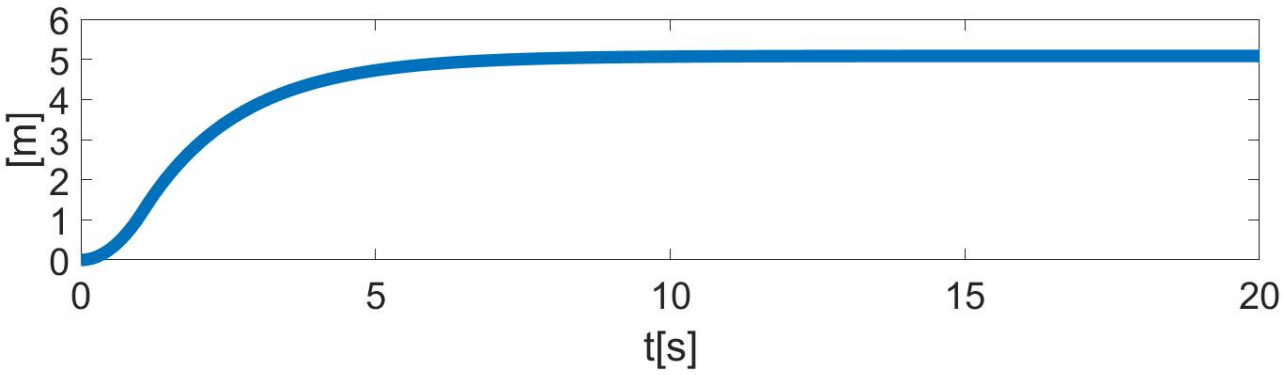


Fig. 4.3: Closed-loop behaviour of the UAV with cable-suspended load horizontal dynamics z(t).

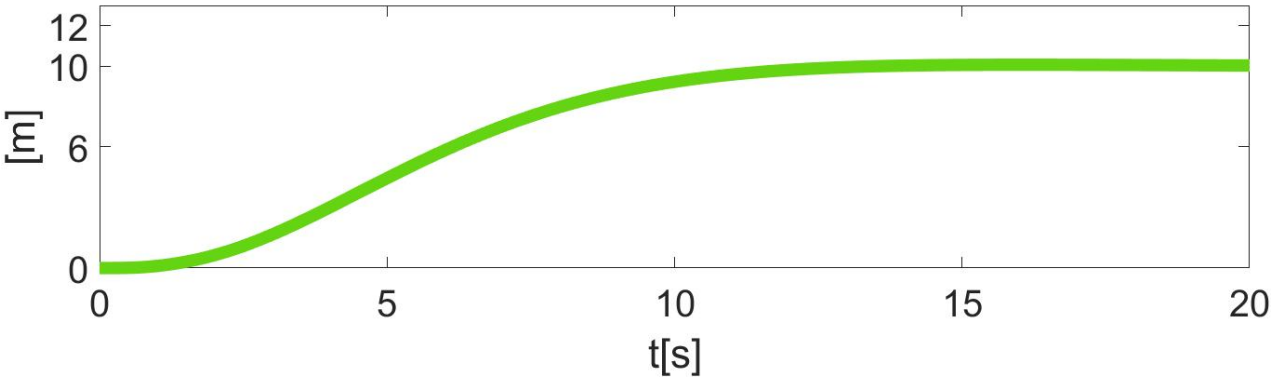


Fig. 4.4: Closed-loop behaviour of the UAV with cable-suspended load horizontal dynamics x(t).

The first scenario is the regulation of the desired position. The Figs. 4.3, 4.4 shows the evolution of the position of the quadrotor. The control is capable of stabilising the aircraft. The Figs.

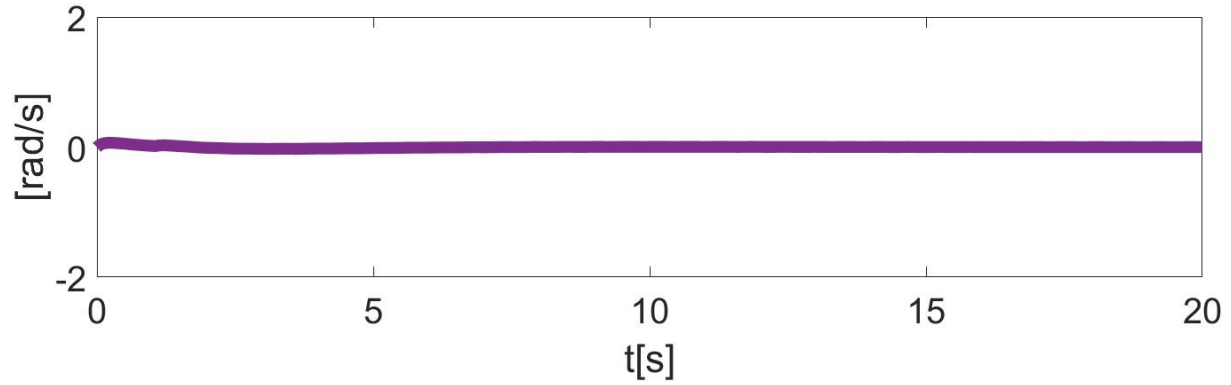


Fig. 4.5: Closed-loop behaviour of the UAV with cable-suspended load horizontal dynamics $\phi_Q(t)$.

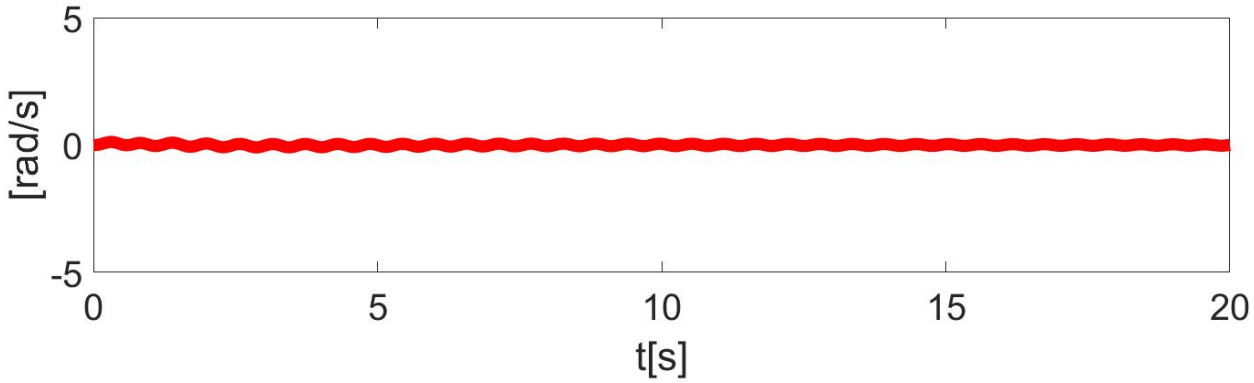


Fig. 4.6: Closed-loop behaviour of the UAV with cable-suspended load horizontal dynamics $\phi_L(t)$.

4.5 and 4.6 show the evolution attitude and angle of the load; the proposed control is robust to disturbance and dynamics not modelled. It presents a smooth response without over-elongation with a favourable settling time and has no error in steady-state. The desired position has been set to $(x_d, z_d) = (5m, 10m)$.

The second scenario compares the control without the extended state observer. Figs 4.7 and 4.8, show the evolution of the position of the quadrotor. The Figs. 4.9 and 4.10 show the evolution of the state variables; the proposed robust control reduces the oscillation of the dynamics of the cable load, simultaneously stabilise the UAV to the desired position (hover) with conditions far from the equilibrium point for the suspended load. Compared to non-observer control that fails to stabilise the load to equilibrium.

Comparing the results obtained with other works, the proposed control algorithm does not require knowledge of the dynamics of the load coupled to the UAV. Comparing the results obtained in [92],

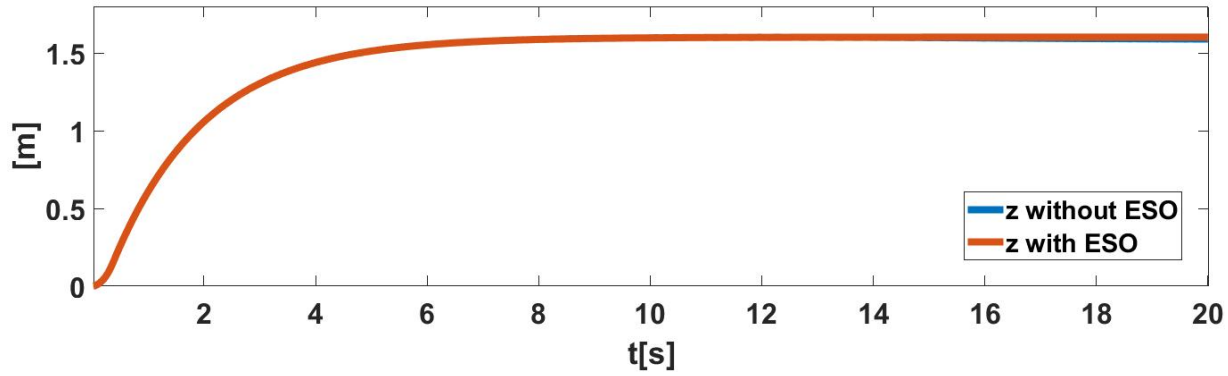


Fig. 4.7: Closed-loop behaviour of the UAV with cable-suspended load horizontal dynamics z(t).

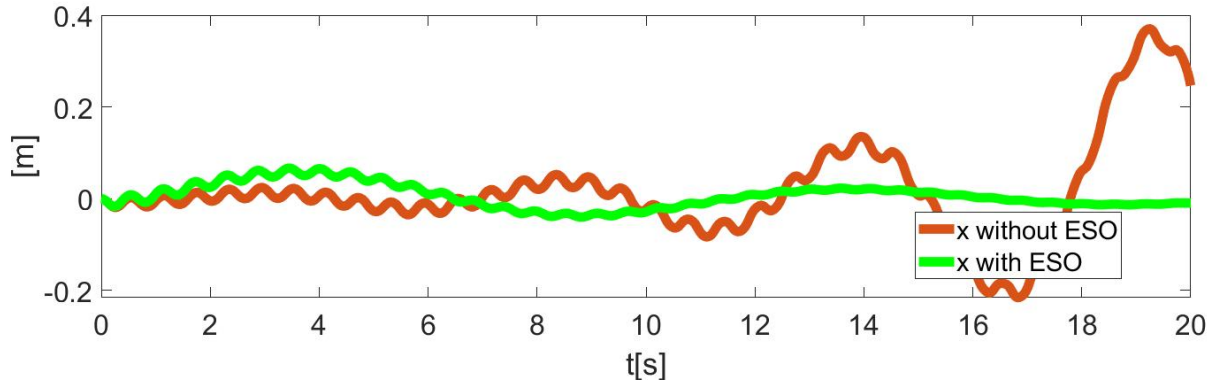


Fig. 4.8: Closed-loop behaviour of the UAV with cable-suspended load horizontal dynamics x(t).

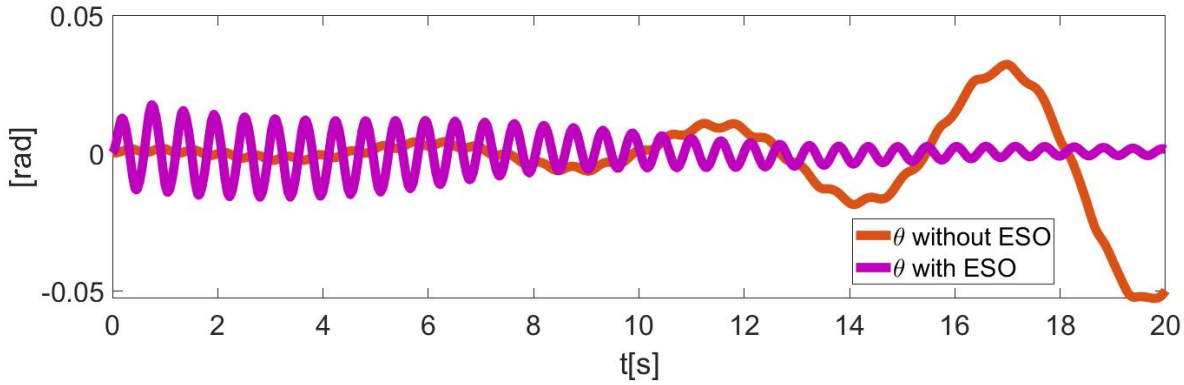


Fig. 4.9: Closed-loop behaviour of the UAV with cable-suspended load horizontal dynamics $\phi_Q(t)$.

for the first scenario, we observe that the knowledge of the dynamics of the load coupled to the UAV is avoided; it also shows a smooth response and simultaneously achieves the stabilisation of the oscillation of the load and the position of the aircraft. A video of the operation of the orientation

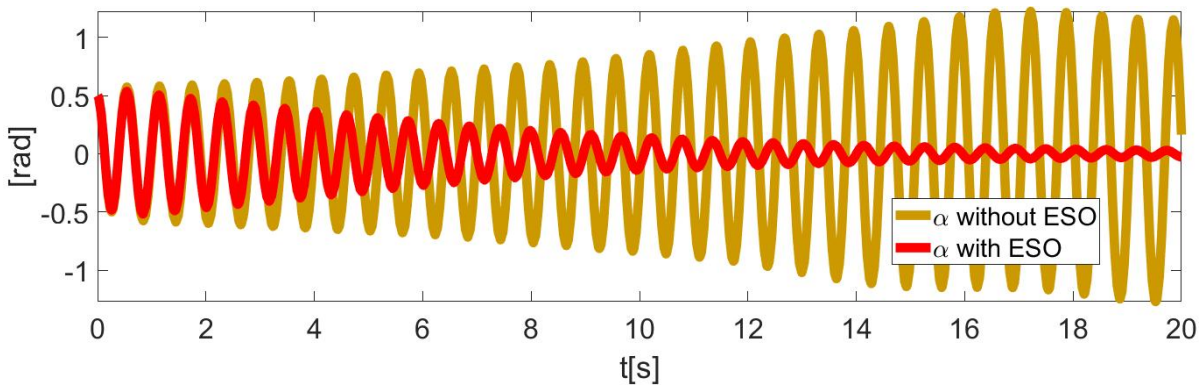


Fig. 4.10: Closed-loop behaviour of the UAV with cable-suspended load horizontal dynamics $\phi_L(t)$.

control implemented in the Pixhawk autopilot is shown in, `https://www.youtube.com/watch?v=EmOCTXBu4rI`.

4.4 Conclusions

The proposed control algorithm shows the stabilization of the UAV to the desired position while simultaneously aligning the position of the load to the balance point (vertical axis). This new control approach does not need extra sensors to estimate the position and speed of the suspended load. The simulations show a smooth response without over-elongation, with a favourable settling time. There is no error in steady-state, and it is robust to disturbances and dynamics not modelled. The proposed approach is compared with different control laws proposed in the literature showing better stability without the need to feedback the dynamics of the suspended load.

Chapter 5

Robust Trajectory Tracking ADRC-Based Control approach

I do not think there is any thrill that can go through the human heart like that felt by the inventor as he sees some creation of the brain unfolding to success... such emotions make a man forget food, sleep, friends, love, everything.

– **Nikola Tesla**

5.1 Introduction

This chapter addresses the problem of proposing a novel robust nonlinear position trajectory tracking control for an unmanned aerial vehicle (UAV) with vertical takeoff and landing (VTOL) capabilities under multiple time-varying perturbations. The trajectory tracking position control algorithm is decoupled into two systems: altitude (z) and horizontal position (x-y). For altitude control, a robust control fusing the principles of active disturbance rejection control (ADRC) with the ground effect (IGE) model is designed. A non-linear extended state observer is created on the vertical axis, taking attitude and altitude measurements. Then, the forces generated by low-altitude flight,

ground effect, and other external disturbances are estimated and used (as a feedback term) with a nonlinear control law (feedback term) to reject them. Tests are performed under different manoeuvres that depend on the proximity of the ground, such as takeoff, landing, inspection, and hover. An improved control scheme based on the Active Disturbance Rejection Control (ADRC) strategy is proposed regarding the horizontal dynamics. Multiple disturbances are considered parametric uncertainties, ground effects and crosswinds. A nonlinear extended-state observer (ESO) estimates aerodynamic effects by taking into account external, time-varying, state-dependent perturbations, and bandwidth is assured based primarily on initial disturbances. Endless experimental tests are performed in real-time. Different trajectories are made by comparing the proposed control with the traditional ADRC methodology. The main contribution of this investigation is adding a function on the dynamics of the error and the coefficient of the sufficiency of control, which help the convergence of the estimation error and its possible saturation. This new methodology stimulates future interactions and cross-fertilisation of ideas between the various disciplines of automatic control.

1. An unsimplified model is established in the presence of endogenous and exogenous disturbances.

2. A novel robust altitude control scheme is designed to stabilise the altitude of the multirotor subject to unmodeled dynamics, external disturbances and ground effect.

3. A nonlinear extended state observer (ESO) is designed on the vertical axis to estimate the ground effect at low altitudes in hovering flights.

4. A novel robust trajectory tracking control scheme is designed to stabilise the position of the multirotor subject to parametric uncertainties, ground effect and wind gusts.

5. A non-linear extended state observer is designed for the multirotor position, taking measurements of the Euler angles (pitch and roll) and the altitude.

6. The algorithm is physically implemented and is shown to be lightweight and efficient.

7. A formal closed-loop control-observer stability test is performed in the ISS sense.

8. An integral square error performance index compares the classic ADRC and the proposed control.

9. Experimental tests on a physical UAV platform validate the proposal.

5.2 Robust Altitude Trajectory Tracking Control

This section describes the structure and design of the proposed control algorithm for the multirotor. Most of the time, the controller is based on a cascade structure; the outer loop controller (dynamics of position) and the inner loop controller (attitude dynamic). The outer loop control calculates the desired angles $\Theta_d(t)$ and the desired thrust $f_d(t)$ according to the desired trajectory position $p_d(t)$ of the UAV. The inner loop control computes the desired moments $\tau_d(t)$ according to the desired angles Θ_d. Furthermore, the outer controller can be broken into two subsystems: altitude and horizontal position control.

The main objective of the present work is to design a precise altitude robust control for a multirotor subject to external disturbances, ground effect dynamics, and parametric uncertainties. The purpose of the altitude controller is to stabilize the aircraft to the desired reference position z_d, sufficiently near to the ground or rigid surfaces, with the desired velocity $v_{zd}(t) = 0$. Mathematically we can express the above according to:

$$z(t) \to p_d(t),\ v_z \to v_{zd}(t) = 0, \quad \text{as } t \to \infty. \tag{5.1}$$

Assumption 5.2.1 *The disturbance function ξ_z is uniformly absolutely bounded, i.e. $\|\xi(t)\|_\infty \leq \bar{\xi}$.*

5.2.1 ADRC-based Altitude Controller

The present investigation aims to combine the ideology of ADRC design with the in-ground-effect model. For that, a nonlinear extended state observer is designed on the vertical axis by taking measurements of the Euler angles (pitch and roll) and the altitude. Then, the forces generated by the flight at low altitudes, the ground effect, and other external disturbances are estimated simultaneously experiencing ground effect and blade damage the IGE model is used as a multiplicative factor in the control gain to compensate for the insufficient disturbance compensation ability of ADRC. Even so, it is difficult for the above mentioned techniques to be robust enough to eliminate multiple

endogenous and exogenous disturbances simply by feedback control. An extended-state nonlinear observer (ESO) is designed on the vertical axis by taking measurements of the Euler angles, the designed thrust, and the nonlinear dynamics of the multirotor to estimate endogenous and exogenous disturbances such as ground effects. Let us define the following variable, $\Lambda(\psi,\theta) = \frac{\cos\psi\cos\theta}{m}$ which is assumed to be known. Then, the dynamics along the z-axis given by (C.8) can be re written as follows

$$\Sigma_z := \begin{cases} \dot{z} = v_z \\ \dot{v}_z = \Lambda(\psi,\theta)T + \xi_z(t) \end{cases} \tag{5.2}$$

In order to estimate the endogenous and exogenous disturbances lumped in the term ξ_z, we introduce the following extended state observer (ESO):

$$\Sigma_{ESO} : \begin{cases} \dot{\hat{z}} = \hat{v}_z + l_2\,(z-\hat{z}) \\ \dot{\hat{v}}_z = \Lambda(\phi,\theta)T + \hat{\xi}_z + l_1\,(z-\hat{z}) \\ \dot{\hat{\xi}}_z = l_0\,(z-\hat{z}) \end{cases} \tag{5.3}$$

where l_1, l_2 and l_3 are the observer's gains which are chosen such that the characteristic polynomial of the observer's error dynamics matches those of the desired polynomial $p_d(\lambda) = (\lambda^2 + \frac{2\zeta\omega_n}{\epsilon} + \frac{\omega_n^2}{\epsilon^2})(\lambda + \frac{p_0}{\epsilon})$. Then, one has

$$\begin{aligned} l_2 &= \tfrac{1}{\epsilon}\left(p_0 + 2\zeta\omega_n\right), \\ l_1 &= \tfrac{1}{\epsilon^2}\left(2p_0\zeta\omega_n + \omega_n^2\right), \\ l_0 &= \tfrac{1}{\epsilon^3}\left(\omega_n^2 p_0\right), \end{aligned}$$

with $\epsilon \in +$ sufficiently small and $\omega_n, \zeta, p_0 \in +$.

Proposition 5.2.2 *The estimation error $e = z - \hat{z}$ satisfies the following perturbed linear differential equation*

$$e^{(3)} + l_2\ddot{e} + l_1\dot{e} + l_0 e = \dot{\xi}_z(t) \tag{5.4}$$

if the observer's gains are chosen as previously mentioned, the trajectories of the estimation error globally converge toward a small as desired sphere of radius ρ centered at the origin of the estimation error phase space $\{e, \dot{e}, \ddot{e}\}$ where they remain ultimately bounded.

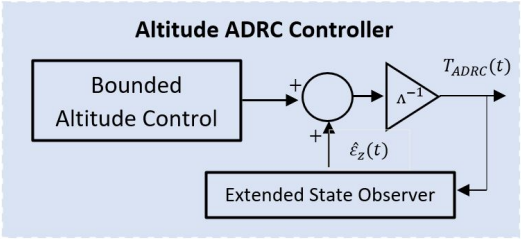


Fig. 5.1: The block diagram of the Altitude ADRC controller.

The proof follows the one presented in [49, 95].

Remark 5.2.3 *Note that the observer* (5.3) *could estimate the term of gravity g together with the perturbation* ξ_z *using only one lumped perturbation term* $\hat{\xi}_z$.

Definition 5.1 *Given a positive constant* M, *a continuous and non-decreasing function* $\sigma_M : \mathbb{R} \to \mathbb{R}$ *is defined as:*

$$\begin{aligned} &(1)\sigma_M = s \text{ if } |s| < M; \\ &(2)\sigma_M = M \cdot \text{sign}(s) \ ; \end{aligned} \tag{5.5}$$

Now, we are ready to announce the control law.

Proposition 5.2.4 *Consider* (5.2) *with the following bounded control input*

$$T_{ADRC} = -\Lambda^{-1}(\phi, \theta) \left(\hat{\xi}_z + \sigma_{M_2}(a_1 v_z + \sigma_{M_1}(a_2 v_z + a_1 a_2 (z - z_d) \right) \tag{5.6}$$

where σ_{M_1} *and* σ_{M_2} *are saturation functions such that* $M_2 > 2M_1$, $a_1, a_2 \in_+$ *are tuning parameters.* $\hat{\xi}_z$ *is the estimation of the unknown disturbance* ξ_z. *Then, the trajectories of the system converge toward a small sphere centered at the origin of the phase space* $\{z, v_z\}$ *where they remain ultimately bounded.*

The closed-loop system becomes:

$$\begin{aligned} \dot{z} &= v_z \\ \dot{v}_z &= -\sigma_{M_2}(a_1 v_z + \sigma_{M_1}(a_2 v_z + a_1 a_2 (z - z_d) + \xi_z(t) - \hat{\xi}_z \end{aligned} \tag{5.7}$$

Let us define the following linear transformation

$$z_1 = a_1 a_2 (z - z_d) + a_2 v_z; \quad z_2 = a_1 v_z, \quad \text{with, } a_1, a_2 \in_+$$

then, the close-loop system (5.7) becomes

$$\begin{aligned} \dot{z}_1 &= a_2 z_2 - a_2 \sigma_{M_2}(z_2 + \sigma_{M_1}(z_1)) + a_2 \tilde{\xi} \\ \dot{z}_2 &= -a_1 \sigma_{M_2}(z_2 + \sigma_{M_1}(z_1)) + a_1 \tilde{\xi} \end{aligned} \tag{5.8}$$

with $\tilde{\xi} = \xi_z(t) - \hat{\xi}_z$. In order to analyse the closed-loop stability, we begin by considering the evolution of the state z_2. Consider the Lyapunov function $V_2 = \frac{1}{2} z_2^2$. The derivative of V_2 is given by

$$\dot{V}_2 = -a_1 z_2 \sigma_{M_2}(z_2 + \sigma_{M_1}(z_1)) + a_1 z_2 \tilde{\xi} \tag{5.9}$$

Assuming that $|z_2| > 2M_1$, it follows that $|z_2 + \sigma_{M_1}(z_1)| > M_1 + \varepsilon$, with ε sufficiently small. Then, $z_2 + \sigma_{M_1}(z_1)$ has the same sign than z_2. From Proposition 5.2.2, $\tilde{\xi}$ remains bounded with bounded denoted $\bar{d}$, then

$$\dot{V}_2 \leq -a_1 |z_2| \, \sigma_{M_2}(M_1 + \varepsilon) + a_1 |z_2| \, \bar{d} \tag{5.10}$$

assuming that $\bar{d} < \min(M_2, (M_1 + \varepsilon))$, one can assure the decrease of V_2, i.e., $\dot{V}_2 < 0$. Consequently, z_2 enters $\Phi_2 = \{z_2 : |z_2| \leq 2M_1\}$ in finite time t_1 and remains in it thereafter. Therefore, (5.8) becomes

$$\begin{aligned} \dot{z}_1 &= -a_2 \sigma_{M_1}(z_1) + a_2 \tilde{\xi} \\ \dot{z}_2 &= -a_1 (z_2 + \sigma_{M_1}(z_1)) + a_1 \tilde{\xi} \end{aligned} \tag{5.11}$$

now consider the evolution of z_1. For that, let $V_1 = \frac{1}{2} z_1^2$ be a Lyapunov function. The derivative of V_1 is given by

$$\begin{aligned} \dot{V}_1 &= -a_2 z_1 \sigma_{M_1}(z_1) + a_2 z_1 \tilde{\xi} \\ &\leq -a_2 |z_1| \, (M_1) + a_2 |z_1| \, \bar{d} \end{aligned} \tag{5.12}$$

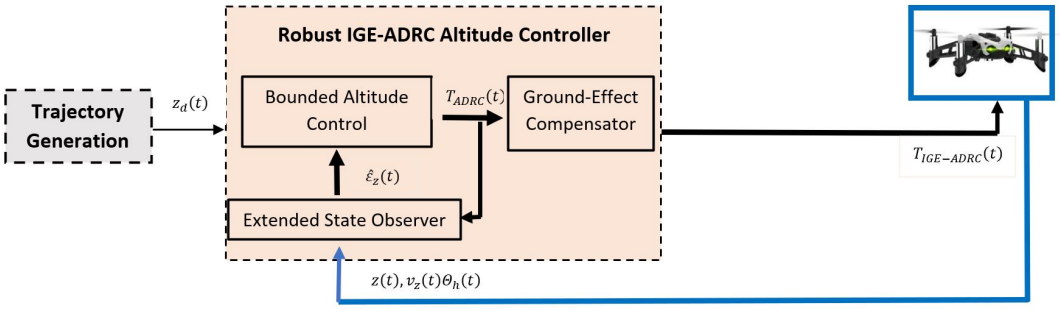


Fig. 5.2: Block Diagram of the Robust Altitude Controllers.

assuming that $\bar{d} < M_1$, then $\dot{V}_1 < 0$. Consequently, z_2 enters $\Phi_1 = \{z_1 : |z_1| \leq M_1\}$ in finite time t_2 and remains in it thereafter. Consequently, (5.8) becomes

$$\begin{aligned} \dot{z}_1 &= -a_2 z_1 + a_2 \tilde{\xi} \\ \dot{z}_2 &= -a_1 z_2 - a_1 z_1 + a_1 \tilde{\xi} \end{aligned} \tag{5.13}$$

Note that (5.13) represent a stable perturbed linear system. Then, the trajectories of the system converge toward a small sphere of radius $\bar{\rho}$ centered at the origin of the phase space $\{z_1, z_2\}$ and, due to the linear transformation, of the phase space $\{z, v_z\}$ where they remain ultimately bounded.

5.2.2 ADRC-IGE-based Altitude Controller

The proposed ADRC control algorithm does not consider the ground effect; however, if a known ground effect model is established, it can be used to generate a compensator for the already established ADRC control.

$$T = -\tilde{\Lambda}^{-1}(\phi, \theta, z) \left(\hat{\xi}_z + \sigma_{M_2}(a_1 v_z + \sigma_{M_1}(a_2 v_z + a_1 a_2 (z - z_d) \right) \tag{5.14}$$

where $\tilde{\Lambda}(\phi, \theta, z) = \frac{\Lambda(\phi,\theta)}{1-\rho\left(\frac{r}{4z_r}\right)^2}$ is an altitude-dependent trust coefficient generated mainly by the ground effect and the lateral movement of the aircraft. Fig (5.2) shows the block diagram of the proposed control strategy. This diagram shows the interaction between the extended state observer ESO, the bounded control and the ground effect compensator.

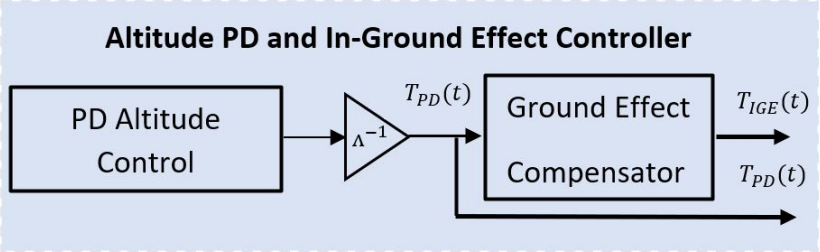


Fig. 5.3: The block diagram of the Altitude in-ground effect IGE controller.

5.2.3 Simulations

In this section, we will perform numerical simulations considering the ground effect, forces induced by wind gusts, and uncertainties. The ESO has some advantages over other observers, high convergence and a wide frequency band to estimate different types of disturbances. Two scenarios are addressed, rejection of external disturbances such as the force exerted by constant gusts of wind in hovering flights and low-altitude flights. The proposed control algorithms, namely, ADCR and IGE-ADRC, will be analyzed and compared with a conventional PD controller and a PD controller using IGE. The conventional proportional-derivative control defined by (5.15), and the controller based on an IGEC ground effect compensator is defined by (5.16).

$$T_{PD} = \frac{-1}{\Lambda(\phi,\theta)}\left(a_2 v_z + a_1(z - z_d)\right) \tag{5.15}$$

$$T_{IGE} = -\frac{\left(1 - \rho\left(r/4z_r\right)^2\right)}{\Lambda(\phi,\theta)}\left(a_2 v_z + a_1(z - z_d)\right) \tag{5.16}$$

The IGEC controller can keep the aircraft at low altitudes by mitigating ground effects. However, it cannot deal with external additive disturbances. The block diagram of the altitude proportional-derivative and ground-effect controller is depicted in Fig. 5.3.

The physical parameters used in the simulation correspond to the Mambo multirotor designed by the French company Parrot and whose values are depicted in Table 5.1.

In this first scenario, we focus on estimating and rejecting the force induced by wind gusts in hovering flight and the ground effect disturbance. A ground effect model [75] is added to the dynamic model of the multirotor with a coefficient $\rho = 3.4$ experimentally determined. The frequency range of the wind gusts is approximately 0.005-0.14 Hz according to [53],[96]. In this scenario, the disturbance

Table 5.1: Physical parameters of the mambo multicopter.

Parameter	Description	Units
m	Mass	0.635 kg
l	Distance	0.185 m
r	Propeller radius	0.04 m
v	Flight speed	30 km/h

generated by the wind gusts only affects the z-axis in the time range of 10 to 40 seconds, and is described by the (5.17).

$$\xi_T(t) = A \sin 2\pi f_w t \tag{5.17}$$

where $A = 0.4$ and $f_w = 0.1$ Hz, which is inside the frequency range of the wind. $a_1, a_2 \in {}_+$ are tuning parameters whose used values are $a_1 = 3$ and $a_2 = 5$. In order to compare the proposed control scheme, the squared integral of the error ISE has been introduced, which is calculated using the following

$$ISE(t) = \int_0^t e_z^2 \, dt \tag{5.18}$$

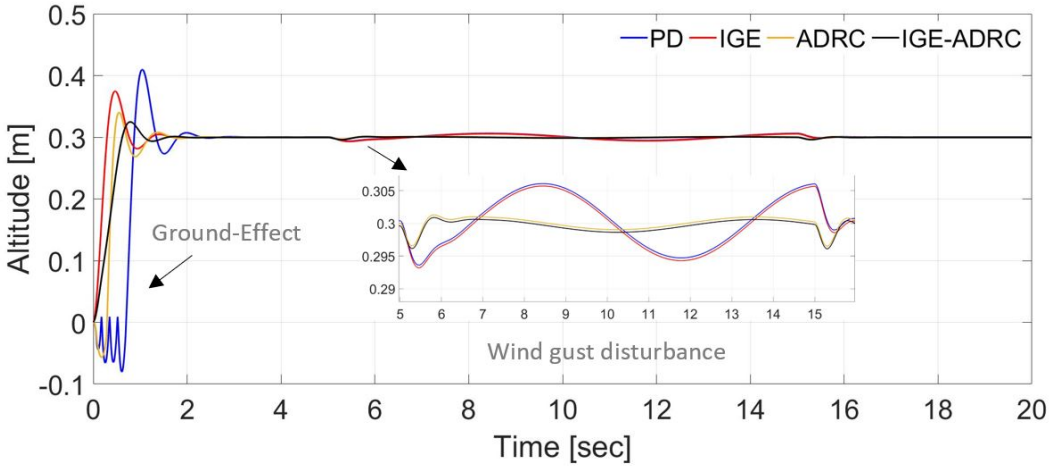


Fig. 5.4: Fist scenario: Altitude response.

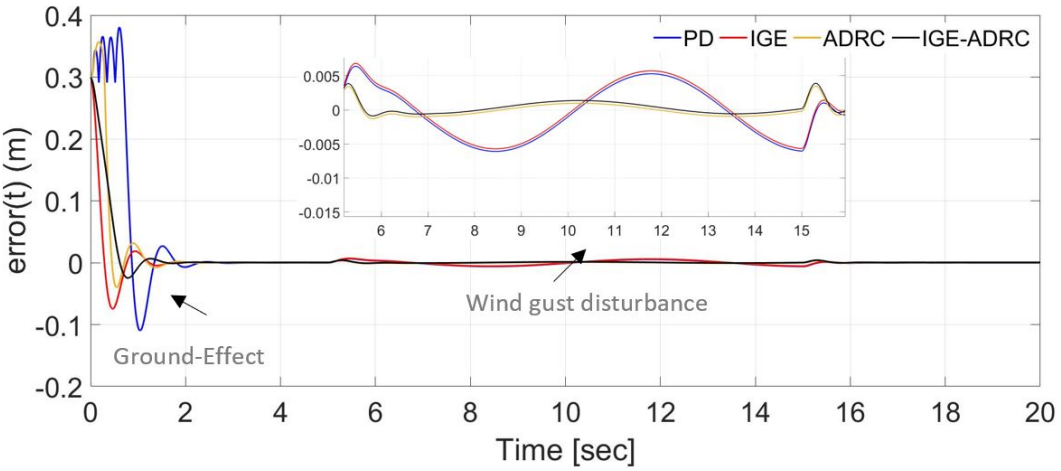


Fig. 5.5: Fist scenario: error response.

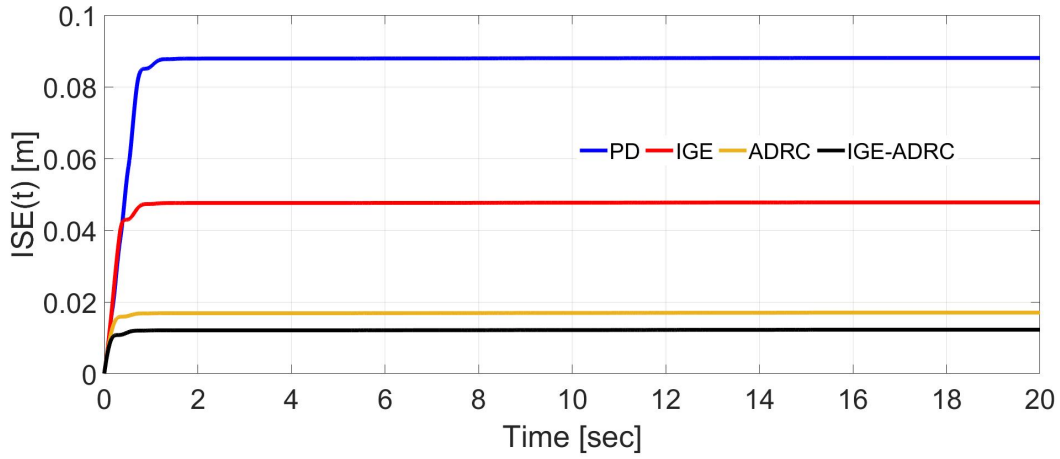


Fig. 5.6: Fist scenario: Integral square error response.

As analyzed in Figs. 5.4, 5.5 and 5.6 the proportional-derivative control stabilises the system at the desired reference $z_d = 0.3$. However, during $t < 2$ second the PD controller cannot attenuate the ground effect causing oscillation which could cause instability. It was also susceptible to external disturbances such as moderate wind gusts, as seen in the time $(5 - 15)$ seconds. The IGE controller is capable of ultimately attenuating and eliminating ground effects. However, it is not robust against additive disturbances such as wind gusts added to the system. It is worth mentioning that using the compensator in the control algorithm is very simple; however, obtaining the coefficient ρ through experimentation can be tedious. In turn, the ADRC control shows good behavior since one assumes not knowing the ground effect's forces when the UAV is close to a surface. Furthermore, it has a

smooth, fast, and error-free response in a steady state. Additive disturbances such as gusts of wind are wholly attenuated. However, the ground effect is not entirely attenuated, mainly because the ground effect can be seen as a multiplicative disturbance. The proposed ADRC-IGE-based altitude control tries to improve the previous control schemes. This control algorithm compensates for the ground effect and eliminates gusts of wind, obtaining a fast, smooth, and error-free response in a steady state. Unlike the conventional ADRC that saturates due to the disturbance generated by ground effect in some maneuvers, takeoff, and landing, the IGEC-ADRC can eliminate these disturbances.

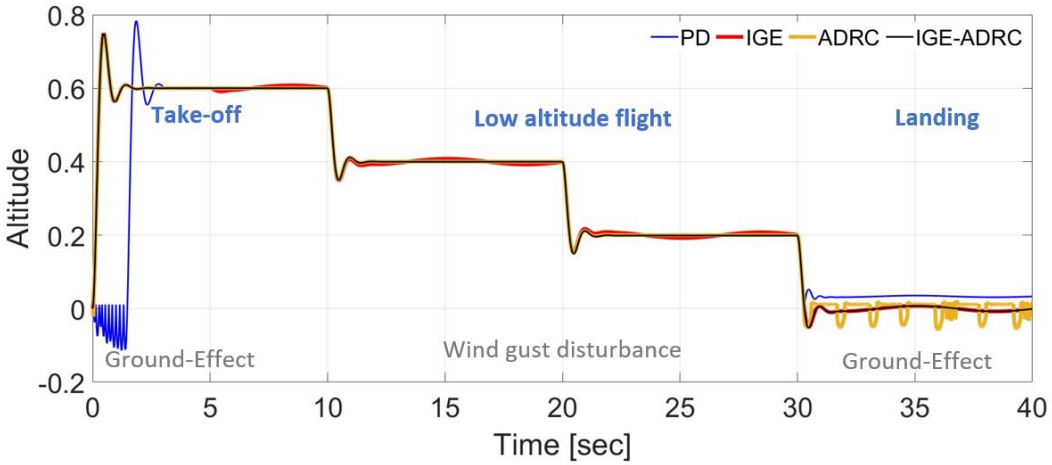


Fig. 5.7: Second scenario: Altitude response.

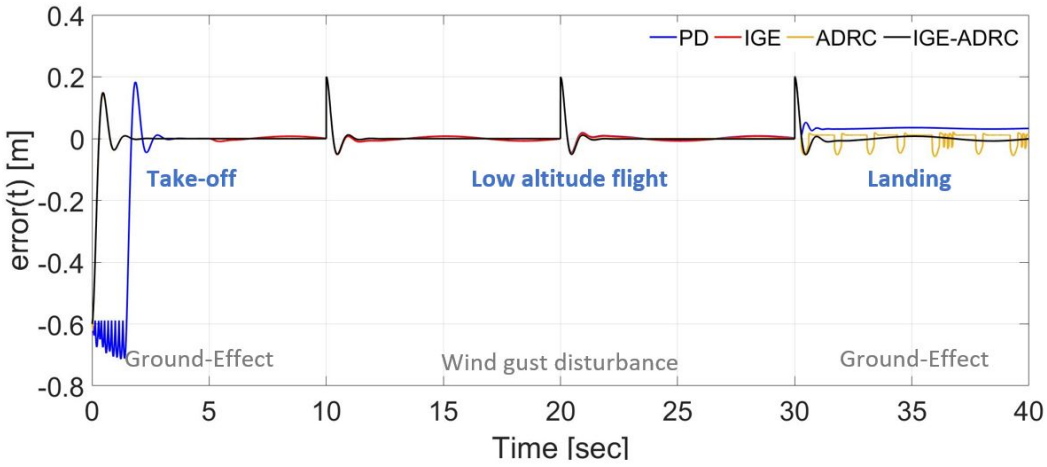


Fig. 5.8: Second scenario: error response.

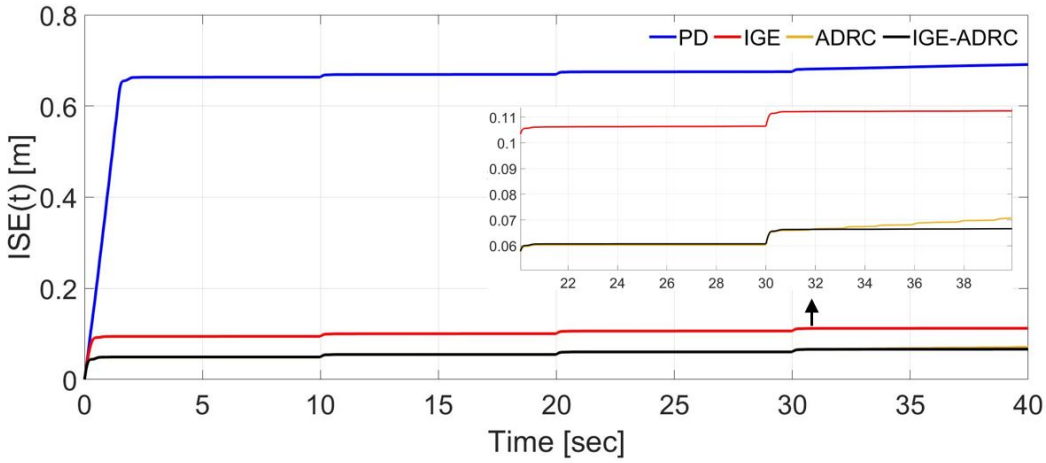
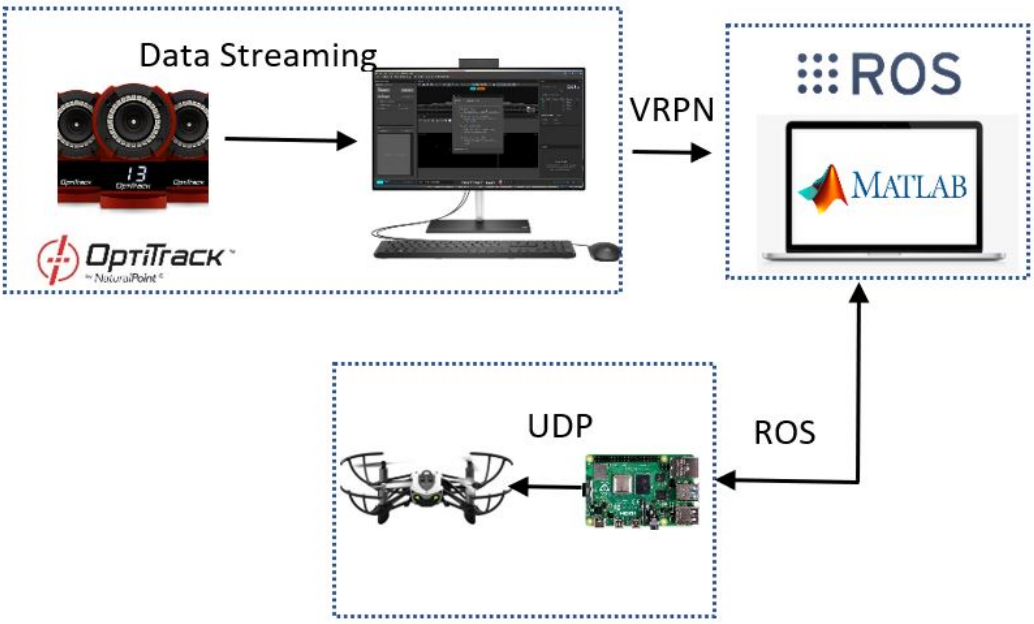


Fig. 5.9: Second scenario: Integral square error response.

In a second scenario, the ground effect at low altitudes is addressed. Since the multirotor is more susceptible to ground effects in maneuvers, such as automatic takeoff and landing, the behavior of the control algorithms is analyzed. Since ground effect can be significant up to 6r above the ground, we have considered four different altitudes $(0.6, 0.4, 0.2, 0)\, m$. In this second scenario, the PD control cannot attenuate the ground effect in the takeoff and landing maneuvers, as shown in Fig. 5.7. The multirotor cannot land due to the extra thrust generated by the ground effect. The IGE control behaves better at low-altitude flights, as shown in Figs. 5.7, 5.8 and 5.9. The ground effect is completely attenuated, and the multirotor can accurately land. The only problem lies in external disturbances affecting this control's performance. The ADRC control and the IGE-ADRC have similar behavior; both allow estimating and eliminating additive disturbances. However, the conventional ADRC is affected by the ground effect in the landing and takeoff maneuvers. This effect is because the ESO tries to compensate for these multiplicative disturbances; however, as there is no knowledge of this model, the estimate of the ESO grows until it saturates the actuators. For this reason, it is necessary to integrate the ground effect compensator to solve the problem. In addition, this ground effect model can be added to the ESO dynamics to obtain a more accurate response.



tw

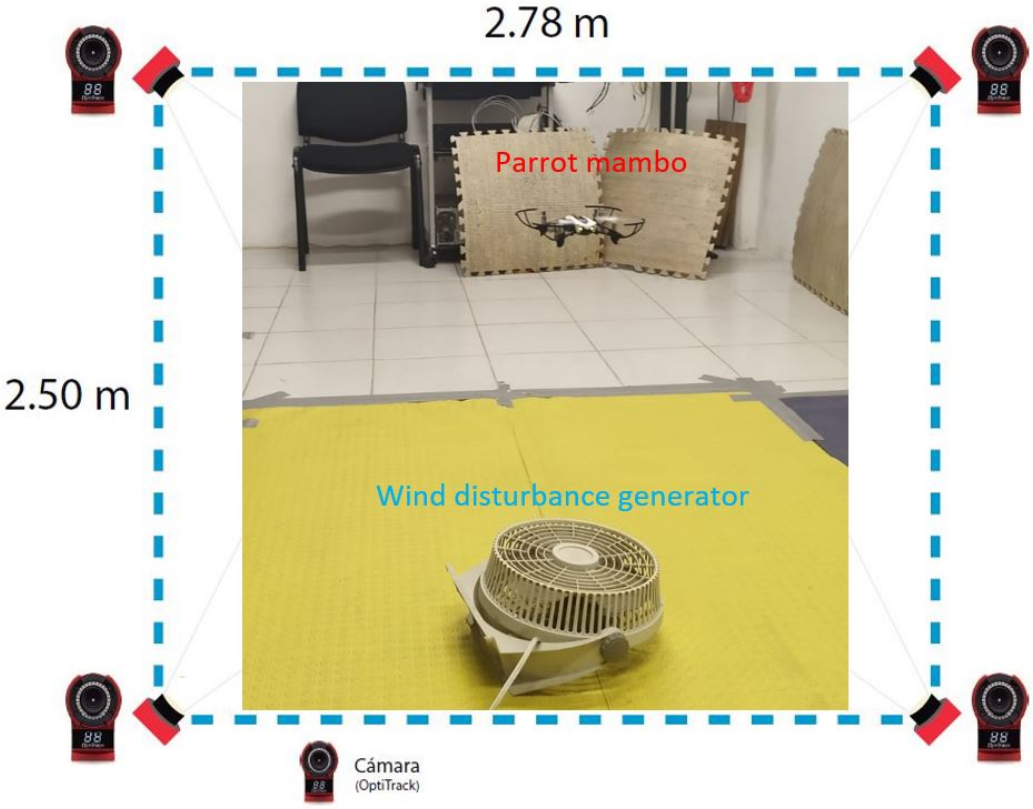
Fig. 5.10: General diagram of the experimental platform.

5.2.4 Experimental Results

The experimental results were generated through a test bench using a camera-based motion capture system, a communication network for transmitting information through ROS, and various specialized software. The unmanned aerial vehicle used in conjunction with the test bench is the Parrot Mambo. The choice of the vehicle focuses on its small dimensions and its performance for handling indoors. Table 5.1 shows some of the characteristics of the Mambo system. In conjunction with the cameras, the facility allows flights to be carried out in a safe and controlled environment.

The OptiTrack system is used for its accuracy at the millimeter level to estimate any rigid body with six degrees of freedom. Data transmission through a local network is used. The multirotor's estimation of position, linear velocity, and attitude (quaternion) is sent to a ground station via the VRPN ROS network. The processing of the positions, the Euler angles, and the velocities estimation are carried out using MATLAB/Simulink with a sample rate of 100 Hz. The control algorithm is evaluated at this ground station and then sent to a single-board computer, Raspberry Pi, via ROS. Finally, the Raspberry Pi sends the control data to the drone using the same ROS network and UDP packets. Fig. C.11 shows the general diagram of the experimental platform.

A test has been carried out to validate the control algorithm, divided into three regions: takeoff,



tw

Fig. 5.11: Ground effect and wind-like disturbance in flight at low altitudes.

low-altitude flights, and landing. Since the ground effect is more significant in maneuvers such as takeoff and landing, both schemes were addressed in conjunction with low-altitude flights. In this test region, an external disturbance has been added with a fan at low speed. The takeoff maneuver is performed in 0 to 40 seconds at a maximum of 1.4 m. Sequentially, it passes to the region of low-altitude flights where through reference points, where there is a low impact of the ground effect in 1.2 m during the time of 40 to 100 seconds. A medium impact of ground effect in 0.5 m during the time from 100 to 200 seconds and a high ground effect impact for 200 to 300 seconds, finally the landing maneuver is performed for 300 to 320 seconds. The performance of the proposed control scheme has been compared with the PD, IGE, and ADCR-based controllers.

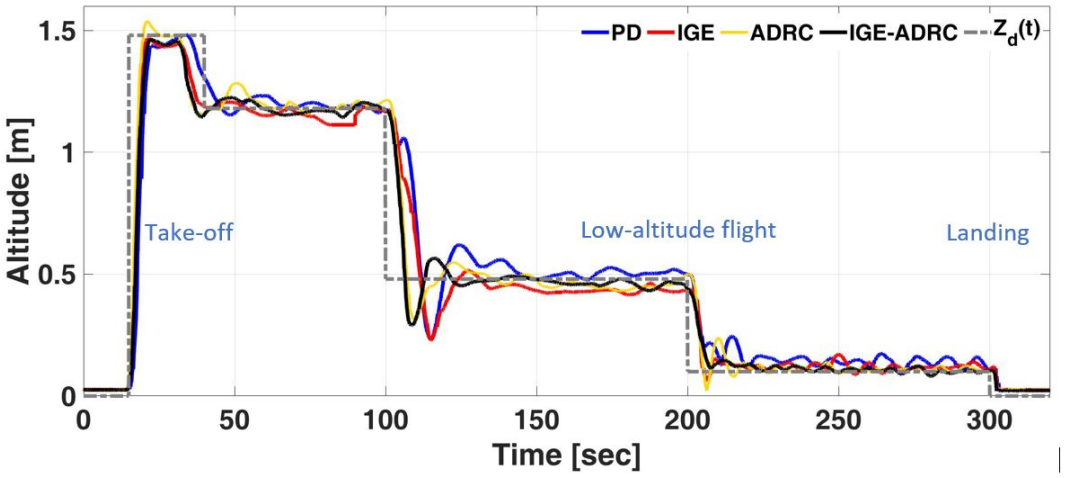


Fig. 5.12: Experimental results: Altitude response.

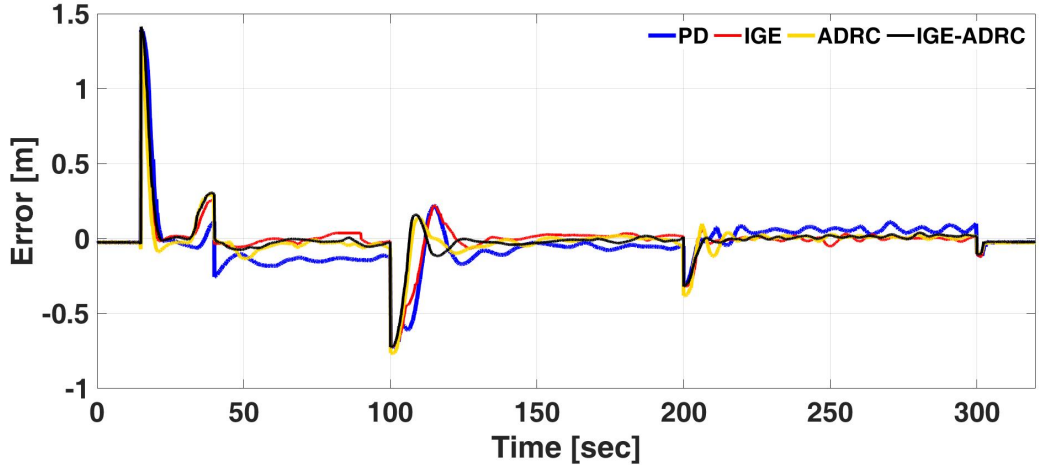


Fig. 5.13: Experimental results: error response.

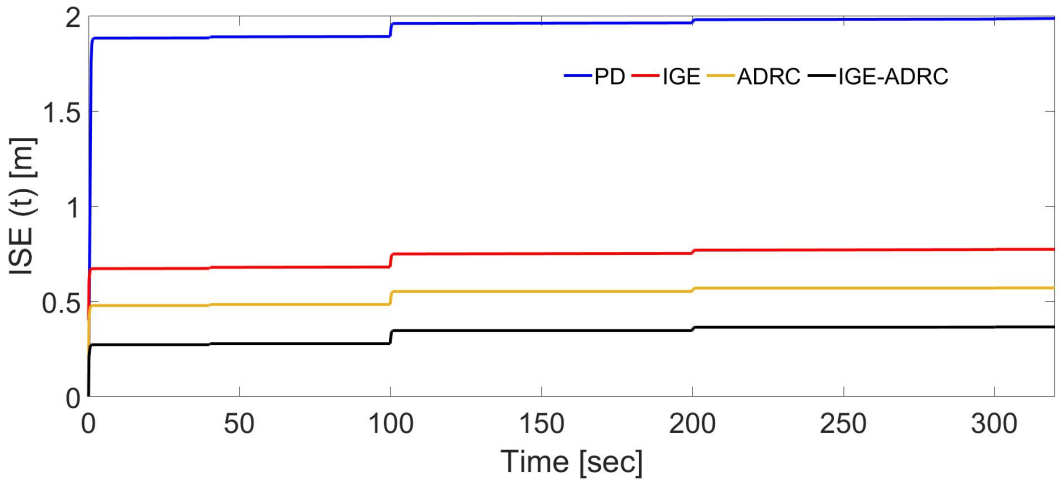
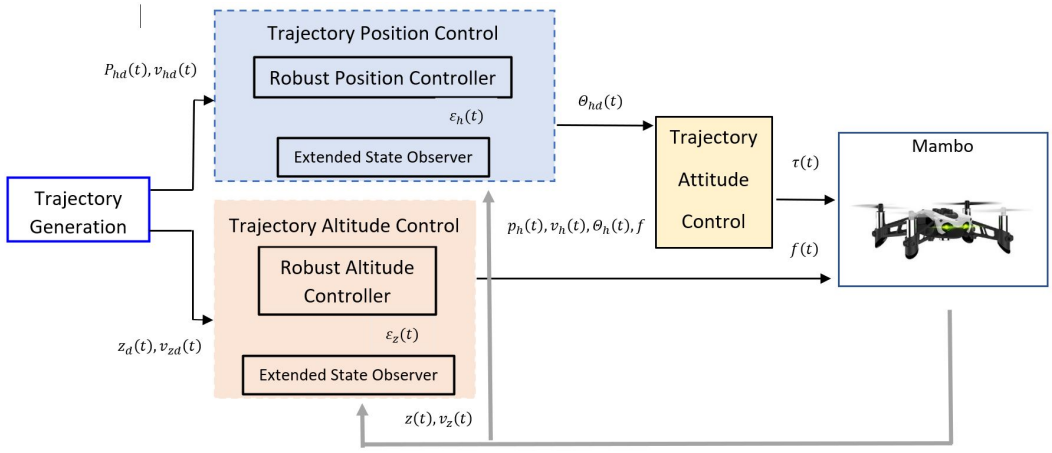


Fig. 5.14: Experimental results: Integral square error response.

Experimentally, the PD control fails to attenuate the ground effect and the external disturbance generated by the fan, showing an oscillatory response with a high degree of transitory error and in a steady state, see Figs 5.12. For the second IGE control scheme, its analysis shows a better performance in takeoff and landing maneuvers, attenuating the ground effect. However, the response oscillates in the low-altitude region where the fan generates the external disturbance. In the third ADRC control scheme, it is analyzed through Fig. 5.13 that it can estimate both disturbances; however, since the ground effect is a multiplicative disturbance, an offset and an oscillation are shown in the takeoff and landing maneuvers. The ADRC-IGE-based control performs better than the previous controllers since the IGE compensates for the multiplicative controller gain, compensating for the insufficient disturbance compensation ability of ADRC.

5.3 Robust Horizontal Trajectory Tracking Control

This section describes the design of the trajectory tracking algorithm in the horizontal plane (x-y axis). A new extended-state nonlinear observer (ESO) is proposed for precise trajectory tracking. On this new observer, a function dependent on the dynamics of the error is added together with a control sufficiency coefficient that helps the rapid convergence of the estimation error and its possible saturation. This new methodology stimulates future interactions and the cross-fertilisation



tw

Fig. 5.15: General diagram of the Robust Horizontal Control.

of ideas between the various disciplines of automatic control.

The main objective of the present work is to design a robust trajectory tracking control for a multirotor subject to parametric uncertainties and uncertain, ground effect and wind disturbance. The purpose of the proposed trajectory tracking controller is to follow a desired reference position $p_d \in \mathbb{R}^3$, with a desired velocity $v_d(t) \in \mathbb{R}^3$. Mathematically we can express the above according to (5.19):

$$p(t) \to p_d(t), \; v \to v_d(t), \quad \text{as } t \to \infty. \tag{5.19}$$

Assumption 5.3.1 *The disturbance functions ξ_h and is uniformly absolutely bounded, i.e. $\|\xi(t)\|_\infty \leq \bar{\xi}$.*

5.3.1 ADRC-based Trajectory Tracking Control

In order to estimate the wind and ground effect disturbances lumped in the term ξ_h in model (C.11), we introduce the following extended state observers (ESO) for the horizontal position dynamics:

$$\Sigma_{ESO_h} \; : \; \begin{cases} \dot{\hat{p}}_h &= \hat{v}_h + L_2 \left(p_h - \hat{p}_h\right) \\ \dot{\hat{v}}_h &= A(\psi, \phi) f \eta + \hat{\xi}_h + L_1 \left(p_h - \hat{p}_h\right) \\ \dot{\hat{\xi}}_h &= L_0 \left(p_h - \hat{p}_h\right) \end{cases} \tag{5.20}$$

(5.21)

Defining the horizontal $e_h := p_h - \hat{p}_h$ and vertical $e_z := p_z - \hat{p}_z$ estimation errors, and following the idea in [97], these errors fulfill the following dynamical equations:

$$\frac{d^3 e_h}{dt^3} + L_2 \frac{d^2 e_h}{dt^2} + L_1 \frac{de_h}{dt} + L_0 e_h = \dot{\xi}_h$$
$$\frac{d^3 e_z}{dt^3} + l_2 \frac{d^2 e_z}{dt^2} + l_1 \frac{de_z}{dt} + l_0 e_z = \dot{\xi}_z$$

Therefore, taking L_0 to L_2 and l_0 to l_2 such that the following matrices A_h (defined below) is Hurwitz, the estimation error behaviour satisfies the ISS property (Input-to-State Stability) [98], *i.e.*, the solutions for the error dynamics converge to a sphere of radius ρ_h and ρ_z, respectively, centred at the origin of the estimation error phase space $\{e_h, \dot{e}_h, \ddot{e}_h\}$ and $\{e_z, \dot{e}_z, \ddot{e}_z\}$. In this sphere, the solutions remain bounded.

$$A_h = \begin{pmatrix} 0_{2\times 2} & I_{2\times 2} & 0_{2\times 2} \\ 0_{2\times 2} & 0_{2\times 2} & I_{2\times 2} \\ -L_0 & -L_1 & -L_2 \end{pmatrix}$$

Given a desired horizontal trajectory p_{h_d}, the horizontal position error dynamics $p_{h_e} = p_h - p_{h_d}$ and $v_{h_e} = v_h - v_{h_d}$ are given by:

$$\Sigma_{h_e} : \begin{cases} \dot{p}_{h_e} = v_{h_e} \\ \dot{v}_{h_e} = A(\psi, \phi) f \eta + \xi_h - \ddot{p}_{h_d} \end{cases} \tag{5.22}$$

For horizontal displacements, let us take:

$$\begin{pmatrix} \phi_d(t) \\ \theta_d(t) \end{pmatrix} = \frac{-1}{f} A^{-1}(\psi, \phi) \Big[\hat{\xi}_h - \ddot{p}_{h_d} + \begin{pmatrix} \sigma_{M_1^x}\left(a_1^x v_{x_e} + \sigma_{M_2^x}\left(a_2^x v_{x_e} + a_1^x a_2^x x_e\right)\right) \\ \sigma_{M_1^y}\left(a_1^y v_{y_e} + \sigma_{M_2^y}\left(a_2^y v_{y_e} + a_1^y a_2^y y_e\right)\right) \end{pmatrix} \Big] \tag{5.23}$$

where M_1^z, M_2^z, a_1^z and a_2^z are strictly positive parameters.

5.3.2 Numerical Simulation

Extensive simulations were made considering external disturbances and parametric uncertainties. We define and compare two control algorithms. The proposed control ADRC and a bounded control without ESO. The desired reference is a vertical helix which is modeled by

$$p_d(t) = \begin{bmatrix} 3\,(1+0.1t)\cos t/2 \\ 3\,(1+0.1t)\sin t/2 \\ 1+0.2t \end{bmatrix} \tag{5.24}$$

The physical parameters of the quadrotor are shown in the table C.1, as mentioned above the mass m and inertia J of the system have an inaccuracy of 30 %, so in the simulation is varied in that margin of error. The disturbance $\xi(t)$ it has been modeled with the following equation: 5.25.

$$\xi(t) = \begin{bmatrix} 0.45\cos 0.2t \sin 5t \\ -0.45\cos 0.2t \sin 5t \\ 0.1\cos 20t \sin 5t \end{bmatrix} \tag{5.25}$$

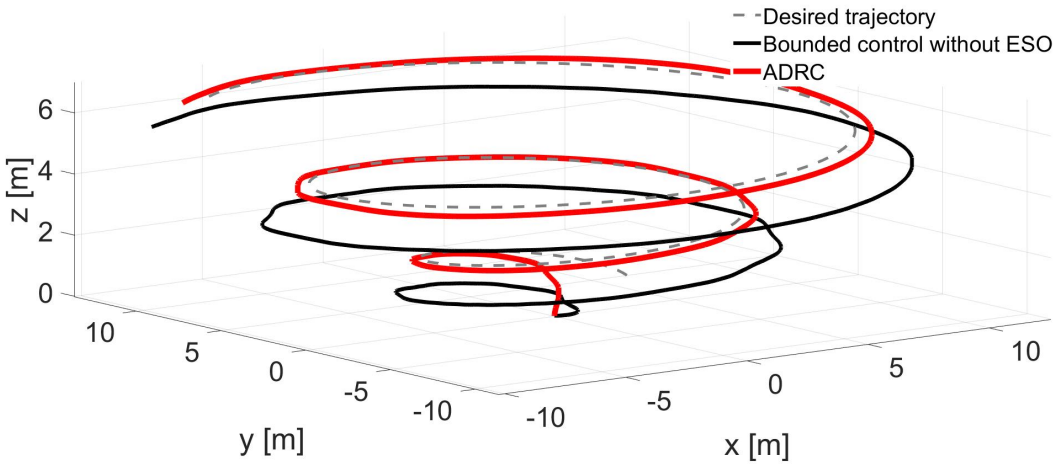


Fig. 5.16: Multirotor trajectory tracking.

The Fig. **??** and Fig. 5.16 shows the evolution of the quadrotor trajectory. The control algorithm shows a smooth response and a high degree of accuracy, and is robust enough to attenuate internal and external disturbances. The control design takes into account the error saturation in the position

and limits the allowable angles. For this application the angles were restricted to ±1 radian.

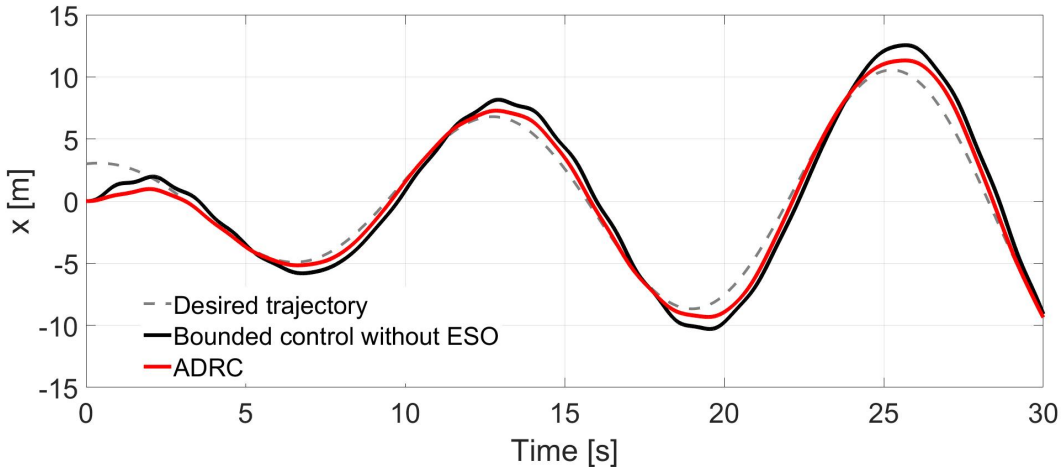


Fig. 5.17: Evolution of the position $p_x(t)$ of the multirotor.

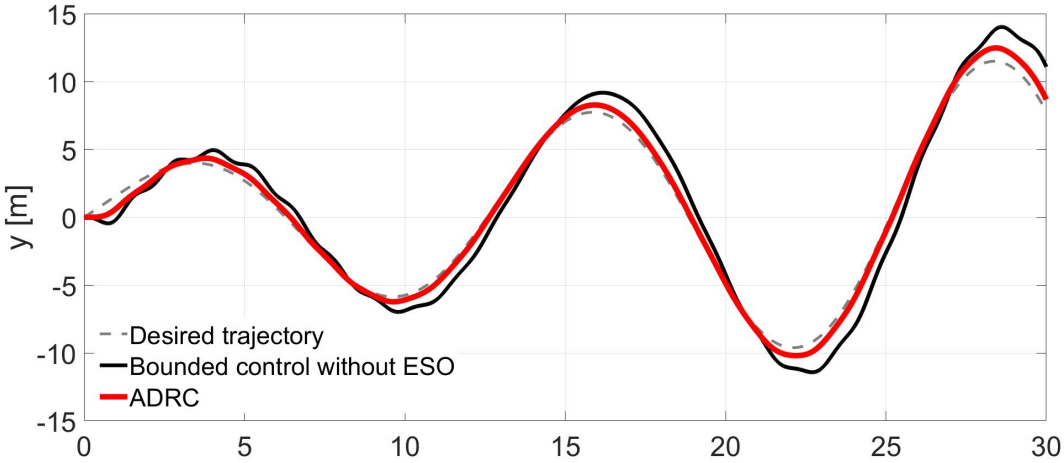


Fig. 5.18: Evolution of the position $p_y(t)$ of the multirotor.

5.3.3 Results

In this subsection, experimental results are performed where the robustness and effectiveness of the proposed IADRC control are demonstrated. The configuration of the flying arena is first described. Next, the trajectories carried out, the conditions and the disturbances used are explained. Different controls are evaluated under the integral of the error (ISE), and finally, the conclusions are given. The flight arena consists of an Optitrack (global sensing) motion capture system, a local server running the Motive software, a PC dedicated to the control algorithms made in Matlab/Simulink and

ROS communication capabilities, finally. , a Raspberry Pi single-chip computer for communication between the server, the PC and the mambo drones. Initially, the mambo pose is estimated by Optitrack motion capture. The speed is numerically differentiated, and the pose is published through a VRPN connection. The data is available to Matlab via ROS; the PC subscribes to this data, executes the control algorithm and publishes the data via the ROS-Matlab bridge at a sample rate of 100 Hz. Finally, the Raspberry Pi is in charge of sending the commands desired to mambo multirotors. Different physical tests have been carried out to demonstrate the robustness of the control. Two trajectories have been chosen, a circular path and a lemniscate path, both at low altitudes, to experience the ground effect; an external disturbance with a low-speed fan (3.5 m/s) has been added. The performance of the proposed control scheme has been compared with a PD and the classic ADRC.

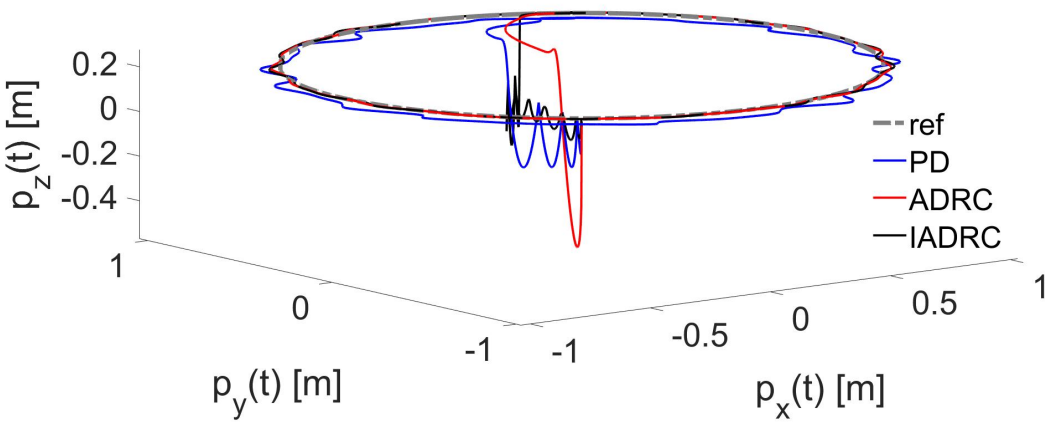


Fig. 5.19: Circle Path 3D.

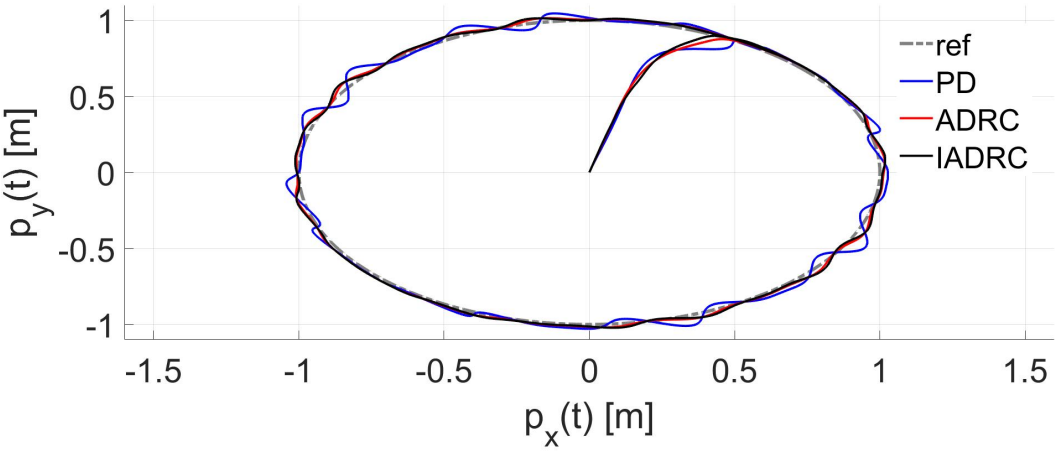


Fig. 5.20: Circle Path 2D.

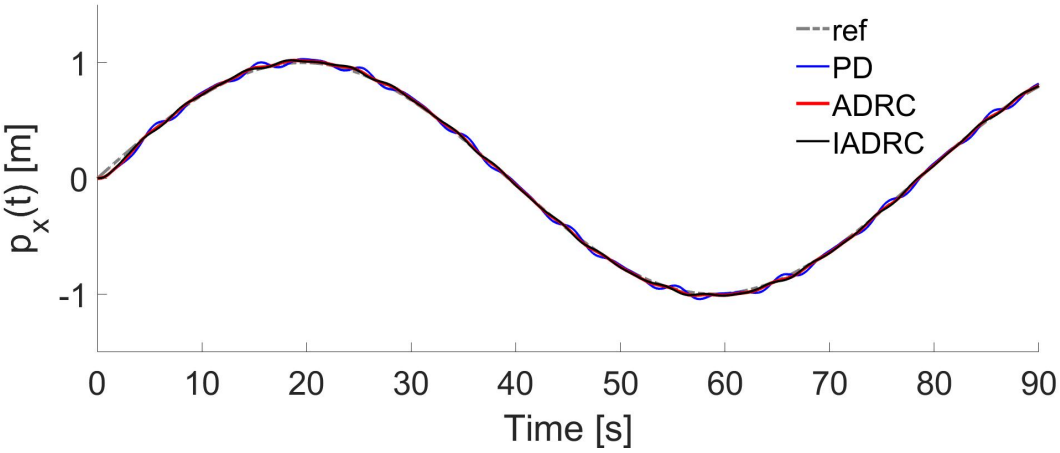


Fig. 5.21: Circle Path: Evolution of the position x $p_x(t)$.

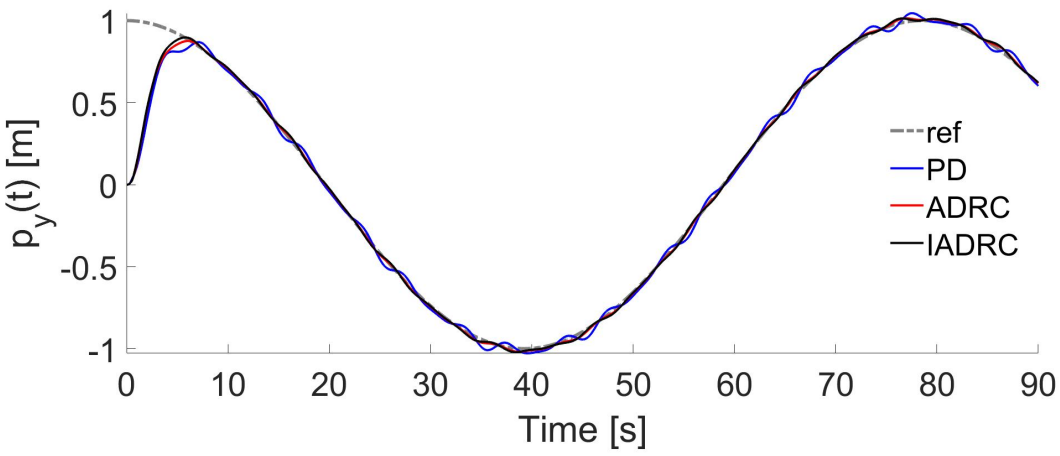


Fig. 5.22: Circle Path: Evolution of the position x $p_y(t)$.

The Figs 5.20, 5.21 and 5.22 show the evolution of the mambo multirotor's position variables in the plane and in 3D. As can be discerned, the PD control shows a highly oscillatory response and cannot deal with both disturbances. The classic ADRC approach can reject disturbances; it has better behaviour than the PD; however, in the takeoff manoeuvre, due to the ground effect, the disturbance is excellent, which estimates it and adds it to the saturating feedback function. To the actuators. In a physical implementation, the IADRC shows better behaviour since it does not saturate the actuators, maintaining a smooth response and without overshooting.

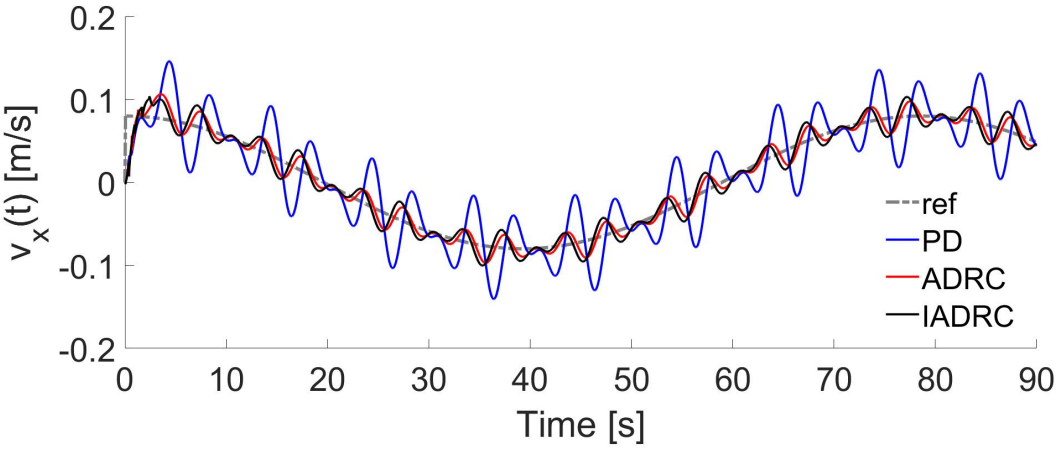


Fig. 5.23: Circle Path: Evolution of the velocity v $v_x(t)$.

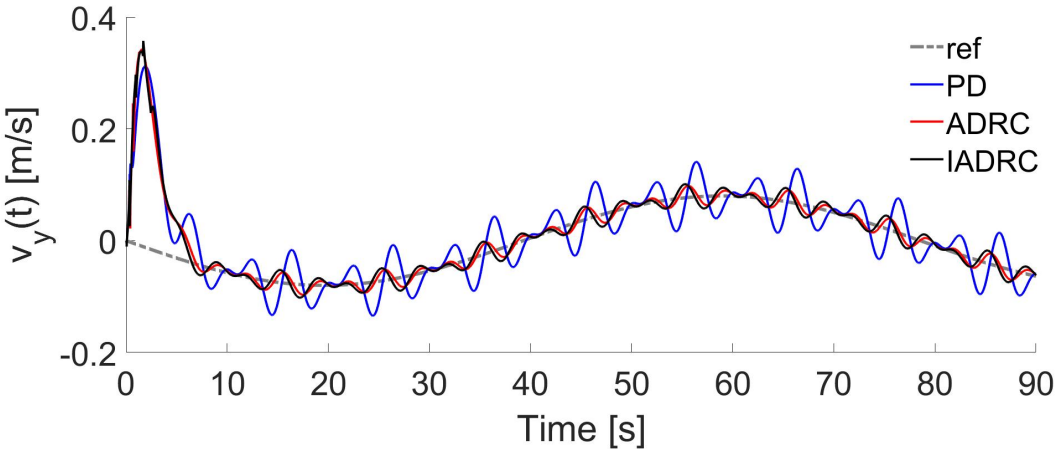


Fig. 5.24: Circle Path: Evolution of the velocity v $v_y(t)$.

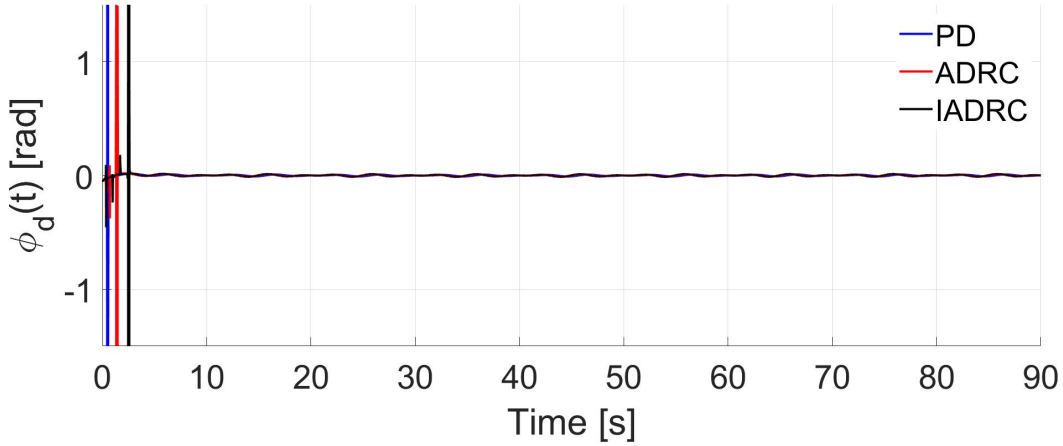


Fig. 5.25: Circle Path: Evolution of the desired command $\phi_d(t)$.

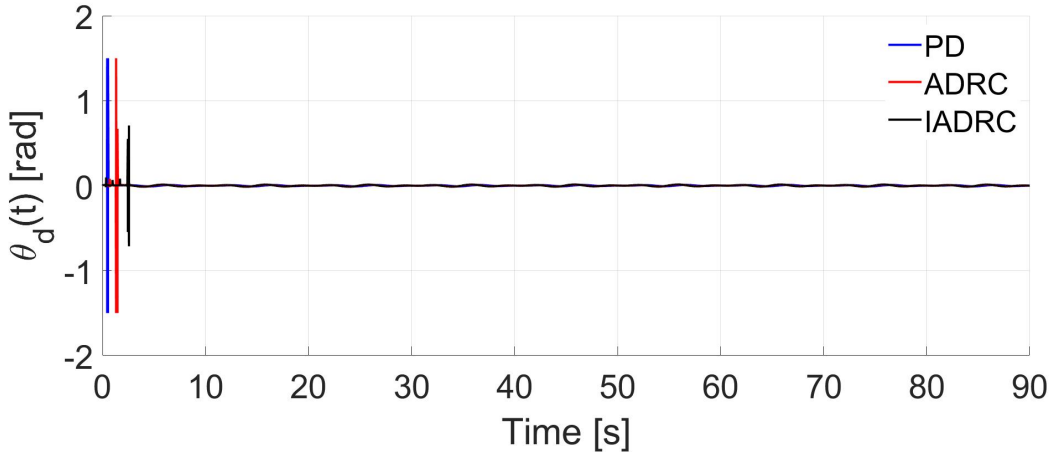


Fig. 5.26: Circle Path: Evolution of the desired command $\theta_d(t)$.

Figs 5.23, 5.24, 5.25 and 5.26 show the evolution of the linear velocities and the desired angles respectively. As we can see, the disturbances exert a deviation in the desired trajectory, affecting the position and orientation of the multirotor. In outdoor environments, it is tough to control the nature of these disturbances; it is, therefore, necessary to use a function based on the estimation error, which allows in a certain way to dose the injection of the same for the elimination of the disturbance.

Chapter 6

Experimental Platforms

I love those who can smile in trouble, who can gather strength from distress, and grow brave by reflection. Tis the business of little minds to shrink, but they whose heart is firm and whose conscience approves their conduct, will pursue their principles unto death.

– **Leonardo da Vinci**

6.1 Introduction

This chapter explains the development of the experimental platforms designed and implemented for the different tests carried out through the thesis research.

The flight arena has been developed for the study of heterogeneous autonomous robots. It comprises a motion capture system, a fleet of robots with wireless communication capabilities (wifi, Bluetooth, Zigbee, etc.), a server, a single chip computer and a middleware based on the Robotic Operating System (ROS). It is worth noting the ease of use, flexibility and modularity of the designed experimental platform. An endless number of experiments and demonstrations are carried out in conjunction with the thesis research.

6.2 Pixhawk and Hardware-in-the-loop Platform

6.2.1 Pixhawk

PX4 is a powerful open-source autopilot flight stack. It powers all vehicles, from racing and cargo drones to ground vehicles and submersibles. PX4 is a core part of a broader drone platform that includes the QGroundControl ground station, Pixhawk hardware, and the MAVSDK for integrating computers, cameras, and other add-on hardware using the MAVLink protocol.

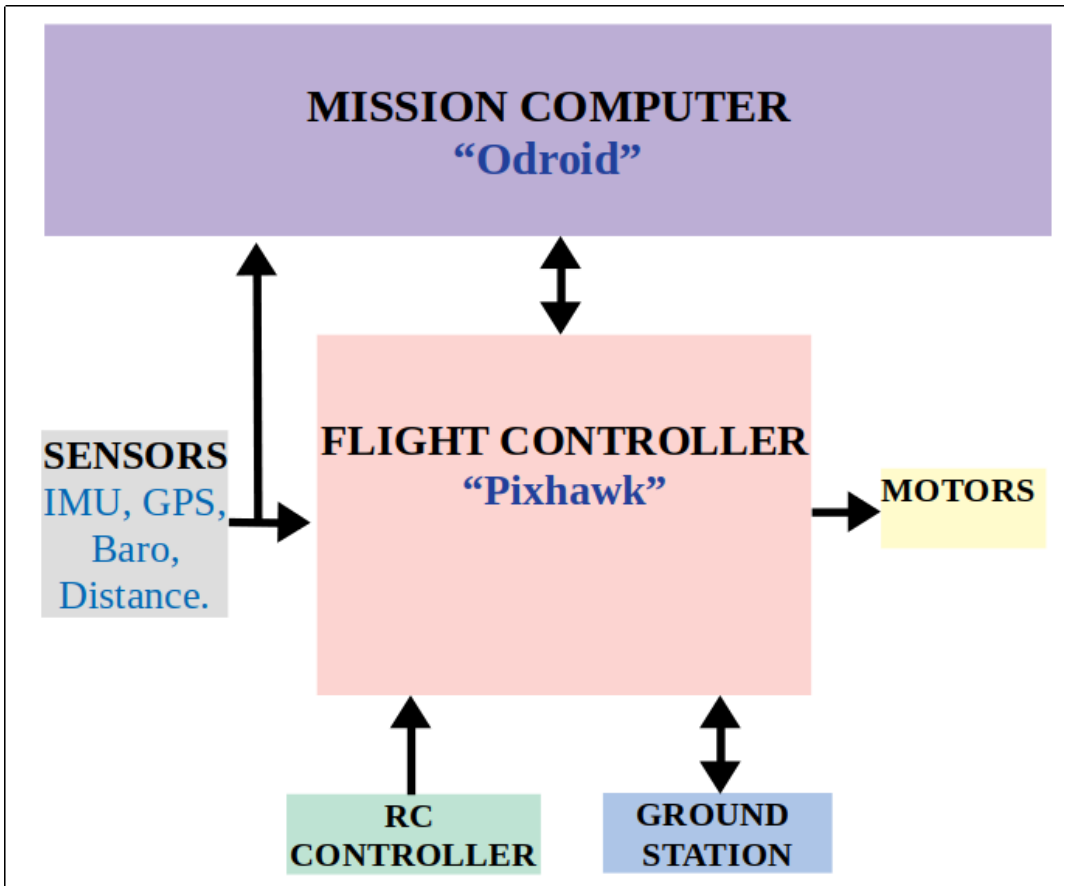


Fig. 6.1: Hardware in the loop VTOL-UAV.

6.2.2 Robotic Operating System (ROS)

Robot Operating System (ROS) is an open-source meta-operating system. Includes hardware abstraction, low-level device control, implementation of commonly used functions, message passing between processes, package management, device drivers, libraries, visualisation tools, messaging,

package management, and more. ROS provides libraries and tools to help software developers create robot applications.

1. **ROS nodes.** Process that use ROS APIs to perform computations.
2. **ROS master.** An intermediate program that connects ROS nodes.
3. **ROS topics.** Named buses in which ROS nodes can send a message. A node can publish or subscribe to any number of topics.
4. **ROS message.** The messages are going through the topic. Existing messages are based on primitive data types, and users can write messages.
5. **ROS service.** The ROS Service has a Request/Reply mechanism. A service call is a function which can call whenever a client node sends a request. The node that creates a service call is called the Server node, and whoever calls the service is called the client node.
6. **ROS package.** ROS is organised into packages. A package may contain a node, a library, a data set, or anything that can constitute a module.
7. **Roslaunch.** Roslaunch is a tool for quickly launching multiple ROS nodes locally and remotely via SSH and setting parameters on the Parameter Server. Roslaunch takes in one or more XML configuration files that specify the parameters to set and nodes to launch, as well as the machines on which they should be run.
8. **Gazebo.** The gazebo is an open-source 3D robotics simulator. Gazebo simulated real-world physics in a high-fidelity simulation. It helps developers rapidly test algorithms and design robots in digital environments. They've branded themselves as robotic simulation made easy, but many tools out there are much easier to use and help speed up the process.

6.2.3 Hardware in the loop

Hardware-in-the-Loop is a simulation mode in which average PX4 firmware is run on real flight controller hardware. This approach benefits from testing most of the actual flight code on real hardware. Hardware-in-the-Loop is a more accurate simulation of the behaviour of the aircraft in

actual flight. With HITL simulation, the PX4 firmware (control algorithm) runs on the virtual flight controller (Pixhawk mini). It communicates with a server or a PC running a simulator (Gazebo) via USB/UART, UDP, Ethernet, or WiFi. In addition, communication between the QGroundControl ground station has been added so that the user can have greater control of the mission to be carried out. A communication API between Matlab-ROS has also been integrated with possibilities for the design of control algorithms, navigation, and reinforced learning, among others. The simulation environment is described below in a simplified way; as shown in Fig. 6.2, this consists of four blocks: a Holybro Pixhak Mini autopilot, a 3D Gazebo simulator, a ground station, and the ROS robotic operating system.

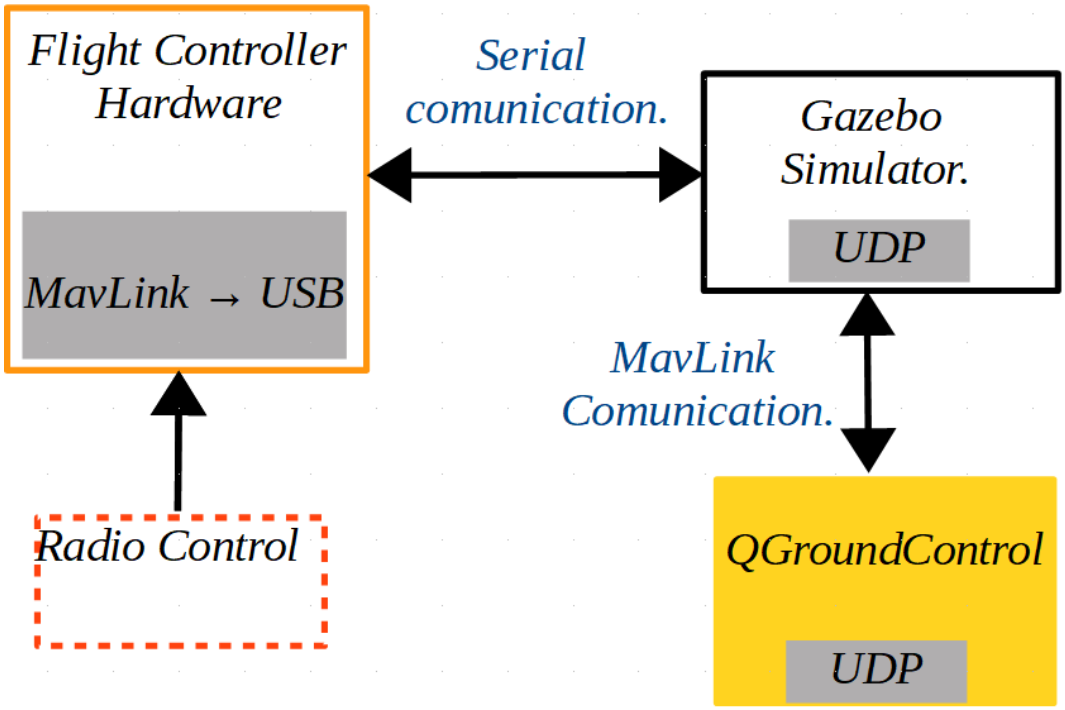


Fig. 6.2: Hardware in the loop VTOL-UAV.

QGroundControl allows monitoring and sending the references to the VTOL-UAV. It contains other utilities such as a route planner through waypoints, supports sending video to the aircraft, saving data for later analysis, etc. The Holybro Pixhawk mini autopilot has two 32-bit ARM processors, the central processor Cortex-M4 running Nuttx RTOS 180 MHz real-time operating system and the IO processor a Cortex-M3 72 MHz. Interfaces such as PPM, Futaba SBUS, UART, PWM, etc. A ROS robot operating system allows communication between the different blocks through nodes.

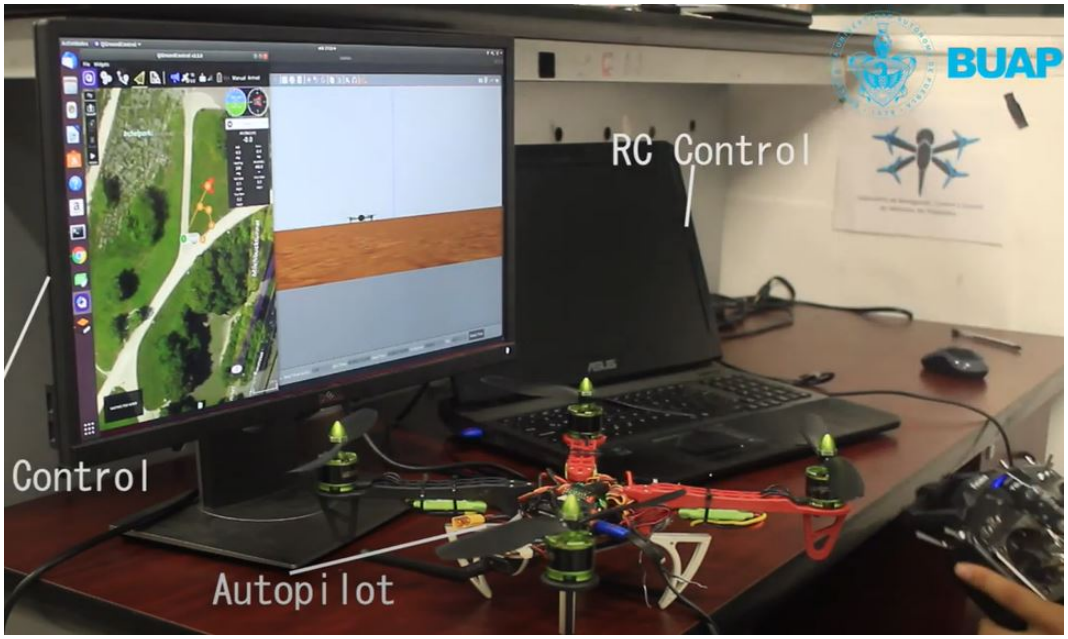


Fig. 6.3: Implementation of the Hardware in the Loop Platform.

The QGroundControl contains an interface to connect a joystick or an RC transmitter. It sends the desired commands to the autopilot; Gazebo models the dynamics of the aircraft in a simulated world, generating the states of the helicopter (linear position, linear velocity, angular position and angular velocity) that will be supplied to the autopilot. Once the control algorithm is programmed into the Pixhawk autopilot, the control signals are automatically calculated based on the signals received by the Gazebo simulator and QGroundControl. The reader can see the operation of the architecture in the following video `https://www.youtube.com/watch?v=AR46Oa-uU5`.

6.3 Mobile and Collaborative Robotics Platform

Indoor research test benches have been critical in developing new algorithms for control, navigation, design, the study of aerodynamic effects, robustness tests, learning schemes, etc. These environments make it possible to avoid the localisation problem and carry out experimentation immediately, safely and efficiently. An interior space of 15.6 cubic meters (2.5 m by 2.5 m by 2.5m) was equipped with four OptiTrack cameras, a server with Motive software, an SBC with Ubuntu Mate 20.04 LTS (ROS Noetic) operating system and a fleet of drones crazyflie 2.0, Parrot mambo drones, open architecture drones based on the Pixhawk autopilot and omnidirectional mobile robots, the latter

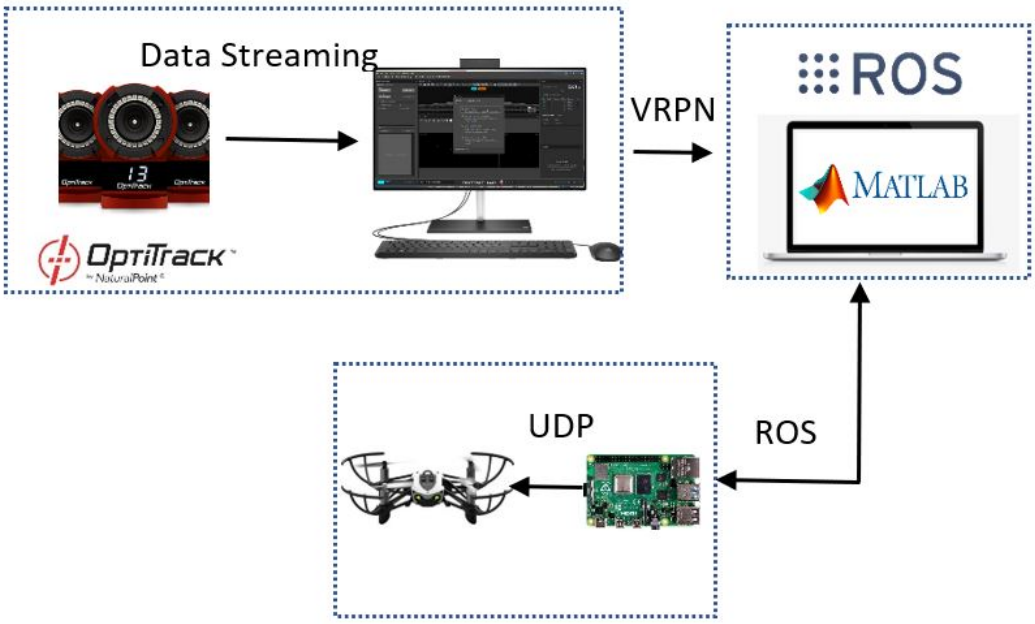


Fig. 6.4: General diagram of the experimental platform.

two made by ourselves. The control laboratory has motion capture cameras from the company Optritrack. OptiTrack includes a high-speed trail camera, motion capture software, and contract engineering services. They are used by facilities worldwide in various markets ranging from film, gaming, biomechanics, and air and ground robotics.

The OptiTrack system is used for its accuracy at the millimeter level to estimate any rigid body with six degrees of freedom. Data transmission through a local network is used. The multirotor's estimation of position, linear velocity, and attitude (quaternion) is sent to a ground station via the VRPN ROS network. The processing of the positions, the Euler angles, and the velocities estimation are carried out using MATLAB/Simulink with a sample rate of 100 Hz. The control algorithm is evaluated at this ground station and then sent to a single-board computer, Raspberry Pi, via ROS. Finally, the Raspberry Pi sends the control data to the drone using the same ROS network and UDP packets. Fig. C.11 shows the general diagram of the experimental platform.

Each of the systems that make up the flight arena are described in detail below.

1. **Mambo Drone.** The Parrot Mambo Drone is a mini drone designed for indoor handling. It has an X configuration, a dimension of 86x86 mm from motor to motor, its weight is 70g and it can reach a maximum speed of 30 km/h, these characteristics are summarized in Table

C.1.

Characteristic	Description
Name	Mambo
Dimensions	185 x 180 x 40 mm
Mass	70 grams
Flight speed	30km/h
Range	Up to 20m
Battery	550mAh

Table 6.1: Main characteristics of the Drone Mambo.

The mambo consists of different sensors. To estimate and control orientation, it uses an accelerometer and a gyroscope with three axes; for altitude control, it uses a pressure sensor and an ultrasound sensor. For horizontal stabilisation, this mini drone uses a low-resolution 60pps camera. It also contains Bluetooth and Wi-Fi, which approximates a radius of 20 meters for its correct operation.

2. **PC (Matlab/Simulink.)** A PC with windows operating system dedicated to the design of control algorithms in Matlab/Simulink with capabilities for communication with ROS has been used in the flight arena.

3. **Raspberry Pi 4, SBC.** A Single Board Computer (SBC) is a complete computer in which a single circuit board comprises memory, input/output, a microprocessor, and all other necessary features. A Raspberry Pi 4 with an Ubuntu Mate 20.04 LTS operating system dedicated to communication between the server, the PC and the heterogeneous robot assembly was used using the ROS middleware and User Datagram Protocol (UDP) communication.

Bibliography

[1] P. Chhikara, R. Tekchandani, N. Kumar, M. Guizani, and M. M. Hassan, "Federated learning and autonomous uavs for hazardous zone detection and aqi prediction in iot environment," *IEEE Internet of Things Journal*, vol. 8, no. 20, pp. 15456–15467, 2021.

[2] Y. Cao, X. Cheng, and J. Mu, "Concentrated coverage path planning algorithm of uav formation for aerial photography," *IEEE Sensors Journal*, vol. 22, no. 11, pp. 11098–11111, 2022.

[3] S. Sawadsitang, D. Niyato, P.-S. Tan, and P. Wang, "Joint ground and aerial package delivery services: A stochastic optimization approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2241–2254, 2019.

[4] P. K. Reddy Maddikunta, S. Hakak, M. Alazab, S. Bhattacharya, T. R. Gadekallu, W. Z. Khan, and Q.-V. Pham, "Unmanned aerial vehicles in smart agriculture: Applications, requirements, and challenges," *IEEE Sensors Journal*, vol. 21, no. 16, pp. 17608–17619, 2021.

[5] S. Sambolek and M. Ivasic-Kos, "Automatic person detection in search and rescue operations using deep cnn detectors," *IEEE Access*, vol. 9, pp. 37905–37922, 2021.

[6] R. Xu and U. Ozguner, "Sliding mode control of a quadrotor helicopter," in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 4957–4962, 2006.

[7] S. Norouzi Ghazbi, Y. Aghli, M. Alimohammadi, and A. A. Akbari, "Quadrotors unmanned aerial vehicles: A review," *International Journal on Smart Sensing and Intelligent Systems*, vol. 9, pp. 309–333, 03 2016.

[8] H. C. T. E. Fernando, A. T. A. De Silva, M. D. C. De Zoysa, K. A. D. C. Dilshan, and S. R. Munasinghe, "Modelling, simulation and implementation of a quadrotor uav," in *2013 IEEE 8th International Conference on Industrial and Information Systems*, pp. 207–212, 2013.

[9] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 2247–2252, 2005.

[10] G. Hoffmann, D. G. Rajnarayan, S. L. Waslander, D. Dostal, J. S. Jang, and C. J. Tomlin, "The stanford testbed of autonomous rotorcraft for multi agent control (starmac)," in *The 23rd Digital Avionics Systems Conference (IEEE Cat. No.04CH37576)*, vol. 2, pp. 12.E.4–121, 2004.

[11] P. Castillo, A. Dzul, and R. Lozano, "Real-time stabilization and tracking of a four-rotor mini rotorcraft," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 4, pp. 510–516, 2004.

[12] R. Mahony and J. Roberts, "Design of a four-rotor aerial robot," pp. 145–150, 11 2002.

[13] N. Guenard, T. Hamel, and V. Moreau, "Dynamic modeling and intuitive control strategy for an "x4-flyer"," pp. 141 – 146 Vol. 1, 07 2005.

[14] J. Nicoud and J.-C. Zufferey, "Toward indoor flying robots," vol. 1, pp. 787 – 792 vol.1, 02 2002.

[15] E. Altug, J. P. Ostrowski, and R. Mahony, "Control of a quadrotor helicopter using visual feedback," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 1, pp. 72–77 vol.1, 2002.

[16] S. Bouabdallah, P. Murrieri, and R. Siegwart, "Design and control of an indoor micro quadrotor," vol. 5, pp. 4393 – 4398 Vol.5, 01 2004.

[17] E. Altug, J. P. Ostrowski, and C. J. Taylor, "Quadrotor control using dual camera visual feedback," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 3, pp. 4294–4299 vol.3, 2003.

[18] S. Bouabdallah, A. Noth, and R. Siegwart, "Pid vs lq control techniques applied to an indoor micro quadrotor," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, pp. 2451–2456 vol.3, 2004.

[19] S. L. Waslander, G. M. Hoffmann, Jung Soon Jang, and C. J. Tomlin, "Multi-agent quadrotor testbed control design: integral sliding mode vs. reinforcement learning," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3712–3717, 2005.

[20] S. Bouabdallah, A. Noth, and R. Siegwart, "Pid vs lq control techniques applied to an indoor micro quadrotor," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, pp. 2451–2456 vol.3, 2004.

[21] B. Erginer and E. Altug, "Modeling and pd control of a quadrotor vtol vehicle," in *2007 IEEE Intelligent Vehicles Symposium*, pp. 894–899, 2007.

[22] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," 08 2007.

[23] A. Benallegue, A. Mokhtari, and L. Fridman, "Feedback linearization and high order sliding mode observer for a quadrotor uav," in *International Workshop on Variable Structure Systems, 2006. VSS'06.*, pp. 365–372, 2006.

[24] V. Mistler, A. Benallegue, and N. K. M'Sirdi, "Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback," in *Proceedings 10th IEEE International Workshop on Robot and Human Interactive Communication. ROMAN 2001 (Cat. No.01TH8591)*, pp. 586–593, 2001.

[25] Y. Yali, SunFeng, and W. Yuanxi, "Controller design of quadrotor aerial robot," *Physics Procedia*, vol. 33, pp. 1254–1260, 2012. 2012 International Conference on Medical Physics and Biomedical Engineering (ICMPBE2012).

[26] J. Escareno, S. Salazar-Cruz, and R. Lozano, "Embedded control of a four-rotor uav," in *2006 American Control Conference*, pp. 6 pp.–, 2006.

[27] P. Castillo, A. Dzul, and R. Lozano, "Real-time stabilization and tracking of a four-rotor mini rotorcraft," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 4, pp. 510–516, 2004.

[28] M. Valenti, B. Bethke, G. Fiore, J. How, and E. Feron, "Indoor multi-vehicle flight testbed for fault detection, isolation, and recovery," 08 2006.

[29] S. Lupashin, A. Schöllig, M. Hehn, and R. D'Andrea, "The flying machine arena as of 2010," in *2011 IEEE International Conference on Robotics and Automation*, pp. 2970–2971, 2011.

[30] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The grasp multiple micro-uav testbed," *IEEE Robotics Automation Magazine*, vol. 17, no. 3, pp. 56–65, 2010.

[31] D. Floreano and R. Wood, "Science, technology and the future of small autonomous drones," *Nature*, vol. 521, pp. 460–6, 05 2015.

[32] M. Labbadi, M. Cherkaoui, Y. E. houm, and M. Guisser, "Modeling and robust integral sliding mode control for a quadrotor unmanned aerial vehicle," in *2018 6th International Renewable and Sustainable Energy Conference (IRSEC)*, pp. 1–6, 2018.

[33] K. Sreenath, T. Lee, and V. Kumar, "Geometric control and differential flatness of a quadrotor uav with a cable-suspended load," in *52nd IEEE Conference on Decision and Control*, pp. 2269–2274, 2013.

[34] F. Goodarzi, D. Lee, and T. Lee, "Geometric nonlinear pid control of a quadrotor uav on se(3)," in *2013 European Control Conference (ECC)*, pp. 3845–3850, 2013.

[35] F. Goodarzi, D. Lee, and T. Lee, "Geometric adaptive tracking control of a quadrotor unmanned aerial vehicle on se(3) for agile maneuvers," *Journal of Dynamic Systems Measurement and Control*, vol. 137, 06 2015.

[36] B. Mu, Y. Pei, and Y. Shi, "Integral sliding mode control for a quadrotor in the presence of model uncertainties and external disturbances," in *2017 American Control Conference (ACC)*, pp. 5818–5823, 2017.

[37] G. Perozzi, D. Efimov, J.-M. Biannic, and L. Planckaert, "Trajectory tracking for a quadrotor under wind perturbations: sliding mode control with state-dependent gains," *Journal of the Franklin Institute*, vol. 355, no. 12, pp. 4809–4838, 2018.

[38] B. Wang, L. Mu, and Y. Zhang, "Adaptive robust tracking control of quadrotor helicopter with parametric uncertainty and external disturbance," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 402–407, 2017.

[39] A. L'Afflitto, R. B. Anderson, and K. Mohammadi, "An introduction to nonlinear robust control for unmanned quadrotor aircraft: How to design control algorithms for quadrotors using sliding mode control and adaptive control techniques [focus on education]," *IEEE Control Systems Magazine*, vol. 38, no. 3, pp. 102–121, 2018.

[40] A. Castillo Frasquet, R. Sanz, and P. García, "Disturbance observer-based quadrotor attitude tracking control for aggressive maneuvers," *Control Engineering Practice*, vol. 82, pp. 14–23, 09 2018.

[41] M. Ramp and E. Papadopoulos, "Geometric surface-based tracking control of a quadrotor uav for aggressive maneuvers," in *2018 26th Mediterranean Conference on Control and Automation (MED)*, pp. 1–6, 2018.

[42] S. Tang, V. Wüest, and V. Kumar, "Aggressive flight with suspended payloads using vision-based control," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1152–1159, 2018.

[43] Y. Chen and N. O. Pérez-Arancibia, "Controller synthesis and performance optimization for aerobatic quadrotor flight," *IEEE Transactions on Control Systems Technology*, pp. 1–16, 2019.

[44] N. Raj, R. N. Banavar, Abhishek, and M. Kothari, "Robust attitude tracking for aerobatic helicopters: A geometric approach," *IEEE Transactions on Control Systems Technology*, pp. 1–15, 2020.

[45] A. Pulido-Flores, J. F. Guerrero-Castellanos, J. Linares-Flores, S. E. Maya-Rueda, J. U. Alvarez-Muñoz, J. Escareno, and G. Mino-Aguilar, "Active disturbance rejection control for

attitude stabilization of multi-rotors uavs with bounded inputs," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1181–1188, 2018.

[46] Y. Ping, J. Qi, and C. Wu, "Quadrotor transporting cable suspended load using adrc," in *2019 Chinese Control Conference (CCC)*, pp. 8008–8013, 2019.

[47] X. Liang and Y. Hu, "Tracking control and differential flatness of quadrotor with cable-suspended load," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 88–92, 2019.

[48] A. Aboudonia, A. El-Badawy, and R. Rashad, "Active anti-disturbance control of a quadrotor unmanned aerial vehicle using the command-filtering backstepping approach," *Nonlinear Dynamics*, vol. 90, 10 2017.

[49] H. Sira-Ramirez, A. Luviano-Juárez, M. Ramirez-Neria, and E. Zurita-Bustamante, *Active Disturbance Rejection Control of Dynamic Systems: A flatness based approach*. 10 2018.

[50] L. Guo and S. Cao, "Anti-disturbance control theory for systems with multiple disturbances: A survey," *ISA Transactions*, vol. 53, no. 4, pp. 846–849, 2014. Disturbance Estimation and Mitigation.

[51] Y. Chen, G. Zhang, Y. Zhuang, and H. Hu, "Autonomous flight control for multi-rotor uavs flying at low altitude," *IEEE Access*, vol. 7, pp. 42614–42625, 2019.

[52] A. Castillo, R. Sanz, P. Garcia, W. Qiu, H. Wang, and C. Xu, "Disturbance observer-based quadrotor attitude tracking control for aggressive maneuvers," *Control Engineering Practice*, vol. 82, pp. 14–23, 2019.

[53] A. Asignacion, S. Suzuki, R. Noda, T. Nakata, and H. Liu, "Frequency-based wind gust estimation for quadrotors using a nonlinear disturbance observer," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9224–9231, 2022.

[54] L. R. Garcia Carrillo, A. Dzul, R. Lozano, and C. Pégard, *Quad Rotorcraft Control. Vision-Based Hovering and Navigation*. 01 2012.

[55] L. V.-P. J.Díaz-Téllez, J.F. Guerrero-Castellanos, "Robust attitude control of mult-rotors for aerial manipulation," 01 2019.

[56] D. M. M. E. S. Richard K. Barnhart, Stephen B. Hottman, *Unmanned Aircraft Systems*. 01 2005.

[57] J. Stafford, "Implementing precision agriculture in the 21st century," *Journal of Agricultural Engineering Research*, vol. 76, pp. 267–275, 07 2000.

[58] G. Rohi, O. Ejofodomi, and G. Ofualagba, "Autonomous monitoring, analysis, and countering of air pollution using environmental drones," *Heliyon*, vol. 6, p. e03252, 01 2020.

[59] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 20–32, 2012.

[60] Y. Zou and K. Xia, "Robust fault-tolerant control for underactuated takeoff and landing uavs," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 5, pp. 3545–3555, 2020.

[61] W. Chung and H. Son, "Fault-tolerant control of multirotor uavs by control variable elimination," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 5, pp. 2513–2522, 2020.

[62] B. Wang and Y. Zhang, "An adaptive fault-tolerant sliding mode control allocation scheme for multirotor helicopter subject to simultaneous actuator faults," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4227–4236, 2018.

[63] B. Wang, X. Yu, L. Mu, and Y. Zhang, "Disturbance observer-based adaptive fault-tolerant control for a quadrotor helicopter subject to parametric uncertainties and external disturbances," *Mechanical Systems and Signal Processing*, vol. 120, pp. 727–743, 2019.

[64] J. Preminger and J. Rootenberg, "Some considerations relating to control systems employing the invariance principle," *IEEE Transactions on Automatic Control*, vol. 9, no. 3, pp. 209–215, 1964.

[65] J. Dou, X. Kong, and B. Wen, "Altitude and attitude active disturbance rejection controller design of a quadrotor unmanned aerial vehicle," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 231, no. 9, pp. 1732–1745, 2017.

[66] Y. Guo, B. Jiang, and Y. Zhang, "A novel robust attitude control for quadrotor aircraft subject to actuator faults and wind gusts," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 292–300, 2018.

[67] J. C. González-Guerrero, J. Díaz-Téllez, J. Estevez-Carreon, R. Mendoza-Vazquez, M. Meraz-Melo, and J. F. Guerrero-Castellanos, "Low altitude control of the vtol uav tolerant to ground effect and actuator failures," in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1504–1509, 2022.

[68] H. Chang, Y. Liu, Y. Wang, and X. Zheng, "A modified nonlinear dynamic inversion method for attitude control of uavs under persistent disturbances," in *2017 IEEE International Conference on Information and Automation (ICIA)*, pp. 715–721, 2017.

[69] A. Pulido-Flores, J. F. Guerrero-Castellanos, J. Linares-Flores, S. E. Maya-Rueda, J. U. Alvarez-Muñoz, J. Escareno, and G. Mino-Aguilar, "Active disturbance rejection control for attitude stabilization of multi-rotors uavs with bounded inputs," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1181–1188, 2018.

[70] H. Yang, L. Cheng, Y. Xia, and Y. Yuan, "Active disturbance rejection attitude control for a dual closed-loop quadrotor under gust wind," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 4, pp. 1400–1405, 2018.

[71] M. A. Lotufo, L. Colangelo, C. Perez-Montenegro, E. Canuto, and C. Novara, "Uav quadrotor attitude control: An adrc-emc combined approach," *Control Engineering Practice*, vol. 84, pp. 13–22, 2019.

[72] J. C. González-Guerrero, J. Díaz-Téllez, J. Estevez-Carreon, R. Mendoza-Vazquez, M. Meraz-Melo, and J. F. Guerrero-Castellanos, "Low altitude control of the vtol uav tolerant to ground

effect and actuator failures," in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1504–1509, 2022.

[73] I. C. Cheeseman, P. D, W. E. Bennett, P. D, and W. E. Bennett, "The effect of ground on a helicopter rotor in forward flight," tech. rep., 1955.

[74] J. S. Hayden, "The effect of the ground on helicopter hovering power required," 1976.

[75] L. Danjun, Z. Yan, S. Zongying, and L. Geng, "Autonomous landing of quadrotor based on ground effect modelling," in *2015 34th Chinese Control Conference (CCC)*, pp. 5647–5652, 2015.

[76] X. Kan, J. Thomas, H. Teng, H. G. Tanner, V. Kumar, and K. Karydis, "Analysis of ground effect for small-scale uavs in forward flight," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3860–3867, 2019.

[77] I. Sharf, M. Nahon, A. Harmat, W. Khan, M. Michini, N. Speal, M. Trentini, T. Tsadok, and T. Wang, "Ground effect experiments and model validation with draganflyer x8 rotorcraft," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1158–1166, 2014.

[78] T. Ryan, "Modelling of quadrotor ground effect forces via simple visual feedback and support vector regression," 08 2012.

[79] E. Davis and P. E. I. Pounds, "Passive position control of a quadrotor with ground effect interaction," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 539–545, 2016.

[80] T. H. Anh, N. T. Binh, and J. W. Song, "In-ground-effect model based adaptive altitude control of rotorcraft unmanned aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 794–801, 2022.

[81] N. Ahmed, A. Raza, S. A. A. Shah, and R. Khan, "Robust composite-disturbance observer based flight control of quadrotor attitude," *Journal of Intelligent & Robotic Systems*, vol. 103, sep 2021.

[82] Y. Younes, A. Drak, H. Noura, A. Rabhi, and A. El hajjaji, "Robust model-free control applied to a quadrotor uav," *Journal of Intelligent & Robotic Systems*, vol. 84, 12 2016.

[83] M. Kothari, I. Postlethwaite, and D.-W. Gu, "Uav path following in windy urban environments," *Journal of Intelligent & Robotic Systems*, vol. 74, pp. 1013–1028, 06 2013.

[84] K. Guo, W. Zhang, Y. Zhu, J. Jia, X. Yu, and Y. Zhang, "Safety control for quadrotor uav against ground effect and blade damage," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 12, pp. 13373–13383, 2022.

[85] X. He, G. Kou, M. Calaf, and K. Leang, "In-ground-effect modeling and nonlinear disturbance observer for multi-rotor uav control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 141, 03 2019.

[86] D. Shi, Z. Wu, and W. Chou, "Anti-disturbance trajectory tracking of quadrotor vehicles via generalized extended state observer," *Journal of Vibration and Control*, vol. 26, no. 13-14, pp. 1173–1186, 2020.

[87] C. Y. Son, H. Seo, D. Jang, and H. J. Kim, "Real-time optimal trajectory generation and control of a multi-rotor with a suspended load for obstacle avoidance," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1915–1922, 2020.

[88] B. Xian, S. Wang, and S. Yang, "An online trajectory planning approach for a quadrotor uav with a slung payload," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 8, pp. 6669–6678, 2020.

[89] D. Allahverdy, A. Fakharian, and M. B. Menhaj, "Back-stepping integral sliding mode control with iterative learning control algorithm for quadrotor uav transporting cable-suspended payload," in *2021 29th Iranian Conference on Electrical Engineering (ICEE)*, pp. 660–665, 2021.

[90] R. Hedjar and M. A. Al Zuair, "Robust altitude stabilization of vtol-uav for payloads delivery," *IEEE Access*, vol. 7, pp. 73583–73592, 2019.

[91] K. Guo, J. Jia, X. Yu, L. Guo, and L. Xie, "Multiple observers based anti-disturbance control for a quadrotor uav against payload and wind disturbances," *Control Engineering Practice*, vol. 102, p. 104560, 2020.

[92] M. M. Nicotra, E. Garone, R. Naldi, and L. Marconi, "Nested saturation control of an uav carrying a suspended load," in *2014 American Control Conference*, pp. 3585–3590, 2014.

[93] M. E. Guerrero, D. A. Mercado, R. Lozano, and C. D. García, "Passivity based control for a quadrotor uav transporting a cable-suspended payload with minimum swing," in *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 6718–6723, 2015.

[94] A. Martinez-Vasquez, R. Castro-Linares, and A. Rodriguez-Mata, "Sliding mode control of a quadrotor with suspended payload: a differential flatness approach," in *2020 17th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 1–6, 2020.

[95] J. Guerrero-Castellanos, H. Rifaï, V. Arnez-Paniagua, J. Linares-Flores, L. Saynes-Torres, and S. Mohammed, "Robust active disturbance rejection control via control lyapunov functions: Application to actuated-ankle–foot-orthosis," *Control Engineering Practice*, vol. 80, pp. 49 – 60, 2018.

[96] T. Tomić and S. Haddadin, "Simultaneous estimation of aerodynamic and contact forces in flying robots: Applications to metric wind estimation and collision detection," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5290–5296, 2015.

[97] J. F. Guerrero-Castellanos, S. Durand, G. A. Munoz-Hernandez, N. Marchand, L. L. G. Romeo, J. Linares-Flores, G. Mino-Aguilar, and W. F. Guerrero-Sánchez, "Bounded attitude control with active disturbance rejection capabilities for multirotor uavs," *Applied Sciences*, vol. 11, no. 13, 2021.

[98] E. Sontag and Y. Wang, "On characterizations of the input-to-state stability property," *Systems and Control Letters*, vol. 24, pp. 351–359, 1995.

Appendix A

Quaternion

The quaternions were discovered by the Irish mathematician William Rowan Hamilton. A quaternion is a four-element vector which consists of one real part and three imaginary parts, i, j and k. Furthermore, these imaginary elements satisfy the following condition.

$$i^2 = j^2 = k^2 = ijk = -1 \tag{A.1}$$

Due to the advancement of computational resources in microcontrollers, digital signal processes and reconfigurable devices, quaternions have received more attention from the scientific community to perform high-performance tasks, aggressive manoeuvres, etc.

Basic concepts and operations are introduced to understand the use of quaternions to represent the aircraft's attitude.

The field of quaternions is written as

$$\mathbb{H} = \{q = q_0 + q_1 i + q_2 j + q_3 k\} \tag{A.2}$$

We can also rewrite the quaternions as a vector

$$\mathbf{q} = \begin{bmatrix} q_0 \\ \mathbf{q}_v \end{bmatrix} \tag{A.3}$$

where $q_0 \in \Re$ is the scalar part of $\mathbf{q} \in \Re^4$ and $\mathbf{q}_v = \begin{bmatrix} q_x & q_y & q_z \end{bmatrix}^T \in \Re^3$ is the vector part. There are many operations between quaternions. However, we will only focus on the processes that will be used for the design of the control algorithms.

1. Multiplication

   The multiplication between two quaternions is

$$\mathbf{pq} = p_0 q_0 - (p_1 q_1 + p_2 q_2 + p_3 q_3) + (p_0 q_1 + p_1 q_0 + p_2 q_3 - p_3 q_2) i +$$

$$+ (p_0 q_2 - p_1 q_3 + p_2 q_0 + p_3 q_1) j + (p_0 q_3 + p_1 q_2 - p_2 q_1 + p_3 q_0) k$$

   represented in matrix form as the product of two matrices

$$\mathbf{pq} = \begin{bmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & -p_3 & p_2 \\ p_2 & p_3 & p_0 & -p_1 \\ p_3 & -p_2 & p_1 & p_0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{A.4}$$

   in a more graceful way we have the product of two quaternions as

$$\mathbf{p} \otimes \mathbf{q} = \begin{bmatrix} p_0 \\ \mathbf{p}_v \end{bmatrix} \otimes \begin{bmatrix} q_0 \\ \mathbf{q}_v \end{bmatrix} = \begin{bmatrix} p_0 q_0 - \mathbf{q}_v^T \mathbf{p}_v \\ \mathbf{p}_v \times \mathbf{q}_v + p_0 \mathbf{q}_v + q_0 \mathbf{p}_v \end{bmatrix} \tag{A.5}$$

   the quaternion multiplication is not commutative, but it is distributive and associative.

2. Conjugate

   The conjugate of a quaternion is defined by

$$\mathbf{q}^* = \begin{bmatrix} q_0 \\ -\mathbf{q}_v \end{bmatrix} \tag{A.6}$$

3. Norm

   The norm of a quaternion is defined by

$$\|\mathbf{q}\|^2 = \|\mathbf{q} \otimes \mathbf{q}^*\| = q_0^2 + \mathbf{q}_v^T \mathbf{q}_v \tag{A.7}$$

4. Inverse

   The inverse quaternion $\mathbf{q}^{-1}$ satisfies

$$\mathbf{q} \otimes \mathbf{q}^{-1} = \begin{bmatrix} 1 \\ \mathbf{0}_{3\times 1} \end{bmatrix} \tag{A.8}$$

   the inverse quaternion can be calculated as

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{\|\mathbf{q}\|} \tag{A.9}$$

5. Unit quaternion

$$\|\mathbf{p} \otimes \mathbf{q}\| = 1\mathbf{q}^{-1} = \mathbf{q}^* \tag{A.10}$$

6. Derivative of the Rotation Matrix

   The derivative of a vector $\mathbf{r}_e \in \Re^3$ which is being rotated by an angular velocity $\omega = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^T$ can be expressed as

$$\frac{d\mathbf{r}_e}{dt} = \omega \times \mathbf{r}_e \tag{A.11}$$

therefore, the derivative is a vector perpendicular to ω and $\mathbf{r}_e$, and normal to the plane that contains them. Let $\mathbf{R}$ be a rotation matrix where $\mathbf{R} \in$ SO3, then using the properties of the cross product the derivative of the rotation matrix is

$$\frac{d\mathbf{R}}{dt} = \mathbf{R}[\omega]_\times \tag{A.12}$$

where $[\omega]_\times$ is defined as

$$[\omega]_\times = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{A.13}$$

is a skew symmetric matrix .

Appendix B

Business Model.

B.1 Description of the Project.

A project based on intelligent agriculture is proposed to manage high-value resources in the field. Its objectives are the study of the soil, optimisation of resources (fertilisers, irrigation, compost, etc.), prior fertilisation and increased productivity. For this purpose, different technologies are used, such as control, artificial intelligence, embedded system design, communication systems and the Internet of Things. In this project's first stage, the realisation of a Drone with aerial manipulation capabilities and active rejection of disturbances is projected. Data acquisition through intelligent sensors in the drone (hyperspectral cameras, C02, temperature, humidity, wind speed, etc.) for the study and management of crops focusing on corn, cane and potatoes. The preceding is based on the UN Agenda for Sustainable Development 2030, the National Development Plan 2019-2024, the State Development Plan 2019-24 and current regulations.

B.1.1 The objective of the project.

Development of intelligent technologies for improving, optimising and automating production processes in the agricultural field.

B.1.2 Scopes.

The project is divided into different stages. The first stage involves developing a robust open-architecture multirotor with aerial manipulation capabilities such as pesticide spraying, automatic seeders and different robotic mechanisms. Also incorporating the active rejection of disturbances, sophisticated accompanying computations, IoT sensors and hyperspectral cameras. The functions that the multirotor will have are:

1. Mapping of agricultural fields.
2. Crop surveillance and monitoring.
3. Pest and disease monitoring.
4. Harvest Yield Estimation.
5. Leaf area index.

B.1.3 Problem to be addressed.

The problem to be addressed is the low productivity of the production units in the Pueblan countryside. This is mainly due to low investment by the government, overpopulation, abandonment of the country, inflation and international price escalation. It is, therefore, necessary to seek technological solutions to achieve food sovereignty security in Mexico. As the population and urban areas are increasing, inequality rates are high, and poverty, malnutrition, and hunger are still present; therefore, the food system must meet the needs of all and all sustainably. , seeking health, integrity, the economy, and the preservation of agrobiodiversity and the environment.

B.2 Canvas.

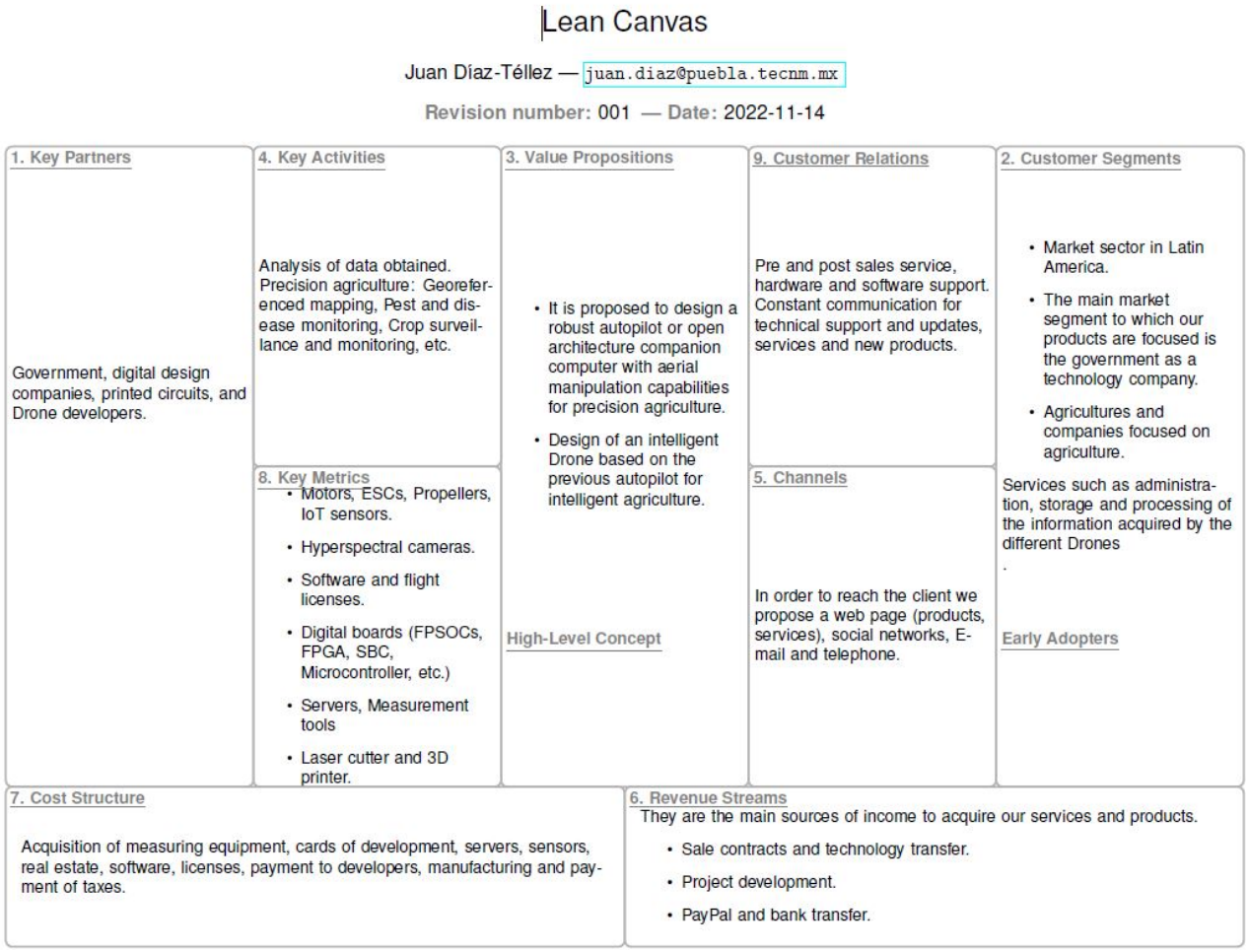


Fig. B.1: Canvas Model.

Appendix C

Manual Técnico de la arena de vuelo del laboratorio de control avanzado y sistemas ciber-físicos

En este capítulo se explica el desarrollo de la plataforma experimental diseñada. La arena de vuelo ha sido desarrollada para el estudio de algoritmos de control, aprendizaje automático y navegación de robots heterogéneos. La plataforma experimental tiene una arquitectura abierta, lo cual da la posibilidad a que cualquier robot manipulador, terrestre o aéreo con capacidades de comunicación ROS pueda integrarse a la arena de vuelo. Contiene diferentes software especializados que hacen el rápido prototipado, facilidad de uso, flexibilidad y modularidad. La plataforma diseñada se compone de un sistema de captura de movimiento Optitrack, robots heterogéneos con capacidades de comunicación inalámbrica, un servidor y una computadora de un solo chip. Todos los componentes se encuentran conectados mediante el middleware ROS.

C.1 Conceptos

Se introducen algunos conceptos para el entendimiento de la arquitectura de la plataforma experimental. Iniciamos con los software de captura de movimiento, enseguida explicamos el sistema

operativo robótico (ROS) y finalmente algunos términos utilizados en ROS.

C.1.1 Optitrack

$OptiTrack^{TM}$ es el proveedor de captura de movimiento más grande del mundo y ofrece seguimiento óptico de alto rendimiento a los precios más asequibles de la industria. La línea de productos OptiTrack incluye software de captura de movimiento y cámaras de seguimiento de alta velocidad, así como servicios de ingeniería por contrato. Utilizado por instalaciones en todo el mundo en una variedad de mercados que van desde películas y juegos hasta entrenamiento deportivo y biomecánica, los clientes de OptiTrack incluyen lo mejor en sus respectivos campos `https://optitrack.com/`.

C.1.2 Motive

Motive es una plataforma de software diseñada para controlar sistemas de captura de movimiento. Motive permite calibrar, configurar el sistema, crear cuerpos rígidos, transmitir datos entre otros. Además proporciona interfaces para capturar y procesar datos mientras sigue las coordenadas 3D de los marcadores en tiempo real. Motive permite reconstruir múltiples cuerpos rígido para permitir el seguimiento de movimientos en el espacio 3D utilizando marcadores y el proceso de reconstrucción. Los datos capturados se pueden grabar o transmitir en tiempo real a otro sistema como un servidor, robot o computadora en un solo chip.

C.1.3 Sistema Operativo Robótico ROS

Sistema Operativo Robótico (ROS) es un meta-sistema operativo de código abierto. Como cualquier sistema operativo este incluye abstracción de hardware, control de dispositivos de bajo nivel, implementación de funciones de uso común, paso de mensajes entre procesos, administración de paquetes, controladores de dispositivos, bibliotecas, herramientas de visualización, mensajería, administración de paquetes y más. ROS proporciona bibliotecas y herramientas para ayudar a los desarrolladores de software a crear aplicaciones robóticas, soporta una variedad de hardware y lenguajes de programación. Ros tiene su propio acervo digital (wiki), el cual cuenta con una amplia documentación del sistema. Para mayor información se recomienda visitar la página oficial de `http://wiki.ros.org`.

1. **Nodo (node).** Unidad más pequeña de procesamiento en ROS. Este proceso utiliza las interfaces de programación de aplicaciones (API) de ROS para realizar cálculos o algún tipo de computación. Los nodos se combinan dentro de un grafo, compartiendo información entre ellos de forma paralela para crear tareas complejas.

2. **Nodo maestro (Master).** Un programa intermedio que conecta nodos y servicios de ROS, ademas de que administra la comunicación.

3. **Paquete (Package).** ROS está organizado en paquetes. Las aplicaciones de ROS están desarrolladas en paquetes, y un paquete puede contener un nodo, una biblioteca, un conjunto de datos o cualquier archivo que pueda necesario para ejecutar el paquete.

4. **Mensaje (Message).** Un nodo envía o recibe datos entre nodos vía un mensaje. Los mensajes son portadores de información y pueden contener variables de diferentes tipos como enteros, de punto flotante, boléanos y una combinación de ellos. Las estructuras son las más comunes en el envió de información.

5. **Tópico (Topic).** Buses con nombre en los que los nodos ROS pueden enviar un mensaje. Un nodo puede publicar o suscribirse a cualquier cantidad de temas.

6. **Service (servicio).** El Servicio ROS tiene un mecanismo de Solicitud/Respuesta. Una llamada de servicio es una función que puede llamar cada vez que un nodo cliente envía una solicitud. El nodo que crea una llamada de servicio se denomina nodo Servidor, y quien llama al servicio se denomina nodo cliente.

7. **Gazebo.** Gazebo es un simulador de robótica 3D de código abierto. Este simulador ayuda a los desarrolladores a probar rápidamente deferentes tipos de algoritmos para diferentes tipos de robots. Se han calificado a sí mismos como simulación robótica simplificada, muchas herramientas son fáciles de usar y ayudan a acelerar el proceso.

C.2 Descripción de la arquitectura de la plataforma experimental

Las arenas de vuelo han sido fundamentales para el desarrollo de nuevos algoritmos de control, navegación, estudio de efectos aerodinámicos, pruebas de robustez, esquemas de aprendizaje, entre otros. Estos entornos permiten las pruebas en interiores en un ambiente seguro y eficiente. La arena de vuelo desarrollada tiene un espacio interior de 15.6 metros cúbicos, es decir 2.5 metros de largo por 2.5 metros de ancho y 2.5 metros de altura. La arena de vuelo cuenta con cuatro cámaras Flex 13 de 1280 por 1024 de resolución a 120 fotogramas por segundo, un servidor, una computadora personal dedicada al diseño de algoritmos de control, una computadora en un solo chip Raspberry Pi y robots con capacidades de comunicación con el middleware ROS.

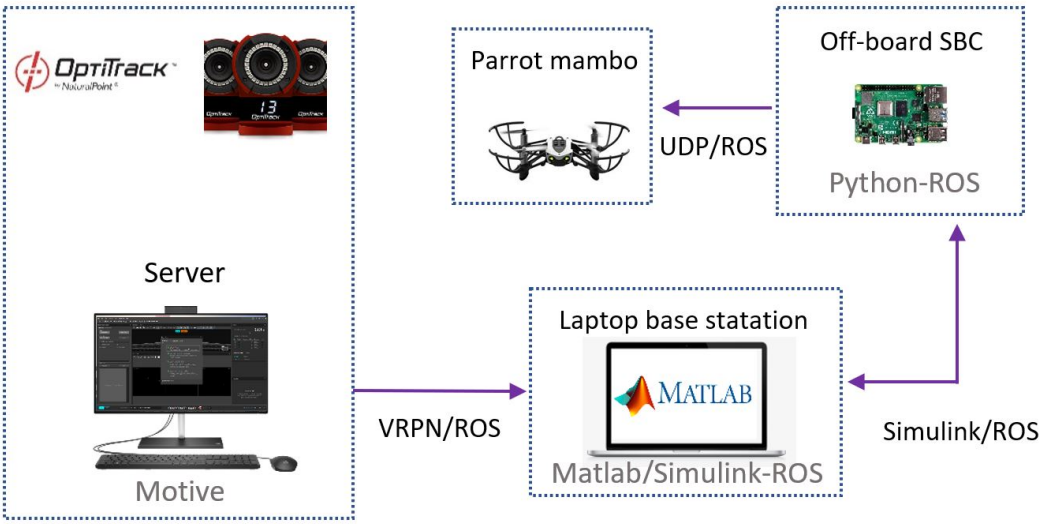


Fig. C.1: Diagrama general de la plataforma experimental.

Los robot con los que cuenta el laboratorio son el crazyflie 2.0, multirotor mambo, drones de arquitectura abierta basados en el piloto automático Pixhawk y robots móviles omnidireccionales, estos dos últimos de fabricación propia. La figura C.2 muestra los diferentes robots utilizados para el uso de la plataforma experimental.

El sistema OptiTrack se utiliza por su precisión a nivel milimétrico para estimar cualquier cuerpo rígido con seis grados de libertad. Se utiliza la transmisión de datos a través de una red local. La estimación de posición, velocidad lineal y actitud (cuaternión) del multirrotor se envía a una

Fig. C.2: Robots del laboratorio de control para el uso de la plataforma experimental.

estación terrestre a través de la red VRPN ROS. El procesamiento de las posiciones, los ángulos de Euler y la estimación de velocidades se realizan utilizando MATLAB/Simulink con una frecuencia de muestreo de 100 Hz. El algoritmo de control se evalúa en esta estación terrestre y luego se envía a una computadora de placa única, Raspberry Pi, a través de ROS. Finalmente, la Raspberry Pi envía los datos de control al dron utilizando la misma red ROS y paquetes UDP. La Fig. C.11 muestra el diagrama general de la plataforma experimental. Los sistemas que componen la arena de vuelo se describen a continuación.

1. **Robots heterogéneos.** El laboratorio cuenta con diferentes robots heterogéneos para poder realizar una tarea de forma conjunta. Algoritmos de control cooperativos, formación y consenso.

    - Robot omnidireccional: son vehículos terrestres que cuentan con una movilidad en cualquier dirección sin tener la necesidad de rotar. El laboratorio cuenta con 4 vehículos de arquitectura abierta con capacidades de comunicación ROS de diseño propio.
    - Vehículos aéreos no tripulados: son vehículos aereos que no llevan una tripulación hu-

Fig. C.3: Plataforma experimental.

mana a bordo. El laboratorio cuenta con multiples drones, 5 crazyflie 2.0 , 3 mambo de la empresa Parrot y diferentes diseños propios de arquitectura abierta con capacidades de comunicación inalámbrica y de ROS.

- Vehículos hibridos: Estas aeronaves combinan las ventajas de los aviones de ala fija y giratoria. Estos aviones pueden despegar y aterrizar verticalmente además de volar a gran velocidad. Su capacidad de carga también se incrementa. Sin embargo, el diseño de estas aeronaves es complejo y su dinámica de vuelo controla tareas desafiantes. Actualmente se cuenta con 2 drones swing de la empresa Parrot con capacidades de comunicación inalámbrica.

2. **Server (Motive).** Dado que la licencia de Motive utilizada en el laboratorio está disponible solo para el sistema operativo Windows 7, es necesario transmitir los datos a través de una red local utilizando el middleware ROS. Una red de área local, un servidor que contiene instalado el software Motive que le permite calibrar, configurar, estimar y procesar los datos a través de marcadores se ha implementado en la arena de vuelo.

3. **PC (Matlab/Simulink.)** Se ha incorporado una PC con sistema operativo Windows dedicada al diseño de algoritmos de control en Matlab/Simulink con capacidades de comunicación con ROS. Es la responsable de publicar y enviar los comandos calculados por los algoritmos de control.

4. **Raspberry Pi 4.** Una computadora de placa única (SBC) es una computadora completa en la que una placa de circuito única comprende una memoria, entrada/salida, un microprocesador y todas las demás funciones necesarias. Se utilizó una Raspberry Pi 4 con un sistema operativo Ubuntu Mate 20.04 LTS dedicado a la comunicación entre el servidor, la PC y el robot mambo utilizando el middleware ROS y la comunicación del Protocolo de datagramas de usuario (UDP).

C.3 Drone Mambo

El Parrot Mambo Drone es un mini dron diseñado para el manejo en interiores. Tiene una configuración en X, una dimensión de 86x86 mm de motor a motor, su peso es de 70g, y puede alcanzar una velocidad máxima de 30 km/h; estas características se resumen en la Tabla C.1.

Características	Descripción
Name	Mambo
Dimensions	185 x 180 x 40 mm
Mass	70 grams
Flight speed	30km/h
Range	Up to 20m
Battery	550mAh

Table C.1: Principales características del multirotor Mambo.

El mambo consta de diferentes sensores. Para estimar y controlar la orientación, utiliza un acelerómetro y un giroscopio de tres ejes; para el control de altitud utiliza un sensor de presión y un sensor de ultrasonidos. Para la estabilización horizontal, este mini dron utiliza una cámara de 60 pps de baja resolución. También contiene Bluetooth y Wi-Fi, lo que aproxima un radio de 20 metros para su correcto funcionamiento.

C.4 Sistema de captura de movimiento Optitrack

Motive no solo permite al usuario calibrar y configurar el sistema, sino que también proporciona interfaces para capturar y procesar datos en 3D. En este capítulo nos enfocamos a la utilización de las herramientas básicas, configuración y navegación del software Motive. Abordamos la creación de cuerpos rígidos, mascaras, filtrados y transmisión de los datos a MatLab.

C.5 Motive Calibración

Motive guarda la mayoría de las configuraciones, definiciones de cuerpos rígidos, datos reconstruidos en 2D y 3D en un archivo (.TAK). Un archivo TAK es una grabación de captura de movimiento única que contiene toda la información para recrear la captura completa del archivo. El archivo calibración

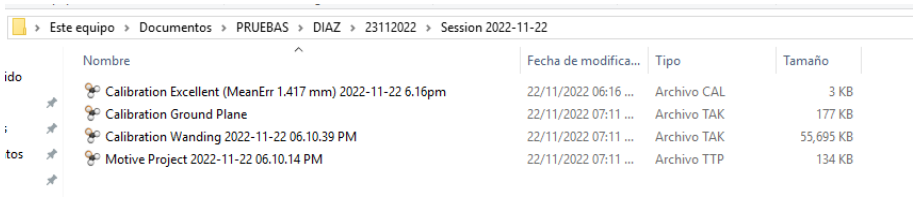


Fig. C.4: Archivos creados en un proyecto de Motive.

de la cámara se guarda con una extensión (.CAL). Contiene toda la información de las cámaras incluidas sus posiciones, orientaciones y los parámetros de distorsión de la lente. Otra extensión que utilizamos es (.motive). Este archivo guarda la configuración del software por ejemplo: configuración de transmisión, sincronización cuerpos rígidos y etiquetado. En la figura C.4 se muestra un proyecto realizado con Motive y sus respectivos archivos. Hay tres pasos principales: calibrar el volumen de captura, colocar marcadores retrorreflectantes (o activos) en los sujetos y crear activos o modelos asociados en Motive.

C.6 Calibración

La calibración es el primer paso para poder trabajar con el sistema de captura de movimiento óptico. Durante esta fase, el sistema calcula la posición y orientación de las cámaras para construir

un volumen del cuerpo bajo estudio. A continuación, se enumeran los pasos para la calibración del sistema de captura de movimiento.

- Aplicar las máscaras necesarias para eliminar los reflejos existentes en el entorno. Lo ideal es eliminar todos los objetos innecesarios que reflejen y causen una estimación errónea.
- Recopilar las muestras de calibración a través de un proceso de mapear la arena de vuelo mediante un barra con marcadores.
- Calcular los resultados obtenidos para aplicar la calibración.
- Establece el plano de referencia inercial para completar la calibración del sistema de captura de movimiento.

C.7 Creación del cuerpo rígido (Ejemplo)

Primeramente, es necesario eliminar todos los objetos reflectantes de la arena de vuelo. Si esto no es posible entonces podemos utilizar máscaras sobre los objetos no deseados para que el software Motive los ignore. La figura C.5 muestra algunos objetos no deseados, esto causa una estimación errónea y degrada el algoritmo de control. Las máscaras son aplicadas haciendo clic en Máscara en el panel de calibración. Los píxeles de las regiones enmascaradas (marcas rojas) serán ignoradas y filtrados. El proceso de recopilar las muestras de calibración se logra mediante un bastón con marcadores preestablecidos por el fabricante. Se agita el bastón sobre la arena de vuelo repetidamente lo que permite que todas las cámaras vean los marcadores de calibración. A través de este proceso, cada cámara captura puntos de datos de muestra para calcular su posición y orientación respectivas en el espacio 3D. Para calibrar presione el botor de *start wanding*. Comience a recolectar muestras sobre la arena de vuelo mediante el bastón de calibración. Se debe mover suavemente mientras dibujamos repetitivamente la figura de un infinito en todo el espacio de la arena de vuelo. En el software de motive aparecerá una tabla que registra todas las muestras y el espacio calibrado. La figura C.6 muestra este proceso. Se debe verificar en el panel de calibración que las muestras se encuentren entre 1000 a 4000. Arriba de este umbral son innecesarios y pueden afectar la precisión

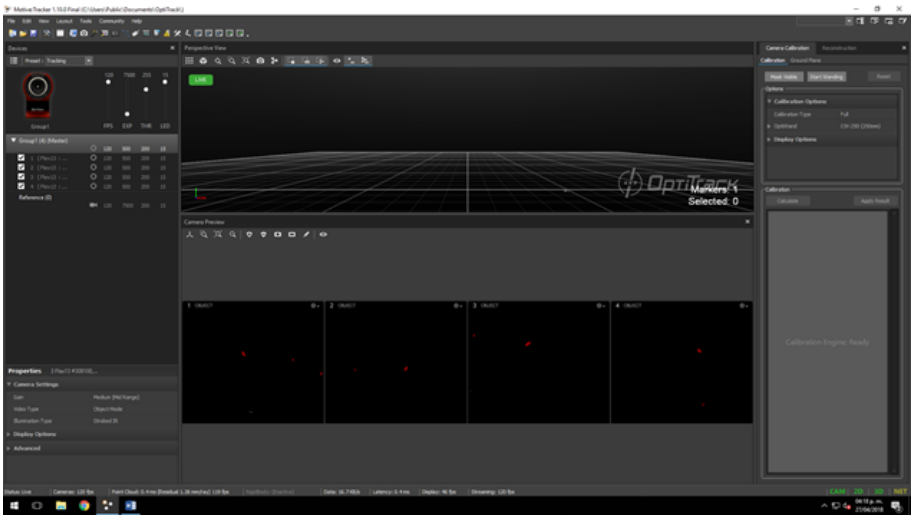


Fig. C.5: Software Motive: Aplicación de mascaras.

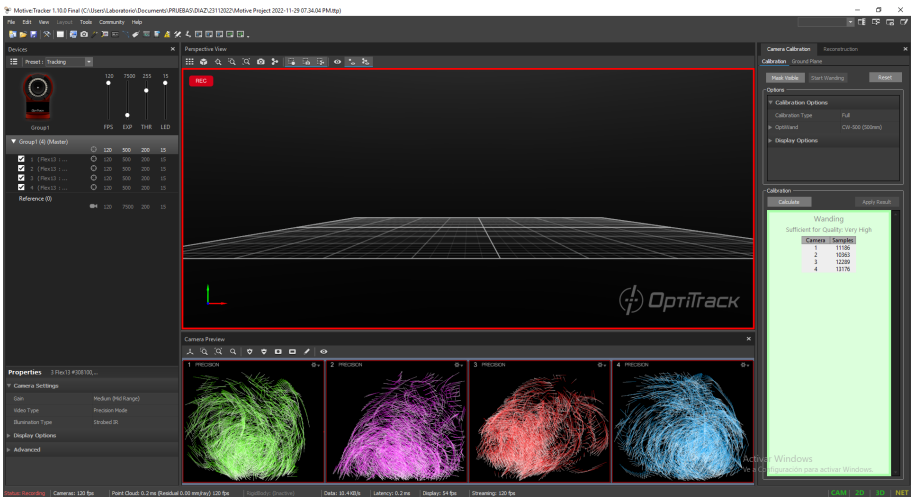


Fig. C.6: Software Motive: Muestras de calibración.

de la calibración. Una vez que se hayan recolectado suficientes muestras de marcador, presione el botor *Start Calculating* para calibrar utilizando las muestras recolectadas ver figura C.7.

Los resultados de calibración se muestran en el panel ver figura, si son aceptables, presionamos continuar, de lo contrario, presionamos cancelar y repetimos el proceso.

Enseguida, se establece el marco de referencia inercial y se elige el origen sobre la arena de vuelo. Se introduce colocando un cuadrado de calibración en forma de regla proporcionado por la empresa. Se coloca en el origen deseado y se establece como el sistema de coordenadas en Motive. El lado más largo del cuadrado de calibración indicará el eje z positivo, y el lado más corto indicará la dirección del eje x positivo. Después de confirmar el posicionamiento del cuadrado de calibración se

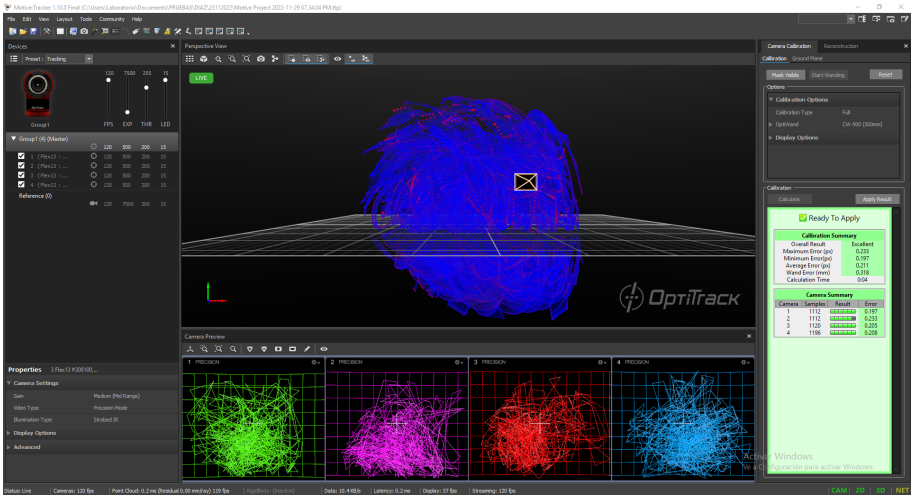


Fig. C.7: Software Motive: Calculo de las muestras obtenidas.

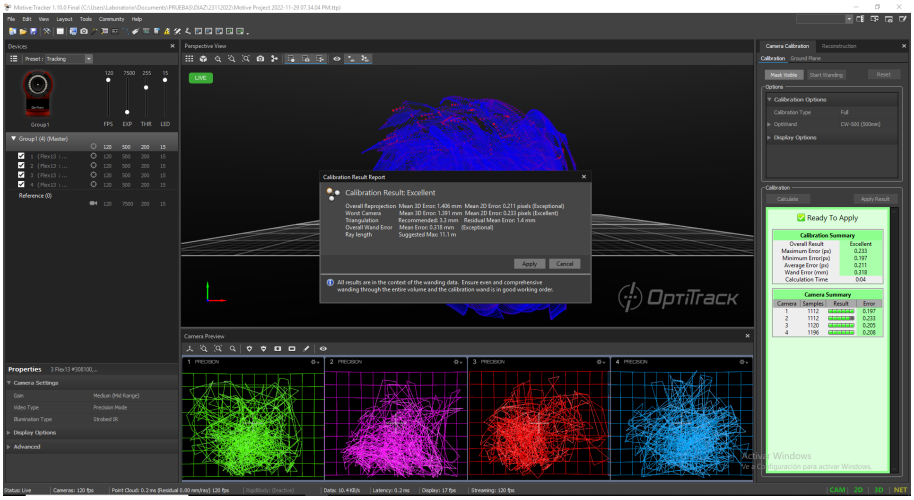


Fig. C.8: Software Motive: Resultados de calibración.

presiona el botón *Set Ground Plane* y guardamos la configuración ver figura C.9. El último paso es la creación de un cuerpo rígido del robot bajo estudio. Estos cuerpos rígidos permitirán rastrear los movimientos, estimar la orientación y la posición del robot. Seleccionamos un grupo de marcadores, un mínimo de tres para poder crear el cuerpo rígido del robot, clic derecho y seleccionamos la opción *create rigidBody*, ver figura C.10. Una vez creado el cuerpo, en el lado izquierdo del programa se genera una lista de los mismo, se despliega su orientación y posición con un sistema de referencia, una etiqueta y un color único. La posición de los marcadores debe de variar en cada robot, además de ser asimétrica. Esto con la finalidad de evitar errores en la captura de la posición de los mismos.

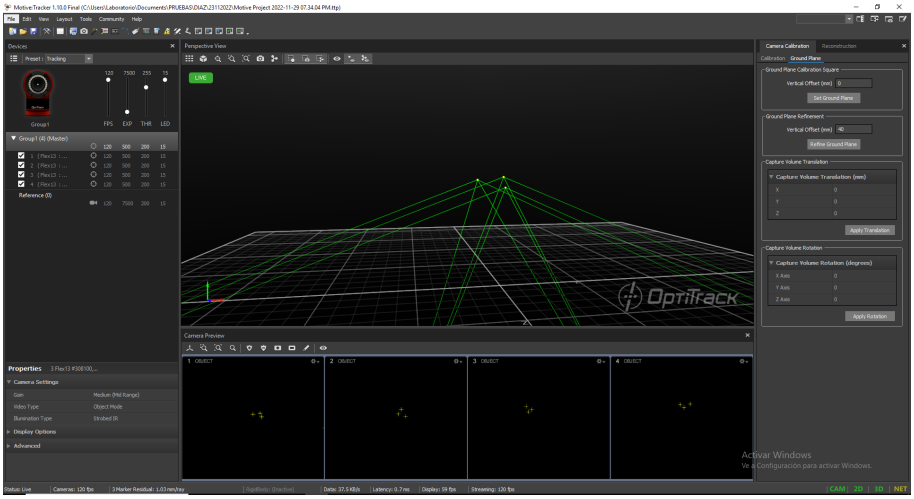


Fig. C.9: Software Motive: Marco inercial.

C.8 Captura de datos y transmisión a Raspberry Pi.

La licencia obtenida en la institución del software Motive solo se encuentra disponible para el sistema operativo Windows 7. Ante esta limitación, se diseña una red local para la transmisión de datos entre el servidor, la raspberry Pi y la computadora dedicada a los algoritmos de control. En Motive se activa el panel *Streaming* en la siguiente ruta: *View* > *Data Streaming*. A continuación, en el panel se selecciona la opción: *Broadcast Frame Data* y en *Multicast Interface*, se coloca la dirección IP de la computadora con la que se realizará la transmisión de los datos en este caso la dirección de la Raspberry Pi. Para recibir datos en ROS desde un sistema OptiTrack que ejecuta el software Motive

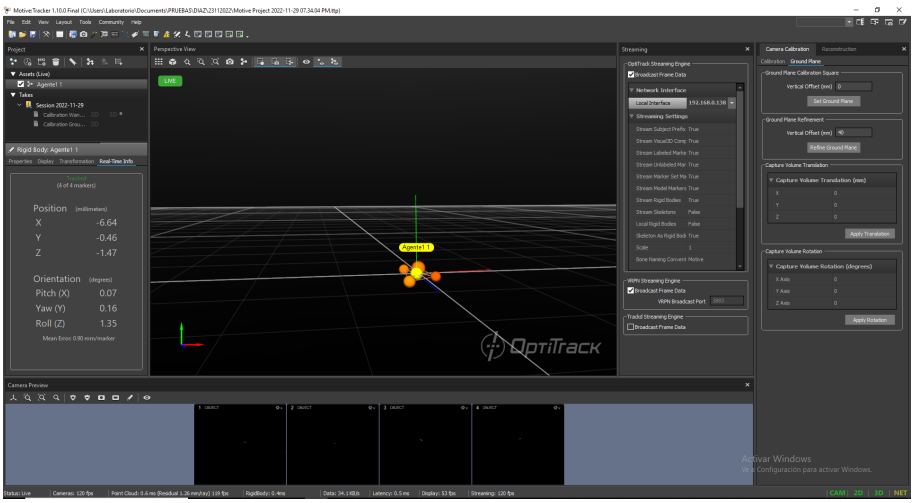


Fig. C.10: Software Motive: Creación del cuerpo rígido.

se ha utilizado el motor de transmisión VRPN con paquete el paquete de ROS *vrpn-client-ros*. El servidor tiene instalador el software Motive y se encuentra conectada a las cámaras de Optitrack a través de Ethernet al conmutador Optitrack. El servidor transmite la información de los cuerpos rígidos a la Raspberry Pi de forma óptima. La Raspberry Pi tiene instalado un sistema operativo Ubuntu Mate que permite la instalación de ROS. La ejecución de ROS en la Raspberry Pi permite la comunicación con el servidor utilizando el paquete VRPN. La figura muestra el paquete VRPN instalado en la raspberry Pi. Generando el paquete vrpn_client_ros, es posible establecer la conexión

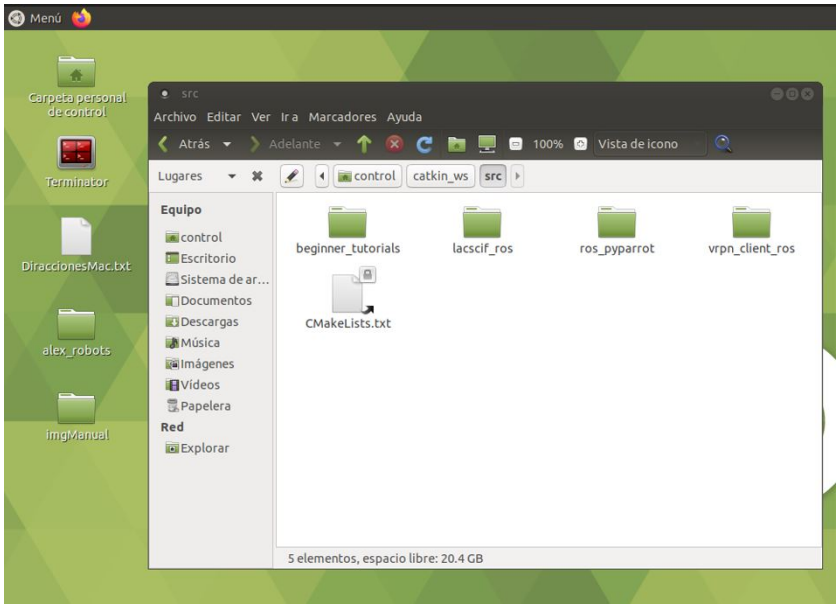


Fig. C.11: Paquetes instalados en el area de trabajo.

con el Motive para obtener los valores de los cuerpos rígidos que se encuentren dentro de la arena. Se ejecuta la siguiente instrucción en la Raspberry Pi: $ roslaunch vrpn_client_ros sample.launch server := 192.168.0.100 donde server es la ip de la computadora en la que se encuentra instalado Motive. La figura muestra la pose del cuerpo rígido publicado a través de ROS. En una segunda terminal, se muestran los tópicos disponibles para suscribirse. El nodo $/vrpn\_client\_node/Dron1/pose$ contiene la pose del robot, es decir la orientación en cuaterniones y su posición.

C.9 Generación de algoritmos de control y visualización de resultados

En este capítulo detallamos el diseño del entorno de comunicación entre el servidor, la PC y la Rapsberry Pi. El capítulo se ha dividido en tres secciones, en la primera sección el diseño de un control Proporcional-Integral-Derivativo (PID) para estabilizar la posición del multirotor. En la sección dos se describe la interfaz diseñada para la comunicación del servidor con la computadora dedicada al desarrollo de los algoritmos de control y su comunicación a la Raspberry Pi. Por último, se implementa un control en estacionario con el multicopter mambo. la comunicación entre la computadora dedicada a los algoritmos de control hacia la Raspberry Pi y el multirotor.

C.10 Diseño de un control PD para el multirotor Mambo.

C.10.1 Modelado matemático del multirotor.

Un modelo matemático exacto es la base para diseñar un algoritmo de control con alta precisión. Sin embargo, un modelo complejo dificulta el análisis de diseño y estabilidad. Por lo tanto, es fundamental considerar la dinámica del sistema más influyente. Como resultado del análisis matemático del multicóptero, se observan características como alta no linealidad, bajo accionamiento y fuerte acoplamiento. Para una comprensión profunda de la trayectoria que toma la aeronave debido a las fuerzas y momentos que actúan sobre ella, es necesario introducir dos sistemas de referencia, el marco fijo inercialmente e $\mathbf{E}^f$ y la aeronave estructura del cuerpo $\mathbf{E}^b$, como se muestra en la Figura **??**.

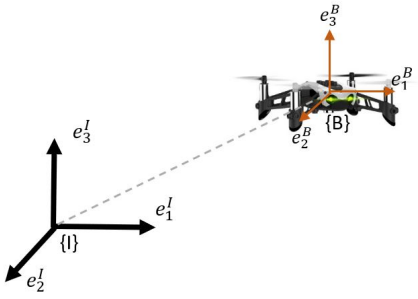


Fig. C.12: Marco de coordenadas inercial.

El multicóptero se puede modelar como un cuerpo rígido con seis grados de libertad y subactuado (tres momentos en sus respectivos ejes y una fuerza de empuje perpendicular a la aeronave). Basado en el teorema de Euler, la rotación del multirotor se puede lograr mediante tres rotaciones elementales alrededor de un punto fijo del marco de inercia. La rotación de las coordenadas de un punto desde el marco del cuerpo $\{B\}$ con respecto al marco inercial $\{I\}$ está representada por la matriz de rotación ortogonal $R \in SO(3) = \{R \in^{3\times3}: R^T R = I_3, \det(R) = 1\}$, donde $I_3 \in^{3\times3}$ es la matriz identidad. Los ángulos de Euler (balanceo, cabeceo, guiñada) son una representación de la actitud que se puede obtener usando el vector $\Theta = [\phi, \theta, \psi]^T$. La matriz de rotación correspondiente se da a través del mapa $R :^3 \to SO(3)$ por

$$R = \begin{bmatrix} \cos\psi\cos\theta & \cos\phi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \cos\theta\cos\psi & \sin\phi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \tag{C.1}$$

La posición del centro de masa del multirotor está dada por el vector $p = [x, y, z]^T$ y está definida en el sistema de coordenadas inerciales $\{I\}$. Las velocidades de traslación $v = [v_x, v_y, v_z]^T$ se pueden relacionar con la derivada relativa al tiempo de la posición como se muestra en la ecuación. (C.8). Las velocidades angulares $\omega = [\omega_x, \omega_y, \omega_z]^T$ medidas en el marco de coordenadas del cuerpo $\{B\}$, relativo al marco de coordenadas inerciales $\{I\}$. La matriz de actitud (C.2) relaciona la tasa de actitud $\dot{\Theta}$ y la velocidad angular del multirotor ω.

El objetivo del control de posición es calcular los ángulos deseados ϕ_d, θ_d de acuerdo a la posición deseada p_d de la aeronave. Por otra parte, el control de la altitud tiene como objetivo calcular el empuje total deseado acorde a la altura deseada del multirotor.

$$W = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & -\sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \tag{C.2}$$

Suponiendo que todas las fuerzas y torsiones actúan sobre la aeronave, usando la segunda ley de Newton, obtenemos la siguiente expresión.

$$\Sigma_T \quad : \quad \begin{cases} \dot{p} = v \\ \dot{v} = \frac{1}{m} R e_3^B f - g e_3^I \end{cases} \tag{C.3}$$

$$\Sigma_R \quad : \quad \begin{cases} \dot{\Theta} = W\omega \\ J\dot{\omega} = -[\omega^{\times}]J\omega + \tau \end{cases} \tag{C.4}$$

Cada rotor produce una fuerza F_i y un par M_i los cuales son proporcionales a la velocidad angular al cuadrado del rotor, matemáticamente lo anterior sé expresar como:

$$F_i = k_F \Omega_i^2,\ M_i = k_M \Omega_i^2 \tag{C.5}$$

donde k_F, k_M son parámetros dependientes de la densidad del aire, forma y ángulo de guiñada de la hélice. Se asume que la dinámica del motor es instantánea, y tomando la configuración del multirotor mambo obtenemos la ecuación que relaciona las entradas de control con las velocidades angulares de los rotores como

$$\begin{bmatrix} f \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} k_F & k_F & k_F & k_F \\ -Lk_F & Lk_F & -Lk_F & Lk_F \\ Lk_F & -Lk_F & -Lk_F & Lk_F \\ k_M & k_M & -k_M & -k_M \end{bmatrix} \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} \tag{C.6}$$

donde L representa la distancia del rotor al centro de masa del cuadricóptero. Para el diseño del control de posición, se ha linealizado el modelo no lineal descrito anteriormente. Los ángulos ϕ y θ se consideran pequeños. Estas suposiciones son escritas matemáticamente como

$$\sin\phi \approx \phi, \cos\phi \approx 1 \tag{C.7}$$

de la misma forma para θ.

C.10.2 Control PD para regulación de la altitud

De acuerdo a las consideraciones anteriores de angulos pequeños, las ecuaciones simplificadas que describen la posición vertical de la aeronave son

$$\Sigma_Z \quad : \quad \begin{cases} \dot{p}_z = v_z \\ \dot{v}_z = \frac{1}{m} f - g \end{cases} \tag{C.8}$$

Se propone el siguiente algoritmo de control PID,

$$f_d = mg - a_{z,1}\tilde{p}_z - a_{z,2}v_z \tag{C.9}$$

donde $a_{z,1}$, $a_{z,2}$ son ganancias positivas. El error de altitud es $\tilde{p} = p_z(t) - p_{zd}$ que se define como la diferencia entre la altitud actual y la altitud deseada. Para llevar a cabo el analisis de estabilidad se obtine la ecuación en lazo cerrado, como se puede analizar, el origen es un punto de equilibrio y es único. Proponemos la siguiente función de Lyapunov,

$$V(\tilde{p}_z, v_z) = \frac{1}{2}mv_z^2 + \frac{1}{2}a_{z,1}\tilde{p}_z \tag{C.10}$$

Se obtiene la derivada de la función de Lyapunov incorporando la aceleración $\dot{v}_z$ de la ecuación en lazo cerrado,

$$\begin{aligned} \dot{V}(\tilde{p}_z, v_z) = mv_z(\frac{1}{m}f - g) + a_{z,1}\tilde{p}_z\dot{\tilde{p}}_z = \\ v_z(-a_{z,1}\tilde{p}_z - a_{z,2}v_z) + a_{z,1}\tilde{p}_z v_z = \\ -ma_{z,2}v_z^2 \leq 0. \end{aligned}$$

cancelando términos se obtiene una función definida negativa y por lo tanto de demuestra estabilidad. La estabilidad asintótica del sistema en lazo cerrado se demuestra utilizando del teorema de Lasalle.

C.10.3 Control PD para la regulación de la posición horizontal

De manera similar al control de la altitud, se consideran angulos pequeños y por lo tanto la dinamica de traslación horizontal se describe como

$$\begin{aligned} \dot{p}_h &= v_h \\ \dot{v}_h &= -\frac{f}{m}\mathbf{A}(\psi)\Theta_h \end{aligned} \tag{C.11}$$

donde $\Theta_h = \begin{bmatrix} \phi & \theta \end{bmatrix}^T$, $p_h = \begin{bmatrix} p_x & p_y \end{bmatrix}^T \in \mathbb{R}^2$, $\mathbf{v}_h = \begin{bmatrix} v_x & v_y \end{bmatrix}^T \in \mathbb{R}^2$ y $\mathbf{A}(\psi)$ está definida por la ecuación (C.12).

$$\mathbf{A}(\psi) = \begin{bmatrix} s_\psi & c_\psi \\ -c_\psi & s_\psi \end{bmatrix} \tag{C.12}$$

Se considera el siguiente algoritmo de control,

$$\Theta_{hd} = \frac{m}{f_d} A_\psi^{-1} \left(-a_{h,1}\tilde{p}_h - a_{h,2}v_h\right) \tag{C.13}$$

donde $a_{h,1}$, $a_{h,2}$ son ganancias positivas. El error de posición horizontal es $\tilde{p}_h = p_h(t) - p_{hd}$ que se define como la diferencia entre la posición horizontal actual y la posición horizontal deseada. Para llevar a cabo el analisis de estabilidad procedemos como el caso anterior. Se obtine la ecuación en lazo cerrado, el origen es un punto de equilibrio y es único. De tal forma, que proponemos la siguiente función de Lyapunov,

$$V(\tilde{p}_h, v_h) = \frac{1}{2}mv_h^2 + \frac{1}{2}a_{h,1}\tilde{p}_h \tag{C.14}$$

Se obtiene la derivada de la función de Lyapunov incorporando la aceleración $\dot{v}_h$ de la ecuación en lazo cerrado,

$$\dot{V}(\tilde{p}_h, v_h) = mv_h(\frac{f}{m}\mathbf{A}(\psi)\Theta_h) + a_{h,1}\tilde{p}_h\dot{\tilde{p}}_h =$$
$$v_h(-a_{h,1}\tilde{p}_h - a_{h,2}v_h) + a_{h,1}\tilde{p}_h v_h =$$
$$-a_2 v_z^2 \leq 0.$$

cancelando términos se obtiene una función definida negativa y por lo tanto de demuestra estabilidad. La estabilidad asintótica del sistema en lazo cerrado se demuestra utilizando del teorema de Lasalle.

C.11 Diseño de Módulos Simulink-ROS Raspberry Pi.

En esta sección introducimos los bloques diseñados en el programa de Simulink para la comunicación entre la Raspberry Pi y la computadora dedicada al algoritmo de control que tiene instalado el software de Matlab-Simulink. Esta comunicación es realizada utilizando el intercambio de mensajes publicador suscriptor utilizando el middleware ROS. El programa se ha dividido en tres módulos para poder diseñar, analizar e implementar algoritmos de control para volar los diferentes robots. Después del vuelo, los datos registrados se pueden visualizar y analizar. Además el middleware ROS permite el soporte de cualquier tipo de robot autónomo, proporcionando comunicaciones internas, externas e integración de hardware.

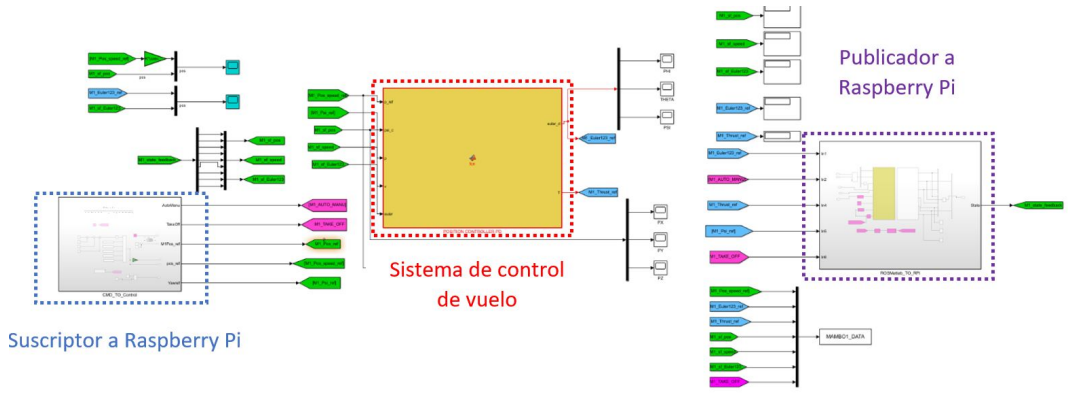


Fig. C.13: Simulink: Diagrama a bloques del diseño propuesto.

Los módulos incluyen la adquisición de la pose de los diferentes robots estimadas con el software

Motive. Un bloque de generación de trayectoria, navegación y el bloque de la ley de control. Por último, un bloque que permite él envió de los comandos obtenidos por el bloque anterior hacia el robot. La figura **??** muestra los bloques propuestos en Simulink.

1. **Suscriptor a Raspberry Pi.** La comunicación es realizada mediante el paso de mensajes asíncronos. Este modulo se suscribe a los nodos de la Raspberry Pi para obtener la orientación y la posición de los robots que se encuentran en la arena de vuelo. Como se muestra en la figura **??**, utilizamos ROS para poder obtener la información de las variables. Es necesario además multiplicar la orientación por un matriz de transformación.

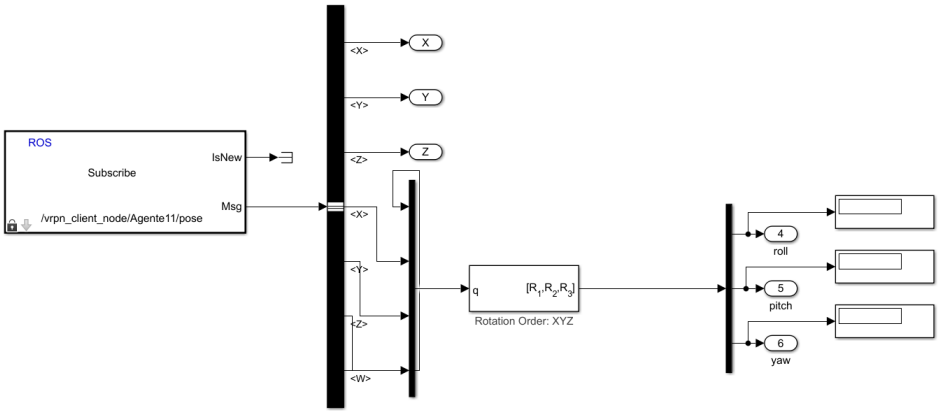


Fig. C.14: Suscriptor a Raspberry Pi.

2. **Sistema de control de vuelo.** Este módulo consiste de una colección de algoritmos de guía, navegación y control para drones autónomos. Los módulos se comunican entre sí a través de un bus de mensajes de publicación-suscripción. Todas las operaciones y comunicaciones están totalmente paralelizadas. La figura **??** muestra el bloque diseñado.

3. **Publicador a Raspberry Pi.** Este modulo publica los comandos deseados calculados por el sistema de control de vuelo. La información se envia a la Raspberry Pi. Debido a su limitada capacidad de hardware, el robot mambo no tiene soporte para ROS. La comunicación utilizada es el protocolo de datagramas de usuario UDP. Ese protocolo permite el envío de datagramas de forma rápida en redes IP sin establecer previamente una conexión. Ver figura **??**.

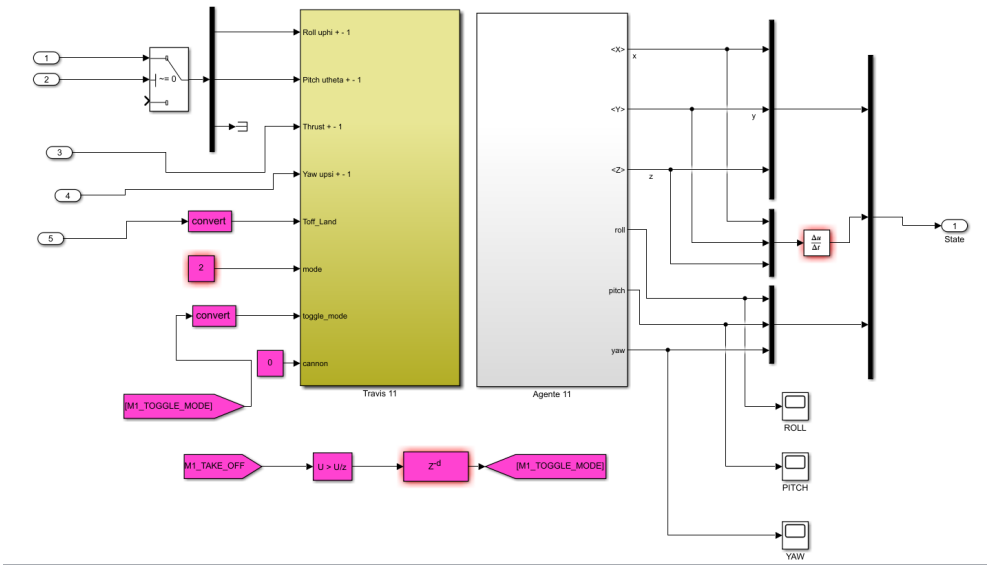


Fig. C.15: Sistema de control de vuelo.

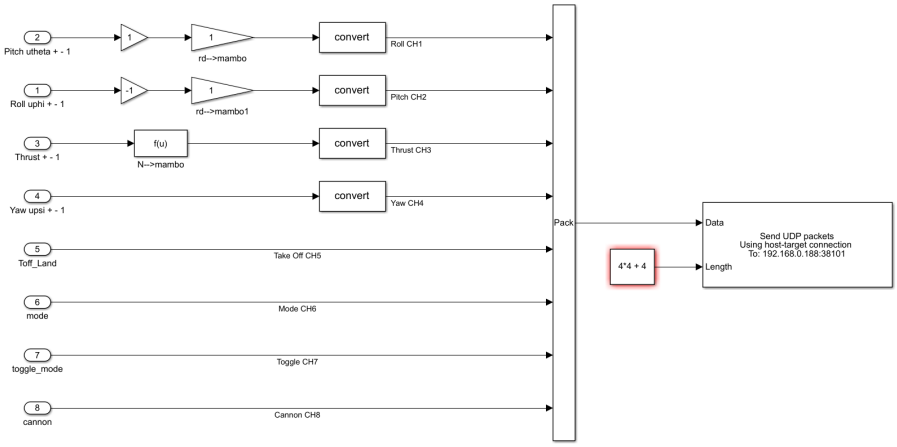


Fig. C.16: Publicador a Raspberry Pi.

El software contiene un scrip para obtener las graficas de todas las variables obtenidas por Motive las cuales se pueden utilizar para analizar el desempeño del controlador.

C.12 Rasberry y ROS

Se ha instalado un sistema operativo basado en Linux. Ubuntu MATE es una distribución ligera óptima para sistemas embebidos. Las características más importantes de esta distribución se listan a continuación:

- Rendimiento optimizado por el equipo de Ubuntu Kernel.
- Parches de seguridad regulares por parte del equipo de seguridad de Ubuntu.
- Expansión automática del sistema de archivos en línea.
- Bluetooth, Ethernet y WiFi (en modelos compatibles).
- Acceso GPIO a través de GPIO Zero y WiringPi.

Por compatibilidad se ha instalado ROS Noetic sobre la Raspberry Pi y además, algunos paquetes y herramientas extra para la comunicación con la plataforma experimental desarrollada. Como se comentó anteriormente se ha instalado el paquete VRPN para establecer la comunicación con el servidor y el software Motive. El paquete ros_pyparrot es un repositorio para minidrones de la

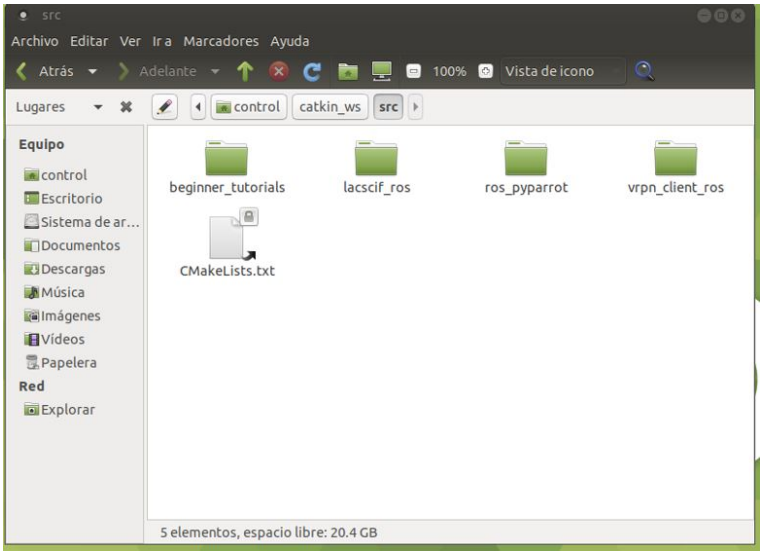


Fig. C.17: Paquetes instalados en el área de trabajo.

empresa parrot. Este paquete permite utilizar todas las interfaces como WiFi, BLE, lectura de sensores y los comandos para navegar desde terminal. El paquete contiene tres carpetas dedicadas a cada robot. Nos enfocaremos en el robot mambo y realizaremos un ejemplo de vuelo estacionario. En este paquete se ha realizado un archivo con la extensión (.launch) que permite ejecutar múltiples nodos ROS de forma simultánea. Permite especificar todos los parámetros a establecer y los nodos a ejecutar, así como las máquinas en las que deben ejecutarse.

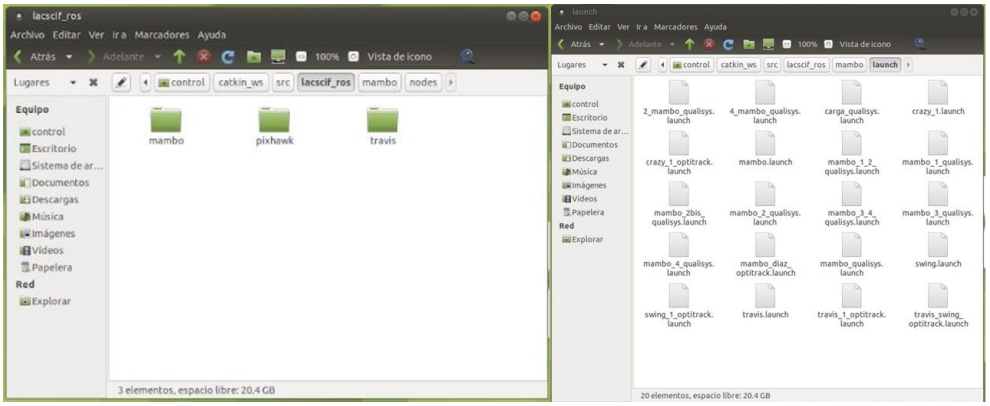


Fig. C.18: Paquete diseñado lacscif_ros.

La instrucción para ejecutar este comando es: $ **roslaunch mambo mambo_diaz_optitrack.launch**. Este archivo ejecuta los nodos necesarios para realizar las suscripciones a los tópicos VRPN, Joystick, Simulink-ROS y las publicaciones a los tópicos udp_to_mambo para la comunicación con el robot mambo.

C.13 Análisis y visualización de resultados

En este capítulo se realiza un vuelo estacionario utilizando el control PID presentado anteriormente. La plataforma experimental diseñada permite el rápido prototipado de algoritmos de control, navegación, estudio de efectos aerodinámicos y aprendizaje automático en un ambiente seguro y controlado. Se demostrarán los diferentes software diseñados para realizar un vuelo estacionario y obtener como resultado un conjunto de gráficas de las variables bajo estudio, las cuales representan la evolución de las posición, velocidad, orientación, velocidad angular y acciones de control de los vehículos aéreos no tripulados. El robot mambo es el vehículo utilizado para realizar las pruebas debido a su reducida dimensión y prestaciones en el manejo de interiores.

En este escenario probamos el vuelo estacionario el cual lo dividimos en tres regiones: despegue, vuelo estacionario y aterrizaje. Se ha añadido una pequeña perturbación externa con un ventilador con una velocidad máxima de 3.5 m/s medido con un anemómetro. La maniobra de despegue se realiza en 0 a 40 segundos a una altitud deseada de 1,5 m. En seguida cambia la referencia deseada a 1 m durante el tiempo de 40 a 200 segundos. Finalmente aterriza en el tiempo de 200 a 220 segundos.

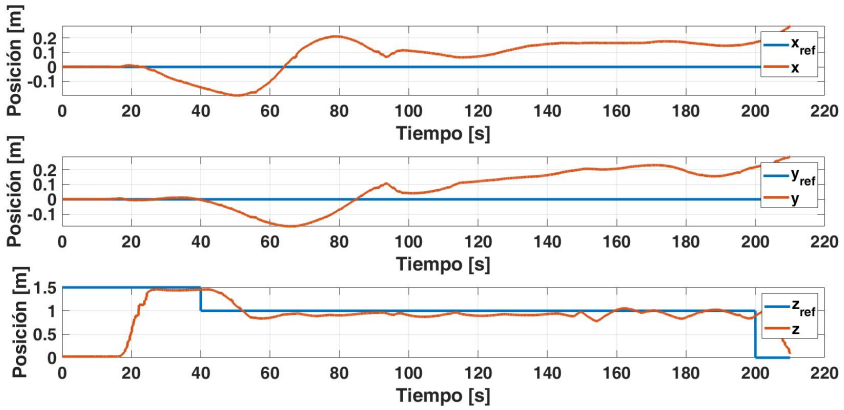


Fig. C.19: Evolución de la posición del robot mambo.

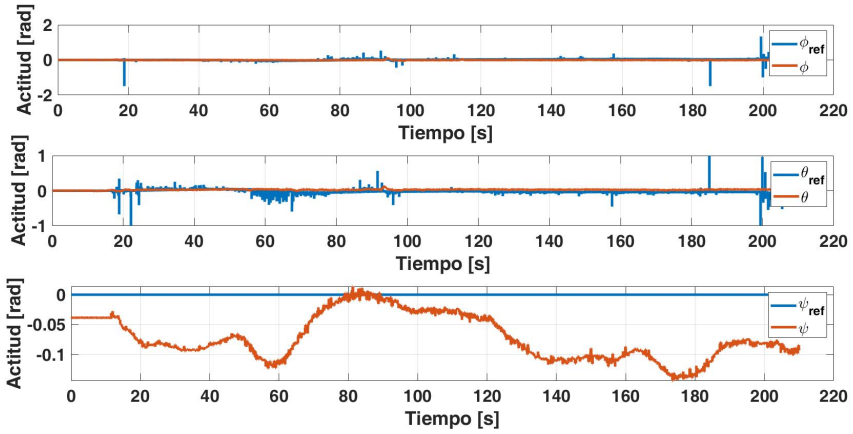


Fig. C.20: Evolución de la actitud del robot mambo.

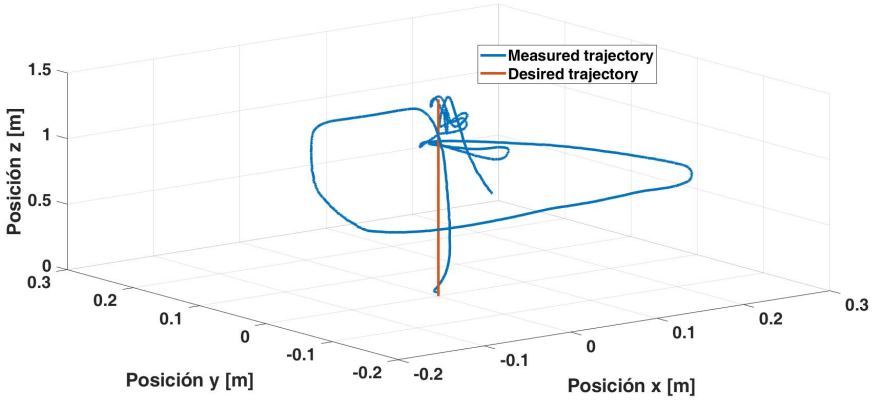


Fig. C.21: Posición en 3D del robot mambo.

La figura C.19 muestra la evolución de la posición obtenida por el software Motive. Se demuestra que a pesar de ser un control PD estabiliza a la posición deseada, además de desviarse en el plano

horizontal. El mejor comportamiento se obtiene en el eje z en donde no existe ninguna perturbación. Las figura C.20 muestra la evolución de la orientación deseada, es decir, los valores calculados por el control de posición. A pesar de ser un control PD la evolución de los ángulos resulta ser suave y sin oscilaciones. La figura C.21 muestra la evolución en el espacio de la posición del robot mambo. Como es de esperarse, el control lineal PD se muestra una desviación en la posición horizontal del robot.

C.14 Conclusiones

Se ha demostrado la funcionalidad de la plataforma diseñada para la enseñanza de la robótica, control, sistemas de navegación y aprendizaje automático. Otra de sus ventajas es que es de arquitectura abierta, es decir que cualquier robot con capacidades de comunicación ROS puede conectarse a esta plataforma robótica, además de expandir sus procesos añadiendo módulos extras para el desarrollo de tareas simultaneas. Lo cual hace modular y reactiva. La paralelización con el middleware de ROS es otro punto a considerar ya que la convierte en una plataforma robusta, segura y confiable capaz de ejecutar múltiples tareas al mismo tiempo.

Appendix D

Publications.

Journal of Intelligent & Robotic Systems

In-Ground-Effect Disturbance-Rejection Altitude Control for Multirotor UAVs

--Manuscript Draft--

Manuscript Number:	JINT-D-22-01001
Full Title:	In-Ground-Effect Disturbance-Rejection Altitude Control for Multirotor UAVs
Article Type:	Full/Regular paper
Keywords:	In-Ground-Effect Control; ADRC; Extended State Observer; Multirotor-UAV; Low Altitude; Wind Gust.
Corresponding Author:	Fermi Guerrero-Castellanos, Ph. D Benemerita Universidad Autonoma de Puebla Facultad de Ciencias de la Electronica Puebla , Puebla MEXICO
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Benemerita Universidad Autonoma de Puebla Facultad de Ciencias de la Electronica
Corresponding Author's Secondary Institution:	
First Author:	Juan Díaz-Téllez
First Author Secondary Information:	
Order of Authors:	Juan Díaz-Téllez
	J. Fermi Guerrero-Castellanos, Ph.D.
	Florian Pouthier
	Nicolas Marchand, Ph. D.
	Sylvain Durand, Ph. D.
Order of Authors Secondary Information:	
Funding Information:	Consejo Nacional de Ciencia y Tecnología, México (FORDECYT-PRONACES:MX-296702 - FR-M18M02) — Prof J. Fermi Guerrero-Castellanos
Abstract:	This paper presents a robust altitude control that merges the principles of active disturbance rejection control (ADRC) with the in-ground-effect (IGE) model. For that, a nonlinear extended state observer is designed on the vertical axis by taking measurements of the attitude and the altitude. Then, the forces generated by the flight at low altitudes, the ground effect, and other external disturbances are estimated and used (as a feedforward term) together with a nonlinear control law (feedback term) to reject them. The closed-loop stability is analyzed in the sense of Lyapunov. Extensive numerical simulation and real-time experimentations validate the proposal. Due to its simplicity, the control algorithm is easy to implement. It can be used for different maneuvers that depend on the proximity of the ground, obstacles, or surfaces, such as takeoff, landing, inspection, monitoring, and hovering.

Powered by Editorial Manager® and ProduXion Manager® from Aries Systems Corporation

Fig. D.1: In-Ground-Effect Disturbance Altitude Control for Multirotor UAVs.

2022 International Conference on Unmanned Aircraft Systems (ICUAS)
Dubrovnik, Croatia. June 21-24, 2022

Low Altitude Control of the VTOL UAV Tolerant to Ground Effect and Actuator Failures.*

J.C. González-Guerrero[1], J. Díaz-Téllez[1,2], J. Estevez-Carreon[1]
R. Mendoza-Vazquez[1] M.A. Meraz-Melo[1] and J.F. Guerrero-Castellanos[2]

2022 International Conference on Unmanned Aircraft Systems (ICUAS) | 978-1-6654-0593-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICUAS54217.2022.9836161

***Abstract*— This paper proposes a novel robust control law to stabilise the altitude of a vertical take-off and landing (VTOL) aircraft subject to ground effect and thrust faults. The VTOL UAV dynamics consider external disturbances, unmodeled dynamics, ground effect, and time-varying additive actuator faults. We perform extensive numerical simulations at altitudes between 0.05-0.2 m for a small Bitcraze Crazyflie 2.0 multirotor through three ground effect models. The VTOL UAV operates at low altitudes close to the ground in a hover state. Numerical simulations show that the control algorithm is precise and robust enough to stabilise the aircraft at low altitudes without including a ground effect compensator. A new form of active disturbance rejection control approach is proposed to guarantee asymptotic stability with endogenous and exogenous disturbances.**

I. INTRODUCTION

The VTOL-UAVs have become a standard platform for aerial robotics research. In particular, the VTOL-UAV multicopter has established itself in the small aircraft market. Its high maneuverability, small size, and hover flight allow navigating indoors, making it possible to solve several problems. Delivery services, aerial photography, surveillance, inspection, disaster relief, pollution monitoring, and search and rescue are among the most common applications of multicopters. From an automatic control perspective, the multicopter is a multi-input, multiple-output (MIMO) nonlinear system, under-actuated due to its six degrees of freedom (DOFs) and only four DOFs. Control inputs are strongly coupled and subject to parametric uncertainties. Several control strategies have been presented in the scientific literature, the investigations initially reported have been dedicated to attitude control. Different control approaches have been used, such as proportional-integral-derivative PID in [1],[2],[3], linear feedback in [5], proportional plus second-order differentiator PD^2 in [4] and nested saturation function in [6]. Most of the works cited above usually make the following assumptions to simplify the control design: small angles, absence of external disturbances, knowledge of the inertia/mass of the system, neglect of gyroscopic effects, or aerodynamic friction torques generated by the aircraft and the rotors. This linear time-invariant (LTI) control is limited to a small region of the state space; conditions far from the equilibrium position or external disturbances and parametric uncertainties can deteriorate the system's performance. Designing control algorithms robust to parametric uncertainties, non-modelled dynamics and environmental disturbances continues to be a current line of research. Many of the autonomous manoeuvres such as takeoff, landing, perching, and hovering directly relate to the interaction of the UAV with rigid or flat surfaces, experiencing the so-called ground effect. The ground effect reduces aerodynamic drag generated by an aircraft's wings or propellers when close to a fixed surface. This effect is a consequence of airflow distortion under such covers attributable to the proximity of the ground [7]. The aircraft experiences an increase in the thrust generated by the rotors when it is close to a surface. Therefore, generating trajectories in ground effect (IGE) can be more efficient than running out of ground effect (OGE) [7], [13].

Autonomous flights in cluttered environments, proximity to objects, walls and surface interaction can occur at low altitudes. Most of these applications require an approach to the object of study and interaction with its environment, such as package delivery, surveillance, an inspection of structures and damage location, to name a few. The ground effect has been extensively studied in helicopters in hover mode [9],[10]. However, it has been shown that these models do not accurately describe the multirotor ground effect. In multirotor, the ground effect is more stronger at higher altitudes than models predicted in helicopters. Models for single-rotor vehicles cannot be applied directly to multirotors ones mention [13]. Very few works have studied the ground effect in multirotors. In [8] a robust control based on PID combined with a robust compensator to uncertainties and a ground effect compensator based in [9] with a correction coefficient is proposed. The scheme is capable of track a trajectory and land precisely. The problem of landing a multicopter on an oscillating surface is addressed in [11]. An adaptive robust controller ARC is proposed to deal with ground effect. The ground effect is estimated online to later use it in the control action. In [12] also addresses the problem of autonomously landing a multicopter on a oscillanting platform. A smooth sliding controller SSC is proposed to deal with the ground effect. The problem of autonomously landing on a floating platform under ground effect and external disturbances is addressed in [14]. Robust altitude control is proposed along with a landing target status estimator and an autonomous precision landing planner. The control scheme demonstrates the precision of the landing despite the influences of the

*This work was supported by TECNM/ITP and CONACYT. Scholarship holder number 628537

[1]J.C. Gonzalez-Guerrero, J. Díaz-Téllez, J. Estevez-Carreon, R. Mendoza-Vazquez and M.A. Meraz-Melo is with Faculty of Electrical-Electronics Engineering, Tecnológico Nacional de Mexico/Instituto Tecnológico de Puebla, Mexico. juan.diaz@puebla.tecnm.mx

[2] J. Díaz-Téllez and J.F. Guerrero-Castellanos are with Facultad de Ciencias de la Electrónica, Benemérita Universidad Autónoma de Puebla (BUAP), Puebla, Mexico. fermi.guerrero@correo.buap.mx

Fig. D.2: Low Altitude Control of the VTOL UAV Tolerant to Ground Effect and Actuator Failures.

Robust Angular Velocity Control of a VTOL-UAV for Aggressive Maneuvers Flight

J. Díaz-Téllez, J. F. Guerrero-Castellanos, J. C. González-Guerrero(✉), J. Estevez-Carreon, and A. Silva-Juarez

Tecnológico Nacional de México/Instituto Tecnológico de Puebla, Av. del Tecnológico #420. Col. Maravillas, Puebla, Pue, Mexico
juan.diaz@puebla.tecnm.mx

Abstract. This work presents the design of an autopilot for acrobatic maneuvers such as multiple flips, invert itself completely and back-flips. This flight mode is possible by controlling the angular velocity of the UAV instead of the attitude. A robust angular velocity control is approached that allows to follow aggressive references at high speeds to generate acrobatic flights. The algorithm uses the active disturbance rejection control ADRC to estimate and eliminate effects that are generated in this type of flights, such as: unpredictable dynamics, external disturbances and ground effect. The control algorithm is parameterized by quaternion avoiding the gimbal lock singularities, in addition the maximum torques of the rotors are taken into account by means of a saturation function.

1 Introduction

In recent years the development of unmanned aerial vehicles for vertical take-off and landing (VTOL-UAV) have received notable growth. The quadcopter is the standard platform for the development of control systems, trajectory generation, cooperative control, aerial manipulation and aggressive maneuvers. Undoubtedly, its high maneuverability, simple design and reduced size are factors that favor its use in applications such as: search and rescue, precision agriculture, aerial photography, disaster relief, among others. However, designing control algorithms for stabilization of their states is not trivial, since the quadcopter is a nonlinear system, underactuated, perceptible subject to external disturbances and parametric uncertainties. Various control approaches have been published in the scientific literature to perform aggressive maneuvers. For example in [1], the problem of multiple flips is addressed using linear control of angular velocity and iterative adaptation for parameter correction. In [2] the dynamics of the system is simplified by hybrid systems, achievable sets are used to perform an autonomous backflip. A linear control that stabilizes the angular velocity and a trajectory generation for the generation of multiple flips are proposed in

H. A. Moreno et al. (Eds.): LACAR 2021, LNNS 347, pp. 1–8, 2022.
https://doi.org/10.1007/978-3-030-90033-5_4

Fig. D.3: Robust Angular Velocity Control of a VTOL-UAV for Aggressive Maneuvers Flight.

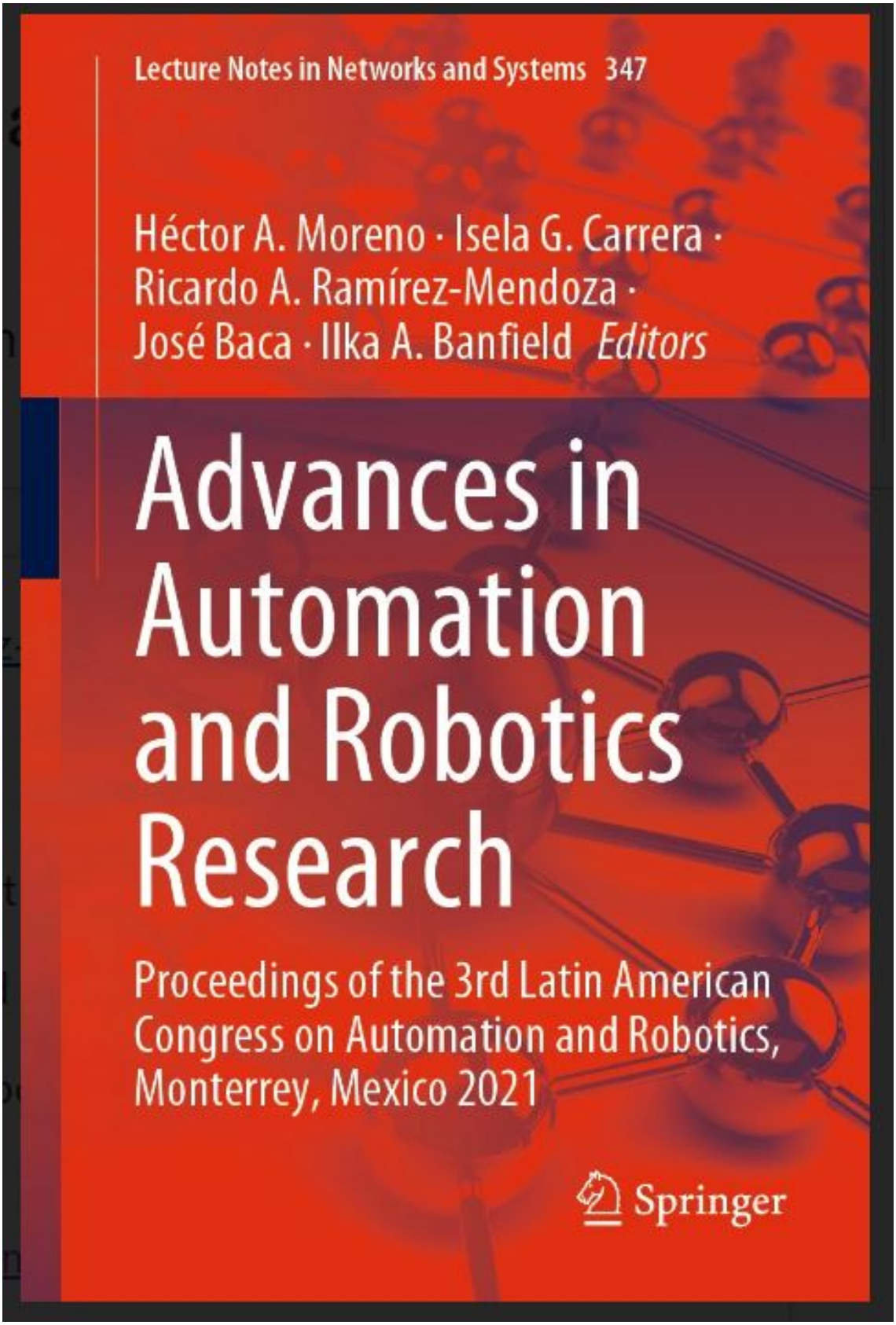


Fig. D.4: Book, Robust Angular Velocity Control of a VTOL-UAV for Aggressive Maneuvers Flight.

VIEP BUAP DITCo

PROGRAMA 2022 DE APOYO AL REGISTRO DE INVENCIONES

De conformidad con la Convocatoria del Programa 2022 de Apoyo al Registro de Invenciones se dan a conocer los resultados de aquellas propuestas que cumplen con la viabilidad de patentabilidad.

PROYECTOS ACEPTADOS

No. Proyecto	Investigador Responsable	No. Proyecto	Investigador Responsable
2022-01	Geolar Fetter	2022-17	Filiberto Candia García
2022-02	Miguel Angel Domínguez Jímenez	2022-18	Filiberto Candia García
2022-03	Filiberto Candia García	2022-19	Víctor González Díaz
2022-04	Albino Moreno Rodríguez	2022-20	Jesús Sandovañ Ramírez
2022-05	Filiberto Candia García	2022-21	Fermi Guerrero Castellanos
2022-06	Albino Moreno Rodríguez	2022-22	Jesús Sandoval Ramírez
2022-07	Luis Beristáin Jímenez	2022-23	Francisco Pacheco Aguirre
2022-08	Claudia Mendoza Barrera	2022-24	Carmen Cerón Garnica
2022-09	Víctor González Díaz	2022-25	Ma. de los Angeles Moyaho Bernal
2022-10	Sofía Schotte Hernández	2022-26	Fermi Guerrero Castellanos
2022-11	Tatiana Pritskij	2022-27	Fermi Guerrero Castellanos
2022-12	Albino Moreno Rodróguez	2022-28	Marino Conde Guevara
2022-13	Griselda Corro Hernández	2022-29	Francisco Pacheco Aguirre
2022-14	Facundo Comba	2022-30	Ángel Picazo Castillo
2022-15	Carlos Ochoa Velasco	2022-31	Jesús Muñoz Pacheco
2022-16	Edgar González Campos		

Atte.
Dr. David Pinto Avendaño
Dirección de Innovación y Transferencia de Conocimiento
Benemérita Universidad Autónoma de Puebla

Fig. D.5: Intellectual Property Product.

Secretaría de Educación
Gobierno de Puebla

CONCYTEP
Consejo de Ciencia y Tecnología del Estado de Puebla

Norma Mexicana en Igualdad Laboral y No Discriminación

H. Puebla de Zaragoza, a 31 de octubre de 2022
Asunto: Aval de proyecto de investigación científica (aplicada)
Dirección General
Oficio No: CCT/DICT/3.4/897/2022
No. Expediente: CCT.DICT.3.4. 1.2022

DR. JESÚS MANUEL MUÑOZ PACHECO
COORDINADOR DEL PROGRAMA DEL
DOCTORADO EN INVESTIGACIÓN APLICADA A LA INDUSTRIA

PRESENTE

Apreciado Dr. Jesús Manuel Muñoz Pacheco, le envío un cordial saludo, a la vez le comento que en el Consejo de Ciencia y Tecnología del Estado de Puebla el cual tengo el honor presidir, hemos recibido en la Dirección de Apoyos a Proyectos y Programas el proyecto del cDr. Juan Díaz Téllez, que corresponde a su trabajo de investigación doctoral, donde resalta su aporte al Segundo y Cuarto Eje de el Plan Estatal de Desarrollo 2019-2024, los cuales versan sobre la RECUPERACIÓN DEL CAMPO POBLANO y la DISMINUCIÓN DE LAS DESIGUALDADES, por tal razón se considera valiosa su investigación ya que aporta a los esfuerzos del Gobierno del Estado.

Sin otro particular, me despido de usted deseándole éxito en sus actividades.

ATENTAMENTE

DR. VICTORIANO GABRIEL COVARRUBIAS SALVATORI
DIRECTOR GENERAL
CONSEJO DE CIENCIA Y TECNOLOGÍA DEL ESTADO DE PUEBLA

JGGA/jgga

DENUNCIAS E INCONFORMIDADES
800 466 37 86
PROINTEGRIDAD
prointegridad.puebla.gob.mx

Avenida 13 Poniente 2904, Col. La Paz,
Puebla, Pue. Tel: +52 (222) 249 76 22
www.puebla.gob.mx

Fig. D.6: Project for the recovery of the field.