NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

ANTÓNIO SALVADOR BOUZA SERRANO DOS SANTOS COSTA

Bachelor in Electrical and Computer Engineering

INTEGRATION AND CONTROL OF A SHUTTLE DRONE FOR COOPERATIVE AND NON-COOPERATIVE CAPTURE

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon

Draft: *June 1, 2023*

DEPARTMENT OF ELECTRICAL
AND COMPUTER ENGINEERING

INTEGRATION AND CONTROL OF A SHUTTLE DRONE FOR COOPERATIVE AND NON-COOPERATIVE CAPTURE

ANTÓNIO SALVADOR BOUZA SERRANO DOS SANTOS COSTA

Bachelor in Electrical and Computer Engineering

Adviser: Bruno João Nogueria Guerreiro
Auxiliar Professor, NOVA University Lisbon

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING
SPECIALIZATION IN ⟨SPECIALITY NAME⟩

NOVA University Lisbon
Draft: *June 1, 2023*

Contents

List of Figures

1 Introduction

With the great advancements in today's modern technology, we are now capable of achieving more and quicker in the most different areas, creating, upgrading, and imagining ways to progress in a piece of specific knowledge. Therefore, align with the tools already available and the constant seeking to do better, we are facing exponential growth in this search for development.

In this way, one of these areas, aerial technology, has come from the simple idea of throwing objects into the air, to pilot them and now we are capable of programming them to fly by themselves, not needing a human pilot onboard. On this last achievement, Unmanned Aerial Vehicles (UAVs), also commonly called drones, appeared and are growing in usage ever since [1] due to their faster and more efficient performance in applications on a variety of important fields of modern society, including: military surveillance, reconnaissance and defense; commercial applications such as package transport and delivery (for example, helping distribute health products in low-income countries in a case of natural disasters, by reaching more easily to affected areas and helping that lack of resources that these countries have to face this issues [2]); agricultural production by not only providing better field management with aerial photography/footage of the vegetation (having better resolution than satellites too), but by also helping in performing precision agriculture, where weed control and fertilizers are more efficiently dispersed with reduced costs and improved crop quality [3]; and even in arts, more spectacles are opting where drones are used as a more sustainable in terms pollution and the environment instead of fireworks [4]. For this reason, the interest in this technology has also increased leading to numerous studies being conducted in order to further increase drones performance and control in the future. Therefore, there are already a large number of techniques and algorithms available for purchase or even customized that already show good results in the challenges that they tackle.

UAVs can also be divided into two main platforms - rotary-wing (RUAV) or vertical take-off and landing (VTOL), such as helicopters and multirotor drones, and fixed-wing (FUAV) UAVs, where the wings are used in a similar way to airplanes - offering different possibilities, depending on the desired outcome. For instance, multirotor UAVs are generally used for more confined environments, where there is less space to take off and to land, and there is also the need to hover to perform slower and more controlled maneuvers, therefore, being more specialized for detailed and scrupulous task like crop spraying [2]. Because of this, when it needed to cover more space and in a faster time, usually required in surveying and mapping missions, a fixed-wing drone is more suited, due to its higher cruising speed and flight range on a single battery. Also, it features higher payload capacity which is important, for example, when performing long-range transport tasks as in pipeline surveys, in which the UAVs can carry gas monitoring sensors safely through large areas, facing most of the times difficult environment conditions where the stability they present is key [5].

1.1 Context

This thesis was developed within the scope of the project CAPTURE [4] that comes with the goal of studying the many challenges behind the capability that UAVs have to interact with their surroundings performing more aggressive maneuvers like capturing and transporting other vehicles or objects. In this way, this project addresses two scenarios in this concept: the first is considered where the shuttle drone cooperates with the vehicles to be captured or launched, by providing information about their planned trajectory or diligently synchronizing its movement to ease the maneuver. This enables the creation of specialized vehicles for a given task, for example, having fixed-wing UAVs with long durability, while possible the vertical take-off and landing of these vehicles with shuttle drones. In a second scenario, the shuttle drone may or may not cooperate with the vehicles that it's trying to capture or launch, either passively or actively. For that reason, the shuttle UAV may act as a security measure to remove drones or objects from restricted areas that may actively avoid being captured.

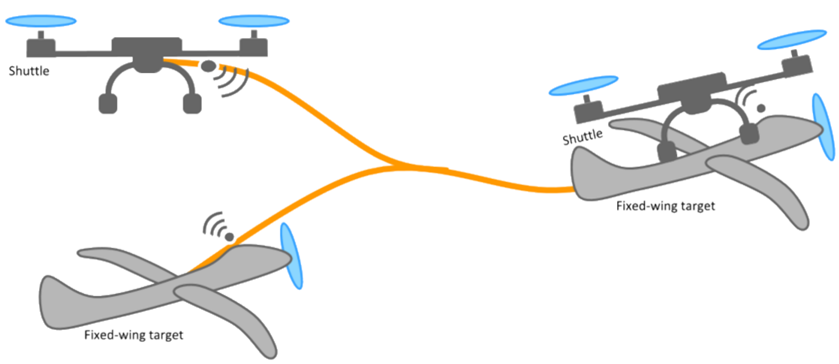

Figure 1.1: Illustration of a capture maneuver of a fixed-wing drone by a shuttle drone (reproduced from [4])

1.2 Motivation

In both approaches, there are a variety of interesting scientific and technological aspects that must be dealt with, including optimal and cooperative planning of trajectories for a team of heterogeneous vehicles; cooperative, hybrid, and distributed control for critical rendezvous maneuvers; cooperative and distributed estimation of the motion of the shuttle drones, other vehicles, and the surrounding environment; as well as dealing with estimation, control and planning strategies based differential games for the non-cooperative scenario. By being capable of performing this kind of maneuver, both types of UAVs can cooperate in handling various specific tasks such as, for instance, package delivery, where the fixed-wing drone has the long reach and high speed to take the package to the destination, and the shuttle UAV can help its landing in more confined areas. Also, in airports, VTOL drones can help the airplanes' launching and landing, possibly redefining airports' landscape, by optimizing the use of the space and infrastructure required.

1.3 Problem statement and proposed solution

This dissertation will deal more specifically with the scenario of the non-cooperative capture by the shuttle drone, where this drone will, therefore, be a quadcopter and it will have to be able to catch a fixed-wing UAV. For that, a variety of stages will take place as the first UAV will have to launch, follow a trajectory, catch the second one and carry it. Consequently, the focus of this thesis is to propose a studied scientifically-backed solution to achieve all these maneuvers efficiently and safely by operating on the shuttle drone.

In fact, this operation brings many challenges, some of them present in the totality of it and others are more specified to each faze of it. For example, the shuttle UAV will face constant environmental and dynamic factors for the entirety of the process, and, in the final stage, when carrying the other drone, will have further challenges of stabilization due to the abrupt weight changes and air dynamics caused by the size differential.

Knowing the various difficulties inserted in this task, the purposed solution is, therefore, divided into 3 major objectives that are essential to achieve the final purpose:

- Design of a model and a controller capable of managing a quadrotor autonomously in basic flight manoeuvres;
- Devise new modeling and controlling techniques for agile maneuvers;
- Design model and control strategies for close proximity flight between two quadrotors;
- Validate proposed strategies in simulation with various scenarios of trajectory following and close proximity flights.

First, a nonlinear controller will be designed and implemented following the example of an already existing one.

Secondly, some modern quadcopter control techniques will be applied to the previous controller to allow the drone to effectively perform more aggressive maneuvers while taking into consideration some classical constraints related to the environment where it is flying and the vehicle itself, such as **wind, drone aerodynamics, and space limitations**.

Then, for the situation where a drone flies above or beneath another one, there is the necessity to develop and implement new modeling and controlling algorithms to deal with the new aerodynamic interactions that take place in these situations.

Finally, simulations are performed aiming to validate the algorithm's trajectory tracking performance and display the aerodynamic interactions present in close proximity flights between drones.

1.4 Outline

The remainder of this document is organized as it follows:

- Chapter 2 is used to review the work already done in this area, more specifically addressing some relevant techniques of drone modeling and control, and aerodynamic modeling.
- Chapter 3 exposes the theoretical background behind some of the necessary concepts to consider throughout this dissertation.
- In Chapter 4, the model of the quadcopter is described alongside a nonlinear controller responsible to guide the drone to follow given reference trajectories.
- Chapter 4.2 presents more robust algorithms for the model and control of the drone, where new aerodynamic conditions and a scenario of close proximity flight between drones are addressed.
- In Chapter 6, the results of simulations to test the developed controller performance in various scenarios are presented.
- Finally, in Chapter 7, some concluding observations are drawn, and future work suggestions are made.

2

State of the Art

Developing and controlling UAVs has seen exponential research and development in the last decade, with some interesting work being done in the rotary-wing drones area. As said before, the scope of the project, which this dissertation is included, covers different topics about this vehicle, from its flight motion and control to its integration. Therefore, this section is divided into various groups: in the first ones, some research is presented in regards to control techniques behind maneuvering a quadrotor in two different motion types - basic and aggressive maneuvering. After that, as the capture maneuver implies close proximity between the shuttle and the target drone, a brief review of the existing work in aerodynamics for quadcopters is also provided. Finally, relevant and recent studies about flight simulation and testing are also addressed, as it was one of the purposes of the **CAPTURE project** to perform real flight tests.

2.1 Drone control techniques for non-cooperative flight

When controlling a single drone in a specific environment, it's important to take into consideration both the vehicle and the environmental constraints. This is due to the fact that different types of UAVs, along with their shape, size, weight, and purpose require different approaches that will also be chosen taking into consideration the vehicle's interaction with its surroundings. Consequently, it's important to note that these procedures often take into account multiple control methods in order to achieve the best solution to the desired requirements.

Furthermore, it's possible to associate a technique with one of the two major categories of control: linear and non-linear control. In fact, these two types derive from the different groups of systems that need to be controlled, also linear and nonlinear ones, where the main distinction is on the principle or rule they follow. For instance, linear systems heed a superposition principle, being determined by linear equations, and, in some cases, their parameters are unchanged over time, being called Linear Time-Invariant systems (LTI). Graphically, the output of these systems forms a line in relation to the input, being the origin of the attribution of the name "linear" to these systems. On the other hand, nonlinear systems are not submitted to these principles and behave in a dynamic way, being, for this reason, the most common to find as they cover a much wider range of real-world situations. Here, a change in the output has no proportional relation to a change made in the input of that system, thus forming a curve. These procedures are supported by nonlinear differential equations, which are much less general and, therefore, are designed to deal with a small portion of systems.

2.1.1 Linear control

Linear control methods are much easier to implement, having the least computational power burden. This results from the fact that these controllers are less complex and, therefore, faster to design and converge to a desired result (when that result is within the control capabilities). Nevertheless, because of this lack of robustness, they come short when having to control underactuated vehicles - systems that have fewer actuators than the degrees of freedom to be controlled - such as quadrotors (where external forces are unable to exert the change in acceleration in an arbitrary direction [6]), more specifically when it's needed to manage them outside their local point of equilibrium and when performing agile maneuvers, due to the difficult task of stabilize the drone. Thus, in these situations, linear controllers are usually just used for attitude stability control. To overcome this limitation, it is necessary to apply linearization to the system, by providing relative equilibrium conditions around a steady-state operating point.

One of the most common linear techniques is **Linear-Quadratic Regulator** (LQR), which consists in a feedback controller that shows some robustness in controlling quadrotors compared to other linear methods. This is due to its capability of handling some nonlinearities such as external disturbances by optimizing its feedback gain, k. This method it's regularly also accompanied by estimators, such as Kalman Filters (KFs), in order to be able to measure the state vector directly all the time, forming a Linear Quadratic Gaussian (LQG). This technique is further explained in [7], where the LQR controller is used to help stabilize a quadrotor at a hovering state by calculating the K value through real experimentations. Figure 2.1 shows a schematic of an LQG controller used in a quadcopter.

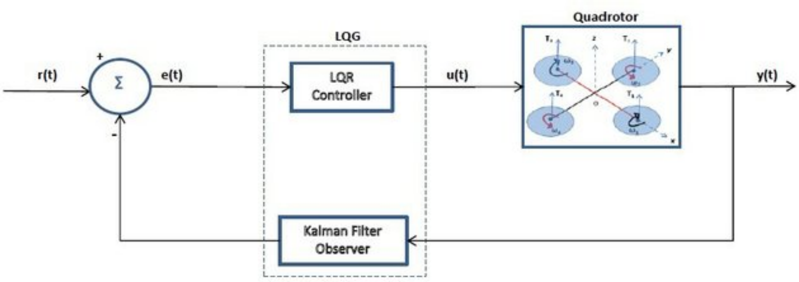

Figure 2.1: Block diagram of an LQR controller with Kalman Filters applied to a quadrotor (Reproduced from [8])

Another regularly chosen method when it comes to linear controllers is **Proportional Integrative Derivative** (PID) control, because of its high reliability and versatility, and also the computational effort it needs to operate and implement. In fact, this technique is already widely used in process industries and other control applications, being also effective for quadcopter hovering stabilization. It fundamentally works by calculating the difference between the values of the desired setpoint and the output and formulating the error e(t). Then, through the mathematical applications of the proportional, integral, and derivative terms it's possible to obtain the optimal response with the new calculated setpoints. More detailed information about the PID controller and a very acceptable example of trajectory following the PID controller of a quadrotor can be seen in [9]. Also, in figure 2.2, a representation of a common PID controller used for quadrotor control can be seen.

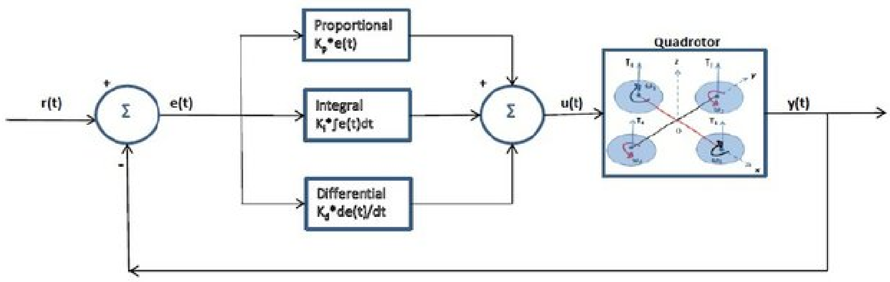

Figure 2.2: Block diagram of a PID controller applied to a quadrotor (Reproduced from[8])

It's important to notice that, although the growing technology has enabled the creation of more robust linear control strategies, few algorithms have proven to be as efficient as well-designed nonlinear solutions when controlling a UAV in a real 3D environment, mainly when dealing in the best possible way with its disturbances. In [8] explanatory and also comparative work about various linear and nonlinear control methods applied to quadcopters can be seen. **However, as is possible to see in many examples provided in the literature, these linear techniques still provide good results in path tracking and, therefore, are included in various projects that use them as low-level controllers (for example, using them to minimize the position error) in a bigger solution that combines linear and nonlinear algorithms.**

2.1.2 Nonlinear control

These types of techniques are much more robust and versatile than the linear ones, offering the possibility of controlling maneuvers far from the equilibrium point while having better stability of the vehicle with no need for linearization. For this reason, although they are more complex and need bigger processing power, they also revealed to be much more suited to manage the complex movement of underactuated vehicles, providing better support for performing aggressive maneuvres - defined in [10] as a sequence of trajectory segments to reach some configuration in state space - which is very relevant for this project.

In this way, there are some algorithms that consistently show good results when implemented in controllers to manage the flight of individual drones in this kind of situation.

Backstepping Control is one of the most common overall techniques used for controlling UAVs, particularly for quadrotors as it fits well with their cascade structure and shows good results in controlling their trajectory and external disturbances. This method operates in a recursive algorithm, dividing the controller into smaller subsystems that work as steps, gradually stabilizing each one of them. This is done by inserting intermediate control laws in order to describe some state variables of the system that is being controlled. Thus, this results in an advantage as these controllers converge quickly, needing fewer computer resources. However, this comes with the cost of robustness, mainly because of the abrupt growth of complexity when performing continuously each differentiation present in the control laws. In the project in [11] this type of nonlinear control is used along with adaptive back-stepping for steering a quadrotor along a trajectory, at the same time it rejects constant force disturbances, showing good results. Furthermore, in [12], backstepping laws are used to integrate a controller capable of tracking the trajectory handling effectively the disturbances by the payload swing.

Adaptive Control Algorithms come with the purpose of helping the controlled systems to adapt to uncertain or varying parameters. This is useful because conventional controllers are only capable of adjusting to the nonlinear system when dealing with fixed conditions that have little to no changes, and, in that way, for the quadrotor controlling scenario, where the UAV sustains uncertain disturbances during flight operations, adaptive controllers prove to be a helpful solution. In this way, when these variances take place, affecting the working capability of the conventional controller that is programmed to maintain the process variable only at a specific set point, an adaptive method can handle the situation where the process starts to operate outside that variable. These situations can happen due to variations in the transfer function due to changes in coefficients or parameters or because of a change in the nature of inputs. In [13] this technique is applied to a trajectory control of a quadrotor, successfully achieving stability and also showing the convergence of the boundedness of the adaption parameters and tracking errors. Also, a comparison between this method and a non-adaptive version of it is done, showing that adding an adaptive algorithm provided better tracking performance.

Slide Mode Control (SMC) - is a very robust and also simple technique to implement with already decades of functional use in a variety of fields like robotics, control, and other scientific and industrial sectors. Its working principles consist of two different phases: firstly, the reaching phase, where the system is stabilized by the controller into a plane trajectory, reaching the sliding surface; and, secondly, the discontinuous control signal is applied to the system in order to instruct it in sliding along its normal behavior (as it's possible to see in 2.3, also altering the dynamics of the system being controlled. Also, while the system is going through this phase, the closed-loop response becomes independent from the exact reference model, because it depends only on the chosen surface where it slides. Thus, this comes with advantages such as better disturbance rejection and lower sensitivity to parameter variations, which are very relevant in drone control. In [14] an adaptive version of this method is used for better tracking and control of a quadcopter and also a PID controller for the sliding surface. Furthermore, in [Robust Integral Sliding Mode Control Design for Stability Enhancement of Under-actuated Quadcopter] an interesting approach is done by matching a backstepping algorithm with an integral sliding mode controller in order to achieve more stabilization of the overall closed-loop and avoid other limitations related to the reaching phase of the SMC method.

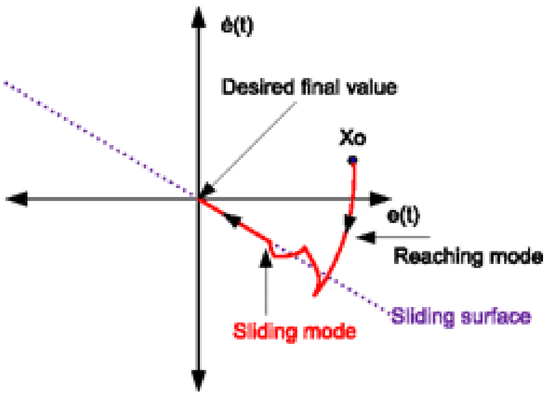

Figure 2.3: Slide Mode Control concept (Reproduced from [15])

Nonlinear Model Predictive Control is one of the two variants - linear (LMPC) and nonlinear (NMPC) - of model predictive control: a more modern method that takes into account a various number of constraints present in the control process (torque limits, roll angles, etc..), and also multiple input and output (MIMO) systems and interaction between them. It works by using an internal model of the dynamic system that is being controlled, the current plant measurements, and the desired targets and limits to predict changes on the dependant variables - which are often control objectives and other process limits. By doing this, the system is able to drive these variables close to the desired values while respecting the other constraints on dependent and independent variables. This preview capability offers the possibility of the UAV adopting the desired behavior with enough time to not have to do it abruptly and rather in a smoother maneuver. The downside of this robustness is that it requires powerful computational process capabilities and memory to find and store the solutions calculated by the algorithm. In [16] a comparison between the two types of MPC is done regarding trajectory tracking in quadcopters, where NMPC nonlinear system models, present better disturbance rejection capability, and step response. Furthermore, to counter the excess of time spent in processing, being difficult to implement for working in real-time. Therefore, to counter this obstacle, in [17] the Newton generalized minimal residual (Newton/GMRES) method is successfully applied to a quadcopter being controlled by a high-level NMPC controller.

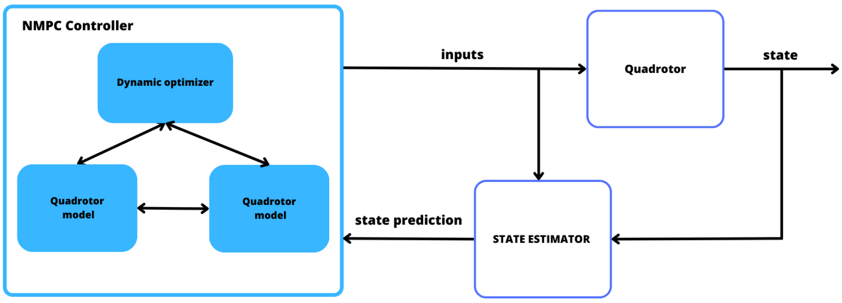

Figure 2.4: Basic NMPC controller loop applied for a quadrotor (Adapted from [18])

Artificial Neural and Networks Another more recent technique implies the use of artificial intelligence algorithms when developing controllers for these operations, such as, for example, neural networks and fuzzy logic, which are the most widely used.

In fact, the first one consists of a complex mathematical model biologically based on the human brain and its nervous system, in order to replicate its learning capacity. In this way, by connecting artificial neurons, or perceptions, that work as similarly as biological ones, and forming a mathematical function that receives a signal from one or more inputs and sums them to produce an output.

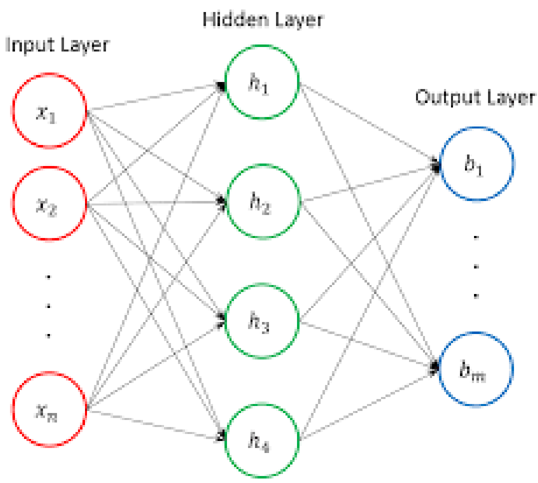

Figure 2.5: A Neural Network (Reproduced from [19])

From this, the neural network is trained by processing examples that contain a known "result" from a known "input", creating probability-weighted organizations between the two that are stored in the net's data structure. Then, the training happens by constantly evaluating the difference between the formed output or prediction, and the target output, defining an error from where the net bases its learning rule, and using it to regulate the weighted associations. In such a manner, repeated alterations will permit the network to approximate its output to the desired one.

For this reason, it results in an acceptable control method for a very nonlinear vehicle like quadrotors, due to the fact that it can better reduce the effects of plant parameters changes and handle environmental disturbances, by compensating in a variety of uncertainties, in relation to more standard controllers that apply fixed values for gain. Nevertheless, implementing this method usually insinuates the need for complex and ample computational resources, being this the main limitation. Examples of this method being applied in quadrotor control can be seen in [20], where successful real-time simulations were performed in a confined space, showing promising results not only in the closed-loop stability and robustness in facing unknown uncertainties but also in creating a real-time model to handle uncertainties sustained during a flight in that specific environment.

2.2 Drone control strategies for aggressive manoeuvres

In the last stage of this project, it will be developed a second controller to tackle some aggressive maneuvers present in the capturing and launching trajectories that must be performed by the quadrotor, such as following another vehicle and transporting it.

Thus, a new method is necessary to deal with the different difficulties present in each stage of the entire flight operation, while also handling, at the same time, present environmental disturbances. This means that the controlling techniques earlier exposed behave as possible solutions to specific required situations and, therefore, can be insufficient to handle this task while presenting good maneuverability and stability. There is a need for more complex and advanced options to increase further controlling robustness and response.

In more recent years, a hybrid control approach for UAVs is being increasingly researched and developed. By implementing this technique on controllers they are capable of performing different interactions with the environment, due to the method's characteristic where it is possible to manage a problem or situation that has different operational stages, containing continuous, logical, and discrete states. In this way, usually, this option is applied to hybrid systems, which distinctive behavior, such as for example, a hybrid UAV - a vehicle that has the ability to change between VTOL and fixed-wing flight mode - and, consequently, needs to be modeled, for better control effectiveness, as a hybrid system. In [21] a hybrid control method is successfully applied to a hybrid drone by implementing 3 different controllers for each existing flight state: horizontal, vertical, and transitional flight; and, then, a fourth one is implemented with the purpose of automatically commuting between them while also maintaining flight stability, obviously.

In the context of this thesis, aggressive maneuvering can also be achieved using this type of possibility, which its use may be justified by the need of handling the distinct phases present in the trajectory that our quadcopter must travel. Accordingly, this shuttle drone must be able, at some point, to follow another vehicle, catch it, carry it, and release it. These cause abrupt changes in its direction, velocity, attitude, weight, and aerodynamics, being necessary for a robust controller to effectively and safely achieve this trajectory. In order to perform capture maneuvers of another UAV, it's pivotal that our drone at some point is capable of following the trajectory of that UAV or intercepting it. For instance, in [22], a method is implemented for n follower quadrotors being able to steer another drone, the leader, while dealing with environmental disturbances in a real-time three-dimensional trajectory. Each follower is able to plan its converging trajectory individually, reducing the need to communicate with other vehicles, and, also, the leader is able to adjust its motion to its followers so that each one of them is able to steer the leader. This approach is also successfully proved by real testing experiments, obtaining more credibility. After capturing another vehicle, the controller implemented on the shuttle drone must be able to handle the abrupt change in weight, when carrying it, maintaining stability in the best possible way during the entirety of this process. For example, [23] exposes a relevant project where a quadrotor is set to exchange a package with another one, needing, as in this project, to carry a payload and coordinate its flight with another drone. To do that, a Hybrid Model Predictive Controller-based system is used to control the trajectory and the gripping mechanism.

2.3 Drone modeling

To better simulate the desired movement scenarios where the drone will operate, it's also very important to design a suitable recreation of the aircraft's motion and physics. However, in most of the literature, the attention given to the complexity of this topic is often forsaken since only the steady-state thrust and reaction torque of a rotor hovering in free space are usually modeled.

2.3.1 Aerodynamics

When studying the interaction between the drone and the surrounding air, it's important to observe the effect the last one has on the rotors of the drone. For that, the work done in [Influence of Aerodynamics and Proximity Effects in Quadrotor Flight] explores this effect in various flight conditions such as the ground effect (proximity to the horizontal surfaces) and in close proximity to other drones. Nevertheless, it only evaluates the effect of quadcopters in the same horizontal plane. Moreover, some of these authors in [24] determine "safe regions" around each vehicle in a team setting, by empirically quantifying the effect of the downwash produced by the rotors of one drone on another flying below the first.

On another perspective, an interesting approach is taken in **[Quadrotor Control in a Wind Field]** where a propeller model is designed combining important concepts such as **Blade Element Theory** with quadrotor classical dynamics to predict the side force, pitching moment and additional thrust generated by each propeller. By also adding the shielding effect (not all rotors are exposed to the wind in the same way), the work produces a satisfactory solution to the problem of the wind effect. In [Dynamic Control of Autonomous Quadrotor Flight in an Estimated Wind Field.], with the objective of having a vehicle flying underneath another, Momentum Theory is applied in order to model the downwash produced by the quadrotor. Here, the designed controller is incorporated with a recursive Bayes filter to estimate the downwash flow field's parameters. When performing the respective simulations, despite the results showing that is possible for a drone to hover below the first successfully, the propeller diameter of the quadcopter is never disclosed, and, therefore, this outcome is somewhat inconclusive.

In **[rotor drag]** it is possible to calculate the rotor drag, where the author proves that a quadcopter subjected to a linear version of this effect is deferentially flat in its position and heading. With this property, feed-forward control terms are computed directly from the reference trajectory and used in a cascaded, nonlinear feedback control law that enables accurate agile flight with quadrotors on a priori unknown trajectories. Therefore, the respective results show that the tracking position error is reduced significantly even in agile maneuvers.

On the other hand, the aerodynamic drag of the drone's frame is also considered by many authors as another important factor to add to the model of the drone, as can be seen in [Quadrotor Control in a Wind Field] and in [Accurate positioning of multirotor UAVs for civil infrastructure monitoring], where this element is studied more in-depth.

2.4 Drone integration

In the project where this thesis is inserted, it will also be required to integrate a quadrotor, which then will be used as the shuttle drone. Therefore, it is important to understand the process behind its instrumentation in order to better obtain the desired drone for our purpose.

In the last 3 decades, the number and variety of quadrotor vehicles have increased considerably, especially since the early 2000s (***FALTA DOCUMENTO***). For this reason, numerous papers have been written regarding their dynamics and design. In fact, idealizing and fabricating a quadcopter is a challenging process as it depends on a variety of important and different factors to have in account (purpose or finality of its use, the budget available to assemble all the parts and components, etc.), but, for this case, we will focus on the more technical idea behind miniature or small UAV (SUAV) quadcopters (2-20Kg [25]) production in general.

Accordingly, in this task, a variety of difficulties are inserted, which can be divided at the component level (e.g developing better sensors and more capable rotors) and at system-level (e.g better aerodynamics), and, in [26], large and detailed information is provided about developing techniques for both in designing a quadcopter heavier than 2 kg and larger than 1m. In addition, as in our project, it is sensible for the drone to achieve high motion capability while also managing its weight, trying to keep it as low as possible [27]. In [28], for example, some key concepts are provided about designing and constructing a quadrotor with 4 kg of weight and 1 kg of payload.

2.5 Flight simulation and testing

Also, to conduct our studies, it's obviously necessary to know how to perform flight tests with quadrotors, studying how to assemble a controlled space where real experimental try-outs can be performed. In this way, this section is aimed to expose relevant work about quadcopters flight simulation and testing procedures.

As it's possible to see in the literature, the great majority of the authors use MATLAB as the chosen method of simulating their algorithms and research for its simplicity and lower computational effort in comparison to other techniques. However, other more realistic environments can be used to devise more credible simulations. For example, [29] (**Tem de ser a do Francisco Matos na pág36**), uses ROS (Robot Operating System)- an open-source framework that allows the writing of robot software, (further information regarding this software can be found in **Robot Operating System (ROS). url: https://www.ros.org/. (Accessed: 20.08.2020)** - is used along with its interface to the simulation environment Gazebo and a firmware PX4 autopilot system that controls the vehicle. The first tool offers an open-source virtual environment well capable of simulating robot behavior by providing a robust physics engine with multiple options that can accurately simulate real-life conditions for both indoor and outdoor scenarios; (more detailed information about this simulator can be seen in **in (Gazebo - Robot simulation made easy. url: http://gazebosim.org/. (Accessed: 01.10.2020).)**. On the other hand, PX4 is a flight control system for low-cost UAVs and, in this case, it connects to the ROS through MAVROS protocol, delivering finer real-time testing samples through its integrated simulator. Further information regarding PX4 and its vast available capabilities can be explored at https://px4.io/). A connection between Simulink and PX4 is also established to test the developed controller. Figure 2.6 depicts the architecture of this system

After finishing the development of **each** controller and their respective virtual simulations, it's necessary to perform experimental flight tests to evaluate their performance on the developed drone in a real 3D environment. For this reason, there are two major categories of tests that can be considered for the project where this thesis is inserted: one to check a single drone's basic flight, performing other slow-speed and low-distance maneuvers indoors, such as take-off, hovering, and landing; and, in an outdoors environment, the second controller responsible for drone aggressive maneuvering will be put to the test. Obviously, each type of test will require separate hardware and software systems.

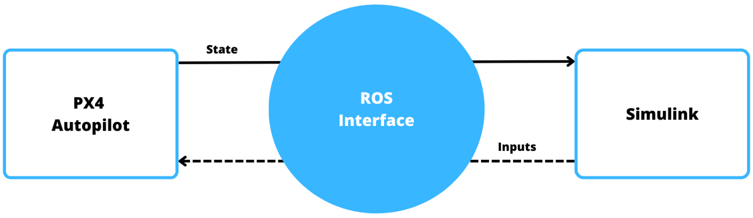

Figure 2.6: Virtual simulations system's architecture

In [29] a good amount of work is done regarding this subject, suggesting a valuable method to develop a safe environment to perform indoor basic flights with one and multiple drones. The UAV simulation flight is done in a controlled environment: a testbed, consisting of a sort of arena with all the sensors and guard measures to provide the best conditions for UAV experimental flights, which is needed for this type of test. In this case, the drone's trajectory is followed with good precision and low latency, using Marvelmind Indoor Navigation System. Then, its trajectory is transmitted to a ROS topic and sent to PX4 using a Wi-fi module, (ESP2866), ensuring this low latency in the transmission.

On the other hand, outdoor quadrotor testing presents more complex challenges compared to the previous scenarios presented. In fact, one of the biggest problems of these types of operations is the unpredictable effects of environmental factors, such as wind, rain, temperature changes, etc., which can significantly affect the performance, battery life, navigation, and control of the quadrotor. Also, if a drone loses control or crashes, these tests can pose safety risks to people and property, ensuring that these procedures are conducted in a safe environment, away from people and objects that could be damaged, and complies with the regulatory restrictions of the location. Moreover, another main issue of outdoor quadrotor testing is its navigation and positioning, which often relies on GPS (Global Positioning System) system. This method can be problematic because the signals used by this system to communicate with the drone can be weak or unreliable in certain environments, such as urban areas with tall buildings or dense forests, disturbing the testing procedure's effectiveness.

The research paper "Autonomous capture of agile flying objects using UAVs: The MBZIRC 2020 challenge" includes a section on outdoor testing. The authors of the paper describe how the MBZIRC 2020 challenge, which focused on the autonomous capture of agile flying objects using UAVs, was conducted outdoors. They provide a detailed description of the outdoor test arena and highlight the challenges of conducting the challenge in an outdoor environment, such as dealing with wind and other weather conditions and ensuring the safety of participants and bystanders. The paper also includes visual materials, such as videos and images, to illustrate the outdoor testing that was carried out during the challenge.

3

Background Theory

In this section, some aspects behind quadcopters, - such as movement and **aerodynamics** - are studied due to their pivotal importance in understanding the work developed in this dissertation.

3.1 Drone movement

Before making assumptions on a control approach for the quadcopter, it's pivotal to study its movement by being familiarized with its general dislocation capabilities and limitations in a 3D environment. Only then, and also knowing the flying goals for the project, one can model this movement correctly, presenting the most approximated mathematical background that represents it. Therefore, in this section, an explanation of the drone motion will be given and then its consequent modulation will also be presented.

Accordingly, this vehicle has six degrees of freedom (DOF), which consists of the quantity of motion that it is able to perform, being, in this case, transitional movement in X, Y, and Z (horizontal, vertical and depth); and rotational movement in pitch, yaw and roll as it is possible to be seen in figure 3.1.

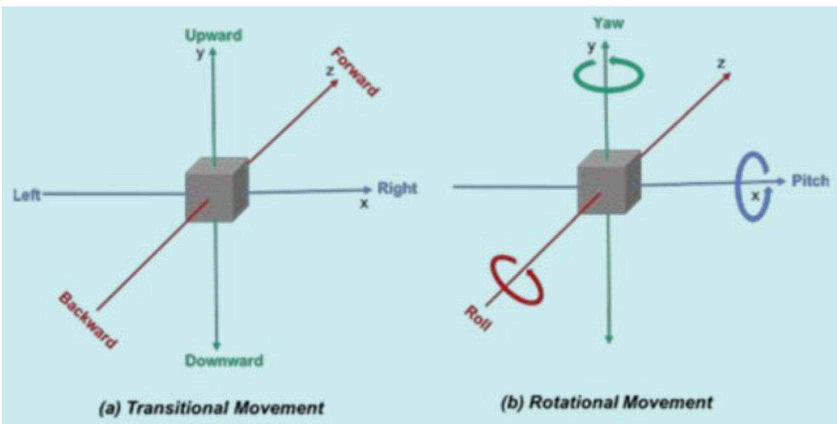

Figure 3.1: Quadrotor's 6 degrees of freedom (Reproduced from [30])

To perform this movement, this type of drone uses its 4 control inputs, representing the speed of each rotor, and by having a lower number of control inputs than degrees of freedom from where it is able to move, the quadrotor can be characterized as an underactuated system. For instance, while the drone is able to move along the z-axis without needing to variate any other state, the same doesn't happen for the x and y axis, as it needs to change its attitude.

The motion in these directions can be achieved by having the following propeller speed variation [31]:

- Changing the speed of all propellers simultaneously will generate vertical z motion;
- Varying speed 2 and 4 rotors inversely will make a roll rotation;
- Varying the speed 1 and 3 rotors inversely will produce a pitch rotation;
- The difference in the counter-torque between each pair of rotors will generate a yaw rotation.

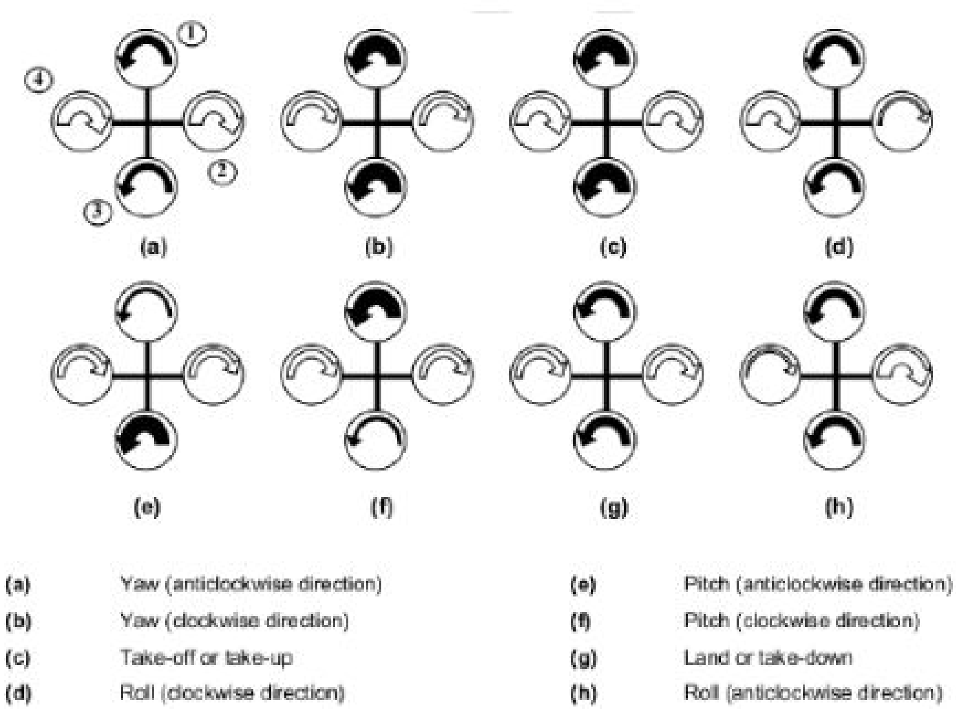

Figure 3.2: Quadrotor operating principle, relationship between propellers speeds and vehicle movements(Reproduced from [32])

This proves that the quadrotor motion is a complex theme and, by adding disturbances of real-world environments, which are usually difficult to integrate and model, the more complex the problem to devise feasible control solutions for quadcopters is. Therefore, in the next chapters, some modeling and control technique's to tackle this problem will be presented and then analyzed to be possible to see their impact.

4

Basic drone modeling and control

As it was previously mentioned, one of the objectives of this thesis is to design and implement a controller that is able to follow desired and more simple trajectories. The algorithm for this controller will serve as the "foundation" for a second and more robust one, which will be later compared for a performance study.

Therefore, this chapter presents, firstly, a description of the model of the quadrotor and then one of its controlling techniques.

4.1 Drone modeling

To represent the complete UAV dislocation in a 3D environment, one must consider two references: one for the environment where the drone is flying, the world frame W, and another for the own body of the vehicle, the body frame B.

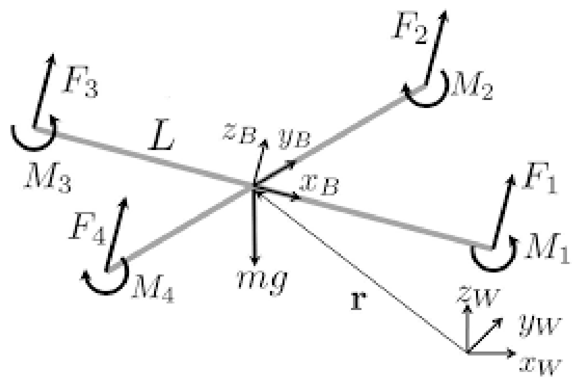

Figure 4.1: Reference frames (reproduced form [33])

Also, to describe the system, it's used a state vector x,

$$x = \begin{bmatrix} p & v & \omega_B & W_{R_B} \end{bmatrix} \tag{4.1}$$

where
$p = [p_x, p_y, p_z]^T$ is the body's position;

$v = [v_x, v_y, v_z]^T$ is linear velocity ;
$\omega_{des} = \begin{bmatrix} \omega_{B_{x,des}} & \omega_{B_{y,des}} & \omega_{B_{z,des}} \end{bmatrix}^T$ is the angular velocity ;
$W_{R_B} = [x_B, y_B, z_b]$ is the rotation matrix that describes the attitude of the body frame in the world frame.

4.1.1 System's dynamics

The system's dynamics will depend on the different trajectory phases because of the different physical influences the aircraft suffers in each one of them, mainly when the drone is flying in close proximity to one another. For that reason, each one of those variations is studied and included in the model of the drone.

In this way, the motion of the aircraft can be described using the following dynamical equations, which can be obtained by deriving each one of the components of the state vector x, being:

$$\dot{p} = v \tag{4.2}$$

$$\dot{v} = -g z_w + \frac{u_1}{m} z_B \tag{4.3}$$

$$\dot{\omega}_B = I^{-1}(-\hat{\omega}_B I \omega_B + u_r) \tag{4.4}$$

$$w_{\dot{R}_B} = w_{R_B} \hat{\omega}_B \tag{4.5}$$

with the moment of inertia I being a 3x3 matrix , gravitational force is $g = 9.81 m/s^2$, $z_w = [0, 0, 1]^T, z_B = W_{R_B} \cdot z_W$ being the z-axis of the body frame, and, finally, $\hat{\omega}_B$ being the skew matrix obtained from ω_B

$$\hat{\omega}_B = \begin{bmatrix} 0 & -\omega_{Bz} & \omega_{By} \\ \omega_{Bz} & 0 & -\omega_{Bx} \\ -\omega_{By} & \omega_{Bx} & 0 \end{bmatrix} \tag{4.6}$$

4.2 Drone control

In order for the drone to perform hovering maneuvers and dislocate from one point to another while maintaining stability, a controller based on the one developed in [34] was implemented. As the majority of controllers when trying to approximate a system to a reference value, this one will do it by trying to guide the controlled state variables to zero by minimizing the error between the reference value and the state's current one, meaning that as the error approaches the value zero, the system follows the path even closer. Also, a quadrotor can roll about any axis in the body's horizontal plane, and thus minimizing the snap or 4th derivative of independent splines in three dimensions will minimize abrupt changes in control inputs needed to follow the path.

In this case, we need to compute the position and velocity errors that can be defined as

$$e_p = p - p_{ref} \tag{4.7}$$

$$e_v = v - v_{ref} \tag{4.8}$$

These values will constantly be used by the controller to calculate the desired force vector F_{des}, which will try to follow, as

$$F_{des} = -K_p e_p - K_v e_v + mgz_W + ma_{ref} \tag{4.9}$$

K_p and K_f controller gains for position and velocity, respectively, are defined as positive definite diagonal matrices. Then, we can obtain the first control input, the thrust one, u_1, by calculating the scalar projection of F_{des} onto z_b, such as

$$u_1 = F_{des} \cdot z_B \tag{4.10}$$

and the other 3 control inputs, u_τ, which relate to the body attitude, are obtained by computing the remaining errors, rotation and angular velocity, $-K_r$ and K_w respectively. To do that, we need to calculate the desired rotation matrix, $w_{R_B} = \left[x_{B_{des}}, y_{B_{des}}, z_{B_{des}}\right]$, operating on its coordinates. First, we are able to obtain $z_{B_{des}}$ by next formula (taking in account that $||F_{des}|| \neq 0$)

$$z_{B_{des}} = \frac{F_{des}}{||F_{des}||} \tag{4.11}$$

Now we will also need to define the intermediate coordinate frame C after the reference yaw rotation angle, ψ_{ref}, in order to obtain $X_{C_{des}}$ as

$$X_{C_{des}} = \begin{bmatrix} cos(\psi_{ref}) & sin(\psi_{ref}) & 0 \end{bmatrix}^T \tag{4.12}$$

With this, we can obtain another component of the desired rotation matrix, the $y_{B_{des}}$ value, by performing the orthogonal between $z_{B_{des}}$ and $X_{C_{des}}$, supposing $\left\| z_{B_{des}} \times X_{C_{des}} \right\| \neq 0$, as

$$y_{B_{des}} = \frac{z_{B_{des}} \times X_{C_{des}}}{\left\| z_{B_{des}} \times x_{C_{des}} \right\|} \tag{4.13}$$

Finally, we can compute the vector, $x_{B_{des}}$, as

$$X_{B_{des}} = y_{B_{des}} \times z_{B_{des}} \tag{4.14}$$

And, also, we are able to obtain the rotation error, written as

$$e_R = \frac{1}{2}(W_{R^T_{B_{des}}} W_{R_B} - W_{R^T_B} W_{R_{B_{des}}})^\vee \tag{4.15}$$

where $\vee$, *vee* map, is the inverse calculation of the skew matrix.

Now, in order to define the desired angular velocity $w_{B_{des}}$, we need to work with the formula of the derivative of the acceleration $\ddot{a}_{ref}$, which is

$$m\ddot{a}_{ref} = \dot{u}_1 z_{B_{des}} + W_{R_{B\,\omega_B}} \times u_1 z_{B_{des}} \tag{4.16}$$

and project it along z_b, knowing that $\dot{u}_1 = z_b \cdot m\dot{a}$, by creating the vector $h_{w_{des}}$ as

$$h_{w_{des}} = W_{R_{B\,\omega_{Bdes}}} \times u_1 z_{Bdes} = \frac{m}{u_1}(\dot{a}_{ref} - (z_{B_{des}} \cdot \dot{a}_{ref}) z_{Bdes}) \tag{4.17}$$

which is the projection of $\frac{m}{u_1}\dot{a}$ on the $x_B - y_B$ plane. From here, using the definition of $h_{w_{des}}$ and the already known values of y_{Bdes} and X_{Bdes}, it's possible to express $w_{B_{x,des}}$ and $w_{B_{y,des}}$ as

$$w_{B_{x,des}} = -h_{w_{des}} d y_{Bdes} \tag{4.18}$$

$$w_{B_{y,des}} = h_{w_{des}} \cdot x_{Bdes} \tag{4.19}$$

and, therefore, calculate $w_{B_{z,des}}$ by resolving the following system which relates the derivatives of the Euler angles and the angular velocity, written as

$$w_{R_B} \begin{bmatrix} w_{B_{x,des}} \\ w_{B_{y,des}} \\ w_{B_{z,des}} \end{bmatrix} = \begin{bmatrix} x_{C_{des}} & y_{B_{des}} & z_{B_{des}} \end{bmatrix} \begin{bmatrix} \phi_{des} \\ \theta_{des} \\ \psi_{des} \end{bmatrix} \tag{4.20}$$

Finally, we can define the desired angular velocity as

$$\omega_{des} = \begin{bmatrix} \omega_{B_{x,des}} & \omega_{B_{y,des}} & \omega_{B_{z,des}} \end{bmatrix}^T \tag{4.21}$$

obtaining, in this way, it's an error, which written as

$$e_\omega = \omega_B - \omega B \tag{4.22}$$

and, with it, we can calculate the remaining control inputs u_τ as

$$u_\tau = -K_r e_r - K_\omega e_\omega \tag{4.23}$$

being K_r and k_ω the orientation and angular rates diagonal gain matrices, respectively.

5

Aerodynamics modeling and control

As previously shown, a quadrotor faces a variety of different aerodynamic factors that affect its motion in different ways, mainly depending on the physical characteristics of the drone, the environment where it is flying, and the flight trajectory that it's following. For this purpose, this chapter is divided into two groups aimed to describe modeling and controlling techniques for aerodynamic interactions present in the scenario of one drone flying and another for two drones flying in close proximity to one another.

5.1 Single drone

5.1.1 Rotor drag

Following the work developed in [35], we take in consideration the rotor drag effect for this model, inferring in the same manner that it is deferentially flat in its position and heading. In this way, some new changes are applied to the last dynamical equations, having now:

$$\dot{v} = -gz_w + cz_B - RDR^T v \tag{5.1}$$

being $c = \frac{u_1}{m} + k_h(v^T(x_B + y_B))^2$ the mass-normalized collective thrust, where the term $k_h(v^T(x_B + y_B))^2$ represents a quadratic velocity-dependent input disturbance. Moreover, $D = diag(d_x, d_y, d_z)$ is the constant diagonal matrix formed by the mass-normalized rotor-drag coefficients. Furthermore, the authors in [35] also present a new formulation for the $\dot{w}$ equation. Still, because of the limited resources and especially time, we don't assume this new format as we are not able to obtain the values for the unknown variables required in this new dynamical equation. Instead, we use the previous one and only consider important the effects on the velocity of the aircraft's frame.

5.1.2 Frame drag

In this type of trajectory, the quadrotor is also subject to wind effects that can come from the atmospheric conditions of the environment and from another drone flying in close proximity. The effect of airflow on a quadrotor's motion is partially caused by the interaction between that airflow and the rotors, and partly by the drag of the drone's frame, which has significant effects on its movement.

To model this effect, we take into account the work developed in [31] where not only the aircraft's airframe drag but also the interaction between the flow passing through its propellers and the wind is considered, although, for this work, only the first one is studied[infraetsructure]:

$$F_{drag} = -\frac{1}{2}\rho C_d A (V_{air})^2 \cdot sgn(V_{air}) \tag{5.2}$$

being Cd the drag coefficient, V_{air} is the relative air speed **(to the body of the drone?)**, the air density and A is the platform's projected area in the corresponding plane and depends on the axes with respect to which it is calculated:

$$A = \begin{bmatrix} A_x & 0 & 0 \\ 0 & A_y & 0 \\ 0 & 0 & A_z \end{bmatrix} \tag{5.3}$$

It's important to note that this reference area remains constant regardless of the orientation of the vehicle because it has been chosen to calculate the aerodynamic drag directly in the body frame. The **drag coefficient** (numerical value taken from [36]), it's assumed that it is the same when the drone is flying in any direction along a body axis, because of the symmetry of the quadrotor body frame. On the other hand, the relative air velocity is calculated as follows:

$$V_{air} = V - V_w - V_c \tag{5.4}$$

where $V_w = W_{R_B} ws$ is the wind velocity felt on the drone's body (ws is the wind speed on vector $[ws_x ws_y ws_z]$) and V is the drone's body speed and V_c is the vertical velocity of the downwash, which corresponds to the additional velocity imparted by the rotor. This last speed can be obtained by taking in account the approach done in **[dynamic]**, which uses momentum theory to describe the downwash of each propeller, defining this vertical velocity at any point z beneath the height of the rotor z_r as [36]

$$Vc = vi + vi \tanh(-k\frac{z_r - z}{h}) \tag{5.5}$$

being h and k shaping parameters that manage how rapidly the area of the stream tube below the rotor converges to its steady-state value. Figures 5.12 and 5.2 display an example of a drone hovering at z = 3.20m and the amount of downwash it produces according to the distance to its rotors, **with v_i varying from 43 m/s to 48 m/s and a rotor radius of 0.15 meters.**

Note that, in the first figure, it's possible to see that, as the distance from the propeller decreases, the intensity of the vertical downwash increasingly gets much higher. However, when we consider a z position below the quadrotor's propellers higher than 2 rotor radii,

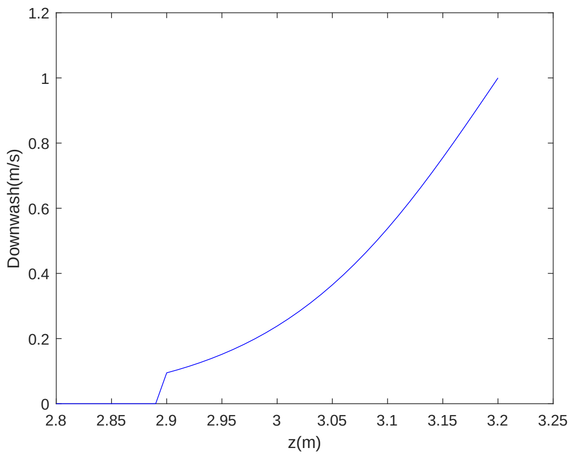

Figure 5.1: Vertical velocity of the downwash using 5.12 as model

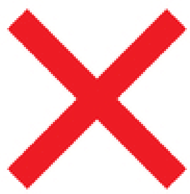

Figure 5.2:

its vertical velocity is considered to be negligible. Furthermore, the second figure depicts the gradual decrease of downwash when the distance in the x and y axis from the center of the rotor increases.

Also, v_i is the induced velocity of a propeller or, in other words, the vertical velocity of the airflow directly underneath it, and can be calculated following the example given in [36] where blade momentum theory, which relates the additional kinetic energy of the air at an infinite distance from the rotor to the thrust, is applied to determine the following expression for the thrust in descent or climb motions:

$$T = \dot{m}w = 2\rho A v|V + v| \tag{5.6}$$

where T is for the rotor thrust, ρ for the density of the air, A is the area swept by the rotor, m is the mass flow through a disc of area A and, finally, w is the velocity of the air at an infinite distance after it has passed through the rotor. Then, to finish the system of equations, blade element theory is used to formulate another equation

$$T = \frac{\rho abcw^2R^3}{4}(\theta_{tip} - \frac{V + vi}{wR}) \tag{5.7}$$

Here the a, b, c, θ_{tip}, R variables are used to describe the geometry of the propellers, more specifically the lift curve slope, number of blades of the rotor, the blade cord, the pitch angle at the blade tip and the radius of the rotor, respectively. w is the rotor speed. Therefore, if $V = 0$, we can obtain the induced velocity as the result of $\sqrt{\frac{T}{2\rho A}}$. Furthermore, having (9) and (10) equations, an angular speed w and a climb velocity V, we can cluster the rotor-describing variables that depend on assumptions on the rotor construction in two groups and rewrite (10) as

$$T = k_1 w^2 + k_2(V + v)w \tag{5.8}$$

being $k1$ and $k2$ associated with the rotor geometry and density of the air. In [37], these values are determined in a laboratory by developing a test rig that collects data from measuring the steady state rotor speed to determine w required to produce a thrust of mg at the given vertical velocity of an ascending and descending micro quadrotor flying at a constant speed. This way, this procedure formulates the best $k1$ and $k2$ that fit the data collected. More details of this approach can be seen in [37].

Once these constants are obtained, by already having thrust T and the drone's velocity V provided by the controller, it's possible to calculate the induced velocity vi. This way, we could, once again, redefine a new dynamical equation for the drone's velocity (already including the rotor drag effect) as

$$\dot{v} = -gz_w + cz_B - RDR^T V - \frac{1}{2}\rho C_d A(V_{air})^2 \cdot sgn(V_{air}) \tag{5.9}$$

5.1.3 Rotor drag compensation

In order to predict and compensate for the effects of the aircraft's propeller drag, we use the feed-forward control terms suggested in [35], which implies that the states $[p, v, R, w]$ and the thrust inputs $[u_1, u_t]$ can be written as algebraic functions of four specific flat outputs and also a finite number of their derivatives. For this purpose, we start by calculating the desired acceleration of the vehicle's body such as

$$a_{des} = a_{fb} + a_{ref} - a_{rd} + gz_w \tag{5.10}$$

being $a_{fb} = -k_p e_p - k_v e_v$ the acceleration that derives from the feed-back control terms determined using the velocity and position errors and $a_{rd} = R_{ref} D R_{ref}^T v_{ref}$ is the acceleration that is caused by the rotor drag effect. After this, we are able to determine the collective thrust input as

$$u_1 = a_{des}^T z_B - k_h(v^T(x_B + y_B))^2 \tag{5.11}$$

Here, kh consists in a constant that inserted in $k_h(v^T(x_B + y_B))^2$ forms a quadratic velocity dependent input disturbance. Additionally, the drag coefficients kh and D used to compute the feed-forward control properties above were chosen taking into account the results obtained from the procedure done in [35]. Here, a Nelder-Mead gradient-free optimization is applied to a finite number of cycles where the drone follows a predefined trajectory, and for each one of them, different drag coefficients are used in order to minimize the **absolute trajectory tracking error**, using it as cost for the optimization.

Furthermore, in [35] a quality comparative study with other controllers that take into consideration the rotor drag effect is made, showing that the proposed controller, which used the **differential flatness property and considered the asymmetry of the vehicles**, had better trajectory tracking performance than the others.

5.1.4 Frame drag compensation

5.2 Two interacting drones

In this part, a solution for modeling the situation of one quadrotor flying above another will be described and analyzed.

For this reason, two better distinguish the mathematical formulas featured in each of the drones' algorithms, we will assume that the aircraft flying above is Vehicle 1 and the other Vehicle 2. Moreover, we take into consideration that the two drones are identical in shape and size, and use the same physical components and materials.

In this way, we start by calculating the downwash produced by vehicle 1 as:

$$V_c = v_i + v_i \tanh(-k\frac{z_r - z}{h}) \tag{5.12}$$

where z_r is the rotor height of that vehicle and k, h and v_i are calculated the same way as the previous scenario. Then, the same process used in the earlier section is used when we obtain the $V_{air} = V - V_w - V_c$ and calculate the frame drag for this drone.

After this, we pass to the second drone and determine the air relative velocity V_2air as it follows

$$V_{2air} = V_2 - V_{2w} - V_{2c} - V_d \tag{5.13}$$

where $V_{2w} = W_{2R_B} ws$ is the wind velocity felt on the drone's body, V_2 is the drone's body speed (ws is the same wind velocity as previously), V_{2c} is the vertical velocity of the produced downwash and $V_d = W_{2R_B} V_c$ is, on the other hand, the felted downwash.

Finally, we are order to determine the frame drag from the second drone, which can be formulated again in the same way as in the individual drone flight.

6

Simulation results and discussion

The effectiveness of the developed solution can be observed in this section, where simulation results regarding various flight situations, involving one or two drones, are displayed and consequently analyzed. Therefore, firstly, the overall setup of this process is briefly described, explaining the implementation of the model and control schematics. Then, the performed simulations are disposed of in two groups: one regarding trajectory following a performance by a quadcopter and another **portrays** the close proximity flight between two quadcopters.

6.1 Environment setup

The quadcopter's controller and dynamic model were implemented in MATLAB using its simulink's feature, therefore, providing the simulation environment to be assembled in this project. In this way, by operating on both reference and obtained values of position, velocity, angular velocity, and rotation matrix, the controller outputs the $[u_1, u_\tau]$ to the drone's model, which is responsible for obtaining the derivative values of the previous parameters.

In order to obtain maximum stability and smoothness through the proposed trajectories, also having a quick rising dislocation, it is needed to adjust the respective gain values k_p, k_v, k_r, k_w, as operating on k_p, k_v gains contributes to better position tracking and k_r, k_w to less oscillation. For this, multiple tests were conducted, empirically obtaining the following values

$$\mathrm{K}_p = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, K_v = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, K_w = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, K_r = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

6.2 Trajectory following by one drone

This section presents various scenarios where one drone follows different paths under various conditions inserted in its modeling and control designs. Therefore, in the chosen

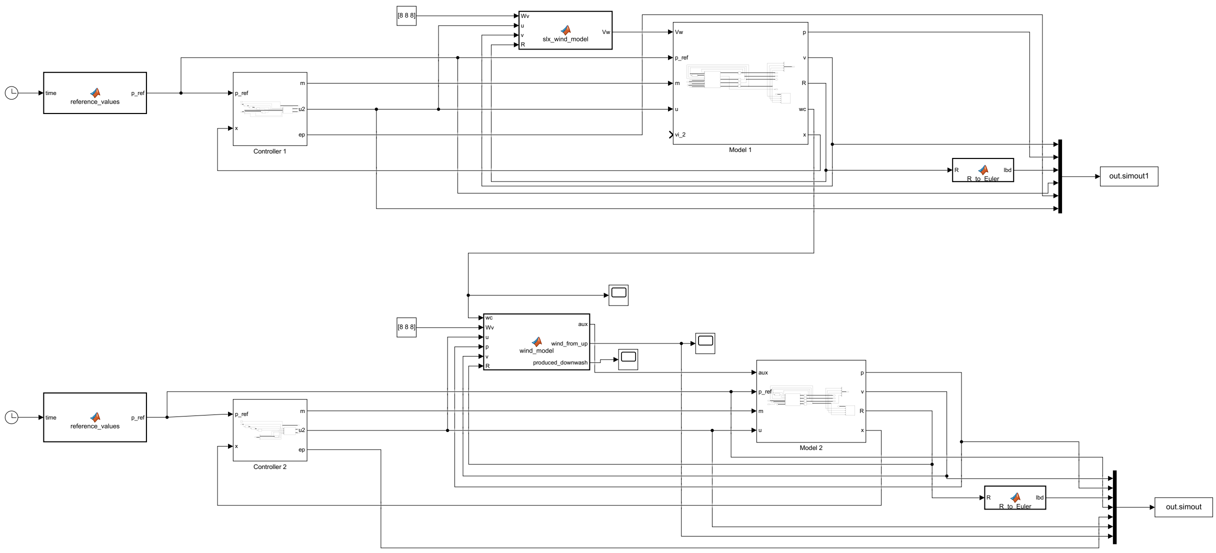

Figure 6.1: Block diagram of quadcopter controller in Simulink

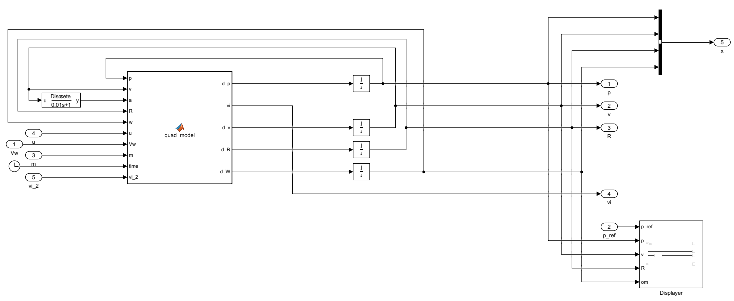

Figure 6.2: Block diagram of quadcopter dynamics in Simulink

trajectories, the rotor drag and the quadrotor's frame drag and the wind effect on the drone will be exposed, through different simulations. For that purpose, we start by testing the most basic version of the controller and then we will add more characteristics, testing their effect.

6.2.1 Basic nonlinear controller

We test the performance of the controller implemented in Chapter 4 by, firstly, applying a step reference signal at t = 2 seconds, from 0 to 4 meters, in all 3 coordinates individually, to test the step response. Secondly, we test the rotation reference tracking by inputting a step at t=2 from 0 rad to $\frac{\pi}{2}$. The respective results are displayed in Figures **...**

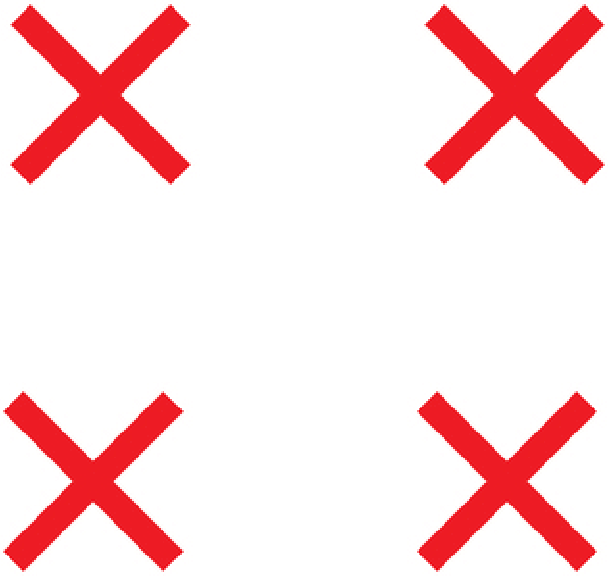

Figure 6.6: Position in X in Gerono lemniscate trajectory

Analysing these figures, it is possible to conclude that the tracking response of this controller presents fast rising and settling times and lack of overshoot, which are good attributes and, therefore, more complex trajectories can be tested while also inputting more aerodynamic **conditions** to the controller.

6.2.2 Rotor drag

In this part, we will test this effect by, firstly, considering it only on the drone model, and, then, we will display the same situation but with the respective compensation on the controller. In this way, for this simulation, a complex route, such as the Gerono lemniscate, is used to test the rotor drag effect and it can be defined as

$$[x(t) = 2\cos\sqrt{2}t; y(t) = 2\sin\sqrt{2}; z(t) = 3]$$

.

being $t \in [0, 30]$ and the vehicles starting position is $p_0 = [0; 0; 3]$. **with a ,a maximum collective mass-normalized thrust of $c =$ and maximum velocity of $v =$.** For the rotor drag coefficients, we used the same as in **[rotor drag]**, where the identical trajectory is simulated. To represent the two situations, there will be vehicle 1, where the rotor drag only affects the model of the aircraft, and vehicle 2, where this condition is present on the model but also compensated by the controller.

Therefore, Figure shows the two drones' trajectory on a 3D plot and Figures **...** display the controller's performance in the x, y, and z axes, respectively.

Figure 6.7: 3D Plot of Gerono lemniscate trajectory

Analyzing these simulations, it is possible to see that both controllers have good reference tracking, but the one operating on the second vehicle, that considers the rotor drag effect, shows significantly less tracking error on the x and y coordinates of the path and also less on the z coordinate. On the ,

6.2.3 Frame drag

For this part, the same simulations

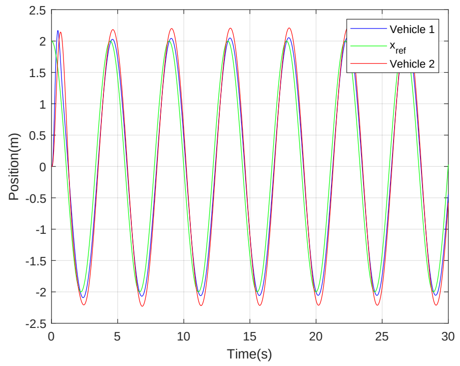

Figure 6.8: Position in X in Gerono lemniscate trajectory

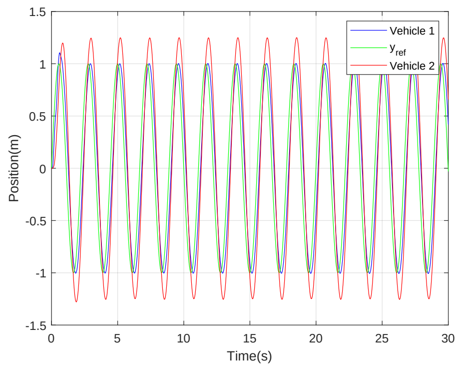

Figure 6.9: Position in Y in Gerono lemniscate trajectory

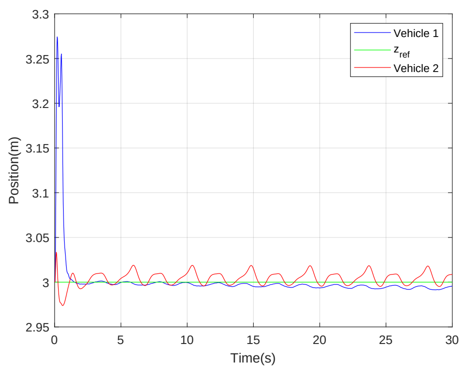

Figure 6.10: Position in Z in Gerono lemniscate trajectory

Figure 6.11: Velocity in X in lemniscate trajectory

Figure 6.12: Velocity in Y in lemniscate trajectory

Figure 6.13: Velocity in Z in lemniscate trajectory

Figure 6.14: Angular velocity in X in lemniscate trajectory

Figure 6.15: Angular velocity in Y in lemniscate trajectory

Figure 6.16: Angular velocity in Z in lemniscate trajectory

Figure 6.17: Rotation in Yaw in lemniscate trajectory

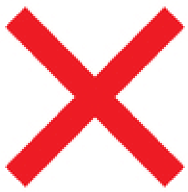

Figure 6.18: Rotation in Roll in lemniscate trajectory

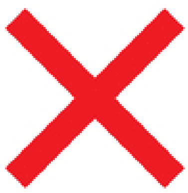

Figure 6.19: Rotation in Pitch in lemniscate trajectory

6.3 Two drones flying in close proximity

On this second group of simulations, there will be the drones 1 and 2, being the first one the vehicle that flies below the second and is affected by the downwash of this last drone. These two UAVs start in the positions $p_{o1} = [0;3;3]$ and $p_{o2} = [0;3;92,5]$, respectively. From here, the first one will follow a reference linear trajectory, which is defined by $[x(t) = t; y(t) = 3; z(t) = 3]$. On the other hand, the second aircraft will fly above the first gradually approaching it and then distancing from it by performing the following path $[x(t) = t; y(t) = 3; z(t) = exp((0.5*(t-15)^2)/5^2)) + 2.00 + h]$, where h is the minimum distance limit between the aicrafts in the z axis. Thus, the shuttle quadrotor will fly above the other quadrotor for 30 seconds, but will do it in close proximity for 10, taking 5 seconds to approach and 5 seconds to distance from it.

Two scenarios will be tested: one where the two aircrafts will be at the minimum distance in the z axis of 0.15 meters and another where that distance will be 0.40 meters. Figures... show..

Figure 6.20: 3D Plot of capture maneuvre

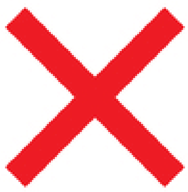

Figure 6.21: Position in X in capture manoeuvre

**ANÁLISE

6.4 Discussion

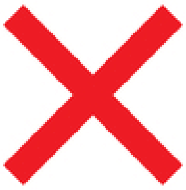

Figure 6.22: Position in Y in capture manoeuvre

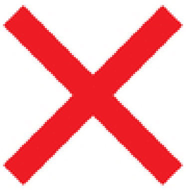

Figure 6.23: Position in Z in capture manoeuvre

7

Conclusions

Concluding, this dissertation focused on studying the problem and developing an algorithm for controlling a shuttle drone to perform capture maneuvers while considering aerodynamic interactions. The document

In fact, the first objective was exploring how quadrotor controllers worked by designing one while following an example of an already developed and tested nonlinear controller. This process is described in Chapter 4. and in Chapter 6 it's possible to see controller was successfully implemented using Simulink, showing good results,

The second objective, as described in Chapter 4.2, aimed to enhance the controller's performance by introducing additional aerodynamic considerations. By incorporating the effects of rotor drag and frame drag into the controller, significant improvements were achieved in robustness and adaptability, enabling the drone to navigate more realistically in various aerodynamic conditions.

Furthermore, once this was effectively accomplished, a scenario where the aerodynamic interaction between two quadrotors flying one above the other is addressed and effectively implemented, as can also be seen in Chapter 4.2.

Finally, in pursuit of the fourth goal, a dedicated chapter was methodically constructed to conduct a set of virtual simulations to evaluate the developed controller's performance over a wide range of scenarios. The results show the controller's competence, reactivity, and flexibility in a variety of environmental circumstances, flight profiles, and complex maneuvers, previously studied in this dissertation. The discussion of the results presented at the end of this chapter shows that...

7.1 Future work

For future work, there are some areas of research that could be investigated to further develop this project.

In fact, the first point could be improving the robustness of the developed model of the quadrotor by implementing the forward flight condition that would better suit the scenario of a capture maneuver, providing more realistic values of the induced speed of airflow produced by the propellers. Moreover, it is interesting to study a control compensation for the downwash effect on the drone flying beneath, in order to cancel the disturbances provoked by the motion of this last drone.

On the other hand, using more realistic simulation environments would significantly increase the credibility of the developed controller, setting it on a better course to later perform real flight experiments. A good example of this simulation technique would be very similar to the one presented in **Tesbed** since it provides better real-time testing samples and more environmentally complex scenarios.

Moreover, it would be interesting to integrate a quadrotor and later use it to perform real flight experiments. This not only would enable us to better validate the aerodynamic effects studied during the course of this thesis but also it would fulfill one of the goals of this project.

Bibliography

[1] M. R. Polaris. *Commercial UAV (Unmanned Aerial Vehicle) Market Share, Size, Trends Industry Analysis Report*. Accessed online: 2022-07-15. 2021. URL: https://www.polarismarketresearch.com/industry-analysis/commercial-uav-market (cit. on p. 1).

[2] B. Rabta, C. Wankmüller, and G. Reiner. "A drone fleet model for last-mile distribution in disaster relief operations". In: *International Journal of Disaster Risk Reduction* 28 (2018-06), pp. 107–112. ISSN: 22124209. DOI: 10.1016/j.ijdrr.2018.02.020 (cit. on pp. 1, 2).

[3] W. H. Maes and K. Steppe. *Perspectives for Remote Sensing with Unmanned Aerial Vehicles in Precision Agriculture*. 2019-02. DOI: 10.1016/j.tplants.2018.11.007 (cit. on p. 1).

[4] S. McDill and A. Richardson. *Could display drones snuff out the firework?* terceira referência
. 2021-07. URL: https://www.reuters.com/business/sustainable-business/could-display-drones-snuff-out-firework-2021-07-28/ (visited on 2022-02-16) (cit. on pp. 1, 2).

[5] "Project CAPTURE – Shuttle drone for launch and capture". In: (). URL: http://capture.isr.tecnico.ulisboa.pt/pt/ (visited on 2022-02-16) (cit. on p. 2).

[6] A. Zulu and S. John. "A Review of Control Algorithms for Autonomous Quadrotors". In: *Open Journal of Applied Sciences* 04 (14 2014), pp. 547–556. ISSN: 2165-3917. DOI: 10.4236/ojapps.2014.414053 (cit. on p. 6).

[7] G. E. M. Abro et al. "Performance Evaluation of Different Control Methods for an Underactuated Quadrotor Unmanned Aerial Vehicle (QUAV) with Position Estimator and Disturbance Observer". In: *Mathematical Problems in Engineering* 2021 (2021). ISSN: 15635147. DOI: 10.1155/2021/8791620 (cit. on p. 6).

[8] B. Aliyu, C. Osheku, and O. Sholiyi. "Conceptual Design, Analysis and Construction of a Fixed-Wing Unmanned Arial Vehicle for Oil and Gas Pipeline Surveillance". In: *Journal of Aircraft and Spacecraft Technology* 1 (2017-01), pp. 18–29. DOI: 10.3844/jastsp.2017.18.29 (cit. on pp. 6, 7).

[9] O. A. Dhewa, A. Dharmawan, and T. K. Priyambodo. "Model of Linear Quadratic Regulator (LQR) Control Method in Hovering State of Quadrotor". In: (). ISSN: 2289-8131 (cit. on p. 7).

[10] S. Abdelhay and A. Zakriti. "Modeling of a Quadcopter Trajectory Tracking System Using PID Controller". In: vol. 32. Elsevier B.V., 2019, pp. 564–571. DOI: 10.1016/j.promfg.2019.02.253 (cit. on p. 8).

[11] D. Mellinger, N. Michael, and V. Kumar. "Trajectory generation and control for precise aggressive maneuvers with quadrotors". In: vol. 31. 2012-04, pp. 664–674. DOI: 10.1177/0278364911434236 (cit. on p. 8).

[12] D. Cabecinhas, R. Cunha, and C. Silvestre. "A nonlinear quadrotor trajectory tracking controller with disturbance rejection". In: *Control Engineering Practice* 26 (1 2014), pp. 1–10. ISSN: 09670661. DOI: 10.1016/j.conengprac.2013.12.017 (cit. on p. 8).

[13] K. Klausen, T. I. Fossen, and T. A. Johansen. "Nonlinear control of a multirotor UAV with suspended load". In: IEEE, 2015-06, pp. 176–184. ISBN: 978-1-4799-6010-1. DOI: 10.1109/ICUAS.2015.7152289 (cit. on p. 8).

[14] R. Perez-Alcocer and J. Moreno-Valenzuela. "Adaptive Control for Quadrotor Trajectory Tracking with Accurate Parametrization". In: *IEEE Access* 7 (2019), pp. 53236–53247. ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2912608 (cit. on p. 9).

[15] R. Findeisen and F. Allgöwer. "An Introduction to Nonlinear Model Predictive Control". In: 2002-01 (cit. on p. 9).

[16] H. L. N. N. Thanh and S. K. Hong. "Quadcopter robust adaptive second order sliding mode control based on PID sliding surface". In: *IEEE Access* 6 (2018), pp. 66850–66860. ISSN: 21693536. DOI: 10.1109/ACCESS.2018.2877795 (cit. on p. 9).

[17] M. Kamel, M. Burri, and R. Siegwart. "Linear vs Nonlinear MPC for Trajectory Tracking Applied to Rotary Wing Micro Aerial Vehicles". In: vol. 50. Elsevier B.V., 2017-07, pp. 3463–3469. DOI: 10.1016/j.ifacol.2017.08.849 (cit. on p. 9).

[18] A. Mekky, T. E. Alberts, and O. R. Gonzalez. "Experimental Implementation of an ANN Controller for Quadrotor Trajectory Control in Confined Environment". In: IEEE, 2019-07, pp. 573–580. ISBN: 978-1-7281-1416-3. DOI: 10.1109/NAECON46414.2019.9058018 (cit. on p. 10).

[19] K. Holkar and L. Waghmare. "Sliding Mode Control with Predictive PID Sliding Surface for Improved Performance". In: *International Journal of Computer Applications* 78 (2013-09), pp. 1–5. DOI: 10.5120/13474-0992 (cit. on p. 10).

[20] H. M. Daniali. *Fast Nonlinear Model Predictive Control of Quadrotors: Design and Experiments*. 2019 (cit. on p. 11).

[21] M. R. Khosravani. “Application of Neural Network on Flight Control”. In: *International Journal of Machine Learning and Computing* (2012), pp. 882–885 (cit. on p. 11).

[22] B. Neves and B. Guerreiro. “Flight control of hybrid drones towards enabling parcel relay manoeuvres”. In: *2021 International Young Engineers Forum (YEF-ECE)*. 2021, pp. 13–19. DOI: 10.1109/YEF-ECE52297.2021.9505161 (cit. on p. 12).

[23] P. O. Pereira et al. “Leader following trajectory planning: A trailer-like approach”. In: *Automatica* 75 (2017-01), pp. 77–87. ISSN: 00051098. DOI: 10.1016/j.automatica.2016.09.001 (cit. on p. 12).

[24] N. Michael et al. “The GRASP Multiple Micro-UAV Testbed”. In: *IEEE Robotics Automation Magazine* 17.3 (2010), pp. 56–65. DOI: 10.1109/MRA.2010.937855 (cit. on p. 12).

[25] F. Matos and B. Guerreiro. “Model predictive control strategies for parcel relay manoeuvres using drones”. In: *2021 International Young Engineers Forum (YEF-ECE)*. 2021, pp. 32–37. DOI: 10.1109/YEF-ECE52297.2021.9505115 (cit. on p. 13).

[26] P. S. Ramesh and M. Lal. “Mini unmanned aerial systems (UAV)-A review of the parameters for classification of a mini UAV”. In: *International Journal of Aviation, Aeronautics, and Aerospace* 7 (3 2020). ISSN: 23746793. DOI: 10.15394/ijaaa.2020.1503 (cit. on p. 14).

[27] I. Robotics and A. Society. *Design Principles of Large Quadrotors for Practical Applications*. [IEEE], 2009. ISBN: 9781424427895 (cit. on p. 14).

[28] P. E. I. Pounds. *Design, Construction and Control of a Large Quadrotor Micro Air Vehicle*. 2007 (cit. on p. 14).

[29] R. Sa, G. Barreto, and A. Varela. “Design and construction of a quadrotor-type unmanned aerial vehicle: Preliminary results”. In: 2012-05, pp. 1–6. ISBN: 978-1-4673-0871-7. DOI: 10.1109/WEA.2012.6220398 (cit. on pp. 14, 15).

[30] H. Alexandre and R. Cabrita. *Design and control of a rotary wing drone testbed*. 2020. URL: https://github.com/joaomlourenco/novathesis (cit. on p. 17).

[31] D. Mellinger and V. Kumar. “Minimum snap trajectory generation and control for quadrotors”. In: IEEE, 2011-05, pp. 2520–2525. ISBN: 978-1-61284-386-5. DOI: 10.1109/ICRA.2011.5980409 (cit. on pp. 18, 24).

[32] S. Bouabdallah, P. Murrieri, and R. Siegwart. “Design and Control of an Indoor Micro Quadrotor”. In: vol. 5. 2004-01, 4393–4398 Vol.5. ISBN: 0-7803-8232-3. DOI: 10.1109/ROBOT.2004.1302409 (cit. on p. 18).

[33] G. Strimel, S. Bartholomew, and E. Kim. “Engaging Children in Engineering Design Through the World of Quadcopters”. In: *Children’s Technology and Engineering Journal* 21 (2017-05), pp. 7–11 (cit. on p. 19).

[34] D. Mellinger. "Trajectory Generation and Control for Quadrotors". In: (2012-01) (cit. on p. 20).

[35] O. Araar and N. Aouf. "Full linear control of a quadrotor UAV, LQ vs H". In: *2014 UKACC International Conference on Control (CONTROL)*. 2014, pp. 133–138. DOI: 10.1109/CONTROL.2014.6915128 (cit. on pp. 23, 26, 27).

[36] N. Sydney, B. Smyth, and D. A. Paley. "Dynamic control of autonomous quadrotor flight in an estimated wind field". In: *52nd IEEE Conference on Decision and Control*. 2013, pp. 3609–3616. DOI: 10.1109/CDC.2013.6760438 (cit. on pp. 24, 25).

[37] N. K. Tran, E. Bulka, and M. Nahon. "Quadrotor control in a wind field". In: *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2015-06, pp. 320–328. DOI: 10.1109/ICUAS.2015.7152306 (cit. on p. 26).