Bruno Miguel Soares Neves

Master of Science

Flight control of hybrid drones towards enabling parcel relay manoeuvres

Dissertation submitted in partial fulfillment
of the requirements for the degree of

Master of Science in
Electrical and Computer Engineering

Adviser: Bruno Guerreiro, Assistant Professor,
NOVA University of Lisbon

Examination Committee

Chairperson:	Doutora Ana Inês da Silva Oliveira
Raporteur:	Doutor Rui Alexandre Nunes Neves da Silva
Member:	Doutor Bruno João Nogueira Guerreiro

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

November, 2020

Flight control of hybrid drones towards enabling parcel relay manoeuvres

This document was created using the (pdf)LaTeX processor, based in the "novathesis" template[1], developed at the Dep. Informática of FCT-NOVA [2].
[1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

Acknowledgements

I would like to thank the FCT NOVA institution for always providing great studies opportunities such as this work. A special thanks goes to Prof. Bruno Guerreiro for providing the needed support whenever necessary and also providing great explanations. Specially in the current time, being able to attend weekly meetings made the process more enjoyable. Also a special thanks to the research partner Hugo, for the long discussions that helped both of us to progress between the many difficulties.

A big thanks too, to all my friends that provided constant motivation, and generated great moments, throughout the course of the past 5 years, specially to my girlfriend Barbara who supported me the most in the final stretch. Finally thanks to my family who have made all this possible, providing constant support and enabling the process to run smoothly without outside interference.

This work was partially funded by the FCT project REPLACE (LISBOA-01-0145-FEDER-032107) which includes Lisboa 2020 and PIDDAC funds, and also project CTS (UID-B/EEA/00066/2020).

Abstract

This work addresses the modeling and controlling process of a hybrid UAV, aimed for parcel relay maneuvers. Hybrid UAVs bring big advantages with the capability of flying in two flight modes, rotary and fixed wing. But with them comes added complexity both in modeling and controlling.

This work is based on a popular airframe, a tilt tri-rotor UAV, containing all the specific system dynamics such vehicle category provides. The model is then validated by designing two separate controllers for both flight modes, capable of trajectory tracking in each mode, making use of a custom hybrid control allocation technique that differentiates the control in three parts: vertical, horizontal, and transitional flight modes.

Finally, a hybrid controller is proposed, using a finite state machine capable of handling logical events, with the aim to provide control logic to perform autonomous mid flight transitions. All the designs system are simulated using a mathematical framework and a power-full simulation tool.

Keywords: Hybrid UAV, UAV modeling, Transitional flight, Control Allocation, Hybrid Control, Finite state machine

Resumo

Este trabalho aborda o processo de modelação e controlo de um veículo aéreo não tripulado híbrido com o objetivo de proporcionar manobras de transição de carga. Drones híbridos trazem grandes vantagem com a sua capacidade de voar em dois modos de voo, de asa rotativa e asa fixa. Por outro lado, estas vantagens adicionam complexidade ao sistema dificultando o processo de modulação e controlo.

Neste trabalho está presente um modelo de um drone tri rotor tendo dois rotores movíveis. Este contém todas as dinâmicas especificas que um sistema deesta categoria de UAV obriga. O modelo é posteriormente validado com dois controladores separados em modo de voo, capazes de proporcionar medidas de seguimento de trajetória em cada modo, usando uma técnica de alocação de controlo personalizada que diferencia o controlo em três partes: vertical, horizontal e de transição.

Por fim, é proposto um controlador híbrido contento uma máquina de estados capaz de tratar de eventos lógicos, de modo a proporcionar transições de modo de voo autónomas em pleno voo. Todos os sistemas propostos são devidamente simulados usando ferramentas matemáticas e também poderosos sistemas de simulação.

Palavras-chave: Drone híbrido, Modelação, Voo de transição, Alocação de controlo, Controlo híbrido, Máquina de estados finita

Contents

List of Figures

Glossary

Actuator settings	Actuator settings are the raw signal inputs given to a set of actuators in order to obtain the desires response.
Euler angles	A three angle system used to describe a body rotation. In this work, as is common in aerial vehicles, we will refer to them as roll, pitch and yaw.
Inertial measurement unit	An inertial measurement unit that is used in all types of UAVs, it combines a set of accelerometers and gyroscopes to provide the measurements of the body specific force, angular rate, and orientation .
Negative transition	Negative transition is a transition from fixed wing to rotary flight mode.
Positive transition	Positive transition is a transition from rotary to fixed wing flight mode.
PX4	PX4 is a open source autopilot software used in wide variety of UAVs.

Acronyms

AoA	Angle of Attack.
BEC	Battery Elimination Circuit.
DHA	Discrete hybrid Automata.
DOF	Degrees Of Freedom.
ESC	Electronic speed Control.
FSM	Finite State Machine.
HITL	Hardware in the loop.
IMU	Inertial Measurement Unit.
LMPC	Linear Model Predictive Control.
LQR	Linear Quadratic Regulator.
MLD	Mixed Logical Dynamical.
MPC	Model Predictive Control.
NMPC	Nonlinear Model Predictive Control.
PID	Proportional Integral Derivative.
ROS	Robot Operating System.
SDC	State Dependent Coefficient.
SITL	Software in the loop.

UAV Unmanned Aerial Vehicle.

VTOL Vertical take-off and landing.

Chapter 1

Introduction

With the recent great developments being made in all the areas surrounding unmanned aerial vehicles (UAV), or drones, such as sensors, computing power, and automotive industry, a UAV is becoming more resourceful and useful for many situations in the most diverse areas. Knowing this, this thesis comes from that fuss and development around UAVs to test and determine their capabilities in one of the most important industries, logistics, more precisely on parcel delivery techniques. Taking on that, this project was made to determine and test the capability and the possibility of creating a parcel delivering robust system, using one or multiple UAVs. UAVs can solve many problems and bring a lot of advantages to ease industries such as efficiency and automation. In short, the project seeks to understand and test all the possible advantages and disadvantages of bringing a hybrid UAV to the parcel delivering game. This creates new scientific challenges on trajectory planning, aggressive maneuvers, localization estimative, control techniques, hybrid control and so others. These indicated, that powerful systems need to be created in order to take the full advantages of using this technology in a new context.

1.1 Motivation

In the scope of this project, the use of the most commonly known drone, a multi-rotor is already being tested and is subject of various analysis. This work builds on the existing work and tries to expand this knowledge of multi-rotor drones to the more recent Hybrid drones also know as Vertical Take-Off and Landing (VTOL) vehicles. This type of UAV is being more used because of its advantage in endurance, efficiency, speed, and manoeuvrability, from that we can complement the use of hybrid drones to built a more robust system using multiple drones as a part of a complex package delivering network. For their use we need to devise strategies for modelling, design and control of hybrid drones

taking advantage of their capabilities.

A hybrid drone main difference is that they contain two flight modes, horizontal and vertical. While the vertical flight mode extends its maneuverability, the horizontal flight mode is used for more speed and propulsion efficiency being able to cover larger distances faster and using less energy when compared to a conventional multi-rotor drone. Some package delivering systems have already been made and tested, and still, one of the main constant problems of using multi-rotor drones is their low autonomy. Depending on their size, battery and also their cargo weigh, the range is normally very low witch is a major limitation for a package delivering system. The idea of using a hybrid drone comes from this problem, as explained before hybrid drones have a major efficiency gain when operating in a vertical flight mode, with this it becomes possible to increase the range and also decrease the time of delivery. This thesis will focus on creating a operational controller for a hybrid drone for further testing.

Nowadays there are a lot of hybrid UAVs in the market, they all have some differences between them including the immense difference in price range. For the purpose of this work, we wanted a drone that would enable us to modify and access its hardware, flight control, IMU (Inertial Measurement Unit) etc, while having most of the advantages of conventional hybrid drones such as being able to perform both flight modes with relative ease. The chosen drone was then the E-FLITE CONVERGENCE,as it meets all of these requirements and its easily accessible.

Figure 1.1: E-FLITE CONVERGENCE [1]

1.2 Objectives and Problem Statement

The main objective of this thesis is the implementation of a non-linear hybrid control system on a Hybrid UAV. A model of the hybrid UAV will be made at first, in order to test and validate all the proposed control designs and, in a final stage its advantages and disadvantages over conventional controllers will be discussed.

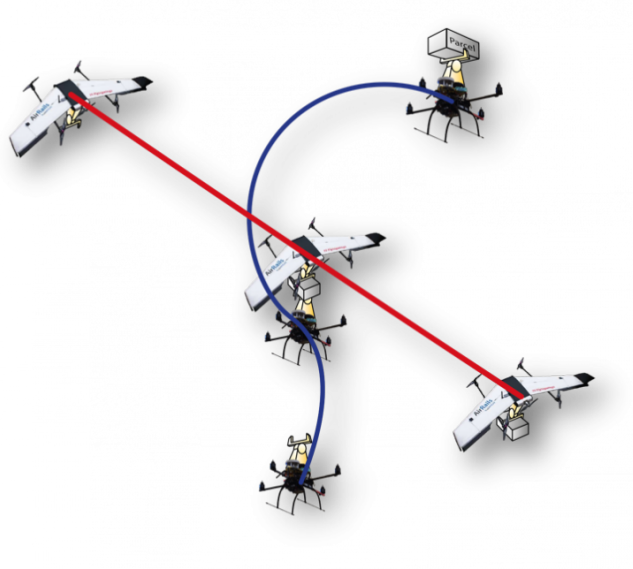


Figure 1.2: REPLACE project Hybrid drone usage example [2]

Figure 1.2 shows one possible hybrid drone usage example in parcel relay maneuvers in the scope of project REPLACE. Here a hybrid drone provides package transition from another UAV.

Controlling a hybrid drone can be developed in various different ways, some of them more complex than others, but all of them serve a purpose. As a great amount of controlling techniques have already been developed in the literature for this type of UAV, the objective of this thesis is to build on this knowledge to make a review of all the possibilities and different techniques available for this purpose, and then develop a state of the art controller capable of:

- Control the chosen vehicle to the level of actuator settings in both flight modes independently;
- Command any state transition when it fits to be the most efficient flight mode, without any outside input beyond the desired flight path;
- Perform smooth transition between flight modes without trajectory oscillations or high altitude drops;
- Use all the hybrid drone capabilities in a optimized technique.

The final controller will join various control components creating a fully operational autopilot capable of following a predefined trajectory with precision, efficiency, speed, and most importantly taking full advantage of the hybrid drone capability.

1.3 Proposed Solution

To obtain the desired outcome one must be able to simulate the system before proceeding to real world tests, in order to regard for the safety of the equipment, and at the same time facilitate the controller design process. Knowing this, this thesis will be separated into two stages, modeling and control.

In a first stage of this work, a closer look will be taken at the drone modelling, where, using a framework proposed for this effect, the hybrid UAV will be modelled as a hybrid nonlinear system. This will ensure a robust mathematical model minimizing the errors later on when testing the control system, and enabling the model to be controlled in both flight modes with the real actuator settings as inputs. This task will be performed in chapter 2 of this thesis.

Stage 2 will contain all the process needed to develop a hybrid controller designed for this problem, evolving hybrid systems and nonlinear advanced control techniques. We will be building on existent work designed for multi-copters and transforming the control system maintaining the same bases as a reference, this will result in separate controllers used for different flight modes independently. Later, a hybrid control system will be introduced to unify both control systems, taking advantage of the versatility of a hybrid UAV in a single autopilot. Chapters 4 and 5 will focus on these tasks.

In the end, attitude, altitude and trajectory tracking systems should culminate in a robust autopilot. This controller is simulated and tested in both mathematical and SITL(software in the loop) simulation environments. The final goal is presenting hard evidences and data indicating all the capabilities of the designed controller.

1.3.1 Thesis outline

The structure of this thesis goes as following:

- Chapter 2 contains a brief review on the most popular UAV types, hybrids and non-hybrids, naming their individual general characteristics and uses. Also the current most used methodologies on UAV controlling/modeling systems are discussed on this chapter;

- In Chapter 3 the mathematical model for the hybrid UAV is presented. It is separated into both flight modes, giving thrust forces and aerodynamics forces equations separately, that in the end will be join together to recreate the full fledged hybrid UAV mathematical model that will be used throughout this work;

- Chapter 4 describes the proposed trajectory tracking control system for both flight modes. In addition, the controller is validated through mathematical simulations for both flight modes presented in this chapter;
- Chapter 5 contains the control actuation scheme developed in this work for the hybrid UAV, so that the actuator settings can be accessed;
- Chapter 6 is related to the main objective of this thesis, the hybrid control system. Here a hybrid controller is proposed in order to handle flight modes transitions. Mathematical simulation on mid-flight transition are also presented and discussed;
- In Chapter 7 the SITL(Software In The Loop) simulations are performed using gazebo as a powerful simulation tool, and using actual PX4 communications techniques that can easily be ported to real flight tests;
- A start to the practical work is presented in Chapter 8, including the instrumentation process of the used UAV. This step enables it to be controlled using the PX4 framework with MAVlink communication;
- Finally, Chapter 9 adds a conclusion note on the achieved goals of this thesis, and ends with some knowledge to aid possible future works.

Chapter 2

State of the Art

For this thesis some diverse areas will be explored and studied, some of them are already very well established due to the immense development that was put into them for an extensive period, but others are in a primordial stage where not many developments or studies are being made, mainly because of the semi-recent technologies that are being brought to the current date and will continue to greatly evolve.

Being this, a recent and trendy technology UAVs and especially Hybrid UAVs, are in constant development leading to plenty of recent published studies. In them, various types of hybrid UAVs are used in conjunction with diverse control techniques. Depending on the specific problem, control methods and the type of UAV may be different. In this sections we will take a look at various studies concerning areas of interest for this work, the most relevant are:

1. **Hybrid UAVs**
2. **Control techniques**
3. **Hybrid Control**
4. **Related work**

From this, all the methodologies to be used for this thesis will be chosen concerning our specific problem with the objective of creating a robust and reliable system with the knowledge provided by the current literature.

2.1 Hybrid UAVs

In the past times there are two, more conventional, types of UAVs that are already very popular, the fixed-wing and the multi-copter (VTOL) UAVs with interest growing in the

second vehicle category.

(a) Multi-copter UAV[3]

(b) Fixed-wing UAV[4]

Figure 2.1: Conventional UAV categories

Multi-copters provide usually four to eight rotors all pointed upward providing a upward lift force. The most common type, the quad-rotor 2.1a, has two rotors pairs rotating in opposite direction(X or + configuration), torque in the three axis of rotation is generated by having differential rotor speeds, they provide good maneuverability being able to hover in a stable position easily and are easily controllable. This makes it highly versatile and used for the most various reasons. The downside of only relying on the force produced by the rotors to maintain altitude creates a low power efficiency when traveling long distances. Fixed-wing UAVs 2.1b, behave like conventional aircraft having usually one to two rotors pointing forwards, providing a thrust force in the direction of movement. They use aerodynamics to maintain altitude, possible by the lift force created by the wings, they also usually have a set of ailerons, elevators, and rudder providing torque forces in all three axis of rotation. Taking advantage of aerodynamics they require low power to maintain altitude and travel long distances efficiently, and with higher speeds when compared to multi-copters. The rotors are unable to provide upward forces so they can only maintain altitude with high speeds and become more unstable at low speeds, also they can´t or hover flight, this fact is a major downside for many applications where maneuverability is key.

Nowadays, UAVs named Hybrid drones are gaining popularity and with that more studies on their advantages and disadvantages are being made. The goal of a Hybrid UAV is to have both the advantages of fixed-wing UAVs such as high cruising speed and increased range, and the advantages of multi-copters UAVs like flexibility, manoeuvrability, and the VTOL(Vertical Take-off and Landing) ability. To fly in both flight modes, hybrid UAVs use various transition systems that enables them to produce thrust force in more than one direction either by rotating the rotors, wing, or completely change their air frame rotation. Some of the most popular are present in Figure 2.2.

The tail-sitter 2.2a achieve such transition by tilting the entire air frame 90°, it lands and takes-off on its tail in vertical flight mode and tilts the entire vehicle mid flight to a horizontal position, poiting the two rotors to the direction of movement for a fixed-wing like flight mode. This fact enables it to have a reduced number of rotors lowering the total weight. As it rotates the entire air frame, the possibility to carry payloads

(a) Tail-sitter UAV[5]

(b) Dual-system UAV[6]

(c) Tilt-rotor UAV[1]

Figure 2.2: Conventional hybrid UAV categories

become much more challenging or even infeasible. Dual-system 2.2b UAVs solve the problem by having rotors in both directions of flight constantly, generally four rotors are mounted upward, providing vertical flight mode and a additional one or two rotors generate thrust force parallel to the ground enabling horizontal flight mode. It takes off and lads using the upward facing rotors and once in the air turns on the additional rotors generating a forward thrust. Tilt-rotors UAVs 2.2c have a tilt mechanism that is able to rotate all or some of its rotors to the direction of flight on any occasion. They start the flight with all rotors facing upwards in a vertical flight mode, and when desired tilt the rotors forward enabling horizontal flight mode. In horizontal flight mode they are controlled like a convectional fixed-wing making use of aerodynamic forces. They are one of the most popular hybrid UAV types because they offer very good manoeuvrability in vertical flight mode, but achieve an easy transition to horizontal flight mode whenever necessary, making it very versatile. With this, the control system becomes more complex and requires more computational power in order to account for the added dynamics.

For this work, a tri-rotor having two tilt rotors UAV will be the vehicle of choice, so the reminder of this thesis will focus on this type of drone, including the non-linear controller and control allocation techniques. It was the best option because of the reasons discussed before among other advantages [7], and for the fact that this type of UAV is easily accessible and, can be controlled by low-cost auto-pilot systems (e.g., PixHawk4).

2.2 Control Techniques

There are many ways and different laws to implement a controller, in this case, depending on the UAV model and his purpose different control technique will perform better in some tasks than others. For hybrid UAVs, as they have a more versatility flight envelop, three modes can be considered, horizontal flight mode, vertical flight mode, and a transition mode, each one can be controller using a different control technique, depending on the desired output. Below are some of the most popular control laws implemented in hybrid UAVs and UAVs in general.

2.2.1 Linear control laws

Linear control laws are the most easy to implement and have the least computational power need, but of course they offer some drawbacks. As previously mentioned Hybrid UAVs are non-linear systems so in order to use linear laws a linearization is needed, this is commonly done by applying relative equilibrium conditions around a steady state operating point.

1. **PID:** Proportional-Integral-Derivative is probably the most common type of controller for any type of UAV, however for its use in this case a linearization of the system is recommended to decouple the states. It is commonly made by using three separate linear controller for each flight dynamic, Horizontal, Vertical and transition. This makes the controller much simpler and easy to implement however, in certain conditions it has a poor robust ability compared with other controllers. This controller method is used in [8, 9];

2. **LQR:** Linear Quadratic Regulator is also one of the most common Linear laws used in hybrids UAVs. The main advantage of using LQR is its power to handle complex dynamic systems and multiple actuators systems like a Hybrid UAV, with this is possible to achieve a more robust control system that has larger stability margins to errors in the loop, very useful in hybrid UAV controllers. As a downside, for it to be applied it requires a access to the full state of the vehicle witch may not be accessible in certain conditions.

2.2.2 Nonlinear control laws

Although more complex, non-linear control laws are more robust and versatile, offering the possibility of controlling aggressive manoeuvres, far from the equilibrium point, without the stability problem that linear controller have, and without the need of linearization. They can only be used assuming that a proof of stability can be made, which in some cases, with non-linear systems, can be hard task.

1. **Back-stepping**: Is a non-linear control law commonly used for this type of non-linear systems. This method is often coupled with the use of Euler-Lagrange approach for the dynamic modelling [7]. Using this non-linear control law will bring robustness to the system that generally overcomes linear control laws, at the cost of computational power;

2. **Adaptive back-stepping**: Adaptive back-stepping is the common methodology to be used, it can rest on a estimator with integration control to archive a disturbance rejection effect. This change may cause a problem, as the estimate can grow unbounded depending on the initial conditions of the system. An easy solution would go by using a projection operator to constrain the parameter creating another whole problem because of its discontinuity and out continues time system. To overcome this problem in [10] a smooth projection operator is proposed, to generate estimates with enough smoothness to continue the back stepping procedure. Some UAV controllers using similar back-stepping techniques cases can be found in [10, 11].

2.2.3 Model Predictive Control

All of the control methods mentioned above have a common practice, as almost any type of drone is input constrained, the controllers work by feeding them inputs in a post-facto state [12]. MPC (Model Predictive Control) in the other hand provides a framework that is capable of directly include constraints in the optimization problem, being also capable of predicting future events and provide the according actions.

MPC methods use the dynamic model of the UAV to predict the systems behaviour in a finite horizon (prediction horizon) allowing the controller to know the future outputs of the system, the algorithm then takes them into account when formulating an optimal trajectory tracking controller bringing the system to the desired state. It does this at each time step, computing the optimal control inputs given the prediction calculated in the current step.

Two types of MPCs are usually used, a linear LMPC and a nonlinear NMPC. For LMPC, the nonlinear UAV dynamic equation need to be linearized around a operation point, in a simple quad rotor or any type of VTOL aircraft this point is normally its hover state. Knowing the dynamic model, a set of constraints is also added to the MPC framework, for the input it is necessary to constrain the maximum force each rotor is able to produce, this is known as a hard constraint representing the limits of the physical system, as for the output, the constraints tend to be imposed by limiting the angles of the drone (roll,pitch and, yaw). Finally an objective equation is given to the MPC in order for it to minimize, generating the optimal solution in each time step. In trajectory tracking controllers this function has the reference trajectory as an input and its solution should drive the UAV to the desired state and also minimize the controller effort required.

For NMPC there is no need for the linearization dynamic model of the UAV, but instead a state-dependent coefficient (SDC) factorization is used with the nonlinear dynamic model to obtain pseudo-linear system matrices. The main algorithm basically stays the same as the one used in the LMPC, the main difference is upon the objective function, where now it depends on the current status of the system, that needs to be calculated at each sample step.

Some various tests have been done to validate this controlling methods in small UAVs, and all of them arrive to the same conclusion. In [12, 13] both MPC algorithms were tested back to back and both authors conclude that nonlinear MPC clearly outperform the linear controller in aspects like, disturbance rejection, step response, tracking performance, better transient performance.

2.3 Hybrid Control

As introduced before creating, a fully operation controller for a hybrid drone has its added complexities, to simplify the process the techniques to be used will most likely generate a separate controller for each flight mode, horizontal, vertical and transitive. This will create a problem, the transitions of each controller must be smooth operations in order to create a robust system capable of achieving some semi-aggressive manoeuvres smoothly, for that a hybrid control system is the perfect method to be adopted. Hybrid control is a recently new topic, being studied more severely in the recent years. Some cases already, show us the power of modelling a Hybrid system.

This method can produce a very robust model and controller for a Hybrid System, that's why its of some importance to this work, a Hybrid UAV can be modelled as a hybrid system easily. Most authors tend to create a dynamic Hybrid Model of the Hybrid UAV, implementing and separating different flight modes and dynamics in the same model, even when there is no intention to use this controlling method.

A hybrid system is simply a system that interacts in different types of dynamics,[14], in our case this means a system that interacts between continuous and discrete states creating a hybrid dynamic system, the definition of this states are:

1. **Continuous**, any system that takes values in Euclidean space $\mathbb{R}^n$ where $n \geq 1$. This system can change their value in continuous time according to a differential equation. In our case this will represent the dynamic model of our Hybrid UAV given by a differential system of equations for both flight modes;

2. **Discrete**, any system that takes values according to a predetermined finite set and no more than those. This systems can only change their value through "jumps"between states when some event triggers them to jump. For our case, the discrete system will be the flight modes that the Hybrid UAV is capable of, the finite set of values would be $q \in [Vertical\ flight, Horizontal\ flight, Trasistion\ to\ vertical, Transistion\ to\ horizontal]$. Note that these sets of values may vary further through this work;

3. **Hybrid**, this would be our final system, that takes both values in continuous states as well as discrete states. Involving both of these dynamics, the system becomes hybrid and is subjected to this control method.

In short, these systems are used when it is convenient to model any discrete components that would implicate a instantaneous change in the continuous components. Our dynamical hybrid system will consist on a set of continuous components, nonlinear differential equations representing the drone movement, those equations will be switched between them when a jump happens in the predetermined discrete states. The goal of the method is to find an optimal solution that is capable of handling this instantaneous changes in the discrete states without hurting the continues components of the system and minimize the effort of this changes.

2.3.1 HYSDEL

Modelling a hybrid system has its added development steps, that's why it isn't common to see this type of modelling and control used in hybrid drone control, it comes with a series of challenges and it is harder to predict the advantage over a more common control technique.

Knowing this, a new form of modelling language was proposed in 2004, [15], Hybrid System Description Languages(HYSDEL), it allows engineers to model classes of hybrid systems represented by linear dynamic systems and logic conditions, once the hybrid system is modulated as a discrete hybrid automata(DHA) this high-level modelling language can produce mixed logical dynamical(MLD) or piecewise-affine used to solve any control and optimization problem.

For a better understanding of HYSDEL and its usages let us describe a MLD system. These systems are used to describe the behaviour of linear systems with logical rules, in other words a hybrid system composed of continuous and discrete states(logical rules). The time evolution of that hybrid system as described by MLD should follow the following form [16]:

$$\begin{aligned} x(k+1) &= Ax(k) + B_1 u(k) + B_2 \delta(k) + B_3 z(k) \\ y(k) &= Cx(k) + D_1 u(k) + D_2 w(k) D B_3 z(k) \\ E_2 \delta(k) + E_3 z(k) &\leq E_1 u(k) + E_4 x(k) + E_5 \end{aligned}$$

where $x(k+1)$ is the state-update equation, $u(k)$ the input equation, $y(k)$ the output equation, and the linear inequality represents the switching conditions of the system modes. HYSDEL is then capable of producing a system representation of this form given the DHA modelled in HYSDEL language, like shown in Figure 2.3.

Although HYSDEL gives us a powerful modelling language that would help to solve our hybrid system, it has a major drawback for our problem. HYSDEL does not support variables in the form of vectors or matrices and misses some important instructions, manly loops. This would lead to major complications when using this language.

For those reasons we will also focus on another method, MINLP solvers.

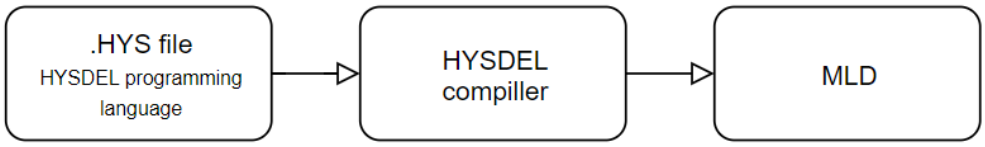


Figure 2.3: MLD generation process using HYSDEL compiler

2.3.2 MINLP

Mixed integer nonlinear programming its a optimization method that derives from the Mixed Integer Linear Programming (MILP), where that solution can only be integers, being the difference, now continuous and discrete variables can exist, apart from integers, in the objective function and constraints, creating a very versatile modelling framework. MINLP, by using integers variables enables the model to incorporate discrete decisions, complementing this with the possibility to use nonlinear functions makes a perfect framework to create am accurate model of our problem.

A MINLP problem is described as, [17]:

$$\begin{cases} \quad min\ f(x,y) \\ x \in X \subseteq \mathbb{R}^p \\ y \in Y \subseteq \mathbb{Z}^n \end{cases}$$

This type of problem is usually hard to solve, that's why it has been subject to various works and progress in the last years, culminating in what we have available in this day, powerful solvers. MINLP solvers enables a problem to be optimized quite fast when compared to other methods, they can have a crucial value to this work. Some of the popular MINLP solvers are, CPLEX(by IBM), XPRESS, BARON and in open-source, GLPK and Cbc. Because of their complex form and large theoretical background, for further more knowledge refer to [17] [18].

2.4 Related Work

A great amount of articles and papers address similar problems to the baseline of this thesis. Trajectory tracking controller come in a array of shapes and sizes and there is a lot of diverse methods adopted from each author. As we seen before, there is no such thing as a perfect method for a trajectory tracking controller, different methods have different advantages, that's the reason engineers choose a method based on the specific problem and there is no global formula.

As one of this thesis main goals is developing a non-linear trajectory tracking controller, we will focus our attention in this category of controllers, more specific, the back-stepping method. As we know, this method involves more computational power and technically difficulty, but rewards us being a very robust controller. Some authors already used this method for quad rotors and some hybrid drones, although it seems difficulty to find in the literature a work that uses this method for controlling a tri-rotor hybrid UAV, or that focus on the transition flight mode. In this section, we will take a look at various control methods present in the literature in order to familiarize ourselves to the subject in question.

2.4.1 Linear Controllers

Linear controllers usually require lower computational burden, and for that reason they are very popular on simpler applications, as the technology evolve, nowadays it is possible to have a very robust trajectory tracking controller using this method. Knowing this some authors used linear methods like PID or LQR when developing control methods for quad rotors due to its simpler system dynamics. The controller formulated in [19] where the test resulted in a position error of about 10 *cm* indoor and 50 *cm* outdoor, with the throwback of only being effective at relative low speed. In this solution, the more simplest method of Euler angles representing roll, pitch, and yaw was used along with some other rotor techniques to control the quad-rotor in each direction/rotation. The use of a linear PID was made possible by the feedback linearization, forming an attitude controller which later on would be used alongside a tracking controller to orient the rotor thrust, providing the desired lateral acceleration. Note that a great advancement was made here by adding an extra zero in the root locus by using the acceleration feedback, allowing gains to rise, yielding higher bandwidth. This continued to be a great controller method and was used later on a updated version in [20] where, for obtaining even greater precision, the quad-rotor momentum, blade theory, and advanced aerodynamic were taken into account for creating a improved model. With the goal of increasing speed and outdoor precision with a sub-meter positioning quad-rotor, enabling it to fly around multiple vehicles in a small area. Effects such as, angle of attack, lift force, speed on total thrust, blade flapping effect and moments acting on the UAV were accounted when creating the model, all of these verified with a static thrust test. The model was then used to derive non-linear control laws, that were later used to create a closed loop controller structure composed of simple PID controllers similar to the one formulated in[19]. With this powerful aerodynamic model and given precise state estimation this simple control law produces great accuracy with increased speed.

Continuing in the field of linear controllers, they are also appreciated in the Hybrid UAV department. Although in this case, the increase complexity of the vehicle, with different flight modes, more advanced aerodynamics and so on, tends to make authors prefer the use of nonlinear control methods, taking advantage of the full capabilities of

a Hybrid UAV. Despite that, some still use linear control methods in cases where the accuracy isn't as demanding.

Beginning with a simplified version of a hybrid UAV in [8] the author takes advantage of the two servo motors connected to the two front rotors of a tilt-tri-rotor UAV to create a simple controller capable of moving the drone in vertical flight mode only. Here, the rolling motion is controlled by the differential rotor thrust, the pitch motion is controlled by the tail rotor thrust and tilt angle, but the big difference happens in controlling the yaw motion, where the tilt of each rotor changes the direction of thrust enabling it to produce torque around the yaw axis. For the use of a linear controller to be possible, the system is linearized around its hovering point, obtaining then a control equation for each degree of freedom. With a series of PID loops that were fine-tuned, a controller for attitude and altitude was formulated. Using this method the Hybrid UAV is not used in all its capabilities being able only to fly in vertical mode. creating a more robust controller involves a lot more system complexity.

Note that the main goal in this work is to take advantage of all the capabilities of the hybrid drone. The first methods that were explored too achieve this were based on formulating a controller for each flight state, horizontal, vertical and in some cases a transition state. A good example of that technique being used is shown in [9] where a controller is designed to take full advantage of a hybrid drone different flight modes, combining a rotary wing and a fixed wing aircraft in the same package, as hybrid drones were intended for. Here a tilt-tri-rotor having the front two rotors connected to the same tilt shaft operated by a servo motor, was modelled using Newton-Euler equations with six degrees of freedom, containing the transnational and rotational motion, thus creating a singularity in the attitude equation. This problem was solved by limiting the roll of the UAV to 45º. For greater results, the external forces and torques represented by a vector, and each flight mode as a different source. In vertical flight, the forces and torques are produced only by the thrust of the three rotors, in horizontal and transition state the increased speed of the UAV develops greater air resistance, combining it with the high angle of attack, the aerodynamically forces become relevant so they are added to both equations. For controlling purposes, and one of the main factors to review this work, is the different phases presented by the author here they are, vertical take-off, hover, transition to horizontal and to vertical flight mode, vertical flight and vertical landing. To formulate the controller a nonlinear PID method was used, where the parameters are tuned every loop. This is done by a neural network that outputs the optimal control parameters at every loop, therefor, resulting in a system that is capable of utilizing both flight modes of the Hybrid UAV and doesn't need a strong computational power to support it. Note that in this work it was assumed that there was no wind or any external force countable, making this just suitable for certain application.

2.4.2 Non-Linear Controllers

Moving on to non-linear control techniques, they are a step-up over linear techniques providing more robust control systems and can be very modular when it comes to different real world scenarios, all this at the cost of complexity and computational power. With these techniques the objective function, almost always non-linear, can stay that way without the need of any type of linearization, meaning that the constraints that the linearization process adds to the system is no longer present. Using nonlinear techniques its possible to operate the aircraft far from the equilibrium point, imposed by the linearization, and can also perform more agile maneuvers, crucial for some specific problem.

In terms of modelling few differences were made to adapt to non-linear control techniques, the modelling process basically stays the same as previously, where some authors use Euler representations and some moved on to Quaternion representation, usually when more aggressive manoeuvres are needed, avoiding singularities.

One of the most common techniques used in nonlinear control of any type of UAV is back-stepping, this method is normally not applied to under actuated system, like some UAVs, but a more simplified model can be obtained by linearization of the input and output. This was first shown in [21] and is commonly used since then on VTOL vehicle control. Enabling the model to be stabilized by the back-stepping method. This method was also used in [10] in order to obtain a robust control system for a quad-rotor. Here a quad-rotor is controller by a back-stepping control law in angular velocity and torque, both well known parameters as nowadays drone manufactures use small foot-print sensors to measure angular velocity, and electric motors are controlled independently by speed witch leaves access to torques measurement being equivalent to the added exerted force in each motor. The UAV was modelled as a rigid body capable of generating force on the z axis(perpendicular to ground plane), also a common technique. The proposed controller was based on [22] work, in short, a translational subsystem back-stepped by a angular subsystem resulting in the final controllers operate by outputting desired angular velocity and torque.

Using such control method comes with the advantage of the possibility, of adding disturbance rejection with some easiness, this was also added to the work [10], being able to stabilize the closed-loop system in the presence of force disturbances caused by external forces such as wind or mass imbalance within the drone. This force rejection technique was obtained by a adaptive back-stepping method that uses a estimator to obtain this effect. Using this type of method comes with a problem, the estimate parameter can grow unbounded without any priori bound, that has to be known. A projection operator is normally used to correct this problem but it adds discontinuity to the system, a major draw back. To suppress this problem, authors in previous works came with smooth projection operator witch can produce sufficient smooth parameters to be used in continuous systems. A controller with all this capabilities is capable of following pre defined smooth trajectories as shown in [10], this work contains a lot of important knowledge and

techniques that may be used in the development of this thesis.

Normally, the main reason why authors choose to use nonlinear control methods as a way to formulate trajectory tracking controllers in a UAV is the fact that they are aiming to develop a more high performance and accurate control system being usually the only objective of their work, this act, that already adds to the complexity of the work, leads authors adding methods to improve the performance of their controller, by increasing speed, increasing precision, developing disturbance rejections systems etc. A recent breakthrough motivated by the seek of high performance controllers, uses so called Hybrid systems, that are no more that two or more control methods/laws applied to the same system, hence taking the advantages of more than one technique.

Work developed in [23] uses this to formulate a hybrid system that uses MPC method and fuzzy logic control. It is able to propose a control system capable of trajectory tracking and disturbance rejection, wind effects and cross-coupling disturbances. MPC is able to predict the system behaviour in a finite interval, this property helps the system dealing with multiple value constrains, adding to this fuzzy logic control brings the capability for the system being able to coupe with imprecise values, common in real world scenarios. Here, three MPCs loops deal with XYZ movement while a fuzzy interference system, FIS, acts as a feed forward compensator to the attitude control loop. The FIS works by comparing the sensed acceleration in the z axis with the acceleration predefined set-point and then contracting the impressions caused by wind. Also, the FIS takes the measurement of pitch and roll to eliminate the effect of cross-coupling, this effect happens when pitch and roll combined create a secondary effect on altitude. With this method the autopilot showed a noticeable performance difference when compared with more traditional linear laws.

2.4.3 Non-quad-rotors

All of the previous works were modelled after a quad-rotor UAV, the most common drone as we mentioned before, but not the objective of this thesis. On that note, it is also important to take a closer look at other works made with different types of drones to see the changes that will need to be implemented in this work. One of the main differences between a quad-rotor and different types of drones with less rotors is the way they are able to perform roll, pitch and yaw moments. In a quad-rotor this is made fairly easy by combining pairs of rotors thrust to obtain roll and pitch, as well as using the difference between them to obtain yaw. When less rotors are available, a mechanism normally controlled by a small servo motor is able to tilt two or three rotors independently, or in a joint motion(connected by a shaft), this mechanism is the reason tri-rotor UAVs are able to perform rotation in the three axis.

In [11], the author presents a controller for a two-tilt rotor aircraft using a back-stepping method. A pair of servo motors can rotate the rotors in two axes, laterally and longitudinally, enabling the drone to perform all moments, roll by creating a speed

difference between the motors, yaw by rotating the propellers longitudinally and pitch by rotating them laterally, note that this is made possible by shifting the centre of mass below the tilt axes providing a greater pitching movement. The dynamics of this aircraft hence become more complex, and for simplification it was modelled as a rigid body using Newton-Euler equations resulting in complex nonlinear model that was made simpler by constraining the tilting action by 15º. Using then a back-stepping procedure to formulate a trajectory tracking controller, was concluded that its indeed possible to control a UAV having only two rotors.

From this relatively simple UAV model we can go to a different approach with increased difficulty. In [24] a tilt-rotor having three rotors is subject to a trajectory tracking controller where the two flight modes, vertical and horizontal, are being controlled by a nonlinear dynamic inversion. Due to its complexity, a non-linear model was used, so that a great amount of aerodynamic factors could be taken into account. The transition mode of such drones are severe challenges to control due to its uncertainty, here most authors chose to use gain-scheduling witch is simpler but ignores the system non-linearities. That's why in this work, the authors used a more complex approach, NDI [25], witch takes those non-linearities into linear dynamics.

The model used in this work is of extreme aerodynamic precision, taking into account most aero resistive factors into the model equation. The UAV is modelled as a rigid body using kinematics theory, it also separated the forces that exerted force on the drone body into two parts, forces caused by the UAV body and forces caused by the two tilt-rotors. So that the UAV used all the control inputs possible for such drone, conjugating aircraft inputs like aileron, rudder and elevator to more common tilt-rotor inputs, rotor speed, tilt angle and the respective differentials. For control purposes of the NDI method, the system was divided into fast and slow loops, then through the equations(obtaining the tilt angle), a feedback function was found in order to get control laws for each loop. Another great technique used was a dynamic control allocation where a weight factor is tuned depending on the state of the drone, giving more or less importance to the aerodynamics forces and thrust forces. All of these resulted in a robust autopilot capable of tracking a trajectory and handling the transnational state smoothly.

The problem of the transition between flight modes must be subjected to great importance in this work, being one of the most demanding and complex problems to be solved and a important part of this work. Some previous works had already found ways to make this transition efficient and fast most of the times using one or more nonlinear control methods, the work presented in [26] shows us a different control strategy in order to perform such transition and also attitude control of a tilt quad-rotor UAV. For such operations the drone flight envelope was divided in three conditions, obtained by the displacement state of the rotors/tilting mechanism, hover flight where the motion relies on the thrust created by the rotors and aerodynamics are ignored, fast forward flight here the UAV is controlled like a typical aircraft, and a transition mode to link the two flight conditions. This three flight modes are differentiated by the position of the four rotors,

that are all connected to a tilt mechanism that can be between 0º and 90º.

The innovation was in the control strategy, two separate controllers were formulated for two different actions using two nonlinear control laws. The first controller is responsible only for the transition mode, being in charge of the attitude dynamics and the tilting mechanism, and the controller was made using a back-stepping procedure aimed to act on thrust, elevator angle and torque variables. The second controller goal is to stabilize both altitude and velocity for defined reference values in order to make the transition from hover to fast forward mode maintain constant altitude, this was possible assuming the pitch angle of the aircraft to be stable by means of the first controller, taking the rotor tilting angle as a virtual controller and using a nested saturation control approach. The complete controller was tested using simulations procedures and showed the stability of the system while performing the transition maneuver. Such work is very limited being able to perform the transition from hover to horizontal modes but still, is relevant to this thesis showing a robust approach to one of the problems that will be worked on later.

Chapter 3

Modelling a Hybrid UAV

As stated before, one of the main objectives of this work is designing a mathematical model of a Hybrid UAV. In this section, the method used will be presented such as all of the preliminary assumptions for building such model.

For a better distinction between the flight modes in such vehicle, the model will be separated in two parts, the tilt-rotor dynamic model, where the chosen framework will be presented and also all the forces and torques produced by the UAV. This is enough for simulating the drone in vertical flight mode. After that, the Fixed-wing dynamics are introduced, here we take a closer look of the main aerodynamic forces, such vehicle produces in higher speeds, enabling it to fly as a conventional aircraft. Before that, a closer look at the characteristics and actuators to familiarize us with all the physical abilities of the vehicle in question.

Before starting the modeling process, note that some assumptions were made in order to make the process less complex, with minimal influence in most flight conditions.

- The UAV will be considered a rigid body with a fixed center of mass;
- Aerodynamic forces will only be generated by the rotor and wing;
- The fuselage drag forces are neglected;
- Wind will be neglected, so the groudspeed is equal to the airspeed on the body.

3.1 Vehicle characteristics

This work is aimed to be as generic as possible in order to enable its use in any type of hybrid tri-rotor configuration available, but in order to create a case example, we will focus our attention on the UAV that will be used for test flight, as indicated before, the

E-flite convergence, an easily available package. This drone configuration is the most common in hybrid tri-rotors being one of the most versatile also providing a stable flight.

To know its capabilities we will take a look at all of the available actuators and control surfaces providing a simplified explanation of their main uses.

- **Tri-rotors**: Provides all the necessary thrust in all flight modes. Note that in horizontal flight the back rotor will be off;

- **Tilt mechanism**: Both front rotors have an independent servo motor that enables them to rotate in the y body axis. In horizontal flight mode, their position is fixed to 90 degrees (parallel to level ground);

- **Elevons**: Will only be used in transition and horizontal mode. They combine the more common ailerons and elevators (present in aircraft) features through a mixing algorithm that generates roll plus pitch moment on the drone body.

Throughout this work, some geometric constants will be brought upon, that are relative to the geometry of the drone, mainly the position coordinates of the three rotors and tilt angles and deflection angles. They are all present in Figure 3.1 and Figure 3.2 where all the lengths are relative to the center of mass, represented by COM.

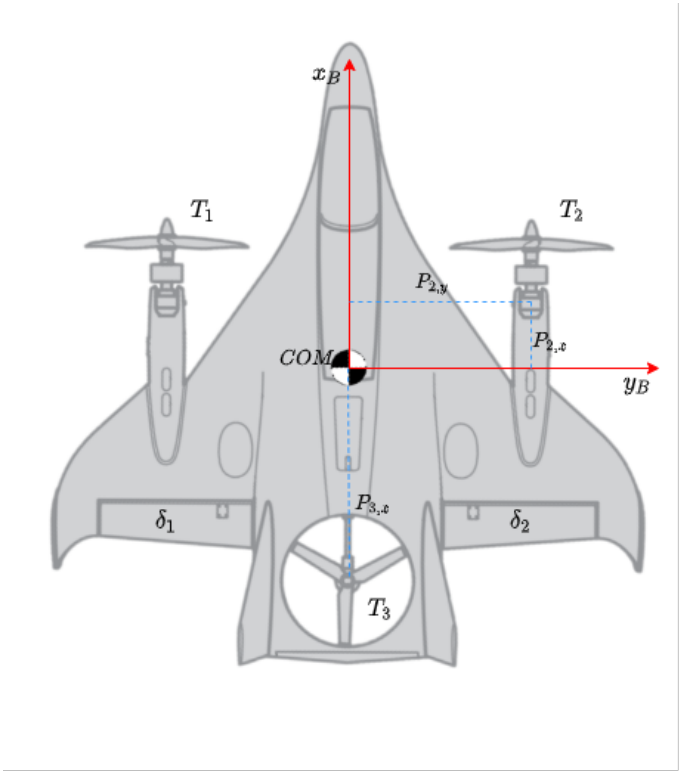


Figure 3.1: Top view representing all actuators and Body axis

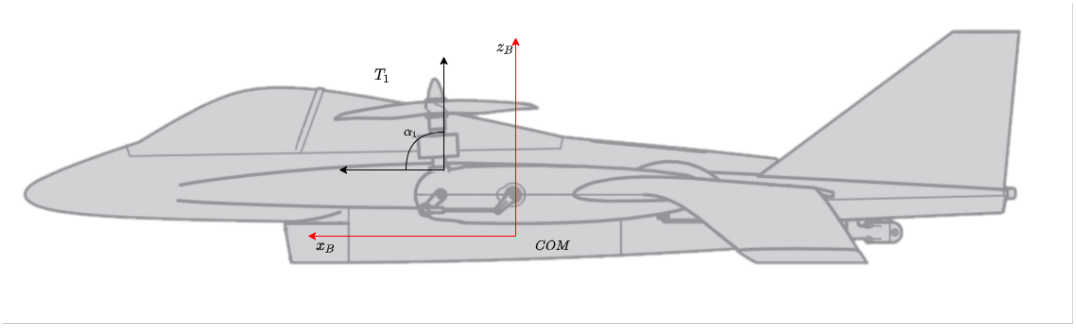


Figure 3.2: Side view representing tilt mechanism

3.2 Tilt-rotor Dynamic Model

For better representation, two coordinate systems will be used: $W = \{W_x, W_y, W_z\}$ representing the stationary world frame in order with the ENU(East-North-UP) convention, with W_z being perpendicular to the ground plane and pointing upward; $B = \{B_x, B_y, B_z\}$ represents the body frame attached to the tilt-rotor centre of mass with B_x being the forward direction and B_z perpendicular to the drone plane in the upward direction. The rotation between the world frame and body frame depend on the Euler angles such that, roll ϕ, pitch θ and yaw ψ represent the rotation in x , y, z axis respectively, it will be donated by, ${}^W R_B$, a rotation matrix obtained from multiplying each rotation matrix Z-X-Y in this specific order, the final matrix is given by

$$
{}^W R_B = \begin{bmatrix} cos(\psi)cos(\theta) - sin(\psi)sin(\phi)sin(\theta) & -sin(\psi)cos(\phi) & cos(\psi)sin(\theta) + sin(\psi)sin(\phi)cos(\theta) \\ sin(\psi)cos(\theta) + cos(\psi)sin(\phi)sin(\theta) & cos(\psi)cos(\phi) & sin(\psi)sin(\theta) - cos(\psi)sin(\phi)cos(\theta) \\ -cos(\phi)sin(\theta) & sin(\phi) & cos(\phi)cos(\theta) \end{bmatrix}. \tag{3.1}
$$

Next, the angular velocity of the body frame relatively to the world frame is denoted by $\omega_B = [p, q, r]^T$, these values are the derivatives of roll ϕ, pitch θ and yaw ϕ such that

$$
\omega_B = \begin{bmatrix} cos(\theta) & 0 & -cos(\phi)sin(\theta) \\ 0 & 1 & sin(\theta) \\ sin(\theta) & 0 & cos(\phi)cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \tag{3.2}
$$

As we know, this tilt-rotor consists of three rotors capable of producing a lifting force in the positive B_z axis. Two of these rotors can be rotated independently in the B_y axis and the produced angle is given by α_1 for rotor 1 and α_2 for rotor 2. Each motor is able to produce a force, F_i, with its angular speed ω_i given by

$$
F_i = k_F \omega_i^2 \tag{3.3}
$$

being k_F a system constant relative to the used motor model. The resulting force exerted on the body is expressed by

$$F_1^B = \begin{bmatrix} F_1^{B_x} \\ F_1^{B_y} \\ F_1^{B_z} \end{bmatrix} = \begin{bmatrix} F_1 \sin(\alpha_1) \\ 0 \\ F_1 \cos(\alpha_1) \end{bmatrix}, \tag{3.4}$$

$$F_2^B = \begin{bmatrix} F_2^{B_x} \\ F_2^{B_y} \\ F_2^{B_z} \end{bmatrix} = \begin{bmatrix} F_2 \sin(\alpha_2) \\ 0 \\ F_2 \cos(\alpha_2) \end{bmatrix}, \tag{3.5}$$

$$F_3^B = \begin{bmatrix} F_3^{B_x} \\ F_3^{B_y} \\ F_3^{B_z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ F_3 \end{bmatrix}. \tag{3.6}$$

Each motor also produces two forms of moments exerted on the hub of each motor. The first one is due to the propeller drag, $M_{D,i}$, given by

$$M_{D,i} = k_M \omega_i^2. \tag{3.7}$$

The second form appears when a rotor is commanded to tilt, thus producing a small gyroscopic effect generating the moment $M_{G,i}$ given by

$$M_{G,i} = I_i(\dot{\Gamma}_i \times \omega_i) \tag{3.8}$$

where I_i represents the propeller inertia and $\Gamma_i = [0\ \dot{\alpha}_i\ 0]^T$. These values now need to be expressed relatively to the tri-rotor body rotation. For that a rotation matrix, ${}^P R_B$, converts between the rotor shaft rotation to the body frame like so

$${}^P R_B = \begin{bmatrix} \cos(\alpha_i) & 0 & \sin(\alpha_i) \\ 0 & 1 & 0 \\ -\sin(\alpha_i) & 0 & \cos(\alpha_i) \end{bmatrix}. \tag{3.9}$$

The tri-rotor design of the drone will also have a moment effect relative to the position of the three rotors, that is obtained with the cross product of the force vector and the vector between the centre of mass and the propeller. Lets define $P_i = [P_{i,x}\ P_{i,y}\ P_{i,z}]$ the rotor hub position relative to the body frame and h_P the distance between the hub and the rotor propeller. With this notation, we can present the position vector of each rotor propeller relative to the centre of mass, r_i, expressed in the body frame B

$$r_1 = \begin{bmatrix} P_{1,x} + h_P \sin(\alpha_1) \\ P_{1,y} \\ P_{1,z} + h_P \cos(\alpha_1) \end{bmatrix}, \tag{3.10}$$

$$r_2 = \begin{bmatrix} P_{2,x} + h_P \sin(\alpha_2) \\ P_{2,y} \\ P_{2,z} + h_P \cos(\alpha_2) \end{bmatrix}, \tag{3.11}$$

$$r_3 = \begin{bmatrix} P_{3,x} \\ P_{3,y} \\ P_{3,z} \end{bmatrix}. \tag{3.12}$$

Now we can obtain both net force and the total moment vector expressed in the body frame accounting with all the forces introduced to the system by controller command inputs, F^B and M^B

$$F^B = \sum_{i=1}^{3} F_i^B \tag{3.13}$$

$$M^B = \sum_{i=1}^{3} {}^P R_B (M_{D,i} + M_{G,i}) + \sum_{i=1}^{3} (F_i^B \times r_i). \tag{3.14}$$

Note that this model is only precise in vertical flight mode where speed is reduced and the aircraft aerodynamic don't play a big part on the forces exerted on the drone, because of these, aerodynamic forces like the aileron force are ignored. Later in this thesis, a more precise model will be formulated to account for the transitions mode and the horizontal flight mode, where some additional forces are present and need to be accounted for.

3.3 System dynamics

Knowing the forces exerted on the drone's body, by Newton's equations of motions, its possible to obtain the system dynamics containing the linear acceleration $\dot{v}$ and angular acceleration $\dot{\omega}_B$ in the following equation:

$$\begin{aligned} \dot{p} &= v \\ \dot{v} &= -g W_z + \frac{1}{m} {}^W R_B F^B \\ \dot{\omega}_B &= I^{-1}(-\hat{\omega}_B I \omega_B + M^B) \end{aligned} \tag{3.15}$$

Where g is the gravitational acceleration, m is the body mass, I the moment of inertia, and $\hat{\omega}_B$ is the skew matrix resulting from ω_B. The vehicle position is $p = [p_x, p_y, p_z]^T$ and the velocity $v = [v_x, v_y, v_z]^T$ of the body frame described in the world frame. These equations describe the system state, with them it is possible to obtain the updated body states in each step from the controller inputs.

The next step from here would be implementing this model in *Matlab*. From there the controller development can begin and will result in a complete simulated autopilot system ready to be simulated using GAZEBO environment with PX4 framework.

It's important to acknowledge that, working on a hybrid vehicle has its added complexity, since the UAV has more than one mode of operation where its dynamics, exerted

forces, and control mechanisms behave in different ways. Knowing this, in order to simplify the modeling process, the model will be separated into two components: standard tri-rotor and fixed wing (horizontal flight mode). Both of them will apply the same dynamics equations as (3.15), being the only difference the input forces, F^B, and moments, M^B, as they are calculated in distinct ways depending on the flight mode.

3.3.1 Tri-rotor dynamics

In this mode, the tri-rotor acts as a conventional drone (quad-rotor). Since the air velocity is low, the drag and aerodynamic forces can be ignored. The thrust vector contains now two components, being able to generate force in the z_B and x_B body axis direction due to the tilt rotors. Torques are always compensated through the mixer and controlled by the desired angular velocity, to be sent to the UAV by the proposed controller.

Here the three rotors are operational, the input force of the system, F^B will be the sum of the three rotors forces (Equation (3.13)). The torque system inputs M_B, will be directly fed from the controller to the dynamic system, the reason being that no external forces will actively change the rotation of the aircraft. For this purpose, the input moment will be denominated u_t, where u_t is the desired torque calculated by the proposed controller.

The model dynamics for the vertical flight mode can be expressed with the following motion equation

$$\begin{aligned} \dot{P} &= v \\ \dot{v} &= -gW_z + \frac{1}{m}{}^W R_B F^B \\ \dot{\omega}_B &= I^{-1}(-\hat{\omega}_B I \omega_B + u_t) \\ {}^W\dot{R}_B &= {}^W R_B \hat{\omega}_B. \end{aligned} \tag{3.16}$$

This sub-model will be used in junction with a separate controller developed for the horizontal flight mode of the UAV, in order to test and validate the controller in computational simulation environment.

3.3.2 Fixed wing dynamics

Going forward to the horizontal flight mode, where the two front rotor are at 90º angles and the back rotor is turned off, a more complex model will be needed since the air velocity in this phase is significantly higher. The aerodynamic forces can no longer be discarded and are in fact the reason why the drone is be able to maneuver in this flight mode. Now the hybrid UAV will function similarly to a conventional fixed wing aircraft.

Firstly, the thrust vector is now always pointing in the x_B body axis direction, as the back rotor is not operational and the tilt mechanism is fixed to the 90° angle. The thrust force produced by the rotors will be

$$F_t^B = F_1 x_B + F_2 x_B. \tag{3.17}$$

Note that in this state the back rotor, T_3, is not generating any force or torque.

Now that we are present with a vehicle behaving like a conventional aircraft, aerodynamics play a big role in keeping it in the air. Two main relevant forces are exerted on the body of the UAV: the drag force, D, and the lift force L. These forces are dependent on a series of characteristics of the drone's body shape, they can be simplified using a drag and lift coefficients, C_D, C_L. Both of these coefficients are dependent of the angle of attack(AoA), α, angle between the chord line of the airfoil and wind velocity vector.[27]

For relatively small roll and pitch angles, where is planed the drone to be within, the values can be simplified into two curves dependent on the AoA[28].

$$\begin{aligned} C_L &= C_{L_0} + C_{L_\alpha}\alpha \\ C_D &= C_{D_0} + C_{DC_L}C_L^2 \end{aligned} \tag{3.18}$$

where, $C_{L_0}, C_{L_\alpha}, C_{D_0}$, are fixed values representing lift and drag at zero values and the respectively curve slope, C_{DC_L}, the induced drag derivative. From this, the simulated drag and lift forces can be easily calculated, being them, a direct product between the wing surface area, S, and the dynamic pressure, $\bar{q}$.

$$\begin{aligned} D &= \bar{q}SC_D \\ L &= \bar{q}SC_L \\ \bar{q} &= \frac{1}{2}\rho V_a^2 \end{aligned} \tag{3.19}$$

being ρ the air density and V_a the wind speed.

This concludes the aerodynamics forces, F_a^B, acting on the center of mass of the UAV in a stability body axis.They now need to be mapped to the body frame, B, in the following form

$$F_a^B = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 0 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} -D \\ 0 \\ L \end{bmatrix}. \tag{3.20}$$

The total force generated by the UAV results from the thrust produced by the rotors and aerodynamics. The force applied to the center of mass expressed in the body frame becomes

$$F^B = F_t^B + F_a^B \tag{3.21}$$

Note that the last equation only expresses forces produced by the UAV.

Lastly the torques produced by the elevons, they are able to produce roll and pitch moments, expressed in the center of mass. They are

$$M_a^B = \bar{q}SL \begin{bmatrix} C_r \\ C_p \\ 0 \end{bmatrix} \tag{3.22}$$

where L is the position relatively to the center of mass of the elevons

$$L = \begin{bmatrix} E_y & 0 & 0 \\ 0 & E_x & 0 \\ 0 & 0 & E_z \end{bmatrix} \tag{3.23}$$

and C_r, C_p the roll and pitch aerodynamic coefficients. Similar to drag and lift coefficients, they are calculated in the following manner

$$\begin{aligned} C_r &= C_{r_\delta}\delta_a \\ C_p &= C_{p_0} + C_{p_\alpha}\alpha + C_{p_\delta}\delta_e \end{aligned} \tag{3.24}$$

being C_{r_δ}, C_{p_0}, C_{p_α}, C_{p_δ} aerodynamics coefficients and δ_a, δ_e aileron and elevator deflection angles respectively, which can be translated to elevon deflection so that the left(δ_{le}) and right(δ_{rh}) elevon deflection angle are

$$\begin{aligned} \delta_{le} &= \delta_a - \delta_e \\ \delta_{rh} &= -\delta_a - \delta_e \end{aligned} \tag{3.25}$$

The total torque matrix applied to the the center of mass is a sum of the aerodynamic torques and rotor torques expressed as

$$M^B = \sum_{i=1}^{2} {}^P R_B(M_{D,i}) + \sum_{i=1}^{2}(F_i^B \times r_i) + M_a^B. \tag{3.26}$$

Knowing this, the full fixed wing motions equations will result in the following equation.

$$\begin{aligned} \dot{P} &= \nu \\ \dot{\nu} &= -gW_z + \frac{1}{m}{}^W R_B F^B \\ \dot{\omega}_B &= I^{-1}(-\hat{\omega}_B I \omega_B + M^B) \\ {}^W\dot{R}_B &= {}^W R_B \hat{\omega}_B. \end{aligned} \tag{3.27}$$

This concludes the dynamic model of the UAV in fixed wing flight mode. Now, to have a running mathematical simulation, it becomes a question of transforming the output of the proposed controller into the forces and torques they would generate and use Equation (3.27) to give the vehicle state in each step. More of this will be discussed in chapter 5.

Chapter 4

Control Design

In this section, lays the implementation of the autopilot controller proposed in order to command the UAV in both flight modes. The objective of such controller is to, knowing the state x,

$$x = [p, v, \omega_B, {}^{W}R_B]^T \tag{4.1}$$

of the vehicle and the desired trajectory, output the desired thrust and torque components, u

$$u = [u_1, \tau_2, \tau_3, \tau_4]^T \tag{4.2}$$

that will enable the UAV to follow the desired flight path.

Because of the major difference between control methods in the two main flight modes of the UAV, two separate control algorithms with the same base approach yet with different outcomes will be presented further on. One of them will focus on horizontal flight mode as for the second, it will just be used when the vehicle is in vertical flight mode. This solution discards, for now, the problem that will appear when the UAV is in a transition mode, later, on this work we will see that this problem can be overcome with a semi-hybrid control output involving both of these controllers.

4.1 Controller Design

The proposed controller implementation will be based on the work [29], it contains a trajectory tracking controller designed specifically for a quad-rotor UAV. The bases of the controller still apply for a tri-rotor UAV. With some alterations and different tuning it can be perfected for our case. As for the fixed wing controller, a separate controller with further design modification will be proposed in order to account for the special system dynamics in such flight mode.

The controller operates with two main loops: an outer loop that will calculate the desired force acting on the vehicle body, F_{des}, from the velocity and position errors given by v_{ref} and p_{ref},respectively; And the attitude loop, where the desired body orientation, ${}^{W}R_{Bdes}$, and the necessary body moments are computed, yielding all the necessary input variables, $u = [u_1, u_2, u_3, u_4]^T$. The trajectory will be generated by the derivatives of, p_{ref} and ψ_{ref}, the reference position and yaw respectively, as represented in the controller diagram in Figure 4.1. Both versions of the proposed controller are presented in detail in the following sections.

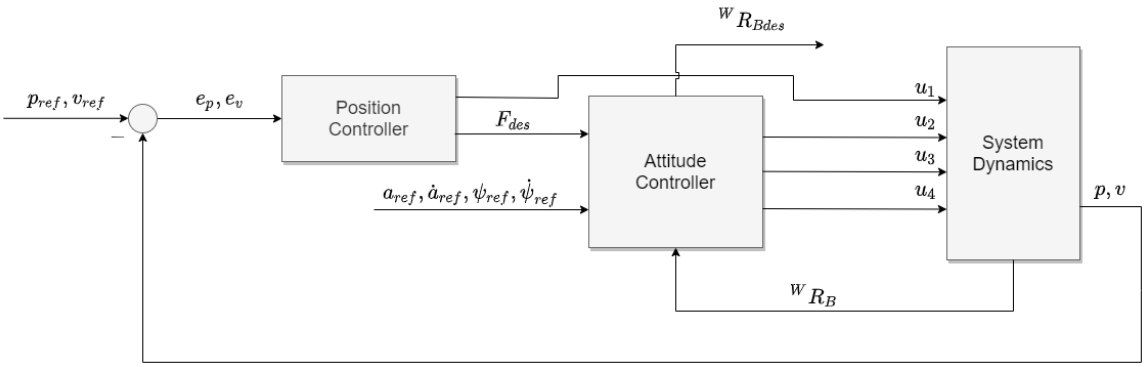


Figure 4.1: Control loop diagram

4.1.1 Tri-rotor Control Law

In this mode, the base design of the work [29] is kept almost intact, only a different tune will be used.

The first step in the position loop is to compute the position and velocity errors as

$$\begin{aligned} e_p &= p - p_{ref} \\ e_v &= v - v_{ref} \end{aligned}. \tag{4.3}$$

Adding now a proportional gain ,K_p, K_v, as positive gain matrices the desired force vector is calculated as

$$F_{des} = -K_p e_p - K_v e_v + mgz_W + ma_{ref} \tag{4.4}$$

assuming $\|F_{des}\| \neq 0$. The first input u_1(thrust command) ,can be computed as a projection of the desired force vector into the body frame z_B coincident with the drone thrust vector

$$u_1 = F_{des} \cdot z_B. \tag{4.5}$$

This concludes the position loop, next being the attitude loop where the desired orientation and orientation error, R_{des} and e_R, respectively, are computed. Firstly the desired rotation matrix is filled with the desired body axis as

$$R_{des} = [x_{B_{des}}, y_{B_{des}}, z_{B_{des}}]^T. \tag{4.6}$$

As we already know the body axis Z_B is along the thrust vector being $z_{B_{des}}$ obtained as the force vector direction

$$z_{B_{des}} = \frac{F_{des}}{\|F_{des}\|}. \tag{4.7}$$

Assuming now a intermediate coordinate frame, representing the orientation of the world frame, W after the reference yaw rotation given by

$$x_{\psi_{des}} = [\cos(\psi_{ref}), \sin(\psi_{ref}), 0]^T \tag{4.8}$$

Knowing $x_{\psi_{des}}$ it is now possible to compute $x_{B_{des}}$ and $y_{B_{des}}$ as

$$y_{B_des} = \frac{z_{B_des} \times x_{\psi_des}}{\left\|z_{B_des} \times x_{\psi_des}\right\|} \tag{4.9}$$

$$x_{B_des} = y_{B_des} \times z_{B_des} \tag{4.10}$$

assuming $z_{B_{des}} \times x_{\psi_des} \neq 0$. This gives us the desired body orientation R_{des}. To continue to compute the angular velocity, w_{Bdes}, the rotation error e_R is calculated as

$$e_R = \frac{1}{2}(R_{des}^T {}^W R_B - {}^W R_B^T R_{des})^\vee \tag{4.11}$$

being $R_{des} = {}^W R_{Bdes}$, the desired body rotation matrix previously calculated and $\vee$ represents the vee map operand(skew matrix inverse). For the desired angular velocity, w_{Bdes}, the jerk(acceleration derivative) equation can be manipulated such as

$$m\dot{a}_{ref} = \dot{u}_1 z_{Bdes} + {}^W R_B w_{Bdes} \times u_1 z_{Bdes}. \tag{4.12}$$

Like u_1 (4.5) $\dot{u}_1$ can be expressed as the projection through the Z_B body axis $\dot{u}_1 = m\dot{a}_{ref} \cdot z_{B_{des}}$. Substituting $\dot{u}_1$ we get h_{wdes} defined as

$$h_{wdes} = {}^W R_{Bdes} w_{Bdes} \times u_1 z_{b_{des}} = \frac{m}{u_1}(\dot{a}_{ref} - (z_{B_{des}} \cdot \dot{a}_{ref}) z_{Bdes}). \tag{4.13}$$

Defining $w_{Bdes} = [p_{des}, q_{des}, r_{des}]^T$, the first two components can be calculated

$$p_{des} = -h_{wdes} \times y_{Bdes}, \quad q_{des} = -h_{wdes} \times x_{Bdes}. \tag{4.14}$$

The third component r_{des}, is found by solving the equation relating both derivatives of Euler angles $\dot{\phi}_{des}, \dot{\theta}_{des}, \dot{\psi}_{des}$ and angular rates

$${}^W R_{B_{des}} \begin{bmatrix} p_{des} \\ q_{des} \\ r_{des} \end{bmatrix} = \begin{bmatrix} x_{\psi des} & y_{B_{des}} & z_W \end{bmatrix} \begin{bmatrix} \dot{\phi}_{des} \\ \dot{\theta}_{des} \\ \dot{\psi}_{des} \end{bmatrix} \tag{4.15}$$

with this the desired angular velocity vector is completed and its error can be computed

$$e_R = w_B - w_{Bdes} \tag{4.16}$$

Finally the last three inputs are given by multiplying two positive gain matrices K_r and K_w with both errors

$$u_t = [u_2, u_3, u_4]^T = -K_r e_R - K_w e_w \tag{4.17}$$

This concludes the fist controller, aimed to control the drone in vertical flight mode. Before entering in the simulation process to validate this model, let us first define the second controller algorithm which is a subtle modification of what has just been explained.

4.1.2 Fixed Wing Control Law

Now, for the horizontal flight mode, the controller needs to be redesigned to cope with the physical changes in the system. The main changes concerning the design of the controller are:

- Only the two front rotors are active and rotated 90°, producing now a forward thrust force in the direction of the body axis x_B;
- Resulting from the increased speed, aerodynamic forces such as lift and drag, now need to be accounted for;
- The yaw angle, ψ is now constrained with the direction of the trajectory of flight and no longer given as a reference. This is due to the flight dynamics of a fixed wing, where yaw is directly correlated to the direction of movement minus minor drifts.

The first step in the position loop still remains the same, calculating the position errors, but now these are calculated in respective to the auxiliary frame A. The auxiliary frame corresponds to the world frame, W, with and added rotation in the z_W axis, so that x_A axis direction is always in the same direction as the current body yaw angle. We define a Rotation matrix that transforms between the world frame and the auxiliary frame,

$${}^W R_A = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.18}$$

The position and velocity errors defined in respect to frame A, are calculated as

$$\begin{aligned} e_p &= {}^W R_A (p - p_{ref}) \\ e_v &= {}^W R_A (v - v_{ref}) \end{aligned} \tag{4.19}$$

This procedure will facilitate later on the tuning process for the roll and pitch moments as now, the desired force F_{des} will be relative to the auxiliary frame A, that is coupled with the body frame B, meaning that roll and pitch will always be correlated with the UAV true axis of rotation. As we have two more main forces acting on the body, drag and lift, the desired force F_{des} will have to account for them, in the mathematical model they

will be the exact force exerted on the body as for STIL and HDIL simulations they will be a approximate calculation as expressed in section 3.3.2. The desired force is computed as

$$F_{des} = -K_p e_p - K_v e_v + mgz_W + m\,{}^W R_A a_{ref} - {}^W R_A (L + D) \tag{4.20}$$

With this, the thrust command can be obtained being, the projection of the desired force onto x_B axis, where the thrust force is directed, concluding the position loop

$$u_1 = F_{des} \cdot x_B. \tag{4.21}$$

For the attitude loop, the objective is once more obtain the desired rotation, R_{des}. Firstly, as a fixed wing generates the majority of the net force in the x_B axis, corresponding to the direction of thrust, meaning, its movement direction in normal conditions is close to the x_B direction(with a low angle of attack)(In other words, the aircraft moves roughly to where the nose points). We assume $x_{B_{des}}$ as the direction of F_{des}

$$x_{B_{des}} = \frac{F_{des}}{\|F_{des}\|}. \tag{4.22}$$

As we know, a fixed wing aircraft achieves a change of direction in the xy plane with a roll moment, a rotation in the x_B axis. The roll rotation will be commanded by the direction of $y_{B_{des}}$ so

$$y_{B_{des}} = \frac{[0, x_{By}, x_{Bx}]^T \times x_A}{\left\|[0, x_{By}, x_{Bx}]^T \times x_A\right\|} \tag{4.23}$$

with $x_A = [1, 0, 0]^T$ and $x_{B_{des}} = [x_{Bx}, x_{By}, x_{Bz}]^T$. Then $z_{B_{des}}$ becomes

$$z_{B_des} = x_{B_{des}} \times y_{B_{des}}. \tag{4.24}$$

Finally, the only step left is bringing the calculated rotation to the world frame again by rotating it over the z_W axis.The required rotation is equal to the desired yaw angle, ψ_{des} defined as

$$\psi_{des} = \begin{cases} -\arccos \frac{v_x}{\sqrt{v_x^2 + v_y^2}} & v_y \leq 0 \\ \arccos \frac{v_x}{\sqrt{v_x^2 + v_y^2}} & v_y > 0 \end{cases} \tag{4.25}$$

Note the yaw angle is relative to the X_W axis belonging to the interval $-180° < \psi_{des} \leq 180°$. Becoming

$${}^W R_{Bdes} = \begin{bmatrix} \cos\psi_{des} & -\sin\psi_{des} & 0 \\ \sin\psi_{des} & \cos\psi_{des} & 0 \\ 0 & 0 & 1 \end{bmatrix} [x_{B_{des}}, y_{B_{des}}, z_{B_{des}}]. \tag{4.26}$$

From this point on, until the last three commands input, u_2, u_3, and u_4, the same tri-rotor control laws applies.

It is important to note that this design assumes a reference trajectory that is inline with a fixed wing aircraft capabilities, a more exotic trajectory will simply don't work with this design. This leaves space for further development of a optimal trajectory generation, that will not be a part of this work.

4.2 Simulation

Having the model and controller design completed, the next step is to validate such system by simulating and testing its results. To do so, both mathematical systems were implemented in *Simulink* enabling us to simulate the UAV in a closed-loop system giving a reference trajectory as the input.

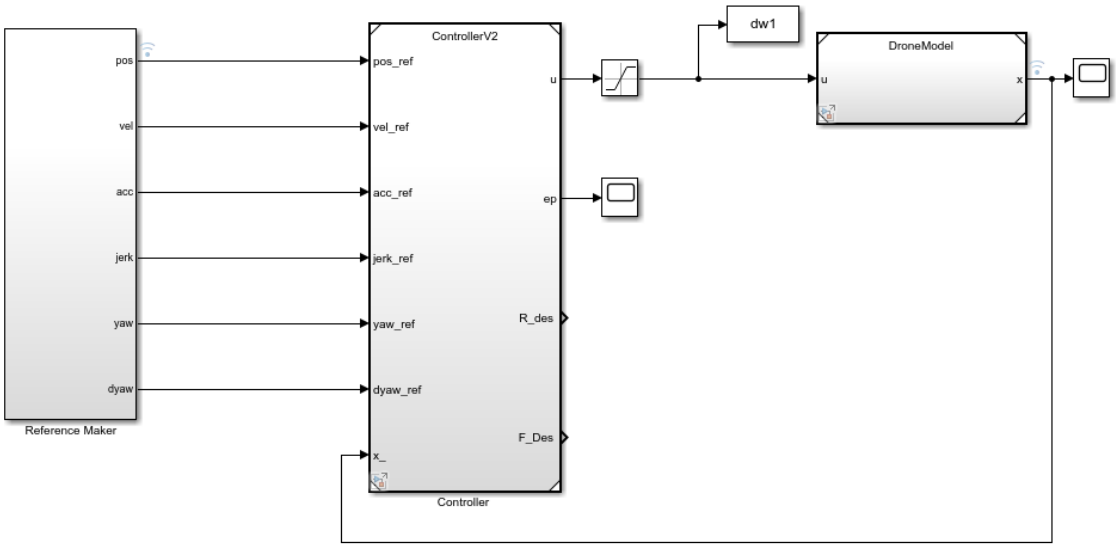


Figure 4.2: *Simulink* block diagram

Before starting the simulation process, a tuning process was made to ensure the controller's maximum performance. Being this a complex nonlinear system, yet with fast simulation trials, no special tuning techniques were used and the process was based on trial an error, knowing the assumptions provided below. Note that as the two controllers were designed for separate flight modes, there will be two tuning processes as well as two sets of gain values.

- $\mathbf{K_p}$: Position gain, higher values generate faster response times but can lead to overshoot or major oscillations;
- $\mathbf{K_v}$: Velocity gain, stabilizes the output at the cost of slower rise times;
- $\mathbf{K_R}$: Rotation gain, higher values creates stability when fixing a desired rotation, yet higher values lead to major rotation overshoots when the reference trajectory changes rapidly;
- $\mathbf{K}_\omega$: Angular velocity gain, preferable lower values than K_R, sensitive to generate rotation oscillations even when hovering.

All of the gain matrices above can provide a separate value for x, y, z components. In VTOL mode x and y references are handled in the same way, to obtain movement in the

xy plane, in other words, the pitch and roll moments are close to symmetric of each other, not being fully symmetrical because of the tri-rotor configuration. This means the x and y gain components are equal to each other. In the other hand, in fixed wing mode roll and pitch moments are used for lateral and vertical movements respectively meaning that they no longer have a reason for being similar.

The tuning process starts by ensuring that a stable hovering position is obtained with a very small position step in the xy plane. From this reference, the objective is to achieve a stable change in the desired rotation, that keeps the drone level to the ground and not oscillating when stationary in mid air, adjusting the attitudes gains, K_R and K_ω, according to the assumption taken before. When this is obtained we pass the focus to the position gain, evaluating the step response in all axis separately, adjusting K_p accordingly and allowing for some overshoot. Next adjusting the velocity gain to reduce or remove completely the overshoot and testing the system response for more complex reference trajectories. As a base reference, for the UAV characteristics in study the gain set that achieved optimal results is

$$\begin{aligned} K_p &= \begin{bmatrix} 12 & 0 & 0 \\ 0 & 12 & 0 \\ 0 & 0 & 15 \end{bmatrix}, \; K_v = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 15 \end{bmatrix} \\ K_R &= \begin{bmatrix} 20 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 20 \end{bmatrix}, \; K_\omega = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}. \end{aligned} \tag{4.27}$$

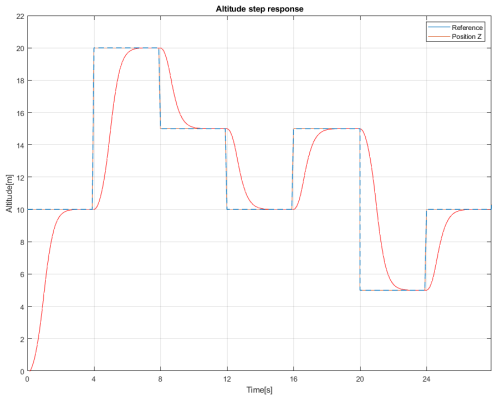
The tuning process for the fixed wing is very similar to what was explained before, being the only difference that the starting state of the UAV, instead of hover, is a sable forward flight with constant speed and a safe altitude. The reference trajectory has a constant froward velocity in order to produce a stable forward flight. This time, the system leans more to a velocity base control instead of position based, this means that heavier velocity gains are used, lowering the static position effect on the control design. The forward flight set of gains are

$$\begin{aligned} K_p &= \begin{bmatrix} 12 & 0 & 0 \\ 0 & 12 & 0 \\ 0 & 0 & 15 \end{bmatrix}, \; K_v = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 15 \end{bmatrix} \\ K_R &= \begin{bmatrix} 20 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 20 \end{bmatrix}, \; K_\omega = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \end{aligned} \tag{4.28}$$

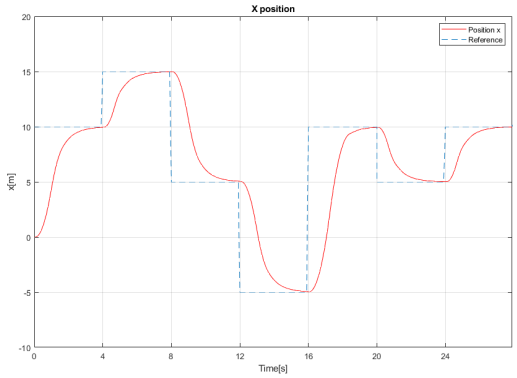
It is important to note that depending on the desired response the gains can be altered to accommodate such needs, e.g. faster response times, no overshoot response, control based on waypoints, control based on velocity, aggressive maneuvering, smoother flight etc. In our case we aimed to a slower but precise response important in testing on thigh spaces e.i a indoor arena.

4.2.1 Simulation Results for Rotary-wing Mode

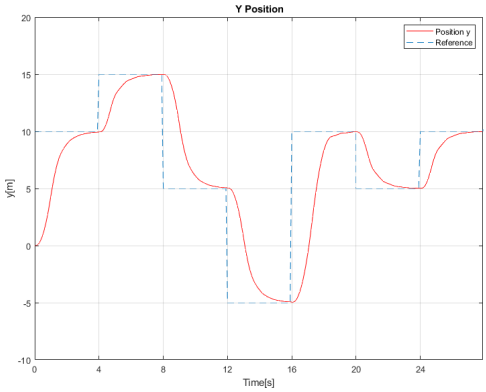
Having adjusted both sets on gain matrices, we proceed to validate them by a set of various simulations, using incremental steps sizes and culminating in a more complex 3D trajectory. Figure 4.3 shows a set of step response for all three axis independently. The tracking controller is able to achieve the desired position with a relative fast time response, for example, a altitude step change of 10 *meters* has a settling time of around 2.5 *seconds* averaging a speed of $4m/s$ in the z_W axis with a approximated overshoot of 0%. In real world testing is safe to assume that settling times will be slower as we are simulating the world without air resistance or any wind perturbations.



(a) Altitude over time



(b) Position in x over time



(c) Position in y over time

Figure 4.3: Position step response in all axis

Having all the gain matrices tuned and validated, we can further test the system subjecting it to more complex 3D trajectory where velocity and acceleration come into play. A semi square trajectory with changes in altitude and velocity was then generated in *simulink* environment and given as a reference trajectory to the system. The 3D plot can be seen in Figure 4.4.

The trajectory last for 30 *seconds* and velocities up to $6m/s$ are achieved in every direction. In the end, the drone is commanded to land in the world frame origin. We can see a good tracking ability, the worst performance occurs when changing direction where

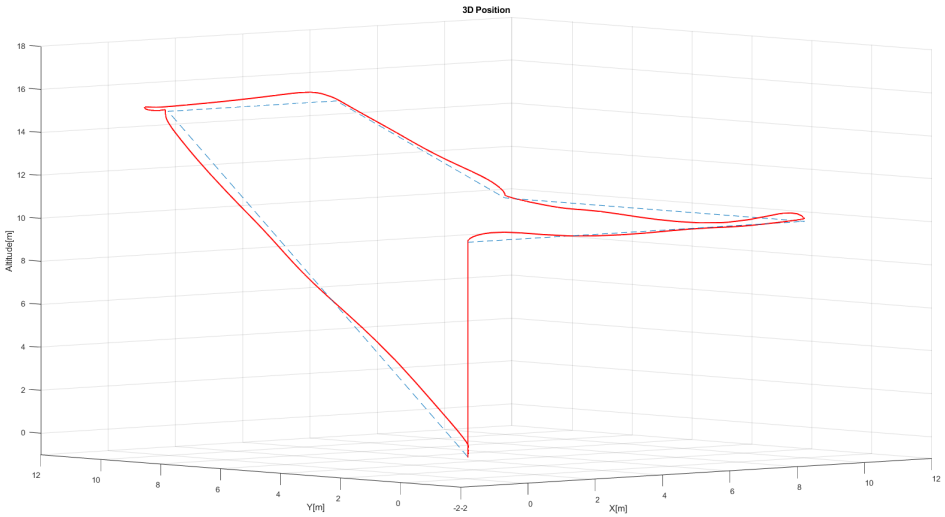


Figure 4.4: System response to a trajectory in 3D space.

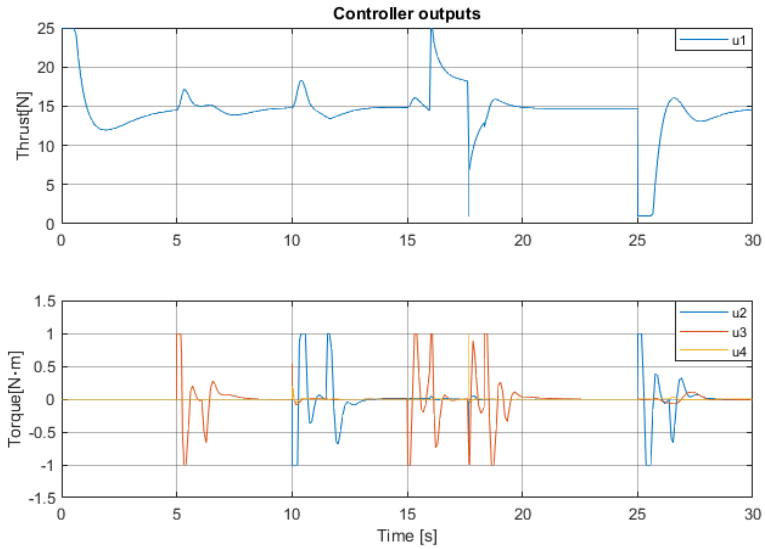


Figure 4.5: Control signal u for VTOL mode

altitude drops quickly about $30cm$ but is able to recover quickly. A explanation for this is based on the configuration of the tri-rotor, for fast rotations the sustainability can be affected and the UAV rapidly looses a small amount force in the z_W axis. Figure 4.5 shows all the controller outputs for the 3D trajectory, the thrust value and respective desired torques.

In addition, we must validate the yaw rotation with a simple ramp. The system response corresponds to the expectations allowing a small overshoot around 3.5%.

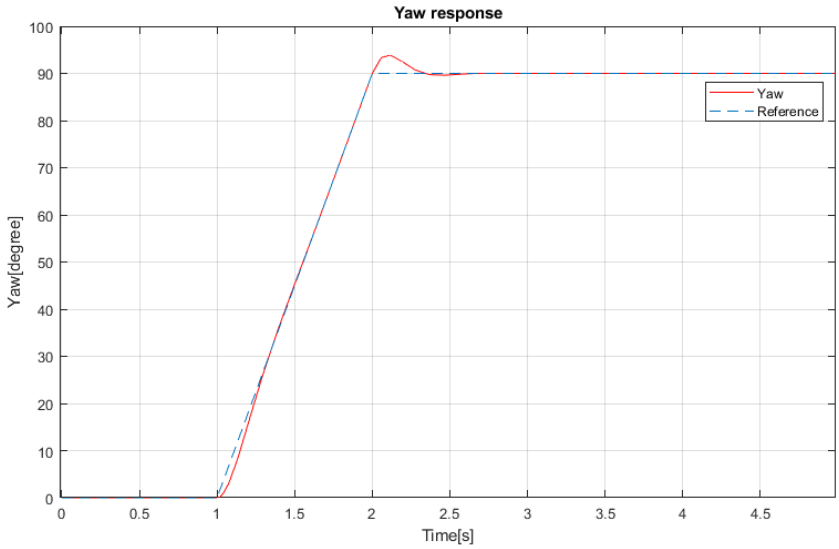


Figure 4.6: Yaw response to a 90 degree ramp

4.2.2 Simulation Results for Fixed-wing Mode

After the validating process for the vertical flight controller, we must also simulate the system in horizontal flight mode and test the proposed controller with the changes proposed above. The simulation process was similar to the previously tests but now the UAV will always start with a groundspeed of $14m/s$ at an altitude of 10 *meters*. This time the controller was tuned with a different set of restrains in mind, as in this state it is important to be sure that the drone is always at a safe altitude and has room for maneuver, because of the speed achieved as it is used to cover longer distances with a relative decent speed. Knowing that normal overshoots values, around 5%, are acceptable giving room for improving the system response and allowing slower steady time also heavier gain our given to velocity being now a more accurate way of controlling the desired trajectory.

Before starting the simulation it is important to know the level stall speed of our vehicle, this is important as we do not want to operate the UAV in speeds lowers than V_{st} as it would obviously not be able to generate enough lift to sustain the UAV, this is also a needed step later on in order to resolve the transition problem. The total lift is calculated in Equation (3.18), to know its maximum we need to know the maximum angle of attack,

α_{max}. As it is not the main focus of this work to have such precise model and knowing that we are using just a simplified equation of the lift coefficient that works on relative stable flight dynamics, this values cannot be easily computed, therefor, we need to set it to a reasonable value for use in this simulation, [27] gives good insight in such problem, from here we assume $\alpha_{max} = 14°$, resulting in

$$mg = \bar{q}SC_{lmax} \iff V_{st} = \sqrt{\frac{2mg}{SC_{lmax}\rho}} \approx 9.5m/s. \tag{4.29}$$

A velocity step response can be seen in Figure 4.7, this shows a good performance in thrust control with minimal overshoot and fast response times, but most importantly note that despite the fast changes in velocity the altitude, also seen in the figure, remains stable, telling that the angle of attack is changing as expected to keep the vehicle altitude constant.

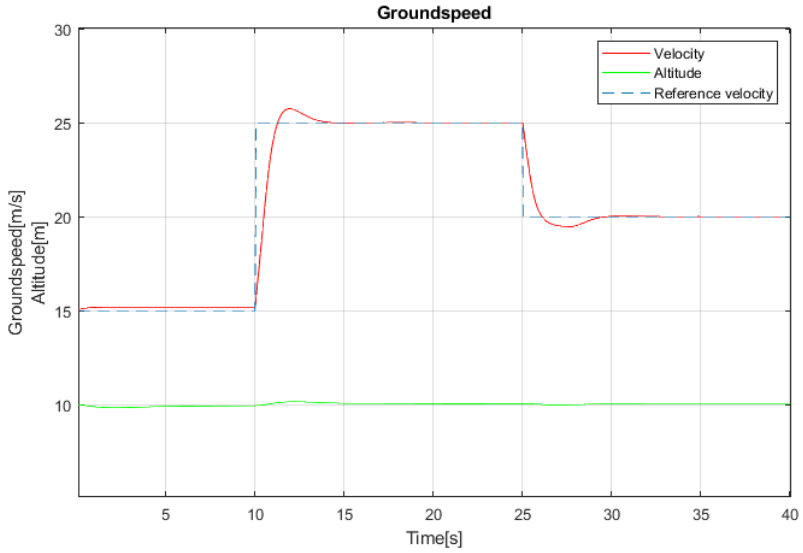
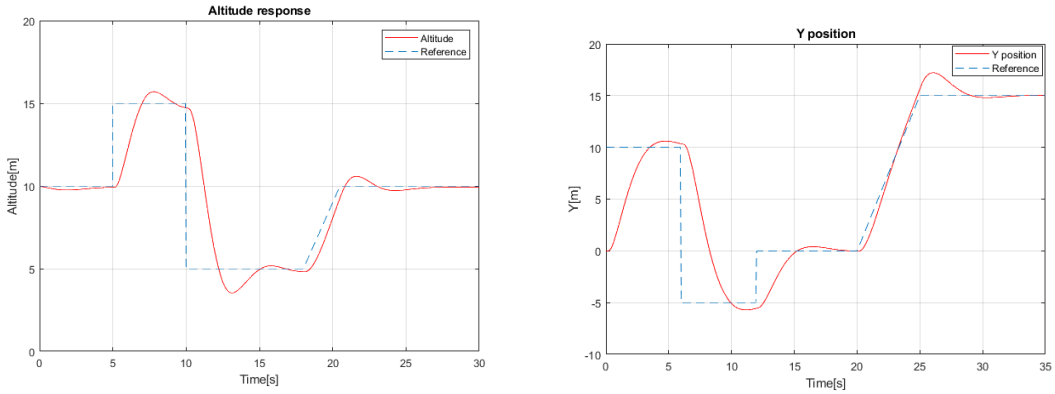


Figure 4.7: Velocity step response

The rest of the simulation plots can be seen in Figure 4.8, where it validates the aimed response constrains, with overshoots averaging 3%, relative small oscillations appearing on the bigger amplitude steps, but they are quick to attenuate and follow the reference precisely without any static error. One can argue that the response time is not quite as good but this decision was made in order to minimize the altitude oscillations when hard banking the UAV, caused by sudden loss of vertical lift. In the other hand, the system responds very well to ramps being able to follow the reference velocity component very well making this controller more stable with complex reference trajectories instead of a waypoint system that works very well in VTOL mode.

Finally, a more complex square trajectory was tested in order to validate all the movements mechanism. Figure 4.9 shows the plotted reference and actual trajectory. The average speed is around 12 m/s constant during the simulation. The overshoots in the xy



(a) Altitude over time

(b) Position in y over time

Figure 4.8: Position step response

plane is explained by the square trajectory, and is expected as fixed wing aircraft can't perform 90° turns, so the UAV performs a steady banked turn. It's also notable the altitude oscillation during the sharper turns, this effect was also explained before and happens due to the sudden loss or vertical lift when the aircraft rolls rapidly to make a turn. This is a common effect on this type of air vehicles and will not have a major effect on the flight trajectory, having the altitude change in this simulation been a maximum of $1.3m$. This could also be improved with a different controller tune. Despite this, the UAV is able to follow the reference trajectory with great performance and relatively low position error.

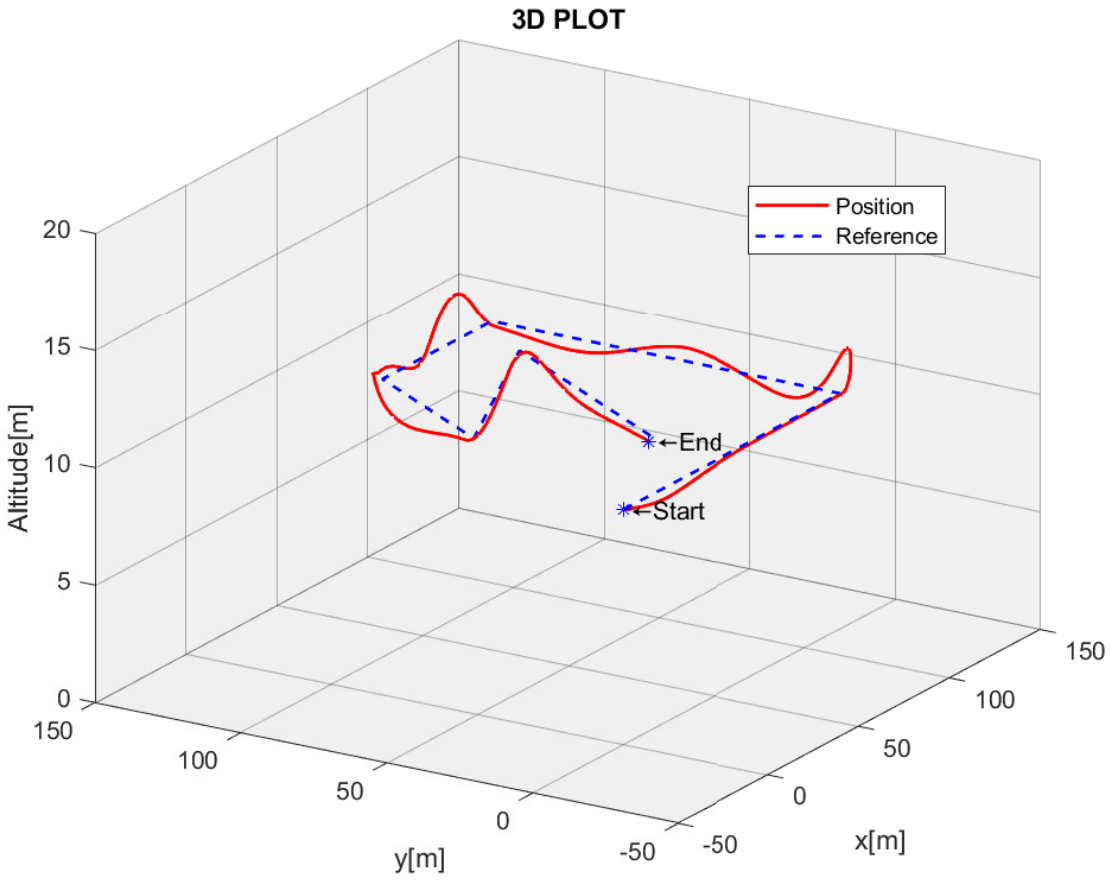


Figure 4.9: System response to square trajectory in 3D space.

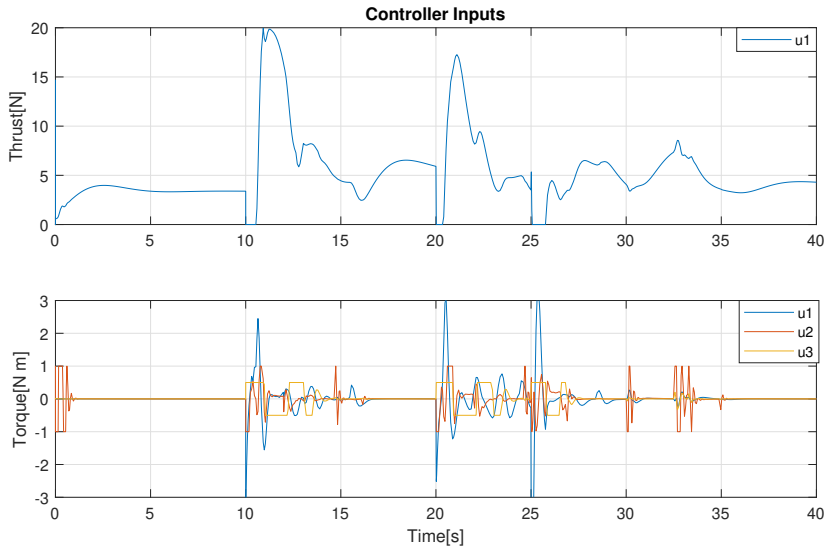


Figure 4.10: Control signal u for fixed wing mode

Chapter 5

Control Allocation

To obtain a full fledged control system one must obtain the desired actuator control signal. To do so a final step, named control allocation, will transform the output of the controller designed in Chapter 4, namely the force and moments values, to real actuator settings, mapping them so that the UAV will produce the desired force and torque. This process is known as control allocation and will be the purpose of this chapter.

The objective is to find the effectiveness matrix, G, that solves the following equation

$$u = Gu_w \tag{5.1}$$

being , $u = [F_x, F_y, F_z, \tau_x, \tau_y, \tau_z]^T$, the desired control input and u_w the actuator vector containing all actuator settings. In this case it contains 7 values, 3 rotor speeds ,w_i, two tilt angles α_i and 2 elevons angles δ_i

$$u_w = \begin{bmatrix} w_1^2 & w_2^2 & w_3^2 & \alpha_1 & \alpha_2 & \delta_e & \delta_a \end{bmatrix}^T. \tag{5.2}$$

Obviously, the effectiveness matrix will only depend on the vehicle configuration and state, representing the way all forces and torques are generated by the UAV actuators. As a reference, a simpler effectiveness matrix belongs to a quad-rotor in the + configuration, and is defined as

$$G = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ k_F & k_F & k_F & k_F \\ 0 & k_F L & 0 & -k_F L \\ -k_F L & 0 & k_F L & 0 \\ k_M & -k_M & k_M & -k_M \end{bmatrix} \tag{5.3}$$

where $u_w = [w_1^2, w_2^2, w_3^2, w_4^2]$ and L is the distance from the rotor hub to the center of mass. The matrices obtained for the Hybrid UAV will be significantly different, due to the more

complex vehicle configuration. After the control allocation, in a HIL (Hardware-In-the-Loop) simulation, the end result is then mapped to PWM values and sent to the UAV controller unit. However, in the mathematical simulation, the actuator values are then mapped again to forces and toques via the equations presented in chapter 3 and sent to the system dynamics loop.

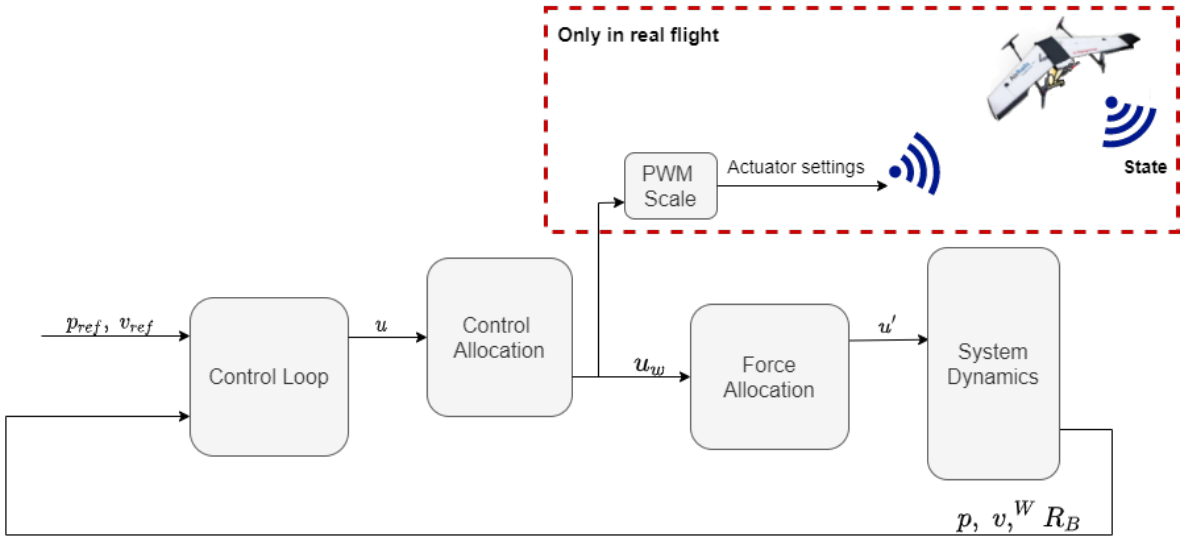


Figure 5.1: Control allocation diagram(Simulation and Real world)

5.1 Effectiveness Matrix

The unique characteristics of a hybrid UAV makes this not a straightforward task. First one must take into account the effectiveness of some actuators in different situations. And, based on that, choose the strategy for each flight state. In the following sections, three different effectiveness matrices will be obtained for different flight modes: tri-rotor, fixed-wing and transition. Each one will use different actuators, being this a over actuated system, with 4DOF(Degrees Of freedom) to seven actuator inputs, although some of the actuators inputs are coupled. This enables the use of various control approaches, note that the ones shown on this work were tested and found to be fit to the problem showing great overall performance.

5.1.1 Tri-rotor

For this VTOL state, both elevons will be ignored since they provide low to none influence at the low speed this flight mode is aimed for. Using only the three rotors will generate an under actuated system, so five inputs will be used: 3 rotors speeds and the tilt angles of the two front rotors. Having this into account, the following equations represent the

output forces and torques connection to the input actuator settings

$$\begin{cases} F_z = k_F w_1^2 cos(\alpha_1) + k_F w_2^2 cos(\alpha_2) + k_F w_3 \\ \tau_x = r_{1y} k_F w_1^2 cos(\alpha_1) + k_M w_1^2 sin(\alpha_1) + r_{2y} k_F w_2^2 cos(\alpha_2) + k_M w_2^2 sin(\alpha_2) \\ \tau_y = r_{1x} k_F w_1^2 cos(\alpha_1) + r_{2x} k_F w_2^2 cos(\alpha_2) + r_{3x} k_F w_2^2 \\ \tau_z = r_{1y} k_F w_1^2 sin(\alpha_1) + k_M w_1^2 cos(\alpha_1) + r_{2y} k_F w_2^2 sin(\alpha_2) - k_M w_1^2 cos(\alpha_2) + k_M w_3^2 \end{cases} \tag{5.4}$$

where $r_{ix/y}$ represents the rotor distance to the center of mass in x and y axis. Looking into the equation system, it's obvious that it is nonlinear and for sure, if arranging the system to accommodate the template in equation 5.1, the effectiveness matrix G would not be invertible. To overcome this problem, the system of equation was linearized trough first order Taylor's series expansion around a equilibrium point. The linearization process can be tedious and quite a stretch to enumerate so the end result, already adapted to the desired template, culminates in the following equation

$$\begin{bmatrix} F_x \\ F_y \\ F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} K_f\,(\,\overline{w_1}\sin(\overline{\alpha}_1)\; + \overline{w_2}\sin(\overline{\alpha}_2)\,) \\ 0 \\ K_f\,(\,\overline{w_1}\cos(\overline{\alpha}_1)\; + \overline{w_2}\cos(\overline{\alpha}_2)\,) \\ K_m\,(\overline{w}_1\sin(\overline{\alpha}_1)\; - \overline{w}_2\sin(\overline{\alpha}_2)\,) + \; K_f\Big(\,\overline{w_1}\cos(\overline{\alpha}_1)\; r_y + \overline{w_2}\cos(\overline{\alpha}_2)\; r_y\Big) \\ r_x K_f\,(\overline{w_1}\cos(\overline{\alpha}_1)\; + \overline{w_2}\cos(\overline{\alpha}_2)\,) \\ K_m\,(\overline{w}_1\cos(\overline{\alpha}_1)\; - \overline{w}_2\cos(\overline{\alpha}_2)\,) + \; K_f\Big(\,\overline{w_1}\sin(\overline{\alpha}_1)\; r_y + \overline{w_2}\sin(\overline{\alpha}_2)\; r_y\Big) \end{bmatrix} +$$

$$\begin{bmatrix} K_f\sin(\overline{\alpha}_1) & K_f\sin(\overline{\alpha}_2) & 0 & K_f\cos(\overline{\alpha}_1)\,\overline{w}_1 & K_f\cos(\overline{\alpha}_2)\,\overline{w}_2 \\ 0 & 0 & 0 & 0 & 0 \\ K_f\cos(\overline{\alpha}_1) & K_f\cos(\overline{\alpha}_2) & K_f & K_f\sin(\overline{\alpha}_1)\,\overline{w}_1 & K_f\sin(\overline{\alpha}_2)\,\overline{w}_2 \\ K_f r_y\cos(\overline{\alpha}_1) + K_m\sin(\overline{\alpha}_1) & K_f r_y\cos(\overline{\alpha}_2) - K_m\sin(\overline{\alpha}_2) & 0 & -K_f r_y\overline{w}_1\sin(\overline{\alpha}_1) + K_m\overline{w}_1\cos(\overline{\alpha}_1) & -K_f r_y\overline{w}_2\sin(\overline{\alpha}_2) - K_m\overline{w}_2\cos(\overline{\alpha}_2) \\ +K_f r_x\cos(\overline{\alpha}_1) & +K_f r_x\cos(\overline{\alpha}_2) & K_f r_{3x} & -K_f r_x\sin(\overline{\alpha}_1)\,\overline{w}_1 & -K_f r_x\sin(\overline{\alpha}_2)\,\overline{w}_2 \\ K_f r_y\sin(\overline{\alpha}_1) + K_m\cos(\overline{\alpha}_1) & K_f r_y\sin(\overline{\alpha}_2) - K_m\cos(\overline{\alpha}_2) & K_m & -K_f r_y\overline{w}_1\cos(\overline{\alpha}_1) - K_m\overline{w}_1\sin(\overline{\alpha}_1) & -K_f r_y\overline{w}_2\cos(\overline{\alpha}_2) + K_m\overline{w}_2\sin(\overline{\alpha}_2) \end{bmatrix} \begin{bmatrix} \delta w_1 \\ \delta w_2 \\ w_3 \\ \delta\alpha_1 \\ \delta\alpha_2 \end{bmatrix}$$

where $\overline{w_i}, \overline{\alpha_i}$ represent the settings for the equilibrium point and $\delta_{w_i}, \delta_{\alpha_i}$ are interval values such that

$$w_i = \overline{w_i} + \delta_{w_i}$$
$$\alpha_i = \overline{\alpha_i} + \delta_{\alpha_i}.$$

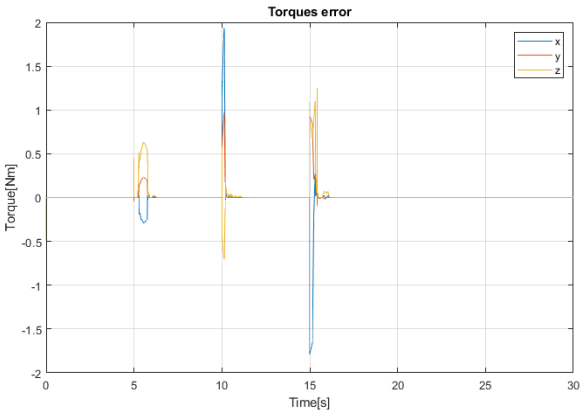
Also for the sake of clarity in Equation (5.1.1) all rotor speeds w_i are already squared, $w_i = w_1'^2$.

For this setup, the equilibrium point used is relative to the actuator settings providing a stable hover flight. Away from hover, the equilibrium point is updated every time step to the last output implying that the effectiveness matrix changes dynamically every time step. Although this method could result in jittering for changing the allocation matrix so fast, the results were found to be very good, in any case. In case it is necessary this process can be changed to only update the equilibrium point when the linearization error overcomes a constant value, saving compute power. In any case, to obtain the desired actuator settings one must invert the effectiveness matrix G, being it not a square matrix

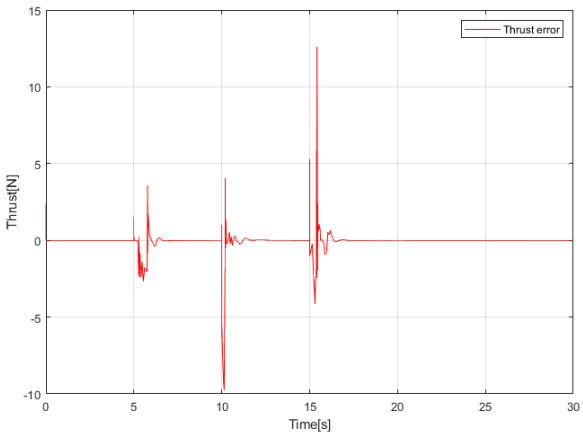
the pseudo inverse operation is used to obtain the same effect

$$u_w = G^+ \begin{bmatrix} F_x \\ F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \tag{5.5}$$

where G^+ stands for the Moore–Penrose inverse. To validate the linearization process, the system was simulated using Figure 5.1 method with a simple waypoint trajectory. The error between the desired control inputs, u, and the force input applied to the system,u_f, produced by the actuator settings was ploted.



(a) Torques error



(b) Thrust error

Figure 5.2: Desired versus actual forces and torques errors

As the plots show, the deduced effectiveness matrix is able to compute all of the actuator inputs precisely to produce the desired forces in most of the flight path. There are, indeed, some significant divergence that happen when the reference trajectory rapidly changes ($t = 5,|;10,\ 15\ seconds$), both in force and torques. These effects happen mostly

because of two reasons. The first one is that the physics limitations of the UAV are not accounted for on the controller part, meaning that high value spikes on the output of the controller can occur in fast trajectory changes that are not physical possible for the UAV to achieve. Mainly because of rotor speed limitation and tilt angle limits, this results on lower achieved values, increasing the error. The second reason has to do with the fast moving trajectory. Requiring sudden changes in rotations and speed means that the equilibrium point where the matrix is linearized rises farther from the desired actuator settings, increasing the linearization error. Even so, the error quickly comes back to zero once a more stable movement and smoother inputs occur. With this method a smother flight path occurs at the cost of slower rise times but more than enough for the intent use of the vertical flight mode.

5.1.2 Fixed Wing

The proposed allocation technique for the horizontal flight mode fixes three controller inputs namely, both tilt servo motors and the back rotor. They will not be used in this mode, this makes the control allocation behave like a conventional aircraft equipped with elevons. Now the velocities achieved are high enough to create lift and produce both roll and pitch moments with only the elevon action. Both rotors are tilted 90° producing a constant force in the positive body axis, x_B direction. They are also able to produce yaw moment and consequently a unwanted minimal roll moment created by the propeller drag that is counteracted by the elevons. Using equation 3.26 and 3.13 it's possible to arrive to the 4DOF equations and then arrive to the effectiveness matrix G. This presumes that force in both z and y body frame axis direction is ignored due to no actuators influencing in them.

$$G = \begin{bmatrix} k_F & k_F & 0 & 0 \\ k_M & -k_M & \bar{q}SE_y C_{r\delta} & 0 \\ 0 & 0 & 0 & \bar{q}SE_x C_{p\delta} \\ r_{1_y} k_F & -r_{2_y} k_F & 0 & 0 \end{bmatrix} \tag{5.6}$$

Note that $\bar{q}$ contains a system variable, airspeed, meaning that the effectiveness's matrix will dynamical change over time. As it is a square matrix it can be easily inverted so the actuator settings are then obtained by solving the following equation.

$$\begin{bmatrix} w_1^2 \\ w_2^2 \\ \delta_a \\ \delta_e \end{bmatrix} = G^{-1} \begin{bmatrix} F_x \\ \tau_x \\ \tau_y - \bar{q}SE_x(C_{p_0} + C_{p_\alpha}\alpha) \\ \tau_y \end{bmatrix} \tag{5.7}$$

5.1.3 Transition

A big step in this work is to design a control allocation technique dedicated to the transition phase of a hybrid UAV. In this phase, there are peculiar dynamics that need to be

taken into account to have an effective control allocation. Most third party autopilots ignore this phase and have a hard coded transition setup that in most cases only allows the transition in hover mode, which is not the objective of this work, being it the possibility for a transition in any state of the system. In this phase the main objective is to bring the tilt rotor angle to a fixed position, 90° for a vertical to horizontal flight transition and 0° for a horizontal to vertical flight transition in the shortest time possible. Not going too deep on this subject, that will be discussed in the next chapter. When a transition is commanded, both tilt servo motors will have a fixed desired angle, meaning that they will not serve as control actuators on this phase. Therefore, the remaining five actuators, $w_1, w_2, w_3\delta_a$, and δ_e will be responsible for the 4DOF system. Another point to consider is the thrust input. On previous flight modes thrust was orientated in the z_B or x_B direction, as the rotor tilt angle will now constantly change, the direction of thrust follows the same direction. So for this allocation we assume the thrust input is already oriented in the actual direction of the tilt angle,$\bar{\alpha}$.

$$\bar{\alpha} = \frac{\alpha_1 + \alpha_2}{2} \tag{5.8}$$

The thrust force vector will be designated by $\vec{F_{zx}}$ so that:

$$\begin{aligned} \left\| \vec{F_{zx}} \right\| &= u_t \\ \theta_F &= \bar{\alpha} \end{aligned} \tag{5.9}$$

With this, the effectiveness matrix can be deduced by combining both the tri-rotor and fixed wing torque equations ending up with:

$$G = \begin{bmatrix} k_F & k_F & k_F cos(\bar{\alpha}) & 0 & 0 \\ r_{1y}k_F cos(\alpha_1) + k_M sin(\alpha_1) & r_{1y}k_F cos(\alpha_2) - k_M sin(\alpha_2) & 0 & \bar{q}SE_y C_{r\delta} & 0 \\ r_{1x}k_F cos(\alpha_1) & r_{2x}k_F cos(\alpha_2) & r_{3x}k_F & 0 & \bar{q}SE_x C_{p\delta} \\ r_{1y}k_F sin(\alpha_1) - k_M cos(\alpha_1) & r_{2y}k_F sin(\alpha_2) + k_M sin(\alpha_2) & k_M & 0 & 0 \end{bmatrix} \tag{5.10}$$

Applying the same process, inverting G, the actuator settings are calculated by solving equation 5.11.

$$\begin{bmatrix} w_1^2 \\ w_2^2 \\ w_3^2 \\ \delta_a \\ \delta_e \end{bmatrix} = G^+ \begin{bmatrix} \left\| \vec{F_{zx}} \right\| \\ \tau_x \\ \tau_y \\ \tau_y \end{bmatrix} \tag{5.11}$$

With this, all three matrices are defined. In the next chapter, they will be used in conjunction to the hybrid controller, to simulate the drone's transitions on the mathematical model.

Chapter 6

Hybrid autopilot controller

The main goal of this chapter is to design a hybrid controller capable of dealing with the three different UAV flight modes, horizontal, vertical and transitional flight. To achieve this the controller must be capable of automatically interchange its scheme based on discrete values. On the previous chapters three different methods were used to allocate the actuator settings, as well as three controllers methods, based on the desired flight mode. In this chapter we will design a hybrid control system that is able to commute amongst them in any system state without creating major flight disturbances. The success of this task will be measured mainly with the position error rise that can occur during transitions.

Most autopilots ignore the transition phase completely and just have a hard coded process that commands the tilt servo motors to the desired position without taking feedback on the current system state. The goal of this chapter is to achieve seamless transition without them being controlled by human operators or by trajectory waypoints. The transition between flight modes is the controller decision, taken based on both trajectory reference and system state, meaning that based on the reference trajectory the controller will decide on which flight mode is best suited for and if needed command the UAV to perform a smooth transition.

6.1 Method overview

The objective of the hybrid control is to accommodate, in this case, three different control schemes (horizontal, vertical, and transitional) in a single controller. Although both separate schemes will work alternating between each other. They will not in any case be used simultaneously. For this reason, methods of hybrid control and modeling such as HYSDEL,[15] or MINLIP, [17] bring few advantages over more generic controllers.

Therefore we chose to use a finite state machine, FSM, to handle all discrete values and logical operations. With them the FSM generates an output signal that serves as a command to switch between control schemes. The FSM can be designed to fit any control method. For this work, we purpose a control based on velocity, the idea is that depending on both the reference velocity and the actual velocity, the controller is able to adopt the most suitable flight mode for the case and if, needed perform a transition. The base idea is shown in Figure 6.1.

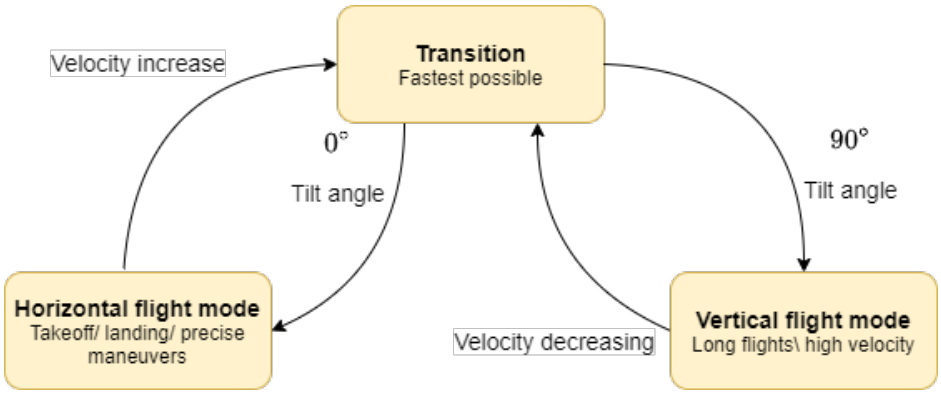


Figure 6.1: Proposed control method block diagram

As mentioned, the chosen method to commute between flight modes is based on velocity, being this the main purpose of having a hybrid UAV. Vertical flight can´t achieve very high speed and, on the other hand, horizontal flight mode can't be used in low speeds due to loss of aerodynamic lift. So a mixing between the reference velocity magnitude and the actual airspeed will act as control variables for the FSM. The last variable is the determinant factor on defining flight modes, the rotor tilt angle, as we assume that a 90° angle corresponds to the horizontal flight mode and an approximate angle of 0° for the vertical flight mode. In sum, the input control variables used for the design of the FSM are:

- $\|\mathbf{v_{ref}}\|$: Reference velocity magnitude, this provides knowledge of the reference trajectory to the FSM in order to determine witch flight mode is most viable for the desired flight path;
- $\mathbf{v_{air}}$: The airspeed, determines the air flow speed on the wings, the crucial factor for generating lift and, therefor, enabling the possibility of horizontal flight mode. This will ensure weather or not a transition can be performed;
- $\bar{\mathbf{a}}$: Rotor tilt angle. When in transitional flight, the tilt angles determines whether the transition ended or is ongoing.

With these control variables, an algorithm was designed to commute between the three possible states, S_1, S_2, S_3, representing each flight mode.

1. $\mathbf{S_1}$: Vertical flight mode, also the entry state, S_0;

State-Transition table			
Input \ Current state	$\mathbf{S_1}$	$\mathbf{S_2}$	$\mathbf{S_3}$
$\|\|v_{ref}\|\| > v_{rt} \wedge v_{air} > v_{vt}$	S_3		
$\|\|v_{ref}\|\| < v_{rt} \wedge v_{air} < v_{ht}$		S_3	
$\bar{a} = \frac{\pi}{2}$			S_2
$\bar{a} = 0$			S_1

Table 6.1: Event based FSM table

2. $\mathbf{S_2}$: Horizontal flight mode;

3. $\mathbf{S_3}$: Transitional flight mode.

Also, one must define velocity threshold values in which the UAV will be commanded to perform a transition. Before this, it's important to note that the horizontal flight mode only operates in certain velocity ranges, where the lift force in enough to compensate for the gravitational force. Equation (4.29) gives us the stall velocity for this type of vehicle configuration, so it is important to note that horizontal flight mode can never be operate below this value and also that enabling vertical flight mode near this velocity range is a wasteful solution. The stall velocity gives us good insights of the velocity ranges of both flight modes and when a transition is viable. For this design three velocity threshold values are used, v_{vt}, v_{ht}, v_{rt}, vertical, horizontal and reference velocities respectively, all can be adjusted to the designer needs, but always inside the stall velocity rule

$$v_v < v_{st} \wedge v_h >> v_{st} \tag{6.1}$$

where v_v, v_h are vertical and horizontal flight velocity. For the current design, in respect to the UAV characteristics, the values used are

$$\begin{aligned} v_{vt} &= 6m/s \\ v_{ht} &= 10m/s \\ v_{rt} &= 12m/s. \end{aligned} \tag{6.2}$$

The event table presented in Table 6.1 shows all of the possible transitions of the FSM, using the three inputs available. The states represented in it are the possible transition. Corresponding to the previous table, the FSM was designed according to the diagram presented in Figure 6.2 and implemented as the hybrid controller.

.

6.2 Transition Simulation

After completing the design of the hybrid controller, it can be tested using the mathematical model previously used. In this section two separate simulations are shown performing

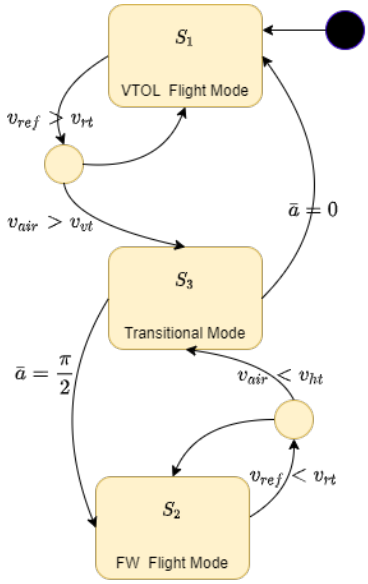
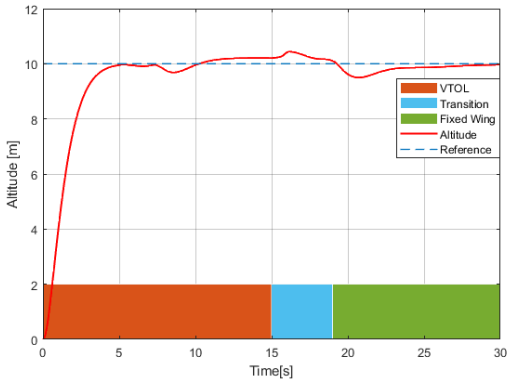
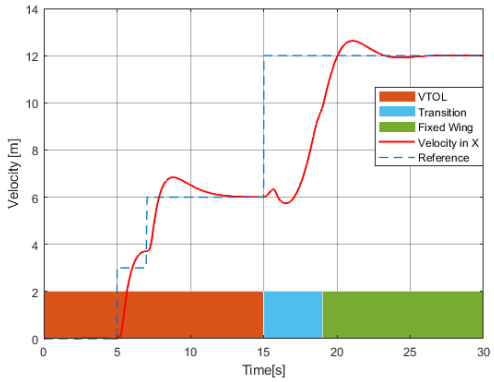


Figure 6.2: FSM diagram



(a) Altitude over time



(b) Velocity in x_W axis direction over time

Figure 6.3: Altitude and Velocity measurements during a positive transition simulation

both positive (vertical to horizontal) and negative (horizontal to vertical) transitions. We expect these transitions to be smooth, without velocity or altitude oscillations and, as a key point, a low altitude drop is a must for this hybrid controller to be successful.

The first simulation procedure will test the performance of the hybrid controller on a **positive transition**, from vertical to horizontal flight mode. The UAV will start at the origin, climb to a stable hover point and then gradually raises its velocity. Set-points will be used for the reference trajectory and then, we observe the UAV behavior when transitioning from VTOL to fixed wing flight. Note that the only human operator input is the pre-defined reference trajectory the hybrid controller handle everything else.

Figure 6.4 shows us, as expected at $t = 15\ seconds$ where the reference velocity exceeds the threshold value v_{rt}, the UVA begins a transition to the horizontal flight mode. Around

$t = 19$ *seconds* the transition is completed, meaning that the rotor tilt angle $\bar{a}$ is fixed at 90°, which is confirmed in Figure 6.4. From there, the UAV is flying steady in the horizontal flight mode achieving the desired velocity of $12m/s$. A good performance evaluation of the hybrid controller performance is the altitude error. As mentioned before, the main objective is to have smooth transitions between flight modes, and one of the most important variables in this type of flight is the altitude drop, minimizing this value becomes one of the main objectives. In this transition, the altitude error was kept between $0.5 < e_{p_z} < 0.5$ *meters*, achieving a maximum value of 0.47 *meters*, a relative low value when compared to the actual altitude this mode is designed to operate. The remaining actuator settings are showed in Figure 6.5

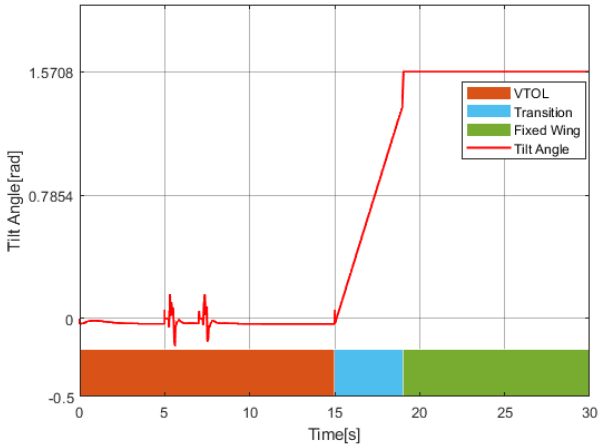
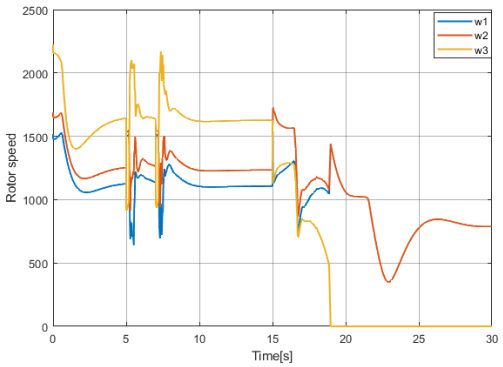
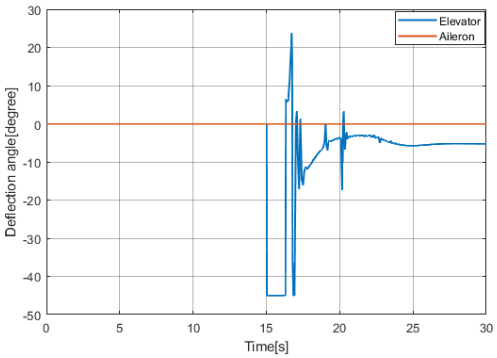


Figure 6.4: Tilt angle $\bar{a}$ during a positive transition simulation

Proceeding with the simulation process, next a **negative transition** was simulated with the UAV, starting in the horizontal flight mode at a reference constant velocity. At $t = 10$ *seconds* a negative step in the reference velocity occurs, bringing the reference value lower than the threshold value, $v_h t$. The UAV then is forced to reduce its speed and



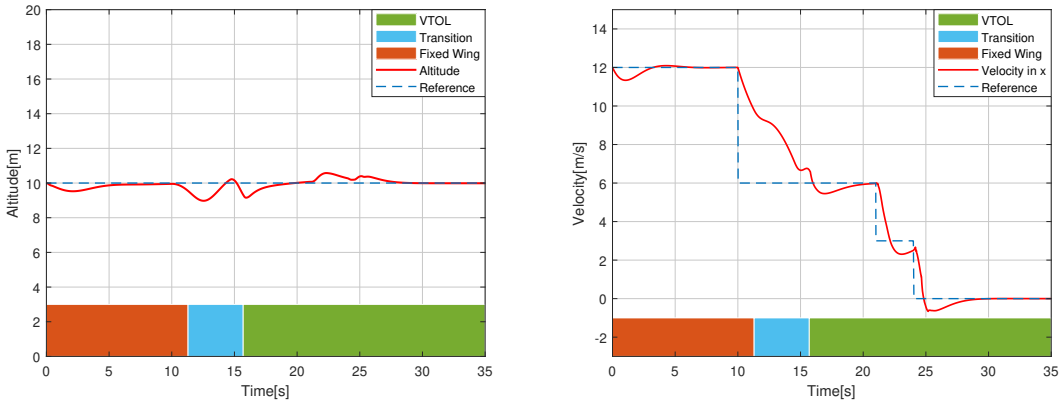
(a) Desired rotors angular speed



(b) Desired elevator and aileron deflection angle

Figure 6.5: Actuator settings during a positive transition simulation

perform a transition to vertical flight mode because of the major lift fore reduction due to the velocity decrees. Again, this transition is purely autonomous, being the only human input, the pre-planed reference trajectory.



(a) Altitude over time

(b) Velocity in x_W axis direction over time

Figure 6.6: Altitude and Velocity measurements during negative transition simulation

This time, the transition although expected, induced a higher altitude drop, figure 6.6, achieving a minimum value of 1 *meters* below the reference altitude. This shows a struggle on the controller side when transiting to rotary wing mode caused by two factors: the changing tilt angle and the lift force oscillation, due to the velocity decrease (also the AoA changes rapidly). The thrust component will struggle due to high side drift, causing a higher altitude drop that is still relatively low.

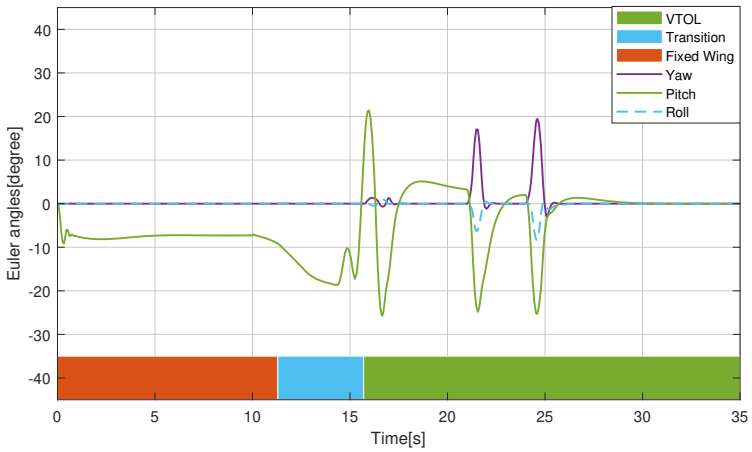
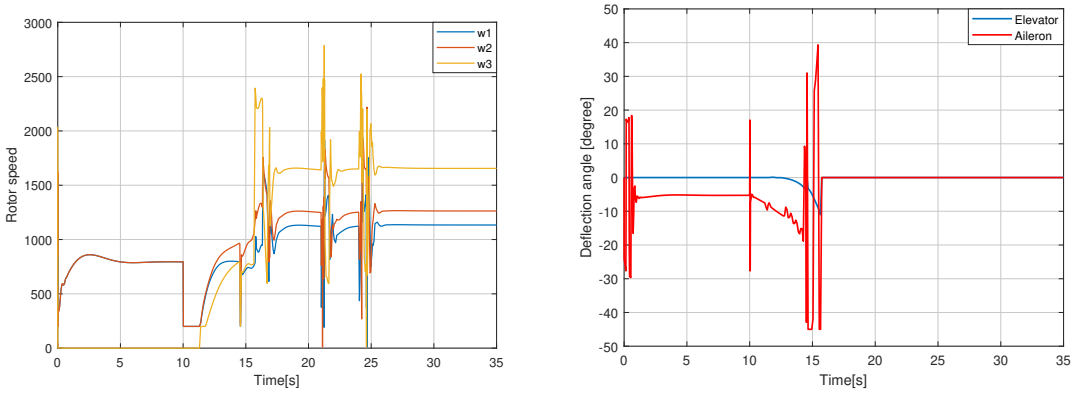


Figure 6.7: Euler angles during a negative transition simulation

As for the UAV velocity, we can notice here a slower deceleration curve, being the wind resistance the only force acting on it in fixed wing mode. And again, near the transition end, $t = 15.7$ *seconds*, the oscillations become more noticeable. This can happen due to a side effect of the pitch angle readjusting. In horizontal flight, mode the pitch tends to

be a few degrees over the xy world frame plane in order to raise the angle of attack. On the other hand in vertical flight mode the pitch is below the reference level so the rotors generate force in the x_W axis enabling the UAV to maintain its velocity. In this scenario the change occurs at a speed that the lift forces are still high enough to be noticeable, with the changing angle of attack the UAV becomes more unstable. A good solution for this would be to have dynamic rotation gains in the hybrid controller so that the UAV had lower toque forces near transitions than in normal conditions.



(a) Desired rotors angular speed

(b) Desired elevator and aileron deflection angle

Figure 6.8: Actuator settings during a positive transition simulation

Chapter 7

SITL simulations

Before real-world testing, as errors can be costly in this type of systems, a software-in-the-loop simulation is the best way to verify the proposed control system in a high precision simulated virtual environment. For this purpose, Gazebo a powerful 3D simulation environment, will be used in conjugation with PX4 autopilot. This is extremely important as PX4 is the same autopilot software that will be used in real-world testing. Being a very versatile open-source system that is possible to aggregate to any kind of UAV, with a custom control option using pre-made PX4 communication protocols and hardware commanding options. Although being easy to operate, a set up process needs to be complete to enable such kind of operation, mostly communications.

7.1 Simulation Setup

Px4 uses MAVlink communication protocol to send and receive data to a off-board API. We will be using ROS (Robot Operating System), so that the drone telemetry and input commands can be handled outside of PX4 itself. Figure 7.1 shows the communication scheme used in this simulation.

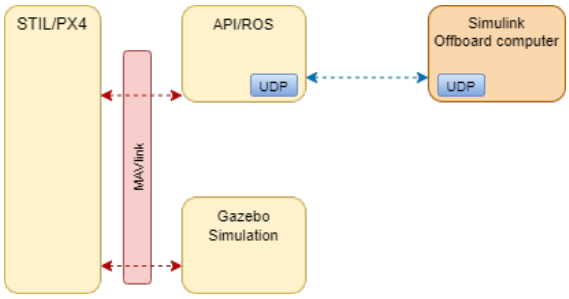


Figure 7.1: Communication scheme

The process is using UDP communication protocol enabling it to be operated remotely from an outside environment, *simulink* in this case. In *Matlab/simulink*, the control commands are packed and sent through UDP, using *simulink's* custom socket, to an off-board computer running ROS that for it self is running a custom UDP *winsocket* client in C++. In ROS, a custom node handles the communications with PX4 enabling off-baord mode [30].

Despite this simulation being as close to reality as one could want , in our case this comes with a few constraints. Mainly the Gazebo simulation does not yet support the vehicle category that this work is based, tilt tri-rotor. This could be overcome with a addition of a custom made airframe, but in this case it would require some additional work that is not under the scope of this thesis. Instead, a similar vehicle type is supported, a tilt quad-rotor, that basically has the same functionalities with just an addition of one extra tail rotor. Even though the difference is almost minimal, this fact makes the control allocation proposed in the previous section unusable due to the different actuator settings, such vehicle (quad-rotor) uses. This fact, will not allow for the validation in SITL simulations of some of the major contributions of this thesis, leaving only the option of real world testing.

Despite this setback, it is possible to validate the proposed controller concept, in a powerfull and realistic simulated environment. Thus one options remains, it's possible to consider the controller design loops up to the computation of the desired rotation, R_{des} and the thrust u_t. With this, converting the desired rotation matrix to quaternion, q_{des}, form and inputting the new control command, $u = [u_t, q_{des}]$ to PX4 framework. This method works for the simple reason that, despite the difference in actuator control, the way both vehicle types perform movement by changing their body rotation in both flight mode, is still exactly the same, meaning that with the same rotation the force produced and induced(aerodynamics) is very similar. With this method we are able to simulate most of the design control modules leaving behind the proposed control allocation.

7.2 GAZEBO environment simulations

The first step, made to ensure the proposed simulation method could function, was to verify both controllers setups separately in order to validate not only the proposed control method but also the simulation setup. The process is similar with the one made in Chapter 4 using roughly the same reference trajectory.

Starting in VTOL flight mode, the results of the simulation can be seen in Figure 7.3. A waypoint based reference trajectory with a duration of 35 *seconds* was used to verify all possible combination of moments and forces. The results are very good, with the UAV getting to the waypoints with little to no overshoot and without any major oscillations.

The 3D trajectory contains some small disturbances expected in such complex simulations tools, where drag and wind disturbances are close to the real world environment, still the results are favorable.

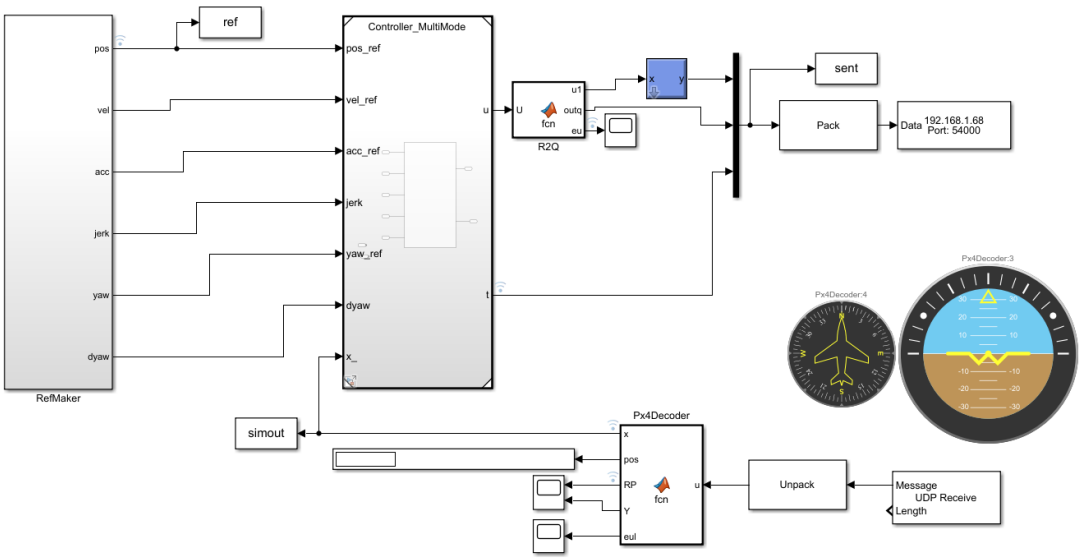


Figure 7.2: Simulink diagram with UDP implementation

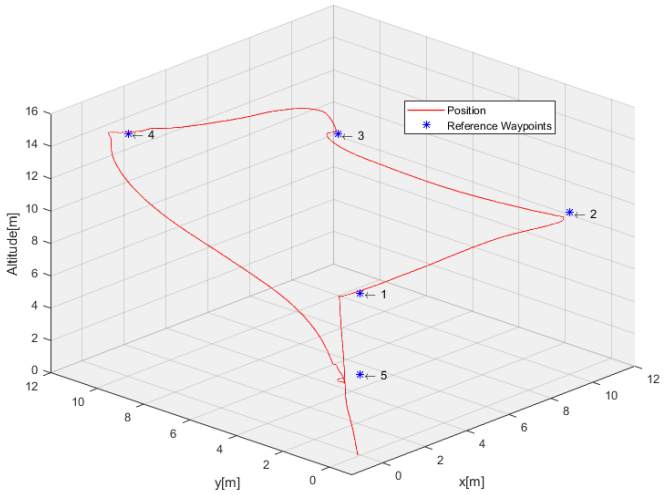


Figure 7.3: VTOL 3d Trajectory

Next, we tested the fixed wing flight controller with a semi-square reference trajectory. The simulation occurred in a 65s time frame with the vehicle starting on a stable flight position and maintaining an average airspeed of 15m/s. Figure 7.4 shows the 3d plot of the simulation from a high level point of view. The UAV was able to follow the reference trajectory relatively well, while also following the reference altitude changes without any problem. Figure 7.5 shows the output of the controller. As discussed before, the controller sends the desired Rotation to PX4, that rotation is represented here in Euler angles, both the output and actual rotation.

The pitch angle shows a greater disturbance, although in small values, (note that the pitch angle varies in a range of only 4 *degrees*) it is still noticeable. Even so, the disturbance was not noticeable in the simulation, as the altitude was maintained without any major issue. In $t = 35$ *seconds* a peak value is obtained due to the change of the reference altitude from 10m to 15m;

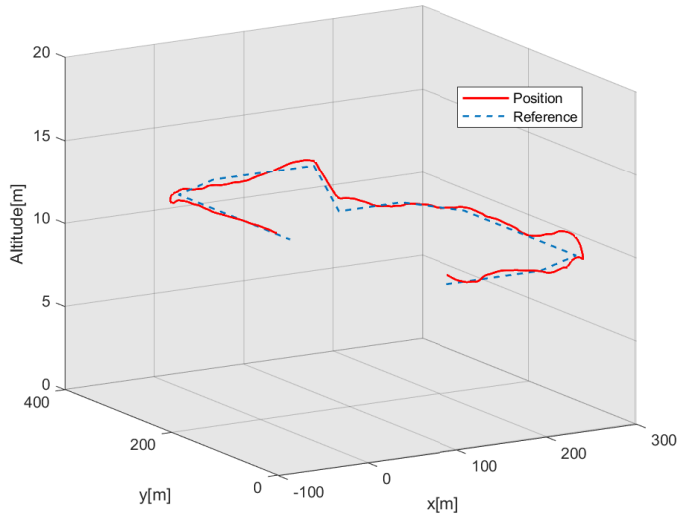


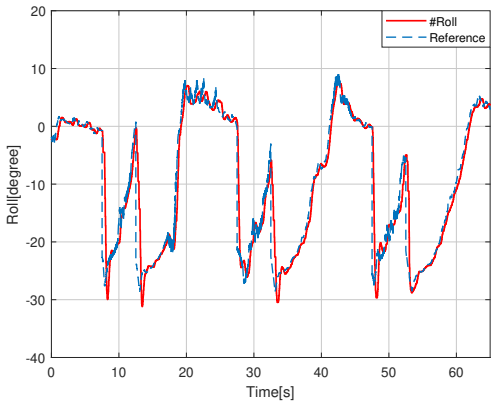
Figure 7.4: Fixed-wing 3d Trajectory

It is then confirmed that both controllers are performing as expected, we can now proceed to the main objective of this work: flight mode transitions.

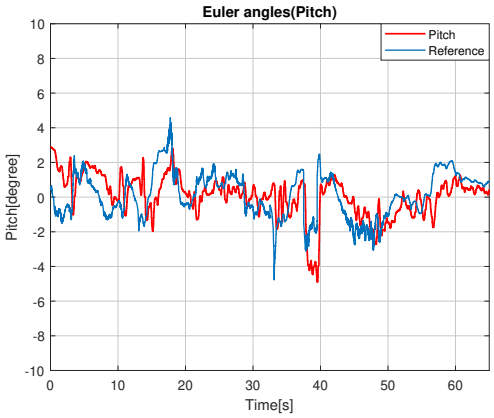
7.2.1 Hybrid control simulations

Before proceeding with the transitions simulations, a few things need to be addressed:

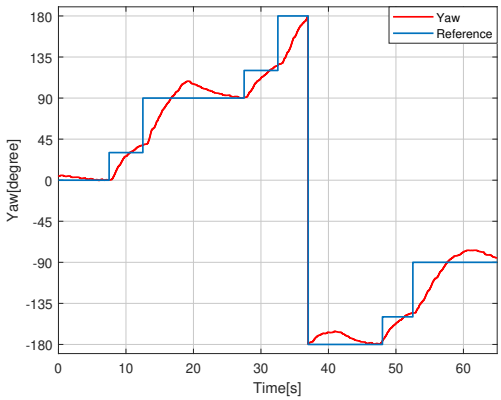
- As pointed before, the Gazebo environment does not support any tri-rotor hybrid UAV, this means that the designed control allocation in Chapter 5 will not be used in these simulations, throwing away important features built during this work. Instead, the control loop will stop when the desired rotation, R_{des}, is calculated, then it is transformed into a quaternion and sent to PX4 along side the thrust component. The torques and actuator settings used in the mathematical simulations will not



(a) Roll angle over time



(b) Pitch angle over time



(c) Yaw angle over time

Figure 7.5: Rotation angles inputs(converted to Euler angles)

be used here. This problem could be overcome by designing a custom vehicle and adding it to the Gazebo library, but this is no easy task and is not within the scope of this work;

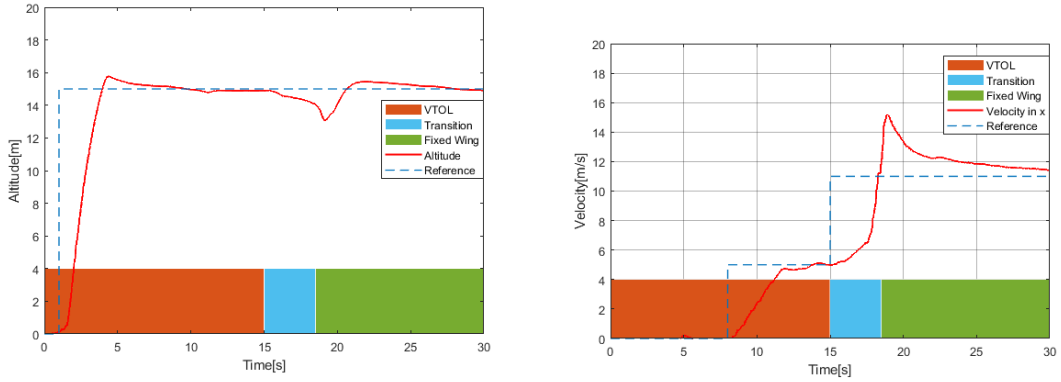
- For this simulation the vehicle used is a quad tilt-rotor UAV, having this configurations a reasonable identical system dynamics, it is able to have the same flight modes and perform identical transitions. The major difference being the four rotors in X configuration instead of three rotors in Y configuration;

- These simulations were performed using PX4 off-board attitude set-point control, meaning that our controller will not directly command the actuator settings. The built in PX4 mixer handles the transformation between the desired attitude and the actuator settings. This creates a problem in transitions, to perform a transition a different control logic message is sent to PX4, from that point the built in PX4 attitude controller handles the drone until the transition is complete. The only off-board input accepted in this transitional mode is the thrust command. This, will result that the hybrid controller made in this work will not be used during this phase,

only the thrust command that generally is coupled to the altitude will be accepted by PX4. The solution for this problem is to control the UAV by RC commands, commanding the actuator settings directly. But again the Gazebo environment does not support the vehicle configuration we aimed for, leaving only real world tests possible to fully operate the UAV with the developed hybrid controller.

Both simulations were performed using similar reference trajectory's, the UAV will start at a fixed altitude set-point and will be commanded to speed up, performing a positive transition, or to slow down, performing a negative transition, both by changing the desired velocity set-point.

Figure 7.6 shows the **positive transition**, starting from take off in vertical mode to gradually increase the velocity and then performing a transition to horizontal flight mode. The transition resulted in an altitude drop of approximate 2 *meters*, a relative high value when compared with the mathematical simulations, and continuous increase in velocity although producing some overshoot not caused by the velocity set-point but by the reference position, still the velocity is quick to settle in the reference value after the transition is ended.



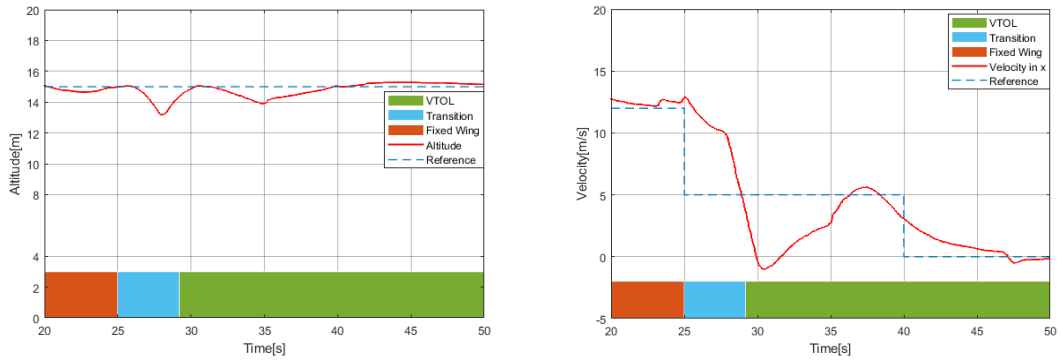
(a) Altitude over time

(b) Velocity in x_W axis direction over time

Figure 7.6: Altitude and Velocity measurements during positive transition SITL simulation

For the negative transition, the results obtained were not the expected due to the problems described before. In particularly when the PX4 is commanded to perform a transition to vertical flight mode, it assumes that the transition ends when the UAV halts in a hover flight, $v = 0\ m/s$, this erases the purpose of the designed hybrid controller. This fact is proven in Figure 7.7, as the UAV is forced to drastically reduce its velocity to perform the transition and then rises back to match the desired velocity. Note that reference trajectory ends with a hover position in vertical flight mode resulting in the ramp depicted in Figure 7.7 .

Despite the setbacks presented before, taking part of the development work made for this work out of the equation for these simulations, it was still possible to confirm that



(a) Altitude over time

(b) Velocity in x_W axis direction over time

Figure 7.7: Altitude and Velocity measurements during negative transition SITL simulation

the controller scheme is functional and provides the expected results. As for the other control components they were validated previously with the mathematical model, and shown good results as well. Altogether it is safe to assume that the final control method presented here,with all is components, is capable of performing mid flight transition as it was intended for.

Chapter 8

Instrumentation

By instrumentation, we mean to reconfigure the hardware and software of a hybrid UAV in order for it to be fully operational and serve our needs. In this case, this means to convert a commercially available hybrid UAV that comes with its own autopilot controller, dismantle it and assemble the a Pixhawk so that it will be in accordance to the PX4 framework.

As previously discussed in this work, the E-Flite convergence mini present in Figure 8.1 will be used for this purpose, as most of the modeling and control allocation work was made specific for this UAV, although being highly modular.

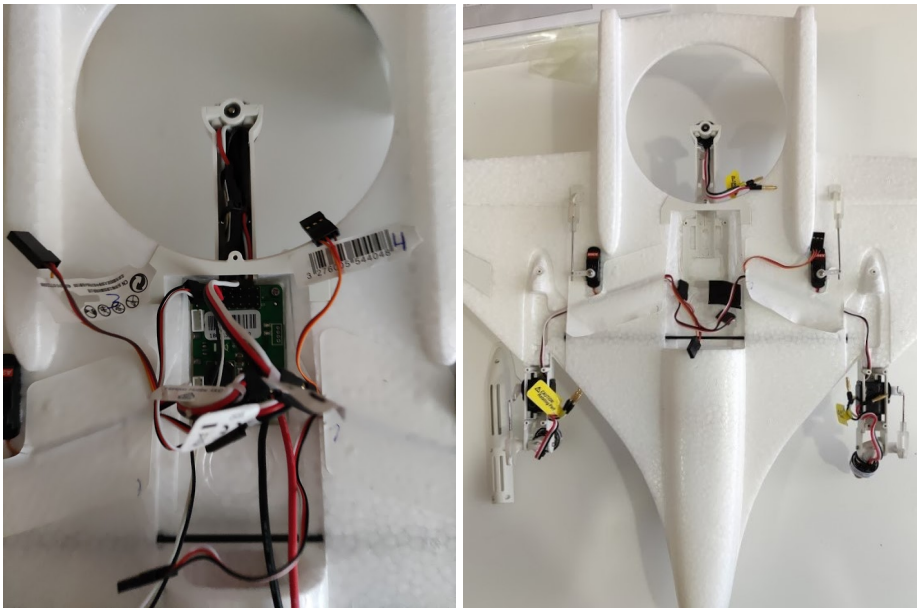
Figure 8.1: E-Flit Convergence mini

This type of conversions is fairly common in the developer aeromodelling world, still,

for this case the work will not be as easy. The convergence mini, is a smaller and underpowered model, with only a wingspan of 410mm and three 6A ESCs(Electronic speed control). When normally this type of controller kit is applied to UAVs using 30A ESCs. With this, the weight and COM shift will be taken into account during this procedure. By the end of this chapter, we should have a fully operational hybrid UAV capable of autonomous and manual flight modes.

8.1 Pixhawk Conversion

To begin with the convergence, first the UAV was stripped in order to remove the OEM(Original Equipment Manufacture) autopilot. This is the only part that will be removed since all the rotors, servos and ESCs will still be used by the Pixhawk controller. This step revealed the lightness of the original control board, weighing around 15 *grams*, a big change when compared to the 37.2 *grams* of the Pixhawk 4 mini plus all the rest of the external components.



(a) Original autopilot control board (b) Convergence control board removed

Figure 8.2: Removing process

After this procedure, it is now possible to star installing all the necessary Pixhawk components. Before demonstrating the final result, let us review the parts used and the reason why.

There are several autopilots compatible with the PX4 frame work, due to availability. For this work it was used the Holybro Pixhawk 4 Mini autopitot system, shown in figure 8.3. It provides all the necessary requirements to enable us to use the PX4 framework

both with an internal or external flight controller system. This autopilot isn't a standalone piece, for it to be fully operational the following components will also be installed in the convergence:

Figure 8.3: Pixhawk 4 Mini[31]

1. **Pixhawk**: The autopilot controller provides all the necessary computing power with an internal FMU processor, and acts as an IMU having one barometer, one Magnetometer and a pair of accelerometer/gyroscopes. It will provide all the necessary PWM control signals;

2. **Power Management Board**: A power distribution board provides the regulated power to the ESCs and to the autopilot. The input of the $800mhA$ battery;

3. **GPS module**: Provides the autopilot with the GPS data, also contains a internal compass;

4. **Battery elimination circuit**: In order to power the servos motors, this circuit is used to provide a regulated $5V$ power source to the servos;

5. **Radio control receiver**: For safety and testing measures, the UAV will also be controllable through a manual transmitter, this component receives the transmitted RC signal and directs it to the autopilot;

6. **Telemetry module**: Enables the communication between a companion computer and the drone while in flight.

To assemble and connect all of the previously components, the standard schematic shown in Figure 8.4 was followed. After completing the wiring connection outside the

vehicle, all of the components were installed in the UAV, for this some structural modifications were made to it so it could accommodate all of the extra components without compromising the center of gravity and aerodynamics. The end result is presented in Figure 8.5.

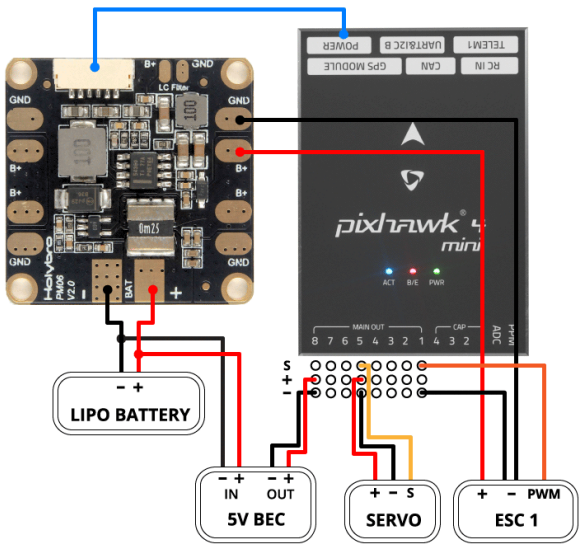
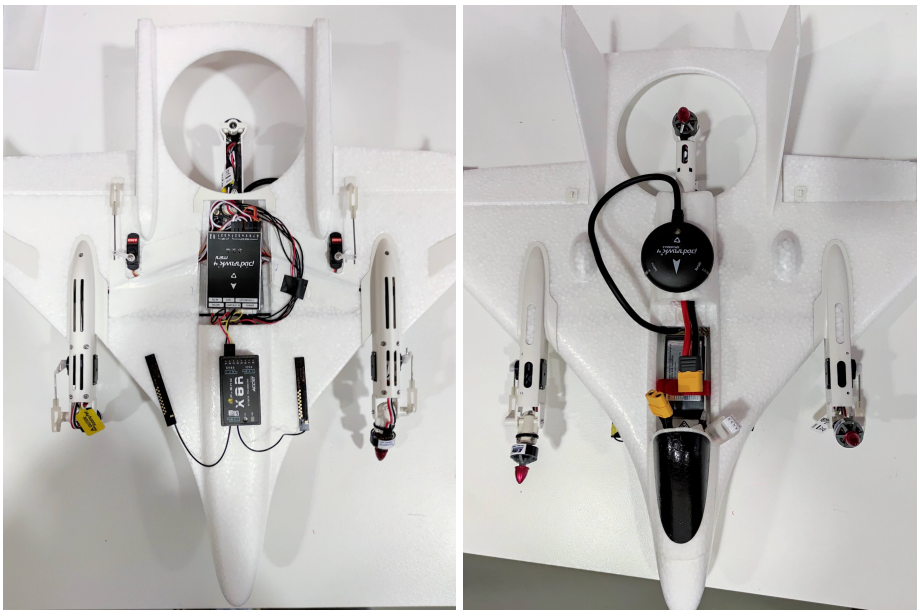


Figure 8.4: Power management high level schematic[32]



(a) Convergence with Pixhawk bottom view

(b) Convergence with Pixhawk top view

Figure 8.5: Convergence with Pixhawk

8.1.1 Set Up and Communication

With the conversion complete, a set up process was made in order to calibrate all of the components, especially the IMU. To do so, QGControll software was used, it is specifically made to work with MAVlink enable UAVs. It provides all of the necessary configurations and will be the link between PX4 and the UAV. There are now two possible ways to communicate and control the UAV from this point, they are:

1. **Manual**: With the RC receiver connected to the autopilot we can enable manual flight mode. The use of this mode is only for safety and testing reasons. It works with a conventional RC transmitter that connects to the receiver;

2. **Autonomous**: With the MAVlink communication protocol, the drone can enter in autonomous mode and the control signals that are generated in a companion computer are sent with MAVlink to the drone via the telemetry radio module. This will be the main communication method, being it the one that will allow us to send PWM direct input signal to the autopilot. This way, not using any preconfigured flight controller software.

With the instrumentation process completed, the vehicle was subjected to quick indoor tests, to check whether all the system were functionally. Firstly by, using RC transmitter in manual flight mode, all actuators showed the expected response, rotors and control surfaces(servos). By changing the UAV orientation manually, all the actuator settings adjusted accordingly, this confirms that the vehicle data is being processed by the autopilot and sent to the actuators accordingly.

The next step was to properly test flight the UAV in both manual and autonomous mode. Due to time and space constrains, no further tests were possible, the vehicle remains fully functional and ready for autonomous control.

Chapter 9

Conclusions

From an generalised perspective, the general goal of this dissertation was to design a versatile controller that was capable of control a hybrid UAV autonomously to its full capabilities using both flight modes and performing seamless transition between them. This was possible by combining techniques of different controller categories and gathering them through a fully operational hybrid controller. Part of this was made possible by implementing innovative techniques suchlike hybrid control allocation in Chapter 5 and a hybrid logical control system in Chapter 6.

The goal was firstly separated into two stages: modeling and control. The first stage, modeling, has the same importance as control and was tackled with great expectations. Although the mathematical model focused on one hybrid UAV category, it can be easily modeled to accommodate any type of small scale aircraft. This is possible by the generalised equation of motions presented in Chapter 3, that account for a variety of motion effects present in such hybrid systems, like gyroscopic effects and aircraft like aerodynamics. The final model is able to receive raw actuator settings and with them provide the resulting movement and rotation subject to the tilt tri-rotor hybrid UAV.

In the second stage, we started by designing a separate control systems for both flight modes, each one sharing the same bases but, providing different results as necessary due to the flight style changing the dynamics of the system drastically. After that, both controllers setups were validated against the designed model, providing great tracking ability in both flight modes, validating then both model and controller. Note that both controllers are versatile and can achieve different results depending on the desired control profile by changing the respective gains. In this work, the vertical flight controller was tuned to achieve great precision with a waypoint based trajectory, as for the horizontal flight mode, is more subjective to velocity steps trajectories. After that, Chapter 5

describes innovative methods of hybrid control allocation, specific for the drone in question. All the methods are meant to be interchangeable and aimed to get the most out of any actuator that has an important effect for the current flight state. Finally, all of the designed control methods are joined together to crate a versatile hybrid controller shown in Chapter 6. Using a finite state machine, the controller showed great capability when transitioning between states autonomously in the mathematical model.

In addition, all of the work described before was then subjected to a powerful simulation tool, GAZEBO. Here, despite the restraints already discussed, both controllers were simulated separately having achieved the expected results with good tracking ability, despite the larger position error window such simulations showed. Unfortunately, GAZEBO coupled with PX4 made the restraints more noticeable in the full hybrid control transition simulations. Here, the results were not the expected but still relatively good, being the drone able to perform both transition with a low altitude drop. The reason being, the restraints discussed in Chapter 7. Still both transitions were possible, controlled by the designed hybrid controller and both showed great potential for improvements if the full fledged system was used. The designed system will need further testing but it's important to notice that it already showed that is possible to have a complete autonomous hybrid UAV performing the diverse maneuvers needed in parcel relay systems, opening the way for hybrid drones to takeover.

9.1 Future Work

Despite the good results obtained, this work creates a desire for further testing that could be done in future researches. For better simulations, one could pursue this path, and design a custom vehicle for the GAZEBO environment with the characteristics of the E-flite convergence. This would enable in-depth simulations to be performed using all the components designed for this work, further validating the system.

A second path can be taken, real world testing. For that, the work is already underway, with the instrumentation completed and validated.The vehicle is fully functional to perform with real world tests, starting by slow velocity tests in hover mode and ending with full mid-flight autonomous transitions. Despite being ready, this procedure will still need some time invested, in order to apply and adjust control gains to ensure the absolute best performance.

Another useful implementation would be to design an optimal trajectory generation system, using MPC control algorithms. Taking advantage of the specifications of the vehicle characteristics, mainly the flight modes, and tuned to run alongside the proposed controller, it would provide an efficient flight path to the system, enabling it to have smother and timely transitions.

Bibliography

[1] *HorizonHobby Convergence Web-page*. Accessed: 2020-04-20. URL: https://www.horizonhobby.com/product/convergence-vtol-pnp-650mm/EFL11075.html.

[2] *REPLACE project*. Accessed: 2020-04-20. URL: https://www.brunojnguerreiro.eu/pt/research/replace.

[3] *DJI Mavic Pro Platinum*. [Online, cited 5 October 2020]. URL: https://pt.wikipedia.org/wiki/Quadrotor.

[4] *UAV Argentine Air Force*. [Online, cited 5 October 2020]. URL: https://commons.wikimedia.org/w/index.php?search=UAV&title=Special\%3ASearch&go=Go&ns0=1&ns6=1&ns12=1&ns14=1&ns100=1&ns106=1#/media/File:Uav_aukan.jpg.

[5] *Heurobotics MKII*. [Online, cited 5 October 2020]. URL: https://www.suasnews.com/2017/04/heurobotics-variable-pitch-tailsitter.

[6] *Arcturus JUMP 15*. [Online, cited 5 October 2020]. URL: https://arcturus-uav.com/product/jump-15.

[7] A. S. Saeed, A. B. Younes, C. Cai, and G. Cai. "A survey of hybrid unmanned aerial vehicles." In: *Progress in Aerospace Sciences* 98 (2018), pp. 91–105.

[8] C. Papachristos and A. Tzes. "Modeling and control simulation of an unmanned tilt tri-rotor aerial vehicle." In: *2012 IEEE International Conference on Industrial Technology*. IEEE. 2012, pp. 840–845.

[9] D. A. Ta, I. Fantoni, and R. Lozano. "Modeling and control of a tilt tri-rotor airplane." In: *2012 American control conference (ACC)*. IEEE. 2012, pp. 131–136.

[10] D. Cabecinhas, R. Cunha, and C. Silvestre. "A nonlinear quadrotor trajectory tracking controller with disturbance rejection." In: *Control Engineering Practice* 26 (2014), pp. 1–10.

[11] F. Kendoul, I. Fantoni, and R. Lozano. "Modeling and control of a small autonomous aircraft having two tilting rotors." In: *IEEE Transactions on Robotics* 22.6 (2006), pp. 1297–1302.

[12] P. Ru and K. Subbarao. "Nonlinear model predictive control for unmanned aerial vehicles." In: *Aerospace* 4.2 (2017), p. 31.

[13] M. Kamel, M. Burri, and R. Siegwart. "Linear vs nonlinear mpc for trajectory tracking applied to rotary wing micro aerial vehicles." In: *IFAC-PapersOnLine* 50.1 (2017), pp. 3463–3469.

[14] J. Lygeros. "Lecture notes on hybrid systems." In: *Notes for an ENSIETA workshop*. Citeseer. 2004.

[15] F. D. Torrisi and A. Bemporad. "HYSDEL-a tool for generating computational hybrid models for analysis and synthesis problems." In: *IEEE transactions on control systems technology* 12.2 (2004), pp. 235–249.

[16] M Kvasnica, M Herceg, M Morari, S Gaulocher, and J Poland. *HYSDEL 3.0–Manual*. Tech. rep. Tech. rep. Slovak University of Technology in Bratislava, ETH Zurich, ABB, 2008.

[17] N. V. Sahinidis. *Mixed-integer nonlinear programming*. 2019.

[18] J. Kronqvist, D. E. Bernal, A. Lundell, and I. E. Grossmann. "A review and comparison of solvers for convex MINLP." In: *Optimization and Engineering* 20.2 (2019), pp. 397–455.

[19] G. Hoffmann, S. Waslander, and C. Tomlin. "Quadrotor helicopter trajectory tracking control." In: *AIAA guidance, navigation and control conference and exhibit*. 2008, p. 7410.

[20] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin. "Precision flight control for a multi-vehicle quadrotor helicopter testbed." In: *Control engineering practice* 19.9 (2011), pp. 1023–1036.

[21] T. J. Koo and S. Sastry. "Output tracking control design of a helicopter model based on approximate linearization." In: *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No. 98CH36171)*. Vol. 4. IEEE. 1998, pp. 3635–3640.

[22] F. Mazenc and A. Iggidr. "Backstepping with bounded feedbacks." In: *Systems & control letters* 51.3-4 (2004), pp. 235–245.

[23] F. Santoso, M. A. Garratt, S. G. Anavatti, and I. Petersen. "Robust hybrid nonlinear control systems for the dynamics of a quadcopter drone." In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2018).

[24] X. Fang, Q. Lin, Y. Wang, and L. Zheng. "Control strategy design for the transitional mode of tiltrotor UAV." In: *IEEE 10th International Conference on Industrial Informatics*. IEEE. 2012, pp. 248–253.

[25] A. Friedman. "Dynamic Inversion and Control of Nonlinear Systems." In: *Mathematics in Industrial Problems*. Springer, 1988, pp. 114–120.

[26] G. R. Flores-Colunga and R. Lozano-Leal. "A nonlinear control law for hover to level flight for the quad tilt-rotor uav." In: *IFAC Proceedings Volumes* 47.3 (2014), pp. 11055–11059.

[27] B. L. Stevens, F. L. Lewis, and E. N. Johnson. *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons, 2015.

[28] W. Rigon Silva, A. da Silva, and H. Gründling. "Modelling, Simulation and Control of a Fixed-Wing Unmanned Aerial Vehicle (UAV)." In: Dec. 2017. DOI: 10.26678/ABCM.COBEM2017.COB17-2703.

[29] D. W. Mellinger. "Trajectory generation and control for quadrotors." In: (2012).

[30] *ROS with Gazebo Simulation*. Accessed: 2020-06-12. URL: https://dev.px4.io/v1.9.0/en/simulation/ros_interface.html.

[31] *Pixhawk 4 Mini*. [Online, cited 5 January 2021]. URL: https://docs.px4.io/master/en/flight_controller/pixhawk4_mini.html.

[32] *Pixhawk 4 Mini Wiring*. [Online, cited 5 January 2021]. URL: https://docs.px4.io/v1.9.0/en/assembly/quick_start_pixhawk4_mini.html.