POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Ingegneria Aerospaziale

Tesi di Laurea Magistrale

Dynamic modeling and control design for a winged eVTOL aircraft

Relatrice:
Dr. Elisa Capello

Co-relatore:
Prof. Philippe Pastor

Candidata:
M. Florencia Lema Esposto

Anno Accademico 2018/2019

Acknowledgements

This project was developed in the Department of Aerospace Vehicles Design and Control (DCAS) at ISAE SUPAERO in collaboration with the start-up NEOPTERA Ltd. I want to thank my supervisor Elisa Capello for accepting this project and monitor our work from Turin. I would also like to thank SUPAERO, specially my co-supervisor Philippe Pastor, for giving me the opportunity of participate in this research and making my master thesis a challenging and constructive experience. I would also like to express my deepest thanks to Arnaud Didey and Renaud Othomene for teaching me how it is to work in real project and making me understand the complexity of carrying it out from the beginning to the end. These months have made me improve both personally and professionally and I am sure they will be of great help for my upcoming working career. I would like to thank also to all the professors and Doctorate candidates that have helped us in this project in times when hope was scarce. My deepest gratitude goes specially to my colleague Zoe Mbikayi whit whom I have spent infinite hours dealing with the most absurd problems but also without whom this experience would not have been the same. Finally, I thank my friends here at Toulouse for making my stay so welcoming and my family for supporting me at each step of the way.

Abstract

VTOL vehicles are of great interest at the moment for urban air mobility because of their versatility. These vehicles have the ability to take-off and land without the necessity of a long runway, being compared to helicopters. However, their appeal is based on the fact that are considered a breakthrough over helicopters due to their efficiency in cruise flight, comparable to fixed wing aircrafts. For these reasons, VTOL vehicles reduce the expenses in complex facilities and improve performances from an efficiency point of view.

Within this new technology field, NEOPTERA Ltd presents itself as one of the companies seeking to develop a civil VTOL vehicle for urban air mobility: the eOpter. NEOPTERA is currently developing different prototypes of the eOpter, one of which was selected for this project. The main objective of the project is the development of a flight simulator and a flight controller for the mentioned prototype, a 60 *cm* wingspan version of the eOpter. The innovative VTOL concept presented by NEOPTERA is a two wings tandem configuration with four electric motors on each wing providing energy to a propeller each.

A VTOL flight can be divided into three modes. A vertical take-off and landing, a transition to cruise flight, and the cruise flight itself. The motivation behind this study is to have proper control laws capable of stabilizing the aircraft dynamics during the three flight operating modes. A remarkable aspect of the project developed is treating the entire flight envelope as a continuous flight path, avoiding discontinuity issues when passing from one flight phase to another.

In order to design the flight control laws of the prototype, it is necessary to design certain mathematical models describing the dynamics of the aircraft and its behaviour during each flight mode. This study goes through the development of these mathematical models, their simulation, a stability analysis of the model created and the design of appropriate stabilizing control laws. The obtained simulation model is validated by simulating the vehicle flight and observing the physical behaviour. The control laws developed allow to stabilize the vehicle while enhancing the flight performances and an integrator controller is implemented for reference tracking purposes, being its robustness tested on the linear models of the aircraft.

The results obtained shows a very reliable simulator for longitudinal flight, while for lateral-directional flight further analysis need to be done. A full mission for longitudinal flight was simulated and the results obtained are very promising. The prototype is currently undergoing some flight tests. It is expected to obtain information from these tests that will help to improve the control system of the vehicle by observing its behaviour in real flight.

Further steps will include implementing the gain scheduling method in the non linear simulator, in order to achieve stability for the entire flight envelope and test the obtained tuned controller on the physical prototype without going through additional tuning campaigns.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1. VTOL Technology

Vertical Take-Off and Landing (VTOL) vehicles have the ability to perform a vertical take-off and landing, interesting capability for the design of practical and versatile aircrafts.

VTOL vehicles has the characteristic of producing lift, in general, in different ways according to the phase of the flight. These type of aircraft is characterized by operate at relative low forward velocities in comparison with civil aircrafts. The main difference with the mentioned group is that VTOL vehicles does not reduce their cruise efficiency even though their flying airspeed is low. As commented, the lift can be produced, as usually, with lifting surfaces as wings or additionally with some kind of propulsion.

A remarkable improvement of VTOL vehicles compared to the classical civil aircrafts is the lack of need of a long runway. Historically, as the cruising speed increased, aircrafts needed longer runways to operate safely. This necessity is not in keeping with the current development of cities and big metropolitan areas. This operational capacity of VTOL vehicles increases the attractiveness of the development of this type of aircraft for urban purposes. Furthermore, VTOL vehicles are not only conceived for civil missions, they are also appealing for warfare tasks. In this field is very valuable the ability to take off and land from any type of soil. They are also able to perform various manoeuvres not possible with a conventional plane, a significant advantage for aircraft in combat situations.

The absence of horizontal landing and take-off entail the reduction of much equipment on board. In the first place, it is not necessary to install a conventional landing gear, which reduces significantly the weight of the aircraft. Instead, a VTOL vehicle generally needs a lighter and technologically simpler landing gear, more similar to those employed by helicopters. This also includes the removal of the correspondent system, hydraulic, electrical, mechanical or pneumatic. Moreover, high-lift devices are not required either; slats, slots, flaps and boundary layer control can be removed, implying a reduction of complexity of the aircraft. VTOL technology can avoid different landing issues such as braking, maintenance of devices and design of the optimal solution to reduce the aircraft speed.

During the past decades there have been developed many concept designs for VTOL vehicles. It is important to clarify that autogiros and helicopters are not included in VTOL aircraft category because of the speed requirement imposed for them. VTOL vehicles are required not to pass a certain minimum speed restriction to be categorized in such group, restriction helicopters and autogiros do not respect. This cruise speed is set comparing the corresponding VTOL with an ordinary fixed-wing aircraft performing the same mission. In the end, VTOL vehicles have the flexibility of a helicopter but with the efficiency of a fixed-wing aircraft. They can take-off and land with the facility of a helicopter and avoiding complex phenomena appearing in fixed-wing take-off and landing operations; and they also can reach fixed-wing velocities during cruise, not having the blade tip compressibility limitation that rotary-wing vehicles have.

Once established the limits of the category "VTOL vehicle", there is a wide range of different configurations developed or on process of development. Its classification can be based on different criteria: rotating parts, energy source, carrying capacity, type of thrust, etc. Hereafter, the most common configurations will be explained.

- **Compound Aircraft**: is a combination of a helicopter and a fixed-wing aircraft. During vertical flight the lift is provided completely by the rotor, but in horizontal flight wings contribute to the lift production and the rotor load is reduced. Thrust is provided by a jet engine or a propeller for horizontal flight, which gives the possibility of leaving the rotor in auto-rotation or even be retracted during this phase of the flight. An example of compound aircraft is shown in Figure 1.1.

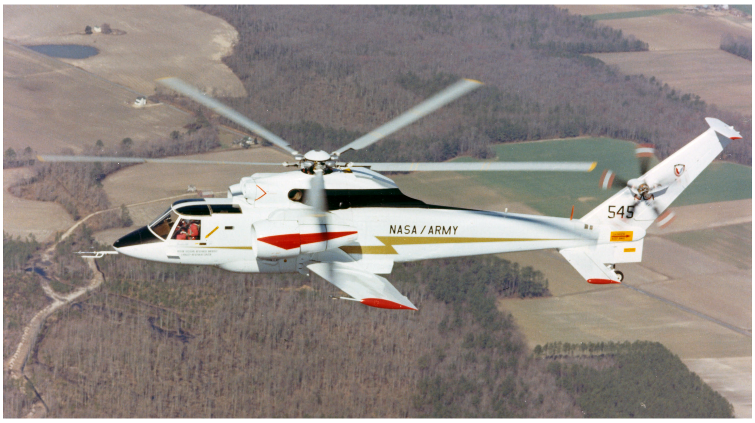


Figure 1.1 Compound aircraft Sikorsky S-72 [1]

- **Tail-sitter**: this VTOL type takes-off and lands on its tail, as their name suggests. However, this is the general definition and there are some vehicles considered tail-sitters that land on their "belly" instead. Tail-sitters have the simplest way to achieve the transition manoeuvre since it does not need extra actuators to perform it. It just does with a differential of thrust in the propellers it has. Tail-sitters development has a long history, many concepts have been considered, such as the Convair XFY Pogo (Figure 1.2) and the SNECMA Coléoptère (Figure 1.3), but their large-scale production is more complicated because of their difficult piloting during vertical flight and transition.

- **Tilt-wing**: This concept of vehicle is based in the fact that the wings and the propellers mounted on the wings tilt 90° while keeping the fuselage horizontal. Tilt-wing vehicles have an important advantage because the major part of the wing is submerged in the propeller slipstream so it does not stall even though the angles of attack reached are very high. On the other hand, rotate a critical part of the aircraft clearly increases the risks of the flight and make it harder to come up with an emergency alternative to follow in case the tilting mechanism fail. If the tilting manoeuvre does not work is hard to regain the control of the vehicle because it is conceived to be handled by the structural part that just failed. However, this configuration has been widely developed and one of the most promising current VTOL technology project is a tilt-wing concept, the Airbus A^3 Vahana shown in Figure 1.4.

- **Tilt-rotor**: this type of VTOL aircraft generates lift and propulsion through propellers that change their orientation in order to perform different manoeuvres. These propellers are

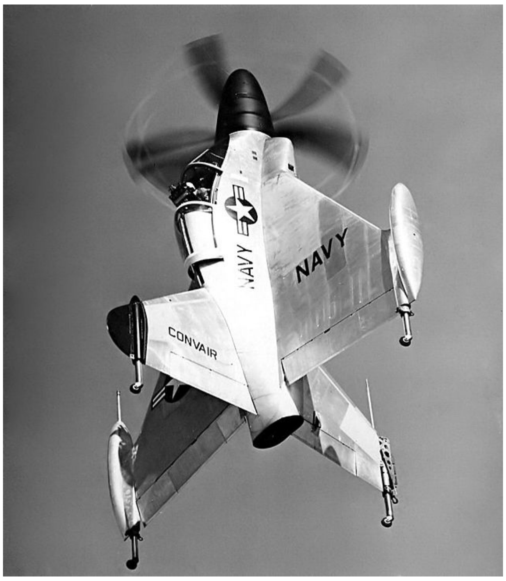


Figure 1.2 Convair XFY Pogo, experimental VTOL fighter

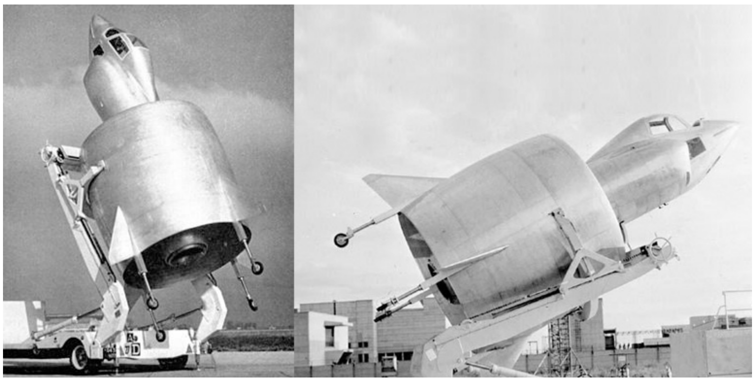


Figure 1.3 SNECMA Coléoptère

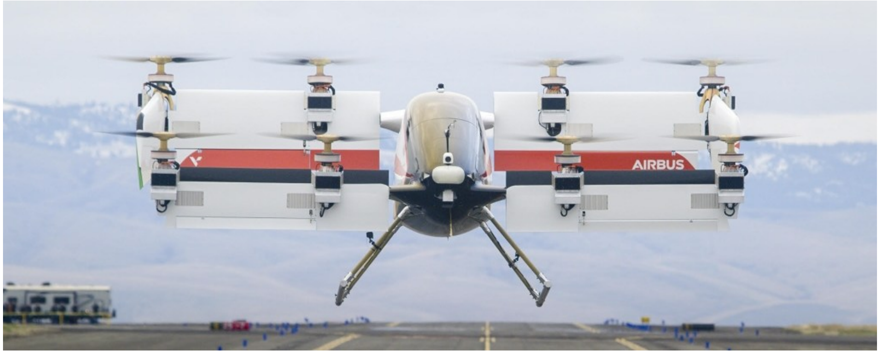


Figure 1.4 Airbus A^3 Vahana tilt-wing concept [1]

usually mounted at the wings or attached to the fuselage, and their tilting permit the transition of the vehicle from vertical to horizontal flight and back. As commented for tilt-wing VTOL vehicles, rotating a crucial part of the aircraft can lead to catastrophic consequences and is harder to guarantee the safety of the flight. The propellers can be rotors, jets or ducted propellers; in all cases the principle is to rotate the thruster producer. This concept has also been extensively studied and some successful projects have gone ahead, such as the Boeing V-22 Osprey, shown in Figure 1.5, which entered service as a military transport aircraft in 2007.

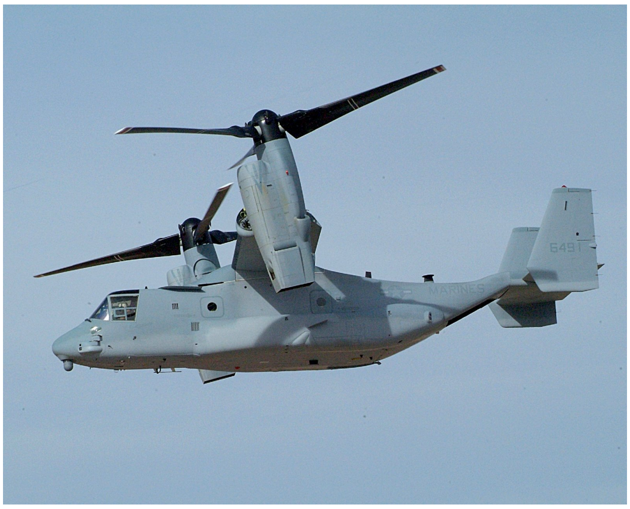


Figure 1.5 Boeing V-22 Osprey, tilt-rotor in service

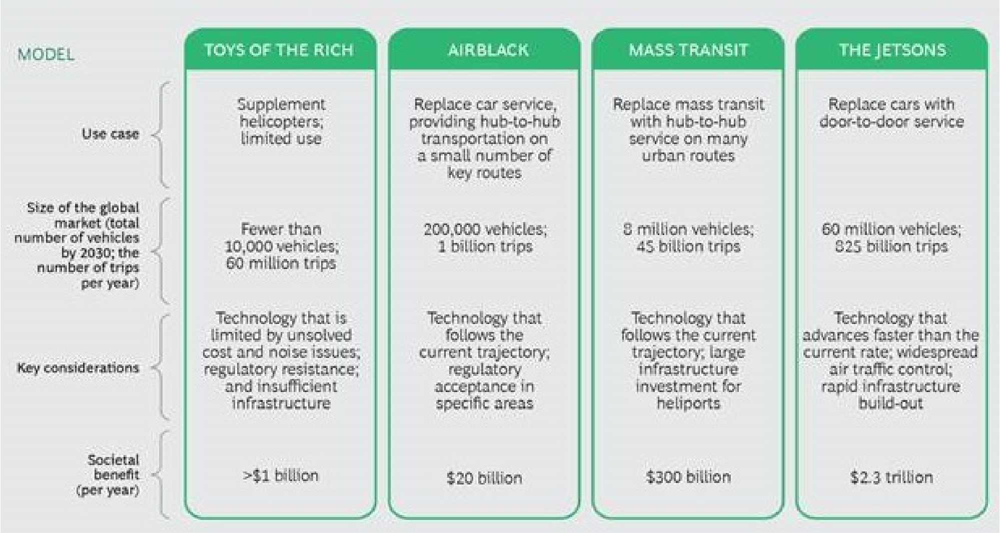
1.2. Current market and forecasts

For decades, it has existed the idea in aerospace industry to generate a market for short-range air transport. In the 1970s there was a big step in this field with the helicopters boom, but for safety reasons, noise problems and high costs, urban helicopters have been limited to a few heliports. The VTOL concept has gained popularity in recent years as a possible alternative to the helicopter. VTOL vehicles have the advantage of combining the flexibility of helicopters with the efficiency of fixed-wing aircrafts. At the moment, most of the VTOL concepts developed have been conceived for military missions, but the idea of an urban vehicle for short-range transport is getting stronger. Nowadays, there are many companies interested in developing a VTOL aircraft for civil purposes and the competition is very tight. Different technologies that have progressed a lot lately, such as electrical propulsion, autonomous flight systems and battery storage, reinforce the idea of VTOL vehicles to be present in our lives in a near future. Even though electric batteries probably need one more step to reach an adequate level of technological maturity to enter in service, commercial VTOL vehicles are expected to be in the market by the mid-2020s. But their role in our lives is still a blurry concept and VTOL vehicles industry evolution is unknown. BCG has analysed VTOL technology future and considers four possible scenarios for this market in 2030 [2], shown in Figure 1.6.

A first possibility is that VTOL vehicles will become the "toys of the rich" and will just substitute helicopters of today owned by wealthy people for private use. This option entails a small production and the fact that this technology will only be enjoyed by an exclusive part of the population.

The next scenario is that they will replace just black-cars, in other words, they will be used only for private business purposes. This scenario increases the market and extends it to more people. However, it is still not the desired situation that aerospace companies are seeking.

The third possible situation is that VTOL vehicles will replace mass transit, that is to say, they will provide transportation point to point for a large amount of people. This requires a big investment in infrastructure but the benefits and the market will be much greater than in the previous cases.



MODEL	TOYS OF THE RICH	AIRBLACK	MASS TRANSIT	THE JETSONS
Use case	Supplement helicopters; limited use	Replace car service, providing hub-to-hub transportation on a small number of key routes	Replace mass transit with hub-to-hub service on many urban routes	Replace cars with door-to-door service
Size of the global market (total number of vehicles by 2030; the number of trips per year)	Fewer than 10,000 vehicles; 60 million trips	200,000 vehicles; 1 billion trips	8 million vehicles; 45 billion trips	60 million vehicles; 825 billion trips
Key considerations	Technology that is limited by unsolved cost and noise issues; regulatory resistance; and insufficient infrastructure	Technology that follows the current trajectory; regulatory acceptance in specific areas	Technology that follows the current trajectory; large infrastructure investment for heliports	Technology that advances faster than the current rate; widespread air traffic control; rapid infrastructure build-out
Societal benefit (per year)	>$1 billion	$20 billion	$300 billion	$2.3 trillion

Figure 1.6 BCG analysis for VTOL vehicles market in 2030 [2]

The last scenario considered is the objective of many companies and consists on the substitution cars for VTOL vehicles in order to reduce the traffic existing in cities today and provide a faster mobility service for the great majority of the population. This reality would be ideal and combining flying cars with conventional cars is the concept that is on the minds of the VTOL technology proponents for the cities of the future.

In order to achieve the forth scenario considered by BCG analysis different prototypes based on the technologies mentioned have been tested and even some of them have done their first flights. Among the most promising projects of today are the following concepts:

- **Lilium Jet**: this German company has in its hands the first electric jet VTOL aircraft, interesting because of its low emissions comparable to an electric car. Their concept is a fixed-wing VTOL vehicle with electric fans on the wings and on the canard. The fans mounted on the canard are supposed to be retracted during cruise while the ones mounted on the wings are situated on a flap that tilts for transition. They have developed several prototypes that have already performed unmanned test flights, and they have even studied the idea of a prototype with folding wings to be also driven as a car. Currently, they are developing a five-seat version and they expect to do a first flight for 2019. From a critical point of view, Lilium project has the drawback of rotating propellers and, as mentioned before, rotating a critical part of the aircraft can seriously compromise the safety of the flight. This issue will probably make it hard to certify the vehicle for urban use due to their incapability to guarantee an emergency solution. Lilium Jet concept is shown in Figure 1.7.

- **Airbus A^3 Vahana**: the giant aerospace company has developed the VTOL vehicle concept in the more advanced stage for the moment. Their submission is an electric 8-prop tilt-wing aircraft to be used as an air taxi for urban mobility. Its first flight with a full-scale prototype took place at Eastern Oregon Regional Airport on January 31, 2018. It was a 53 seconds vertical flight, but on February 12, 2019 they released a video of a new flight of the Vahana including a controlled transition to horizontal flight. This project seems to be in the lead but it has similar inconvenience as Lilium jet: its rotating part is the wing, which make it difficult to control the vehicle in case of emergency and compromise the certification of the aircraft. The full-scale Vahana prototype is shown in Figure 1.8.

- **Bell Nexus**: this concept arises from the association of Uber and Bell Helicopter to develop

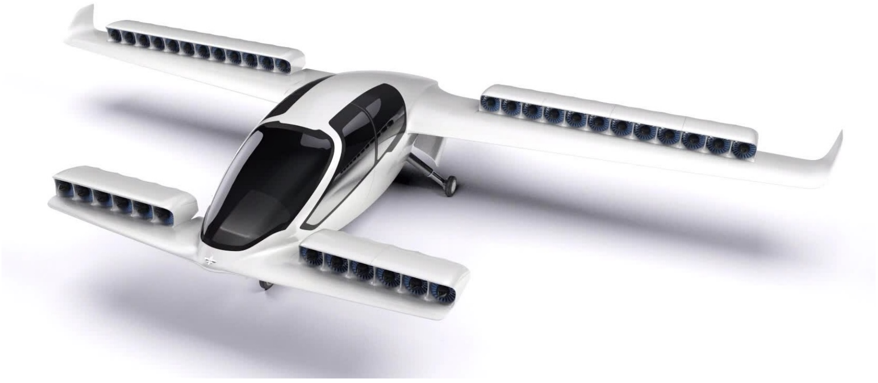


Figure 1.7 Five-seat Lilium Jet concept [1]

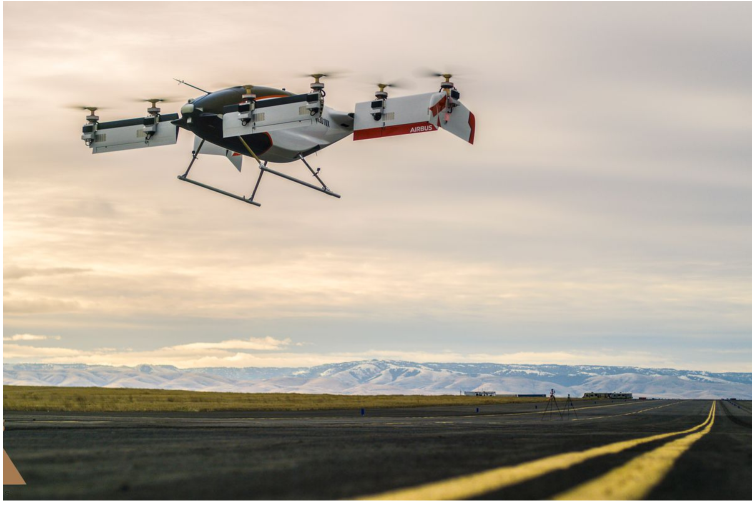


Figure 1.8 Full-scale Vahana prototype

and produce air taxis with VTOL technology. Bell announced at the Uber Elevate Summit on April 25, 2017 that they will work together "to accelerate the eventual large-scale deployment of electric vertical take-off and landing vehicles (VTOLs) in order to fulfil its mission of providing a safe, reliable transportation service to everyone, everywhere". Bell Nexus is a hybrid-electric vehicle with six ducted fans, three on each side of the fuselage, and a vertical stabilizer. It has vectored thrust due to the pitch of the fans allowing to transition from vertical to horizontal flight and vice versa. A full scale model of the Bell Nexus made its first appearance at the Consumer Electronics Show (CES) on January 9, 2019. This project increased its appealing with this spectacular exhibit, but its flying principle has the same disadvantage as the previous concepts. Certification will also be hard to reach for the Bell Nexus and it is one of its main obstacles. Besides, even though they presented a full-scale aircraft there is no proof of any successful flight test, which would be a greater indicator of the technological maturity of the project. Their tilt-rotor aircraft is shown in Figure 1.9.

Figure 1.9 Bell Nexus of Uber and Bell Helicopter association [1]

1.3. eOpter

From a technological point of view, as already seen, some VTOL vehicles are ready to fly, but the main limitation is the certification. In terms of regulation, VTOL technology is still in an early phase and achieving airworthiness certification for urban air mobility and air taxi aircraft is a major challenge these days. This problem has been a motivation for the start-up NEOPTERA Ltd. to create an innovative concept taking into account from the beginning the certification limitations this kind of vehicle will probably have. The eOpter design has been conceived considering at all times the operational aspects certification will probably demand without overlook the efficiency and the ease of use. NEOPTERA.Ltd [3] main objective is developing a light civilian VTOL aircraft for 2 to 5 passengers. eOpter is a tandem-winged vehicle with a fuselage situated between the wings and eight propellers mounted on the leading edge of the wings, four per each. The fuselage maintains its horizontal attitude during the whole flight, while the wings-propellers structure rotates around an axis traversing the fuselage. eOpter concept is shown in Figure refeopter1.

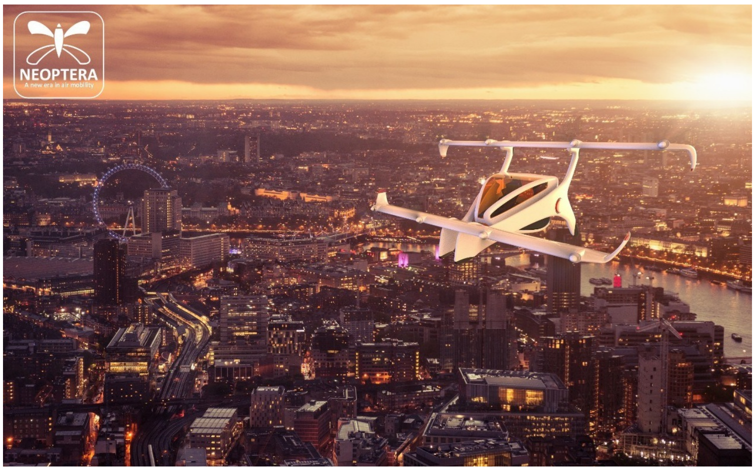


Figure 1.10 eOpter concept in cruise flight [3]

During take-off and landing propellers provide the lift necessary to ascend or descend. During vertical flight mode, the configuration is the one shown in Figure 1.11.

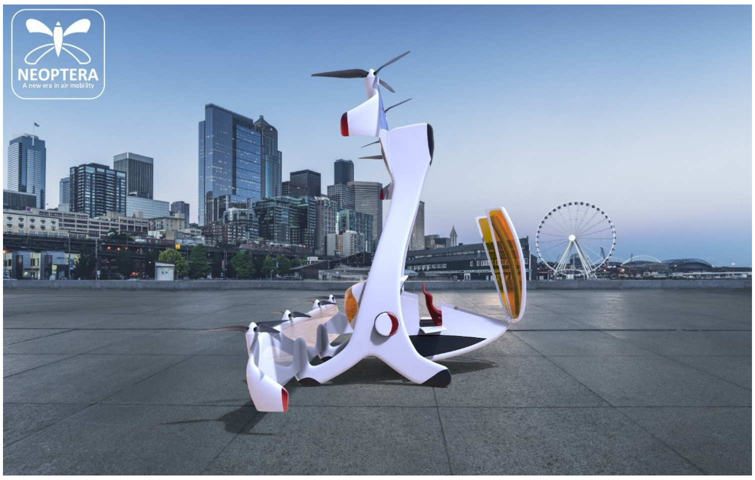


Figure 1.11 eOpter concept in vertical configuration [3]

Once the vehicle reaches a certain altitude and velocity, the transition phase starts. For it to be accomplished, there is applied a differential of thrust between propellers of the lower wing and propellers of the upper wing. For landing, there is again a transition phase that changes the vehicle configuration from horizontal to vertical, also with a thrust differential.

This concept has not the risk of losing such an important component due to rotating malfunctions. In fact, eOpter design allows an emergency landing in horizontal flight configuration if it is necessary, shown in Figure 1.12.

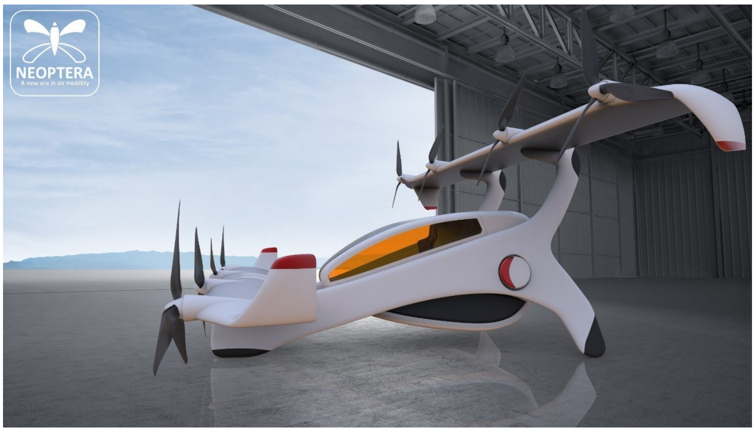


Figure 1.12 eOpter concept in emergency landing configuration [3]

Within the eOpter project frame, NEOPTERA.Ltd is currently working on different scales prototypes of its vehicle. The vehicle in study for this project is a small-scale prototype of 60 *cm* of wingspan. The main objective is to achieve a complete controllable mission which includes a vertical take-off, transition to cruise flight, and vertical landing. To establish a work path, several objectives have been set:

- Build a dynamic simulator for a 60 *cm* prototype of the eOpter.
- Design the flight control laws to stabilize and keep control of the aircraft during the entire flight envelope.
- Test the controller design on the simulator created.

- Perform all the above treating hover, transition and cruise like one continuous flight envelope.

For achieving the goals, a simulator replicating the physical prototype has been developed and tested in order to analyse performances of the vehicle, a stability augmentation system to achieve flying qualities requirements have been built and a reference tracking controller has been designed and tested. Considering the fourth objective, the flight path has been considered continuous, in other words, there is not a discrete division of flight phases. This approach enables a smoother transition and avoids sudden changes in the flight variables.

Finally, a few modifications have been done to the px4 autopilot supported by the Pixhawk flight controller in order to test the prototype. The modifications involve the creation of a new airframe, representing the geometrical configuration of the aircraft and the control mixer. There was also an attempt to add an external Inertial Measurement Unit. But the Pixhawk hardware did not recognize signals coming from the external IMU. After several attempts, several IMU devices tested, several drivers added and/or modified, the Pixhawk did not yield, and so this attempt was paused. However, the new airframe and mixers are currently being tested. The px4 software is complexly structured thus, making the testing and analysis process slow and long.

Chapter 2

The aircraft and its control

The relationship between pilot commands and the attitude of the vehicle is defined Within the Flight Control System (FCS). Generally, this relationship relates pilot actions with the control surfaces, but in the case in study, changes in attitude are achieved through a change of thrust in the propellers.

In the developed simulator the actually changing variable is the propeller rotational speed, which is directly related to propeller thrust. It is necessary to specify the influence each propeller has on the rotations of the vehicle. For that purpose, from now on, engines will be assigned a number to identify them according to Figure 2.1.

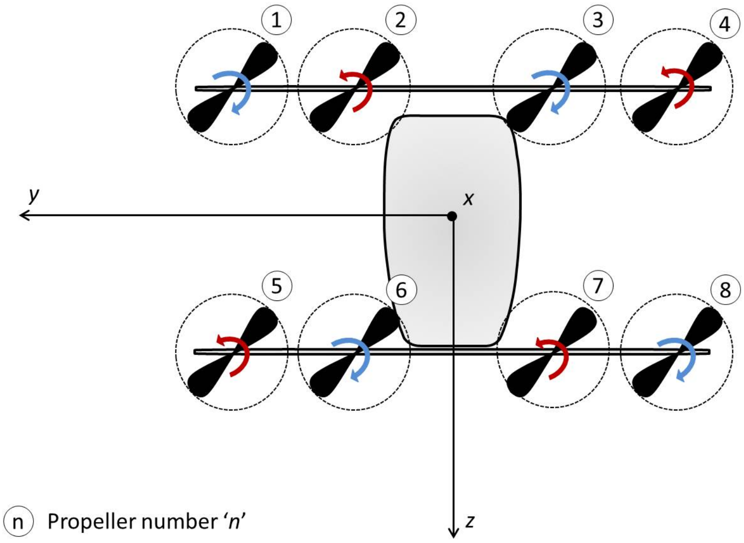


Figure 2.1 Propellers numerical identification (seen from above in vertical mode)

It is important to mention that the axes represented in Figure 2.1, would match a typical aircraft body reference in the prototype horizontal mode. This is, in horizontal mode, the prototype has its $x-axis$ and $y-axis$ both parallel to the ground and $z-axis$ pointing to the centre of the Earth. Because axes rotate in solidarity with the wings, in vertical mode, $x-axis$ points in the opposite direction to the centre of the Earth, $y-axis$ remains in the same position and $z-axis$ changes its attitude and stays parallel to the ground. This is easily to understand by checking Figure 2.2.

As it can be seen, axes remain in the same direction than the wings, which are considered the body reference, not the fuselage as might be guessed.

With this axes configuration, the three rotations roll, pitch and yaw match the typical definition

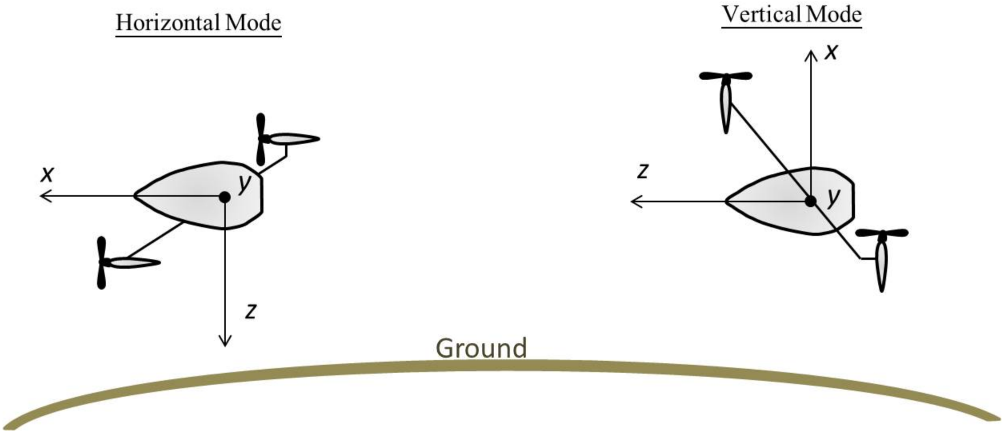


Figure 2.2 Axes configuration in vertical and horizontal mode

in horizontal mode. In vertical mode, yaw would be around $x-axis$, roll would be around $z-axis$ and pitch would still be around $y-axis$, as usual.

As said before, rotations are based on a differential of propellers thrust. A propeller matrix define which propellers must change their rotational speed, if this variation must be an increase or a decrease and the magnitude of the variation.

Different manoeuvres are considered depending on the flight mode, but at the end these manoeuvres will just be related to translations along an axis or rotations around an axis.

2.1. Vertical Mode

- Up & Down: To change the altitude of the vehicle in the vertical mode, the thrust of all propellers must increase to go up and decrease to go down. In this case, all propellers experiment the same change of rotational speed. Going up is moving along $x^{+}-axis$ and going down is moving along $x^{-}-axis$.

- Hover: In this case, the objective is to maintain the vehicle altitude, the logical thing would be to think that all the propellers must give the same thrust, but that is not true. To maintain a balanced position, due to the asymmetry of the vehicle, thrust cannot be the same for all propellers, if so there would be a rotation caused by the difference in distances from propellers to the centre of gravity. This manoeuvre must be controlled by a controller with a difference of thrust from one propellers to the others capable of compensate the moment that would be created.

- Forward: To move forward, there must be a slight rotation around $y^{-}-axis$. In order to get this rotation, propellers 1 to 4 must increase their rotational speed $(\uparrow \omega \rightarrow \uparrow T)$ and propellers 5 to 8 must do the opposite, decrease their rotational speed $(\downarrow \omega \rightarrow \downarrow T)$.

- Backward: This manoeuvre is the opposite from moving forward, so the physical principle to achieve it is to do the opposite from the previous example. This is, decrease the rotational speed of propellers from 1 to 4 and increase the rotational speed of propellers from 5 to 8.

- Yaw: In vertical mode, as already explained, yaw involves a rotation around $x-axis$. If the vehicle is looked from above, a clockwise rotation around $x-axis$ (rotation around $x^{-}-axis$) causes a turn to the right and a counter-clockwise rotation around $x-axis$ (rotation around $x^{+}-axis$) causes a turn to the left.

This manoeuvre is based on the different rotation sense the propellers have and on the actions-reaction principle. To have a balanced quadcopter and avoid the torque effect, propellers must rotate in different directions, half of them in one direction and the other half in the opposite one. This rotational difference can be appreciated in Figure 2.1. In the prototype, propellers 1, 3, 6 and 8 rotate in clockwise direction and propellers 2, 4, 5 and 7 in counter-clockwise direction (looked from above).

Thus, to turn to the right, propellers 2, 4, 5 and 7 must increase their ω while propellers 1, 3, 6 and 8 decrease theirs. The opposite must be done to turn to the left.

- Right: This manoeuvre causes a translation along $y-axis$, for which it is necessary certain rotation around $z-axis$. In this case, there is a differential of thrust between propellers positioned at both sides of the prototype considering plane xz the border. So propellers 1, 2, 5 and 6 must decrease their ω while propellers 3, 4, 7 and 8 must increase theirs. These variations make right side propellers to have a global thrust greater than left side propellers, so a moment around z^+ will appear, as appreciated in Figure 2.3. This can be considered a "roll" for the vertical mode.

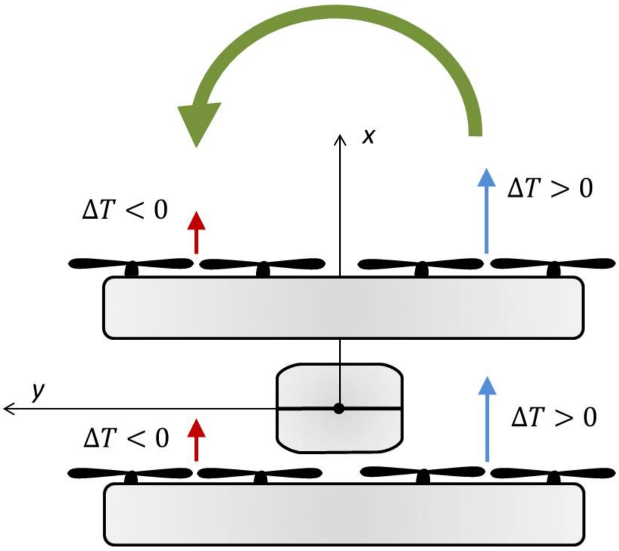


Figure 2.3 Roll rotation in vertical mode

- Left: To go left, the philosophy is the same than for the previous manoeuvre but the effect must be the opposite. Hence, propellers 1, 2, 5 and 6 must increase their ω while propellers 3, 4, 7 and 8 must decrease theirs

2.2. Horizontal Mode

- Forward: This manoeuvre causes a translation along $x^+ - axis$. It is achieved by changing the rotational speeds of all propellers. If these are increased the vehicle will move faster, and if these are decreased the vehicle will move slower, but it will always go forward, never backward. In this case, wings must be taken into account because they will give some lift that will be combined with the propellers to maintain the desired attitude.

- Roll: Roll in horizontal mode is a rotation around $x-axis$, positive to go right and negative to go left. This manoeuvre is based in the same principle than yaw for the vertical mode, because the rotation is around the same axis: x. To turn to the right, propellers 1, 3, 6 and 8 must increase their ω and 2, 4, 5 and 7 must decrease theirs. To turn to the left, the opposite process must be executed. This manoeuvre is represented in Figure 2.4.

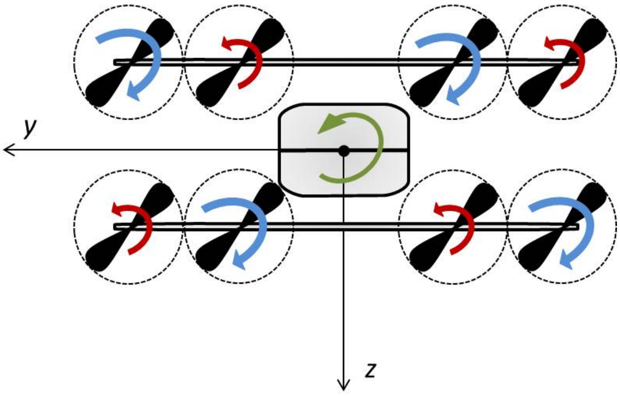


Figure 2.4 Roll rotation in horizontal mode

- Yaw: As in roll for vertical mode (identified as Right and Left manoeuvres in Section 2.1), the propellers are divided in two different groups separated by plane xz. To turn to the right, left side propellers (1, 2, 5 and 6) must increase their rotational speed while right side propellers (3, 4, 7 and 8) decrease theirs. To turn to the left, the variations must be the opposite. This division of two groups corresponds to a top view of the vehicle, as seen in Figure 2.5.

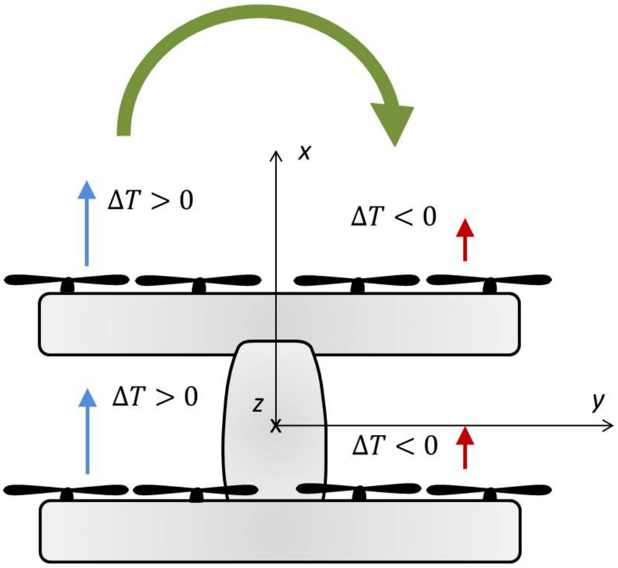


Figure 2.5 Positive yaw in horizontal mode

- Pitch: In horizontal mode, pitch allows the vehicle to go up and down, because of the variation of the angle of attack. To change the pitch of the prototype, propellers are divided in two different groups, separated by plane xy. To cause a positive pitch (turn around y^+), upper side propellers (1 to 4) must decrease their ω while lower side propellers (5 to 8) must increase theirs.

Comparing both flight modes, all the manoeuvres are shown in Table 2.2.

Rotation/Translation	Vertical flight mode	Horizontal flight mode
Translation along $x-axis$	Up/Down	Forward/Backward
Translation along $y-axis$	Right/Left	Right/Left
Translation along $z-axis$	Forward/Backward	Up/Down
Rotation around $x-axis$	Yaw	Roll
Rotation around $y-axis$	Pitch	Pitch
Rotation around $z-axis$	Roll	Yaw

Table 2.1 Manoeuvres for vertical and horizontal flight modes

2.3. Control algorithm

The most critical flight phase of this type of aircraft is the transition between hover and cruise flight. Most VTOL controllers are implemented separately, as multi-rotor in hover, and as a winged aircraft in cruise. These two controllers interchange during the flight, thus creating a temporary transient between the two phases. Besides the separate controllers, different Euler angles definitions are used, in order to avoid the numerical singularity that occurs when the pitch angle reaches 90°. These methods often lead to a temporary loss of control during the transition.

This study goes with a different approach, treating hover, transition and cruise phases as one flight mode. From the pilot point of view, the aircraft is controlled and stabilized without any discontinuities between the phases. And rotations are achieved using quaternions, which are a numerically stable solution (Section 3.2.4).

Besides, the controller is conceived in such a way that it is capable of adapting during the whole flight envelope, and stabilizing the vehicle in a continuous set of forward velocities and pitch angles from zero velocity to cruise speed.

Moreover, the flight control laws need to be tested on a scale prototype, using a Pixhawk flight controller and the PX4 control software. PX4 is a flexible software having different structures, geometries and configurations already defined; therefore can be used for any type of UAV. Figure 2.6 shows the VTOL airframes available on the PX4 software. However, considering the complex configuration of the prototype in study, there is a need to create a new airframe with its geometry.

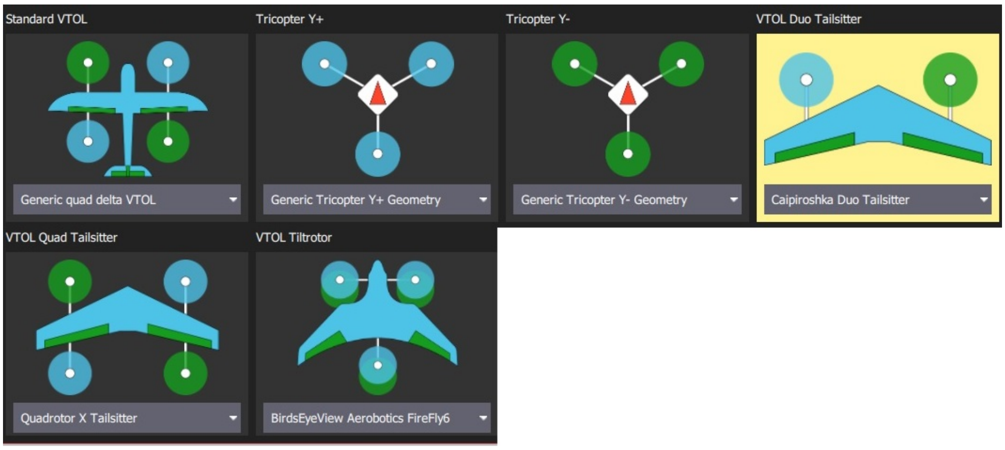


Figure 2.6 VTOL airframes available in the PX4

2.4. Definition of forces and moments

In order to have a close to prototype model, mathematical models based on laws of physics and aerodynamics are built. Modelling the aircraft aerodynamics is undeniably the most important part, in order to achieve a reliable model. The equations of motion that will be developed further will be driven by the forces and moments that are acting on the aircraft. Taking into account and respecting as much as possible the geometry of the aircraft is very important to build a model that replicates the reality. eOpter real prototype is shown in Figure 2.7.

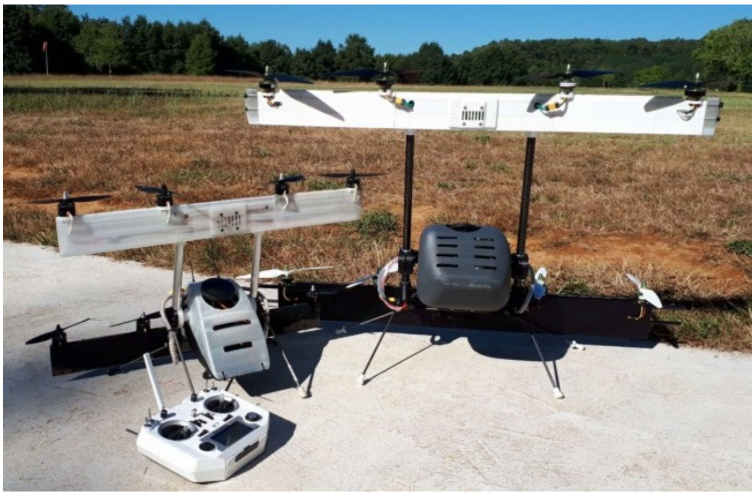


Figure 2.7 eOpter real prototype

An evaluation of all these forces and how do they affect the attitude of the vehicle has been done. The aircraft has two wings producing lift and drag, a fuselage producing only drag, eight electric motors running eight propellers producing thrust, and the aircraft weight. Forces have been separated in different groups according to their nature: aerodynamic forces, weight and thrust.

Theses forces create moments around different axes. Moments depending on dimensionless aerodynamic coefficients found by analytical means and using different computational fluid dynamics tools. However this paper does not go through the computation of these coefficients. And for the purpose of this study, it is assumed that the fuselage does not produce lift.

2.4.1. Aerodynamic Forces

The simulator developed in SIMULINK works in Body Reference Frame, frame considered to be attached to the wings, not to the fuselage. The rotational matrix obtained from [6] and expressed in Equation 2.1 is used to go from Wind Reference Frame to Body Reference Frame.

$$\vec{A}_B = \underbrace{\begin{bmatrix} \cos\alpha\cos\beta & -\cos\alpha\sin\beta & -\sin\beta \\ \sin\beta & \cos\beta & 0 \\ \sin\alpha\cos\beta & -\sin\alpha\sin\beta & \cos\alpha \end{bmatrix}}_{L_{W\to B}} \cdot \vec{A}_W \tag{2.1}$$

where

$$\vec{A}_W = \begin{bmatrix} -D \\ Y \\ -L \end{bmatrix}$$

Being α the angle of attack and β the side-slip angle, known as the aerodynamic angles as they provide the vehicle orientation with respect to the relative wind. Figure 2.8 shows the aircraft with a relative wind $\overrightarrow{V}$, α and β.

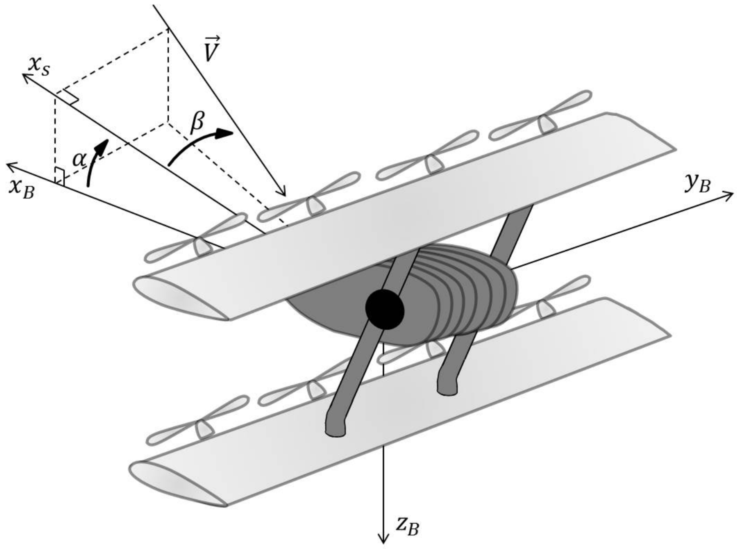


Figure 2.8 Aerodynamic forces and angles

The aerodynamic angles are calculated as shown in Equations 2.2 and 2.3.

$$\alpha = \tan^{-1}\left(\frac{w}{u}\right) \tag{2.2}$$

$$\beta = \sin^{-1}\left(\frac{v}{||\overrightarrow{V}||}\right) \tag{2.3}$$

where u is velocity along $x - axis$, v is velocity along $y - axis$ and w is velocity along $z - axis$ in Wind Reference Frame.

In the first version of the simulator, the fuselage is considered to remain always with the same attitude, so there has not been included the angle relative variation between the wings and the fuselage.

Lift, Drag and the Lateral forces can be modelled as shown in Equations 2.4 and 2.5 and 2.6 respectively, as given in [6].

$$L = \frac{1}{2}\rho V^2 S C_L = qSC_L \tag{2.4}$$

$$D = \frac{1}{2}\rho V^2 S C_D = qSC_D \tag{2.5}$$

$$Y = \frac{1}{2}\rho V^2 S C_Y = qSC_Y \tag{2.6}$$

- $\mathrm{C_L}$: The lift coefficient is a function of α and Mach number (M). Its variation with α is quite linear until near the stall when it drops sharply and then tends to find an equilibrium point. Figure A.4 shows the variation of $\mathrm{C_L}$ with α. This curve is a result of aerodynamic analysis on different CFD tools. Details about these analysis are however beyond the scope of the present paper. The angle of attack used for this calculation is based on the wing-propeller theory developed in Section 2.4.4.

- $\mathrm{C_D}$: In general, the drag force is a combination of friction drag and drag caused when the integral of pressure over the whole surface area of a body is non-zero. It highly depends on the shape of the body, Reynolds number (*Re*), Mach number (*M*) and other parameters. The wing drag coefficient can be modelled as shown in Equation 2.7 given in [7].

$$C_D = C_{D_0} + \frac{C_L^2}{\pi e AR} \tag{2.7}$$

  The efficiency factor (*e*) is calculated using the Shevell method [8]. This method computes the efficiency factor as a function of the wing aspect ratio (*AR*), the sweep of the wing (Λ) and the drag coefficient at zero lift (C_{D0}), parameters defined in Appendix A.

  It is also worth noticing that C_D is a function of C_L, therefore, depends directly on α.

- $\mathrm{C_Y}$: On an aircraft, the lateral force is mainly created by the side-slipping motion and the rudder deflection. However, on this aircraft there is not a rudder. The side-force is created only by the side-slip and the propellers. The coefficient of this force when created by β can be modelled as shown in Equation 2.8, where C_{Y_β} is a function of α and M.

$$C_Y = C_{Y_\beta}(\alpha, M)\beta \tag{2.8}$$

More detailed information about the aerodynamic coefficients is collected in Section A.3 of Appendix A.

The aircraft has additional drag produced by the fuselage. It is also calculated as shown in Equation 2.5, using the surface of the fuselage and a drag coefficient found from CFD analysis of the fuselage (Appendix A). And it is expressed in the Wind Reference Frame as

$$\vec{A}_W = \begin{bmatrix} -D \\ 0 \\ 0 \end{bmatrix}$$

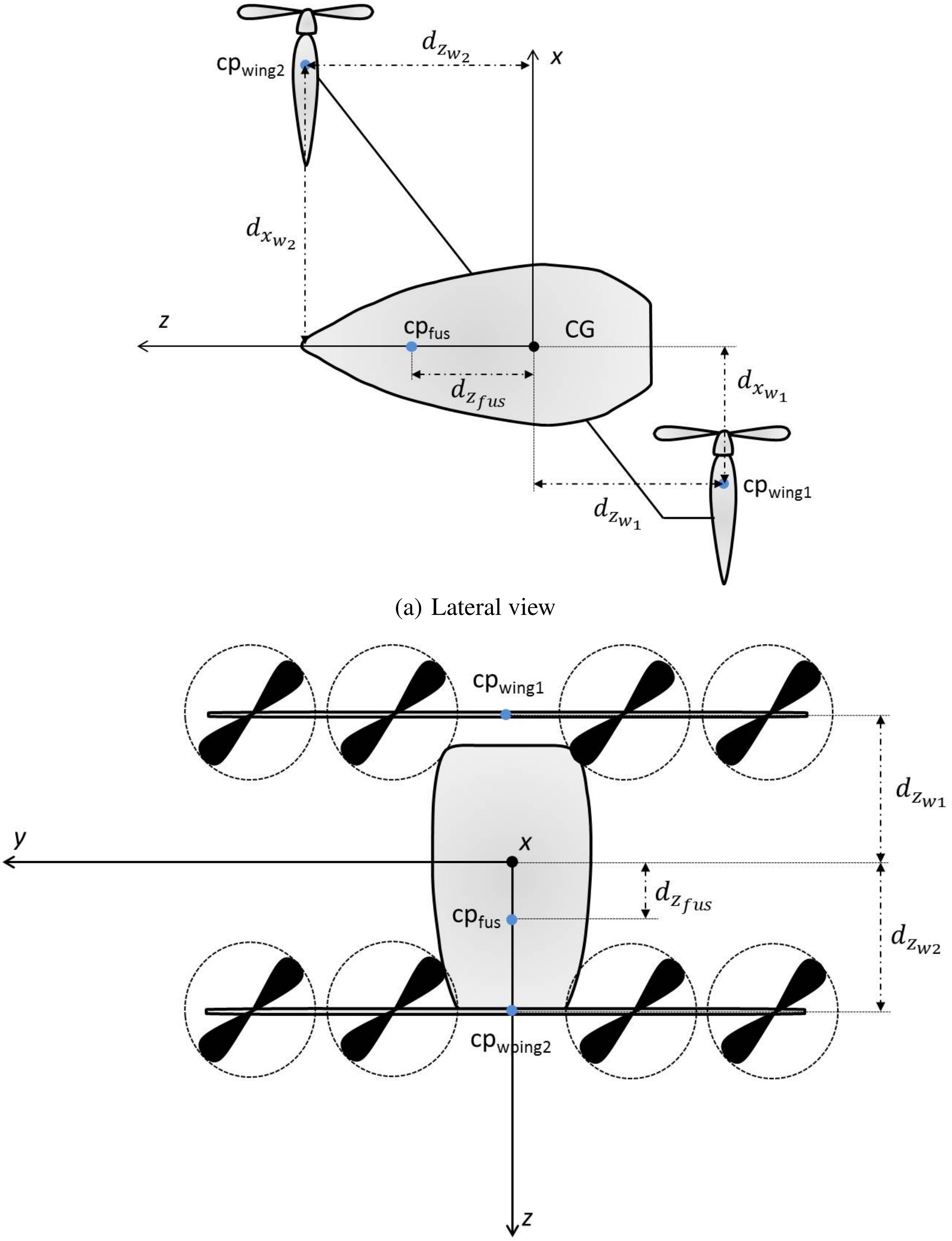
The moments induced by the aerodynamic forces of the wings and the fuselage depend on the distances between the centres of pressure of these bodies and the centre of gravity, considered the first ones for simplicity at 25 % of the chord. These distances are expressed in Figure 2.9. This represents the vertical mode and the centre of gravity is supposed to coincide with the centre of rotation of the wings with respect to the fuselage. In order to simplify the analysis, the centre pressure of the fuselage is considered coincident with the centre of gravity, this is, $d_{z_{fus}} = 0$. Therefore aerodynamic forces from the fuselage do not create any moment.

There are moments about the three axes as shown in Equations 2.9, 2.10 and 2.11.

$$M_x = F_{y_1} \cdot d_{z_{w_1}} - F_{y_2} \cdot d_{z_{w_2}} \tag{2.9}$$

$$M_y = 2 \cdot M_0 + F_{z_1} \cdot d_{x_{w_1}} - F_{z_2} \cdot d_{x_{w_2}} - F_{x_1} \cdot d_{z_{w_1}} + F_{x_2} \cdot d_{z_{w_2}} \tag{2.10}$$

$$M_z = F_{y_2} \cdot d_{x_{w_2}} - F_{y_1} \cdot d_{x_{w_1}} \tag{2.11}$$



(a) Lateral view

(b) Top view

Figure 2.9 Distances from centres of pressure to centre of gravity

such that for each wing

$$M_0 = qScC_{m_0}$$

being S the reference wing surface, c the wing chord and C_{m_0} the pitch moment coefficient at zero lift as a result of various aerodynamic analysis provided by Neoptera. Figure A.2 shows the variation of C_m with α.

The aerodynamic centre of each wing is considered to be in the point situated at one quarter of the chord and the middle line of the span, as shown in Figures 2.9(a) and 2.9(b). This implies that aerodynamic moments around $x-axis$ and $z-axis$ are only created by forces along $y-axis$. The moment around $y-axis$ is the most important factor of the vehicle flight dynamics because of its role in the transition controllability of the aircraft.

It is remarkable that, due to the difference in distances along $z-axis$ between each wing and the centre of gravity, to keep the vehicle balanced, propellers must not give the same thrust. This attitude control will be handled by the controller, focusing on maintain Euler angles in a reference value obtained from the commands of the pilot.

Considering the big distance between the two wings, can be assumed that there is no interaction between the wind flow of the two wings. Therefore, forces and moments of each wing can be modelled separately.

2.4.2. Weight

Total weight is applied on the centre of gravity of the vehicle. In this case, it is necessary to go from Inertial Reference Frame to Body Reference Frame. With this purpose, it is used the matrix expressed in Equation 2.12 obtained from [6].

$$\vec{A}_B = \underbrace{\begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi-\cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi+cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi+\sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi-\sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix}}_{L_{I\to B}} \cdot\vec{A}_I \tag{2.12}$$

where

$$\vec{A}_I = \begin{bmatrix} 0 \\ 0 \\ W \end{bmatrix}$$

Being ϕ roll angle, θ pitch angle and ψ yaw angle.

However, considering the numerical singularity of Euler angles rotation, Equation 2.12 will be replaced by Equation 2.13 which uses quaternions.

$$DCM = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2\cdot(q_1\cdot q_2+q_3\cdot q_0) & 2\cdot(q_1\cdot q_3-q_2\cdot q_0) \\ 2\cdot(q_1\cdot q_2-q_3\cdot q_0) & q_0^2-q_1^2+q_2^2-q_3^2 & 2\cdot(q_2\cdot q_3+q_1\cdot q_0) \\ 2\cdot(q_1\cdot q_3+q_2\cdot q_0) & 2\cdot(q_2\cdot q_3-q_1\cdot q_0) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix} \tag{2.13}$$

The weight is not be considered to generate any moment because it is applied on the centre of gravity.

2.4.3. Thrust

The propeller model is built as given in [9]. Equations 2.14 and 2.15 give respectively the thrust T and the required power P.

$$T = \rho n^2 D^4 C_T \tag{2.14}$$

$$P = \rho n^3 D^5 C_P \tag{2.15}$$

The thrust coefficient C_T, power coefficient C_P and efficiency η of the propellers are given as functions of the advance ratio J given in Equation 2.16.

$$J = \frac{V}{nD} \tag{2.16}$$

The velocity taken for the advance ratio is the component in $x-axis$, which is orthogonal to the propeller plane.

Figures A.5(a), A.5(b) and A.5(c) show the variation of C_T, C_P and η respectively with the advance ratio.

Additionally, propellers produce a torque Q whose sign depends on the rotation direction. This torque can be modelled as shown in Equation 2.17, and is produced around the propeller axis.

$$Q = \rho n^2 D^5 C_Q \tag{2.17}$$

with

$$C_Q = \frac{C_P}{2\pi}$$

As mentioned in Chapter 2, the aircraft does not have control surfaces and is entirely controlled through the thrust. At this stage the moment created around $x-axis$ is given in 2.17. Moments around $y-axis$ and $z-axis$ are directly related on the positions of propellers with respect to the centre of gravity of the vehicle. These distances are expressed in Figure 2.10.

Equations 2.18 and 2.19 shows the resulting moments around $y-axis$ and $z-axis$.

$$M_y = T_1 \cdot d_{z_{w_1}} + T_2 \cdot d_{z_{w_2}} \tag{2.18}$$

$$M_z = (T_1 + T_5) \cdot d_{y_1} + (T_2 + T_6) \cdot d_{y_2} - (T_4 + T_8) \cdot d_{y_1} - (T_3 + T_7) \cdot d_{y_2} \tag{2.19}$$

2.4.4. Wing-propeller interaction

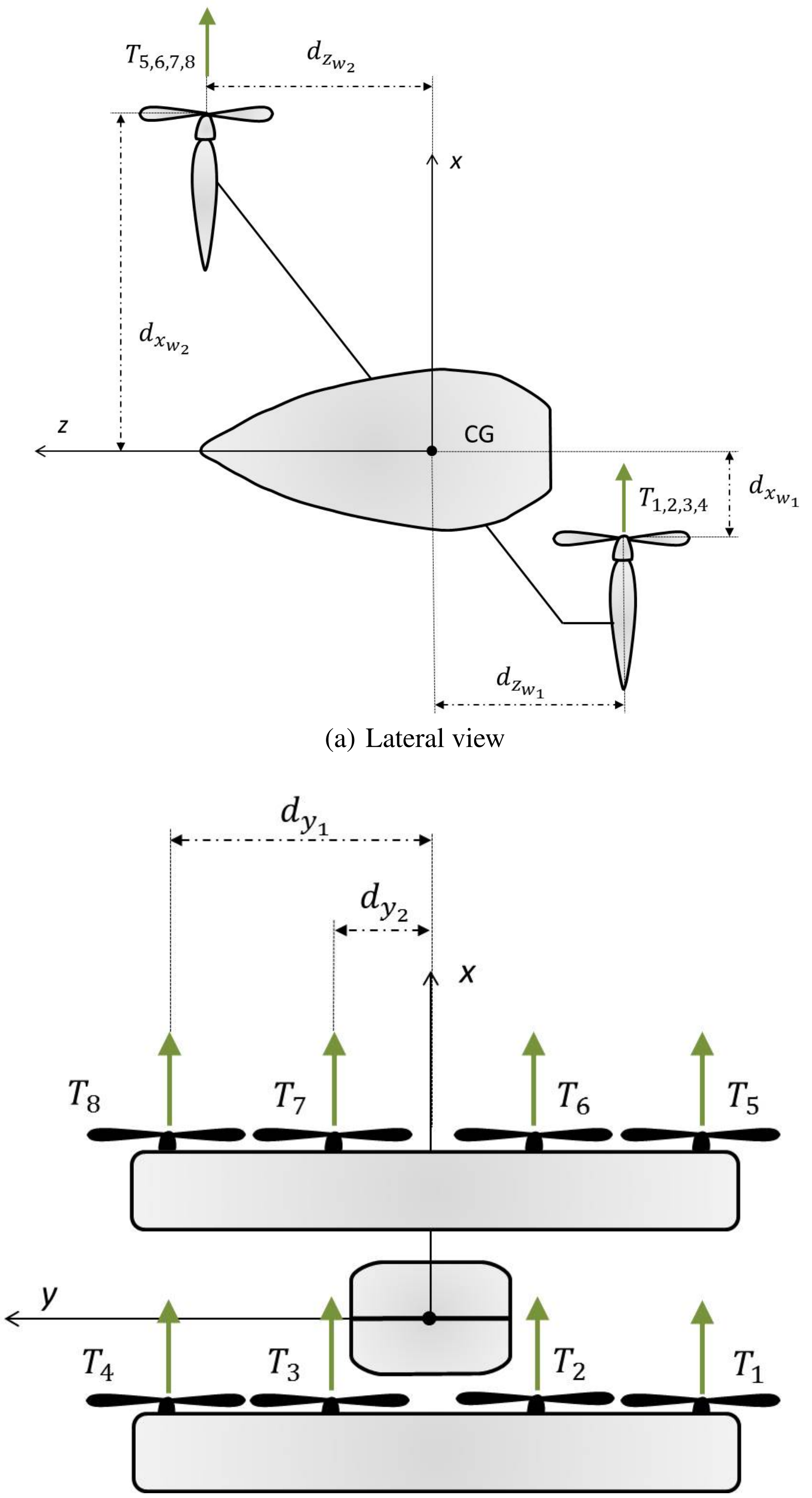
Wing-propeller interactions are modelled using a common approach given in [10] and developed in [11]. The magnitude of the velocity is increased by the propeller downstream, while α and β are decreased, as shown in Figure 2.11.

The prop-wash induced α_i and β_i can be computed as shown in Equations 2.20 and 2.21 respectively.

$$\alpha_i = \tan^{-1}\left(\frac{V^2 \sin\alpha \cos\beta}{V^2 \cos\alpha \cos\beta + \frac{T}{\rho S}}\right) \tag{2.20}$$

$$\beta_i = \sin^{-1}\left(\frac{V^2 \sin\beta}{\sqrt[4]{V^4 + 2V^2 \cos\alpha \cos\beta \frac{T}{\rho S} + \left(\frac{T}{\rho S}\right)^2}}\right) \tag{2.21}$$

Considering that the wings have four propellers each, the four propellers cover the whole wing surface and the small chord, α_i and β_i can be treated as coming from one propeller that covers



(a) Lateral view

(b) Top view

Figure 2.10 Distances from propellers to centre of gravity

the whole wing, so the thrust considered is the addition of the thrust of the four corresponding

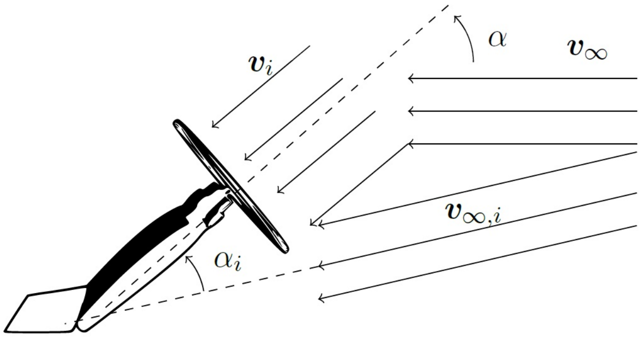


Figure 2.11 Resulting induced angle of attack due to the presence of the propeller, where $v_{\infty,i}$ is the induced relative air velocity, α_i is the induced angle of attack, v_i is the induced propeller wake, v_∞ is the downstream air speed and α is the original angle of attack without propellers

propellers for each wing.

This interaction is very important for transition. In this phase, the angle of attack goes from 90° to 0° or viceversa, which means that is very likely for the wing to stall. The acceleration of the airflow around the wing postpones stall and allows the propellers to have time to take charge of the lift of the vehicle before the wings start to fail.

Chapter 3

Simulator development

Chapter 3 shows the combination of the aerodynamic forces and moments calculated in Section 2.4 with the flat Earth equations of motion to obtain the aircraft simulation models that will be implemented in Matlab and Simulink.

The aircraft is equipped with an electrical propulsion system and a battery, therefore, has a constant fixed mass and a constant inertia matrix.

For this study purpose, the following assumptions are made:

- The aircraft is rigid
- The Earth is flat

3.1. Equations of Motion

The first step to project the simulator is to establish the equations of motion involved in the prototype performances obtained from [12], [13] and [14] and developed following the method in [15].

First of all, Newton equations are defined as

$$\sum_i \vec{M}_i = \frac{d}{dt}\vec{H}$$
$$\sum_i \vec{F}_i = m\frac{d}{dt}\vec{v}$$

where forces are $\vec{F}_i = [F_x \quad F_y \quad F_z]_i = \begin{bmatrix} m\frac{du}{dt} & m\frac{dv}{dt} & m\frac{dw}{dt} \end{bmatrix}_i$ and velocity is $\vec{v} = [u \quad v \quad w]$. Besides, the angular momentum is

$$\vec{H} = I\vec{\omega}_I$$
$$\vec{\omega}_I = [p \quad q \quad r]^T$$

where $\vec{\omega}_I$ is the angular velocity of the vehicle around the centre of gravity expressed in Inertial Frame and I is the inertial matrix defined in Equation A.1.

To express the moments in Body Reference Frame, it is necessary the following transformation

$$[L \quad M \quad N]^T = I\dot{\vec{\omega}} + \vec{\omega} \times (I\vec{\omega})$$

where L, M, N are the moments acting on the vehicle in the three axes.

To obtain the forces in Body Frame it is necessary to express the velocity in this frame, so the following equation is obtained

$$[F_x \quad F_y \quad F_z]^T = m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qw \end{bmatrix}$$

because $\dot{\vec{v}}_I = \dot{\vec{v}}_B + \vec{\omega} \times \vec{v}_B$.

To obtain Euler angles in Body Frame a transformation must be done because vector $[p \quad q \quad r]^T = [\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]^T$ is in Inertial Frame. Then the following transformation is made

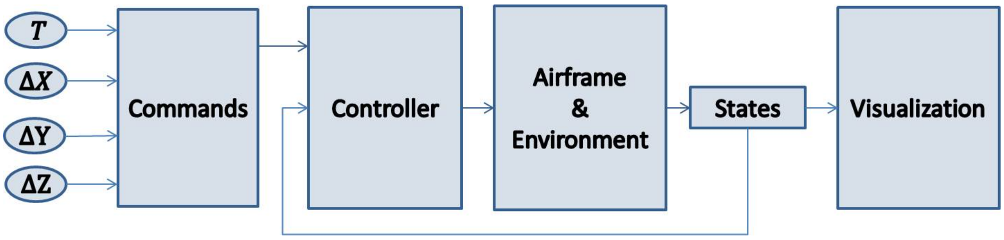
$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \dfrac{\sin\phi}{\cos\theta} & \dfrac{\cos\phi}{\cos\theta} \end{bmatrix} \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.1}$$

After this last step, $\dot{\phi}$, $\dot{\theta}$ and $\dot{\psi}$ are integrated to obtain the Euler angles.

This approach allows us to compute Euler angles starting from the forces and moments acting on the vehicle.

3.2. General structure

The SIMULINK simulator is structured as seen in Figure 3.1.



ing

Figure 3.1 Simulator main structure

Its components are 4 different blocks: *Commands* block, *Controller* block, *Airframe + Environment* block and *Visualization* block.

Explaining the first steps of the simulator as a process, this is:

1. The pilot gives some commands, said to be the reference values.

2. The commands are taken by a controller which, in order to get the angle rates to achieve those reference values, orders some rotational speed differentials on the propellers.

3. The rotational speed variations are transmitted to the propellers.

4. After the propellers had properly changed their rotational speed, the simulator computes the rest of the variables values (*States* and *Environment*) corresponding to this new situation.

3.2.1. *Commands* block

The *Commands* block receives T, ΔX, ΔY and ΔZ commands from a joystick. The first variable T is the rotational speed of basis that all the propellers have according to the thrust lever position. The other three variables express the variation of rotational speed that it is necessary to apply in order to turn around each of three axes, ΔX for a rotation around $x - axis$, ΔY for a rotation around $y - axis$ and ΔZ for a rotation around $z - axis$ (Roll, Pitch and Yaw in horizontal flight mode). A diagram of this part of the simulator is shown in Figure 3.2.

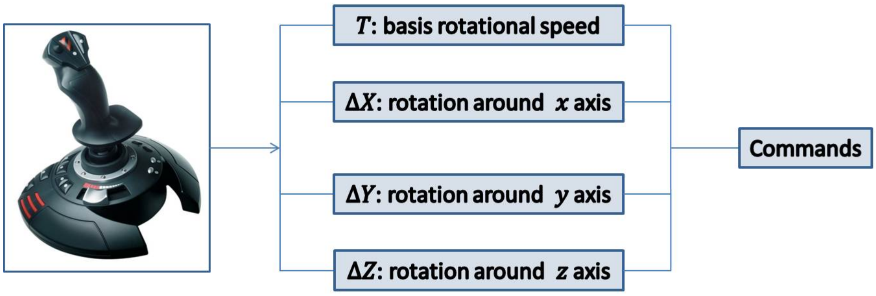

T: basis rotational speed
Δ*X*: rotation around *x* axis
Δ*Y*: rotation around *y* axis
Δ*Z*: rotation around *z* axis
Commands

Figure 3.2 Commands structure

3.2.2. *Controller* block

The *Controller* block transforms the input commands into real orders for the propellers through a matrix that defines the influence of each propeller on the global thrust, which varies from one propeller to another due to their location on the vehicle. This matrix takes the 4 variables mentioned as inputs and its outputs are the pwm signals limited between 0 and 35000.

Combining the 4 commands with a matrix representing the propellers location it is possible to obtain a vector representing the rotational speed that each propeller must have. This operation is expressed in Equation 3.2.

$$[T \quad \Delta X \quad \Delta Y \quad \Delta Z] \cdot [Propellers \quad matrix] = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \\ n_6 \\ n_7 \\ n_8 \end{bmatrix}^T \tag{3.2}$$

In this block for the Longitudinal Model are also included the controllers developed and explained in Chapter 4. Figure 3.3 shows a representative structure of the *Controller* block for the longitudinal model.

The output *pwm* is a vector that contains the rotational speeds pwm signals for all propellers $[n_1 \dots n_8]$.

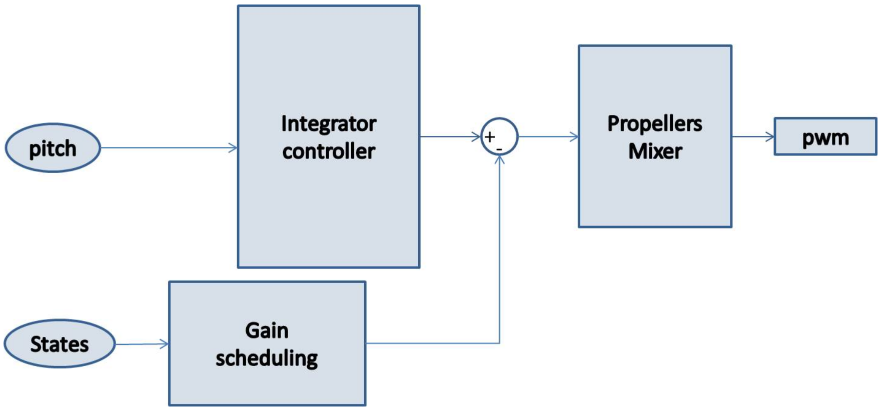


Figure 3.3 *Controller* block for longitudinal model

3.2.3. Airframe + Environment

The *Airframe + Environment* block would correspond to the plant. This represents the vehicle itself and the environmental conditions that will affect the flight of the prototype. The input of this block is the vector with the propellers rotational speeds. And the outputs are two vectors:

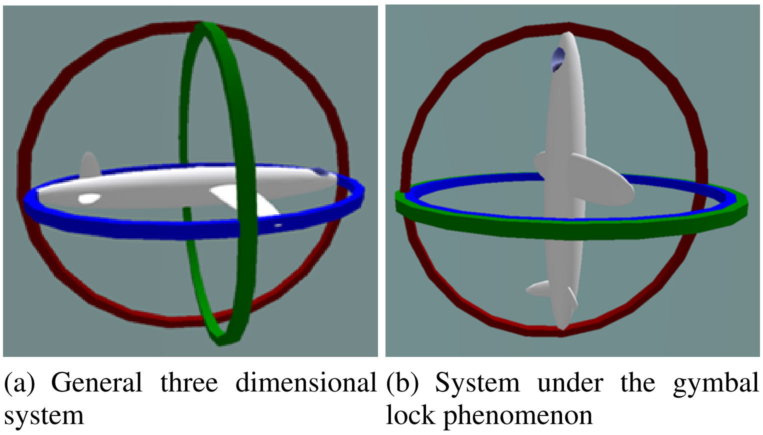
1. **Environment.** This vector contains gravity g $[m/s^2]$, air temperature T $[K]$, speed of sound a $[m/s]$, air pressure p $[Pa]$ and air density ρ $[kg/m^3]$. These variables depend on the vehicle position in the space.

2. **States.** This vector contains latitude, longitude and altitude of the vehicle LLA $[deg \quad deg \quad m]$, velocity in the flat Earth Reference Frame V_e $[m/s]$, position in the flat Earth Reference Frame X_e $[m]$, Euler rotation angles ϕ, θ, ψ $[rad \quad rad \quad rad]$ (roll, pitch, yaw), a coordinate transformation from flat Earth axes to body-fixed axes DCM_{be}, velocity in Body Reference Frame V_b $[m/s]$, angular rates in body-fixed axes ω_b $[rad/s]$, angular accelerations in body-fixed axes $d\omega_b/dt$ $[rad/s^2]$, accelerations in body-fixed axes with respect to Body Reference Frame A_{bb} $[m/s^2]$ and accelerations in body-fixed axes with respect to Inertial Frame (flat Earth) A_{be} $[m/s^2]$.

3.2.4. Quaternions

For the general model, including both longitudinal and lateral-directional models, instead of working with Euler angles it was decided to work with quaternions. The decision of using quaternions instead of Euler angles was taken to avoid the Gimbal Lock singularity. Gimbal lock consists in the loss of one degree of freedom in three-dimensional systems. This occurs when the second angle in the rotation sequence is equal to 90°, which makes two axes align and degenerate into a two-dimensional space. A graphic representation of Gimbal Lock is shown in Figure 3.4.

As seen in Figure 3.4, rotations in the three degrees of freedom are executable: the spin of green ring involves a rotation around $x-axis$, the spin of red ring involves a rotation around $y-axis$ and the spin of blue ring involves a rotation around $z-axis$. However, if the red ring does a rotation of 90°, blue and green rings end up aligned and blocked. This means that, in the situation of Figure 3.4(b), no matter if green or blue ring rotate, it will have the same consequence for the aircraft movement, in this case a rotation around $x-axis$. That is why it is said that a degree of freedom has been lost, in particular rotation around $z-axis$ (yaw). So Gimbal Lock must be avoided to keep the control of the vehicle.



(a) General three dimensional system

(b) System under the gymbal lock phenomenon

Figure 3.4 Gimbal lock singularity

As a first attempt, is was considered the option of forcing the singularity to appear in a position that the vehicle will never reach. For example, $\psi = \pm 90°$ for vertical mode, this is having the vehicle as shown in Figure 3.5, which will never happen in a normal flight.

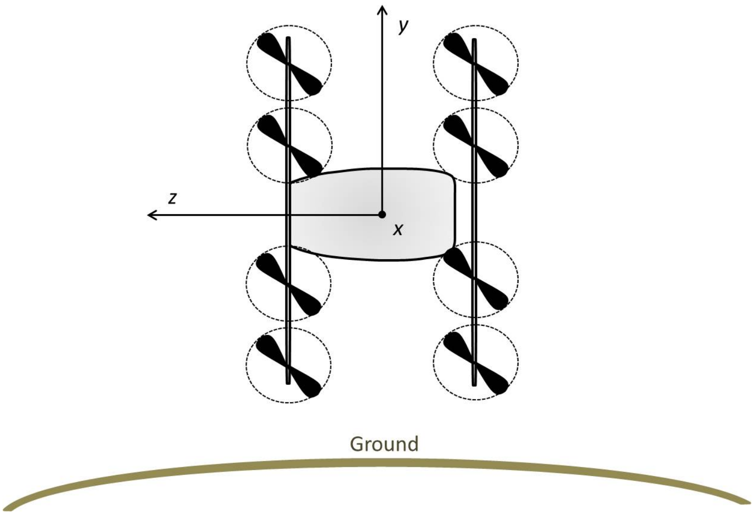


Figure 3.5 Position desired for the placement of the singularity in vertical flight

Despite the fact that this situation will not appear for vertical flight, it can perfectly happen in horizontal flight. For this reason, it was considered to place the singularity for one position in vertical flight mode and for a different position in horizontal flight mode. In horizontal flight mode, this impossible position would be $\phi = \pm 90°$, as shown in Figure 3.6.

Doing some research on VTOL and quadcopter control, it was concluded that trying to solve the singularity problem this way would be harder than the most common solution adopted: the use of quaternions instead of Euler angles [16]. This is because the *impossible* position in vertical flight mode will still appear in horizontal flight mode and vice-versa and this would force the necessity to distinguish somehow the two flight modes, separation which from the beginning was intended to avoid.

In conclusion, for the reasons mentioned above it was decided to use quaternions for the general model. However, for the longitudinal model it is not necessary tu use quaternions because there is only one rotation considered: pitch.

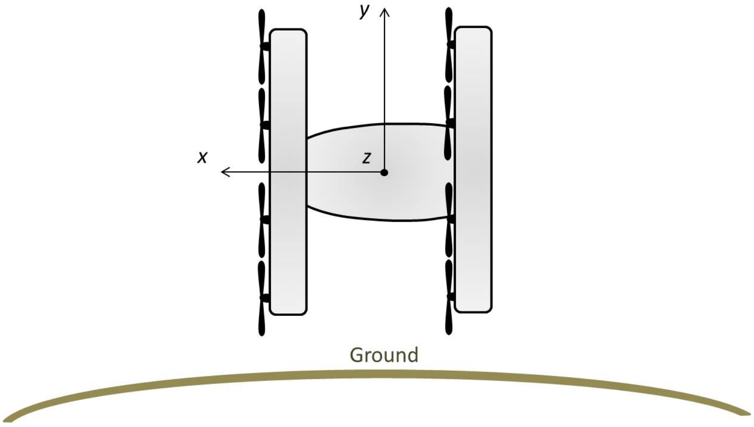


Figure 3.6 Position desired for the placement of the singularity in horizontal flight

Mathematically, it is worth noticing from Equation 3.1 that the expressions of $\dot{\phi}$ and ψ depend respectively on $\tan\theta$ and $\sec q$. Both are undefined for $\theta = \pi/2rad$, which is the aircraft hover flight attitude. This shows clearly the numerical instability of the Euler rotations. As mentioned before, this issue is solved by substituting the Euler angle representation of the aircraft attitude by quaternions, as they are more stable as given in [17].

To obtain the quaternions, first it is necessary to start from rotational speeds p, q, r to calculate the rate of change of quaternions $[\dot{q}_0 \quad \dot{q}_1 \quad \dot{q}_2 \quad \dot{q}_3]$, as expressed in Equation 3.3.

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{3.3}$$

In order to fully enjoy the stability of quaternions, the rotation matrix given in Equation 2.12 is also approximated using quaternions as shown in Equation 3.4 (Section 2.4.2).

$$DCM = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2\cdot(q_1\cdot q_2+q_3\cdot q_0) & 2\cdot(q_1\cdot q_3-q_2\cdot q_0) \\ 2\cdot(q_1\cdot q_2-q_3\cdot q_0) & q_0^2-q_1^2+q_2^2-q_3^2 & 2\cdot(q_2\cdot q_3+q_1\cdot q_0) \\ 2\cdot(q_1\cdot q_3+q_2\cdot q_0) & 2\cdot(q_2\cdot q_3-q_1\cdot q_0) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix} \tag{3.4}$$

The structure of the *Airframe & Environment* block is represented in Figure 3.7.

Being the quaternions modification included in the *Equations of Motion* block.

AC model block

The inputs *Environment* and *pwm* along with the feedback loop *States* are used by block *AC model* to compute the forces $[F_x \quad F_y \quad F_z]$ and moments $[M_x \quad M_y \quad M_z]$ acting on the vehicle.

A global representative view of this block is shown in Figure 3.8.

For the propellers, first of all there is a block to compute the thrust of each of them, which added give a total thrust that contributes to the force along $x-axis$. Besides, there is a calculation of the moments generated by all the propellers according to their relative position.

Propellers characteristics are defined in Appendix Section A.4 and as speed it is used the Body Reference speed in $x-axis$ direction V_{b_x}. The distances used to compute the moments are defined in Section 2.4.3. As mentioned before, moment around $x-axis$ due to propellers thrust times their

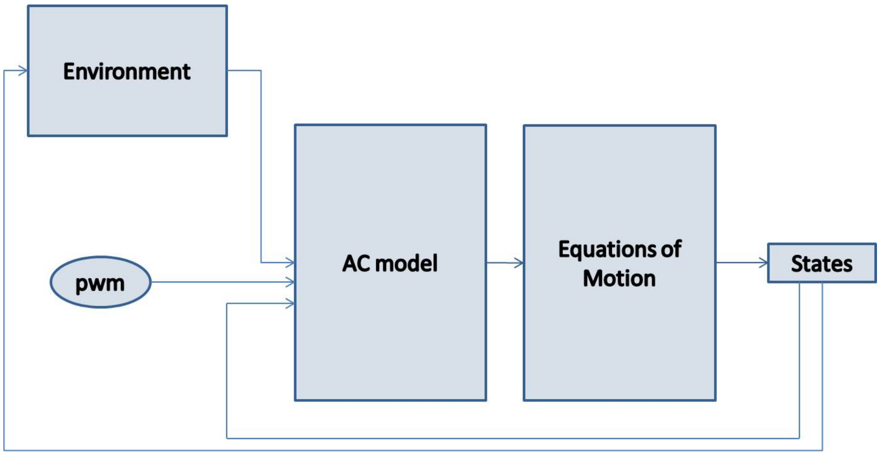


Figure 3.7 *Airframe and environment* block

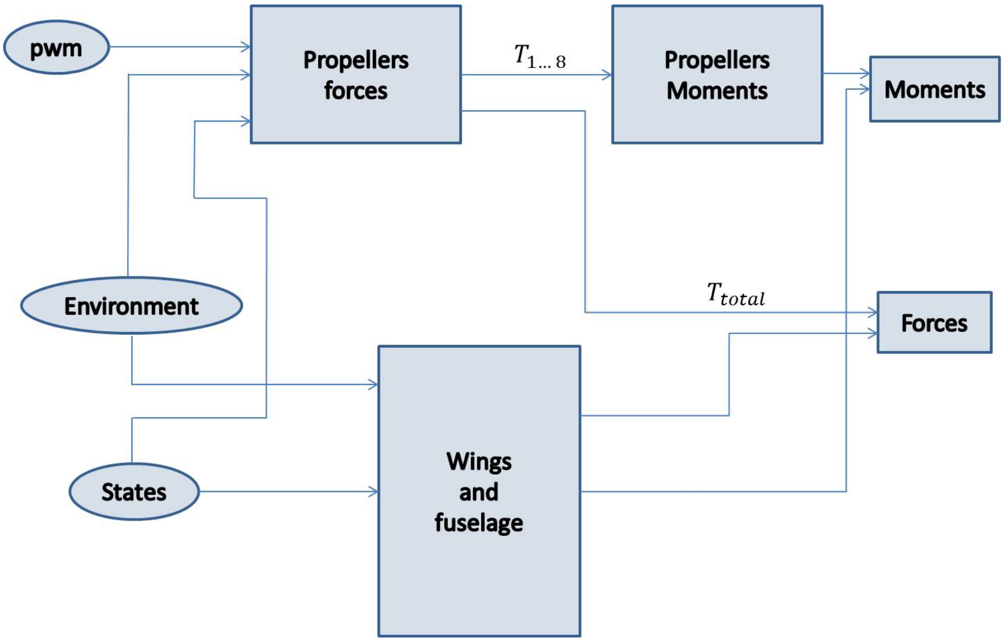


Figure 3.8 *AC model* block

distance to $x - axis$ will always be 0 because this force is parallel to the axis, but there will be a moment around $x - axis$ caused by the torque. This is necessary for the vehicle to achieve a rotation around $x - axis$, which will be yaw in vertical flight mode and roll in horizontal flight mode.

With respect to the wings and the fuselage forces and moments, they are computed separately for each element and then added to have a global moments vector and a global forces vector. These two vectors are later added to the forces and moments created by the propellers.

For both wings, the process is the same, but they have different distances values (Section 2.4.1) and different induced angles of attack. For each wing, the simulator calculates the forces in Body Frame and the moments created by these forces. It starts from an aerodynamic forces vector expressed in Wind Frame transformed to in Body Reference Frame by using the matrix explained in Section 2.4.1.

For now, the coefficient $C_Y = C_{Y_\beta}\beta$ is an estimated value. However, the intention is to do an aerodynamic study with some software to determine a more accurate value for this coefficient because of the presence of the fuselage and also for all the other aerodynamic coefficients.

For C_L, the simulator uses the graphic shown in Section A.3, and C_D is computed with Equation 2.7.

The third part of the *Wings & Fuselage* block is the section related to weight. In this case, weight only contributes to the forces, it does not create any moment because it is supposed to be applied in the centre of gravity, as explained in Section 2.4.2.

For this particular force, the transformation needs to be done from Inertial Frame to Body Frame, as explained in Section 2.4.2.

Finally, fuselage forces and moments are computed. For this block it is done again a transformation from Wind Frame to Body Frame, as explained in Section 2.4.1.

Returning to main block *Wings & Fuselage*, there is a little block called V_b *to* α *and* β. This block includes a logical sequence to deal with certain situations where some of the speed components are zero. To calculate α and β the simulator uses Equations 2.2 and 2.3. According to these equations there will be an indeterminacy when u or V are equal to zero. Because of this, there is a logical sequence to prevent SIMULINK from doing the calculations in those cases.

The sequence built is composed of 10 different cases. Each of them corresponds to a different indeterminacy and if they are not taken into account they will lead to an impossible calculation by SIMULINK. The 9 first cases entail a defined value for α and β, but in the last case instead of be given a value to α and β the angles are calculated by the Equations 2.2 and 2.3, because there will not be a mathematical uncertainty. It is also important to remark that inside this block, every time α or β are computed, there is also carried out a calculation of α_i (2.20) and β_i (2.21) that give a final set of effective angles, which are the ones used.

With this method miscalculations are avoided by SIMULINK and the simulator will not give wrong results due to these indeterminacies.

Besides, each wing separately contains a Matlab function calculating the induced angle due to propellers interaction.

6 *DOF* block

Once all the forces and moments are added and there is a three element vector for both acting variables, all the parameters in the *States* bus can be calculated.

The block used for this is a default block from SIMULINK which computes all the variables starting from the forces and moments applied on the body and expressed in Body Frame. The body is assumed to be rigid and its rotations are about the Inertial Frame. All the equations used by this block are expressed in Section 3.1.

The mass of the body is considered constant $m = 2.626\ kg$ and the inertia tensor is with respect to the centre of gravity of the body (Section A.2).

Before building the *States* bus, there are two variables used by other blocks: position in the Inertial Frame X_e and Euler angles $[\phi \quad \theta \quad \psi]$. The first of them, X_e, is used to determine the latitude, longitude and altitude of the vehicle through the block *Position on Earth*. It is important to remark that the value obtained from the third component of X_e is x_z, which points to the centre of Earth. The altitude h is considered in the opposite direction, that is why it is checked if $x_z > 0$ which necessarily means that $h < 0$ and the simulation needs to be stopped.

The sign of this variable is changed inside *Flat Earth to LLA* block. This SIMULINK block takes X_e and a reference height and calculates geodetic latitude, longitude and altitude. The first two variables are given in degrees and altitude in meters.

Moreover, the Euler angles are limited to a range between $-180°$ and $180°$ so it is easier to read the results.

Environment (Variable)

The altitude obtained from *Position to Earth* block is used by *Environment* block to calculate the current environmental conditions.

The environment model is implemented in two parts:

- Gravity(WGS84): Implemented using the World Geodetic System 1984 [19], which is a mathematical representation of the geocentric equipotential ellipsoid of the World Geodetic System (WSG84), as estimated by the US department of defence in 1984. For the model built, it gives the gravity value at given longitude and latitude coordinates.
- Atmosphere(COESA): Implemented using the U.S. Standard Atmosphere model updated in 1976 by the U.S. Committee on Extension to the Standard Atmosphere [20]. It is a static mathematical representation of how the pressure, temperature, density and speed of sound change over a wide range of altitudes.

3.2.5. Visualization

The *Visualization* block take the current *States* and uses them to represent the different variables as a pilot would see them, with several displays and flight instruments.

3.3. Longitudinal and Lateral-Directional dynamics

In order to analyse more effectively the model created, it was divided in two different models: longitudinal and lateral-directional. This allows to identify the dynamic modes and determine the stability of the aircraft.

3.3.1. Longitudinal dynamics

Considering longitudinal flight involves only take into account forces contained in the plane of symmetry of the aircraft and moments orthogonal to the mentioned plane. There are several assumptions that have been made:

- Aerodynamic forces are contained in the plane of symmetry
- Aerodynamic moments are perpendicular to the plane of symmetry
- Wings are balanced in the local horizontal plane
- The plane of symmetry is vertical and it contains the weight vector

Expressing all the assumptions in a numerical way, this is: $\beta = 0°$, $\phi = °$, $p = r = 0\ rad/s$ and $\psi = cte$.

The 4^{th} order longitudinal equations of motion have been used in order to study the flight dynamic modes. Equation 3.5 shows the states and Equations 3.6, 3.7, 3.8 and 3.9 are the non-linear expressions used to obtain a pure longitudinal model.

$$\begin{bmatrix} \theta \\ u \\ w \\ q \end{bmatrix} = \begin{bmatrix} Pitch \quad Angle \quad [rad] \\ Axial \quad Velocity \quad [m/s] \\ Vertical \quad Velocity \quad [m/s] \\ Pitch \quad Rate \quad [rad/s] \end{bmatrix} \tag{3.5}$$

$$\dot{\theta} = q \tag{3.6}$$

$$\dot{u} = \frac{F_x}{m} - qw \tag{3.7}$$

$$\dot{w} = \frac{F_z}{m} + qu \tag{3.8}$$

$$\dot{q} = \frac{My}{I_{yy}} \tag{3.9}$$

3.3.2. Lateral-Directional dynamics

In the lateral-directional case it is not possible to obtain a model concerning only this motion, it is necessary to include some longitudinal variables such as α, q and V. So it is assumed that these variables are constant, getting the numerical values from a steady-state longitudinal flight condition.

Equation 3.10 shows the states used to represent the lateral-directional flight dynamics and Equations 3.11, 3.12, 3.13 and 3.14 show the non-linear dynamic equations.

$$\begin{bmatrix} r \\ \beta \\ p \\ \phi \end{bmatrix} = \begin{bmatrix} Yaw \quad Rate \quad [rad/s] \\ Sideslip \quad Angle \quad [rad] \\ Roll \quad Rate \quad [rad/s] \\ Roll \quad Angle \quad [rad] \end{bmatrix} \tag{3.10}$$

$$\dot{r} = \frac{I_{xz}Mx + I_{xx}Mz}{I_{xx}I_{zz} - I_{xz}^2} \tag{3.11}$$

$$\dot{\beta} = \frac{F_y}{mV} - r + p\alpha \tag{3.12}$$

$$\dot{p} = \frac{I_{zz}Mx + I_{xz}Mz}{I_{xx}I_{zz} - I_{xz}^2} \tag{3.13}$$

$$\dot{\phi} = p + (q\sin\phi + r\cos\phi)\tan\theta \tag{3.14}$$

Chapter 4

Control

Once the aircraft model is defined, is the moment to design the control laws. These have been applied to the longitudinal model, which has been the one giving good results after the verification tests (Section 5.1.1). Linear control theory approach is the one used in this study to design the controller. For this reason, it is necessary to set different equilibrium points to linearize the model around them.

4.1. Linearization

The linearization of the model is valid within a small region around the equilibrium point and the linearization algorithm can be applied to any non-linear model in the same explicit state-space form that was used with numerical integration.

4.1.1. Equilibrium points

As developed by [12], a Taylor series expansion of a non-linear differential equation $\dot{x}(t) = f(x(t), u(t))$ around a point (x_e, u_e) gives

$$\dot{x} + \dot{\delta}_x = f(x_e, u_e) + \frac{\partial f}{\partial x}\delta x + \frac{\partial f}{\partial u}\delta_u + h.o.t.$$

being $h.o.t.$ "higher-order terms", which are neglected. The partial derivative terms represent Jacobian matrices and the perturbations δ_x and δ_u are small.

Then, if x_e and u_e are equilibrium points obtained from the trim it is satisfied that:

$$\dot{x} = f(x,u) = 0 \tag{4.1}$$

and so

$$\dot{\delta}_x = \frac{\partial f}{\partial x}\delta_x + \frac{\partial f}{\partial u}\delta_u$$

For the case in study, Equation 4.1 is true if: $\sum F(x,u) = \sum M(x,u) = 0.$

The equilibrium points are computed using a numerical algorithm, by forming a scalar cost function from the sum of the squares of the derivatives of the states ($\dot{x}$). A function minimization algorithm is then used to adjust the control variables and the appropriate state variables to minimize this scalar cost. These algorithms are carried out with Matlab *findop* function, which also performs the minimization with some constraints.

The equilibrium points of interest are inside the flight envelope, which in this case is from $\theta = 0°$ to $\theta = 90°$.

4.1.2. Jacobian linearization

Considering a system $\dot{x} = f(x,u)$ with an initial condition $x(t_0) = x_0$ and an equilibrium point compound by a nominal input $\overline{u}(t)$ which corresponds to a nominal state $\overline{x}(t)$, such that $\dot{\overline{x}} = f(\overline{x},\overline{u})$ and $\overline{x}(t_0) = x_0$, the method obtained from [21] says that if a perturbation is applied to input u that creates a perturbation in state x such that

$$\delta_x(t) = x(t) - \overline{x}$$
$$\delta_u(t) = u(t) - \overline{u}$$

And substituting it is obtained

$$\dot{\delta_x} = f(\overline{x} + \delta_x, \overline{u} + \delta_u) \tag{4.3}$$

Then, performing a Taylor series expansion on the left side of the Equation 4.3, neglecting all terms of order higher than 1^{st} and considering that $f(\overline{x},\overline{u}) = 0$ it is obtained the expression in Equation 4.4.

$$\dot{\delta_x}(t) \approx \left.\frac{\partial f}{\partial x}\right|_{(x=\overline{x},x=\overline{u})} \delta_x(t) \quad + \quad \left.\frac{\partial f}{\partial u}\right|_{(x=\overline{x},u=\overline{u})} \delta_u(t) \tag{4.4}$$

This approximation is valid as long as $\delta_x(t)$ and $\delta_u(t)$ remain small. Equation 4.4 is a linear, time-invariant, differential equation, since the derivatives are linear combinations of the δ_x variables and the perturbation inputs, δ_u.

The first partial derivative is a Jacobian matrix of order *n x n* and identified as A. The second partial derivative also corresponds to a Jacobian matrix of order *n x m* and is called B.

Matrices A and B are constant and define the Jacobian linearization of the original non-linear system $\dot{x} = f(x,u)$ around the equilibrium point $(\overline{x},\overline{u})$. This new linear system is given in state space form in Equation 4.5.

$$\dot{\delta_x}(t) = A\delta_x(t) \quad + \quad B\delta_u(t) \tag{4.5}$$

This is an effective way to deal with non-linear systems in a linear manner. However, as mentioned, its limitation is that perturbations must be small and the controller created will work well when the system is operating near the equilibrium point.

The above mentioned algorithm has been performed with Matlab function *linmod*, getting as output a linearized model in state space form.

4.2. Prerequisites

In order to apply a control method it is necessary to check two fundamental concepts that, if not verified, the control methods proposed cannot be implemented.

4.2.1. Controllability

A dynamic system is said to be completely controllable if it is possible to transfer, through the control inputs, the system itself from an arbitrary initial state $\overline{x}_0(t)$ to an arbitrary desired final state $\overline{x}_f(t)$ in a finite interval of time ([22]). If some state variable is independent from the control inputs then it will be impossible to control such variable and the system will not be completely controllable. In other words, it is possible to relate the controllability of a system with the accessibility to the state variables by the inputs of control.

For a linear time-invariant (LTI) system

$$\begin{cases} \dot{x}(t) &= Ax(t)+Bu(t) \\ y(t) &= Cx(t)+Du(t) \end{cases} \tag{4.6}$$

where A^{nxn} is the *state matrix*, B^{nxm} is the *input matrix*, C^{lxn} is the *output matrix* and D^{lxm} is the *direct transmission matrix*. In the case in study, the states vector $x(t)$ is compound of the pitch rate q, the pitch angle θ, and the velocities u and w; the input vector $u(t)$ has two elements: the basis rotational speed for all propellers and D_y that is the differential of rotational speed needed to create a pitch variation; and the output vector $y(t)$ is formed by θ, u, w, q, α, *thrust* and *RPM*, as seen in Figure 5.13.

For the system to be completely controllable it is necessary and sufficient for the rank of the controllability matrix Co to be equal to the number of states n, $rank(Co) = n$ [23]. The controllability matrix is built from A and B as follows,

$$Co = \begin{bmatrix} B & AB & A^2B & \ldots & A^{n-1}B \end{bmatrix}$$

To determine such expression it is considered the generic discrete linear system:

$$\overline{x}(k+1) = \overline{A}\overline{x}(k) + \overline{B}\overline{u}(k)$$

For $k = 0, 1, 2\ldots$ there is the following sequence:

$$\begin{array}{rcl} \overline{x}(1) &=& \overline{A}\overline{x}(0)+\overline{B}\overline{u}(0) \\ \overline{x}(2) &=& \overline{A}\overline{x}(1)+\overline{B}\overline{u}(1) = \\ &=& \overline{A}(\overline{A}\overline{x}(0)+\overline{B}\overline{u}(0))+\overline{B}\overline{u}(1) = \\ &=& \overline{A}^2\overline{x}(0)+\overline{A}\overline{B}\overline{u}(0)+\overline{B}\overline{u}(1) \\ \overline{x}(3) &=& \overline{A}\overline{x}(2)+\overline{B}\overline{u}(2) = \\ &=& \overline{A}^3\overline{x}(0)+\overline{A}^2\overline{B}\overline{u}(0)+\overline{A}\overline{B}\overline{u}(1)+\overline{B}\overline{u}(2) \\ \ldots && \end{array}$$

In general for $k = n-1$ it is obtained:

$$\overline{x}(n) = \overline{A}^n\overline{x}(0) + \sum_{j=0}^{n-1} \overline{A}^{n-j-1}\overline{B}\overline{u}(j)$$

Rearranging the previous expression in matrix form it is:

$$\overline{x}(n) - \overline{A}^n\overline{x}(0) = \underbrace{\begin{bmatrix}\overline{B} & \overline{AB} & \overline{A}^2\overline{B} & \ldots\overline{A}^{n-1}\overline{B}\end{bmatrix}}_{Co}\begin{bmatrix}\overline{u}(n-1)\\ \overline{u}(n-2)\\ \vdots \\ \overline{u}(1)\\ \overline{u}(0)\end{bmatrix}$$

At this moment it is possible to calculate the necessary sequence of input to "move" the system from the generic initial arbitrary state $\overline{x}(0)$ to the generic arbitrary final state $\overline{x}(n)$, in n intervals of time, through the expression:

$$\begin{bmatrix}\overline{u}(n-1)\\ \overline{u}(n-2)\\ \vdots \\ \overline{u}(1)\\ \overline{u}(0)\end{bmatrix} = \begin{bmatrix}\overline{B} & \overline{AB} & \overline{A}^2\overline{B} & \ldots\overline{A}^{n-1}\overline{B}\end{bmatrix}\left\{\overline{x}(n) - \overline{A}^n\overline{x}(0)\right\}$$

All this, however, is possible if and only if $rank(Co) = n$, which implies that the controllability matrix Co has n columns linearly independent. If this condition is verified, then the system is said to be completely controllable.

If $m = 1$ then $\overline{u}(k)$ will be a scalar and the controllability matrix Co will be square matrix (nxn). In general, though, $\overline{u}(k)$ is a vector of dimension m and then Co will be a rectangular matrix of dimensions $nxnxm$. In this case the controllability of the system can be verified analysing one single input $u_j(k)$ at a time. If there exists an input $u_j(k)$ with $j = 1\ldots m$ that verifies the controllability of all the states thus the system turns out to be completely controllable with m inputs.

In this project the controllability matrix has been obtained with Matlab function *ctrb* on the basis of matrices A and B, obtained in a previous step.

4.2.2. Observability

A dynamic system is said to be completely observable if all the states $\overline{x}$ can be determined by the measurement of the output signals $\overline{y}$ in a finite interval of time ([24]). If one or more states cannot be identified from the outputs of the system, then it is not observable. Similarly to what seen for controllability, is given a mathematical definition of observability.

For a linear time-invariant (LTI) system as the one seen in Equation 4.6, it is considered completely observable if the rank of the observability matrix is equal to the number of states n [23]. The observability matrix Ob is build from A and B as follows

$$Ob = \begin{bmatrix}C^T & A^TC^T & (A^T)^2C^T & ... & (A^T)^{n-1}C^T\end{bmatrix}$$

The observability of the system is possible because there are several variables measured by the different sensors on board. These are angular rates $(p \quad q \quad r)$ and accelerations. And the on-board sensors are: 9DOF IMU (accelerometer, gyroscope and magnetometer), GPS and pitot tube.

Numerically, the Matlab function *obsv* has been used to compute the observability matrix.

4.3. Control Method

Control techniques allow to create Stability Augmentation Systems (SAS) and auto-pilots. In fact, the main objective of a regulator is to guarantee the stability of the system in closed-loop making any perturbations or errors from the state with respect to the initial conditions tend to zero. Additionally, controllers must guarantee "good" time responses as well as satisfy the desired specifications.

Hereafter there is a description and detailed explanation of the control laws developed and other possible methods. Besides, it is also explained the stability augmentation system used to meet the flying qualities specifications and keep the controllability of the aircraft during the whole flight envelope.

4.3.1. Pole placement

The full state feedback using pole placement method is chosen as the control method. This choice was made because of the fact that this method allows the placement of the closed loop system poles in pre-determined locations in the s-plane.

The pole placement method can be applied to a system that fulfil the following requirements:

- The system is completely controllable
- The state variables are measurable and are available for feedback

In this case, as explained in Section 4.2.1 and Chapter 3, both conditions are satisfied.

The objective of the pole placement method is to make all closed loop poles lie at some desired locations.

Based on the linear system:

$$\left\{\begin{array}{lcl} \dot{x} & = & Ax+Bu \\ y & = & Cx+Du \end{array}\right. \tag{4.7}$$

the control vector u is designed in the following state feedback form:

$$u = -Kx \tag{4.8}$$

The first step is to compute the control gain matrix K that takes the full state feedback and changes the system dynamic matrix A, such that the closed loop system has poles at specific locations $\mu_1 \ldots \mu_n$, as explained before.

Substituting Equation 4.8 in Equation 4.7 it is obtained the state-space equation for the closed loop system:

$$\left\{\begin{array}{lcl} \dot{x} & = & (A-BK)x \\ y & = & Cx \end{array}\right. \tag{4.9}$$

The stability and performance of the closed loop feedback system are determined by the location of the eigenvalues of the matrix $A-BK$, which corresponds to the poles of the closed loop system. In other words, the gain matrix K must be designed in such a way that:

$$|sI-(A-BK)| = (s-\mu_1)(s-\mu_2)\ldots(s-\mu_n) \tag{4.10}$$

where $\mu_1,\ldots,\mu_n$ are the desired pole locations. By choosing the state feedback gain matrix K the objective is to set the poles in the mentioned locations, as long as the values do not lead to the saturation of the actuators.

Bass-Gura approach

The system is expressed in *companion* form, the state matrix A and the control matrix B are expressed in the following specific form:

$$A = \begin{bmatrix} -a_1 & -a_2 & -a_3 & \dots & -a_n \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{4.11}$$

where a_i are the coefficients of the differential equation, as well as the coefficients of the characteristic equation of the system open-loop are:

$$|sI - A| = s^n + a_1 s^{n-1} + \dots + a_{n-1}s + a_n = 0$$

If the State Feedback control scheme is applied, the state matrix A^* in closed-loop will be:

$$A^* = A - BK = \begin{bmatrix} -a_1 - k_1 & -a_2 - k_2 & \dots & -a_{n-1} - k_{n-1} & -a_n - k_n \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & & \vdots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} \tag{4.12}$$

whereas the characteristic equation in closed-loop with the eigenvalues desired has the form:

$$|sI - A^*| = |sI - A + BK| = s^n + \alpha_1 s^{n-1} + \dots + \alpha_{n-1}s + \alpha_n = 0$$

So then it can be expressed:

$$-a_i - k_i = -\alpha_i \quad \text{with} \quad i = 1, \dots, n$$

from which is obtained:

$$K = [\alpha_1 - a_1 \dots \alpha_{n-1} - a_{n-1} \quad \alpha_n - a_n]$$

Therefore, Bass-Gura method can be used to numerically determine the scalar gains of K matrix for a desired location of the eigenvalues. In general, anyway, the state matrix A of the system is not of the *companion* form and thus the expression used to calculate K is modified as follows:

$$K = [\alpha_1 - a_1 \dots \alpha_{n-1} - a_{n-1} \quad \alpha_n - a_n]\,(VW)^{-1}$$

where a_i are the coefficient of the characteristic equation in open-loop, α_i are the coefficients of the characteristic equation in closed-loop with the desired poles. V is the controllability matrix of the system and W is a triangular matrix of dimensions *nxn*:

$$W = \begin{bmatrix} 1 & a_1 & \dots & a_{n-2} & a_{n-1} \\ 0 & 1 & \dots & a_{n-3} & a_{n-2} \\ 0 & 0 & \dots & \vdots & \vdots \\ \vdots & \vdots & & 1 & a_1 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

LQR (Linear Quadratic Regulator)

LQR technique can be developed using two different feedbacks: output feedback or full state feedback. In this case, as there is access to all states, the second technique it the one considered as a possible implementation option. However, it is easier to explain the full state feedback option from the mathematical process used for the output feedback option.

LQR technique pursue the determination of the gain feedback matrix K of dimensions mxp that minimizes the value of a cost function. The minimization of this function must guarantee the system stability and a good time response and so it is selected from the time domain. The expression of the cost function is of the type:

$$J = \frac{1}{2}\int_0^\infty (\overline{x}^T Q\overline{x} + \overline{u}^T R\overline{u})dt$$

where Q and R are weight matrices, symmetric, non singular and positive semi-definite: this involves a positive value for J and consequently a minimization problem. The fact that J can reach a minimum finite value for an pertinent choose of the input vector $\overline{u}$ implies that the integrand must tend to zero and then the state vector $\overline{x}$ and the control $\overline{u}$ must be zero for high time values, all of it is appropriate for a regulator characteristics.

Going back to the model, the system in closed-loop has the expression shown in Equation 4.9. The cost function can be expressed in terms of K:

$$J = \frac{1}{2}\int_0^\infty \overline{x}^T (Q + C^T K^T RKC)\overline{x} dt$$

In this case, the design of the regulator entails the determination of the gains K that minimizes J with the conditions of dynamic association: $\dot{\overline{x}} = (A - BKC)\overline{x}$.

To simplify the resolution of such problem it is an option to convert the dynamic minimization into an static minimization through the introduction of an auxiliary matrix P constant, symmetric and positive semi-definite that verifies the following expression:

$$\frac{d}{dt}(\overline{x}^T P\overline{x}) = -\overline{x}^T (Q + C^T K^T RKC)\overline{x}$$

Now J can be expressed as

$$J = \frac{1}{2}\overline{x}^T(0)P\overline{x}(0) - \frac{1}{2}\lim_{t\to\infty}\overline{x}^T(t)P\overline{x}(t)$$

With the hypothesis of the system in closed-loop asymptomatically stable, $\overline{x}^T(t)$ fades with time so J becomes:

$$J = \frac{1}{2}\overline{x}^T(0)P\overline{x}(0) \tag{4.13}$$

Combining the expression of P with the equation of the system in closed-loop $\dot{\overline{x}} = \underbrace{(A - BKC)}_{A_{CL}}\overline{x}$ it easy to see that:

$$\begin{aligned} -\overline{x}^T (Q + C^T K^T RKC)\overline{x} &= \frac{d}{dt}(\overline{x}^T)P\overline{x} = \dot{\overline{x}}^T P\overline{x} + \overline{x}^T P\dot{\overline{x}} \\ &= \overline{x}^T (A_{CL}^T P + PA_{CL})\overline{x} \end{aligned}$$

This relation, having to be verified for any initial condition and therefore for any trajectory of the states $\overline{x}(t)$, leads to the following relation:

$$g \equiv A_{CL}^T P + PA_{CL} + C^T K^T RKC + Q = 0$$

known as Lyapunov's Equation with Q and R known and P which must be determined after defining K.

Summarizing , for any K, if there exists a constant matrix P, symmetric and positive semi-definite that satisfies Lyapunov's Equation and if the system in close-loop is stable, then J will be in terms of P as previously shown in Equation 4.13. the matrix P turns out to be completely independent from the states $\overline{x}(t)$ and the cost function can be computed knowing only P and the initial condition of the states $\overline{x}(0)$.

LQR technique applied to full state feedback controllers involves the determination of the optimal matrix K through the solutions of similar equations but much more simplified with respect to the output feedback ($\overline{u} = -K\overline{y} = -KC\overline{x}$). In fact, it is sufficient to consider that $C \equiv I$ in the mathematical reasoning developed for LQR with output feedback.

Ackermann

This method computes the gain matrix K through the controllability matrix using the formula of Ackermann [25], shown in Equation 4.14.

$$K = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix}^{-1} \phi(A) \tag{4.14}$$

where $\phi(A)$ is

$$\phi(A) = A^n + \alpha_1 A^{n-1} + \dots + \alpha_{n-1} A + \alpha_n I$$

and α_i are the coefficients of the desired characteristics polynomial.

This algorithm for a given system and specific poles locations is computed by the Matlab function *place*. The desired closed loop locations are chosen taking into account the flying qualities requirements, without saturating the actuators, and avoiding the poles that might produce gains that would create physically improbable behaviour.

The dynamic behaviour of the aircraft changes for different equilibrium points. Performances get poor or the aircraft gets unstable for different flight conditions. There is therefore a need for a non-linear controller. But these are usually complex and expensive to implement. It was used a method that seems simple and efficient, while being linear.

This method of computing the gain matrix K is the chosen for this project due to its simplicity and the fact that the controllability matrix already computed can be used.

4.3.2. Gain scheduling

Considering the non-linearity of the vehicle, a linear controller will work only within a region around the linear operating point. Gain scheduling is a linear control method using a set of gains that change based on a defined schedule [26]. The schedule is defined based on a set of operating points. This means gain are changing relatively with the states of the aircraft, as they go through different operating points.

For the vehicle in study, the first idea was to use two scheduling variables: pitch angle θ, which defines whether the aircraft is in vertical flight mode, transition or horizontal flight mode and velocity V which changes considerably from take-off to horizontal flight mode, and from horizontal flight mode to landing. But it turned out to be more complicated because of the the lack of a proper relationship between velocity V and θ. So it was decided to use only velocity, taking into account the fact that velocity is small for vertical flight mode and bigger for horizontal flight mode. Some trim points fulfilling this criteria have been selected.

Once there are different gain matrices, a gain interpolation is done between elements of the matrices in order to get a global K matrix depending on velocity.

The idea of using both V and θ as scheduling variables and create with them a gain surface it is suggested for further steps in the project.

Reference Tracking: Integral Control

The control law previously defined stabilizes the aircraft and helps meet the flying qualities requirement. However, in practice, the idea is for the aircraft to follow some input commands.

The use of integral control helps achieve asymptotic tracking. This choice is driven by the fact that integral control is robust and guarantees to obtain asymptotic tracking as long as the closed-loop system is stable. But the inconvenient is that it increases the dimension of the system ([27]).

The equations of a closed-loop linear system can be defined as seen in Equation 4.9.

Augmenting the system with a differential equation representing the integral

$$\dot{\xi} = y - y_d$$

being the initial condition $\xi_0 = 0$ and y_d denotes the desired constant value for the output y to track asymptotically. Since y and y_d are both known, $\xi(t)$ can be determined from ξ_0 with Equation 4.15.

$$\xi(t) = \int_0^t [y(\tau) - y_d(\tau)] d\tau \tag{4.15}$$

The augmented system can be written in the form shown in Equation 4.16.

$$\begin{bmatrix} \dot{x} \\ \dot{\xi} \end{bmatrix} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ \xi \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ -I \end{bmatrix} y_d \tag{4.16}$$

The control law is of the form

$$u = -Kx - K_i \xi$$

If the augmented system, is stable, then $\lim_{t \to \infty} \xi(t) = 0$. This results in $\lim_{t \to \infty} y(t) = y_d$, which means the asymptotic reference tracking is achieved. The robustness is shown by the fact that even if the system is perturbed, as long as the matrices A and B are stable, then $\lim_{t \to \infty} \xi(t) = 0$ still guarantees asymptotic tracking.

The inner loop of the SIMULINK diagram of this method contains the gain matrix K computed using the pole placement method which takes the full states feedback and changes the dynamic of the aircraft in order to stabilize it.

The outer loop takes the variable on which it is desired to track reference, q in this case, and feeds it back to the reference input such that $e(t) = y - y_d$. The error is then fed to the integral controller

Anti wind-up

An unavoidable problem in flight control is the existence of limitations on the actuators. If these limitations are ignored at the design stage, the actual performance of the control system may suffer severe degradation than expected if the control signal reaches its limits. Control wind-up occurs when a controller command exceeds the physical (saturation) limits of the system actuator, and where controller momentum is unable to immediately respond to changes in the control error. There are two ways to deal with wind-up:

- Reduce the requirements on the desired performance, so that the linear controller never exceeds its limits
- Modify the design to take into account the performance limitations

The anti wind-up method used follows the second path. This method will work acceptably if the control signal does not exceed the permissible limits by more than 100 % (moderate saturation). Otherwise, the actuator may need to be replaced by a higher-performance one.

One of the main undesirable effects of saturation on performance is that any controller integrator will continue to integrate even as long as the input is saturated. Thus, the state of the integrator in question may reach excessive values, which will deteriorate the transient response of the system, generally producing large overvalues. This effect is known as *integrator wind-up*.

The anti wind-up method allows to stop the integral output from rising over the actuators saturation. It was used a clamping method that turn the integrator off when the saturation is reached.

Chapter 5

Results and analysis

5.1. Simulator

In the development of the simulator it have been taken into account all the acting forces and moments that influence the flight dynamics. Besides, the geometry of the aircraft is a very important aspect to consider and its correct interpretation makes the simulator be more true to reality. The results obtained with the simulator has been analysed considering the expected behaviour it should have.

5.1.1. Longitudinal model

The simulator has been made in such a way that it replicates as closely as possible the physical model of the aircraft and the aerodynamic interactions that would affect the flight dynamics. This emphasis on the accuracy of the model is important for control purposes and because the next step will be to get a tuned controller from the simulator and plug it in the prototype.

The most important aerodynamic interactions are the ones between wings and propellers. The propeller flow creates an induced angle of attack and obtains an effective angle of attack smaller that the one there would be in the absence of propellers. This phenomenon is interesting during transition because it avoids the wing to enter in stall.

Figures 5.3, 5.2 and 5.1 show this variation of angle of attack for each wing, and a relative angle of attack of the aircraft body without the propellers flow, which corresponds to the angle of attack seen by the fuselage.

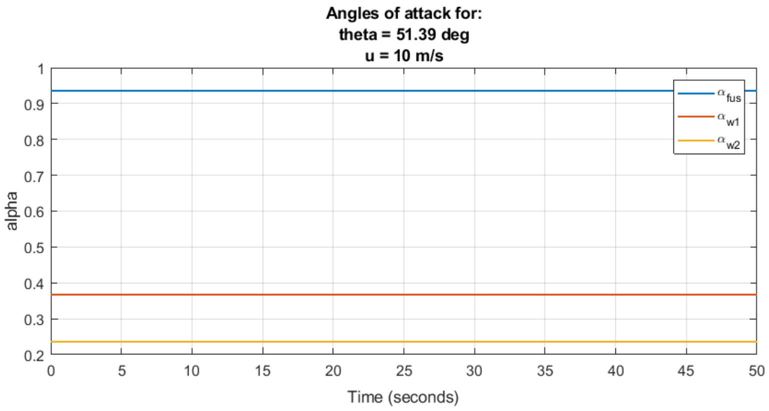


Figure 5.1 α_s for $u = 10\ m/s$ and $\theta = 51.39°$

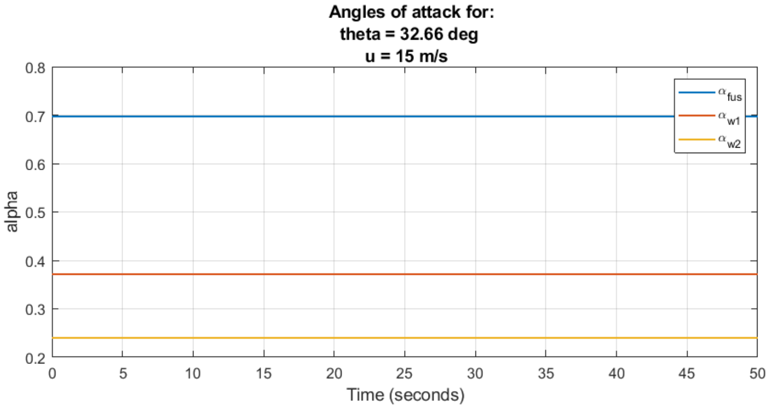


Figure 5.2 α_s for $u = 15\ m/s$ and $\theta = 32.66°$

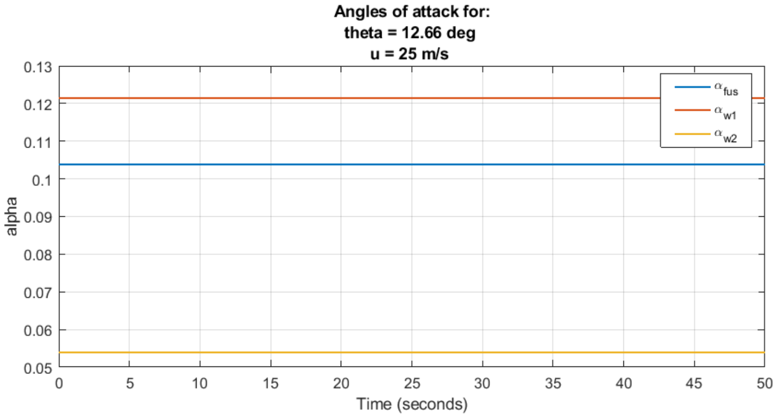
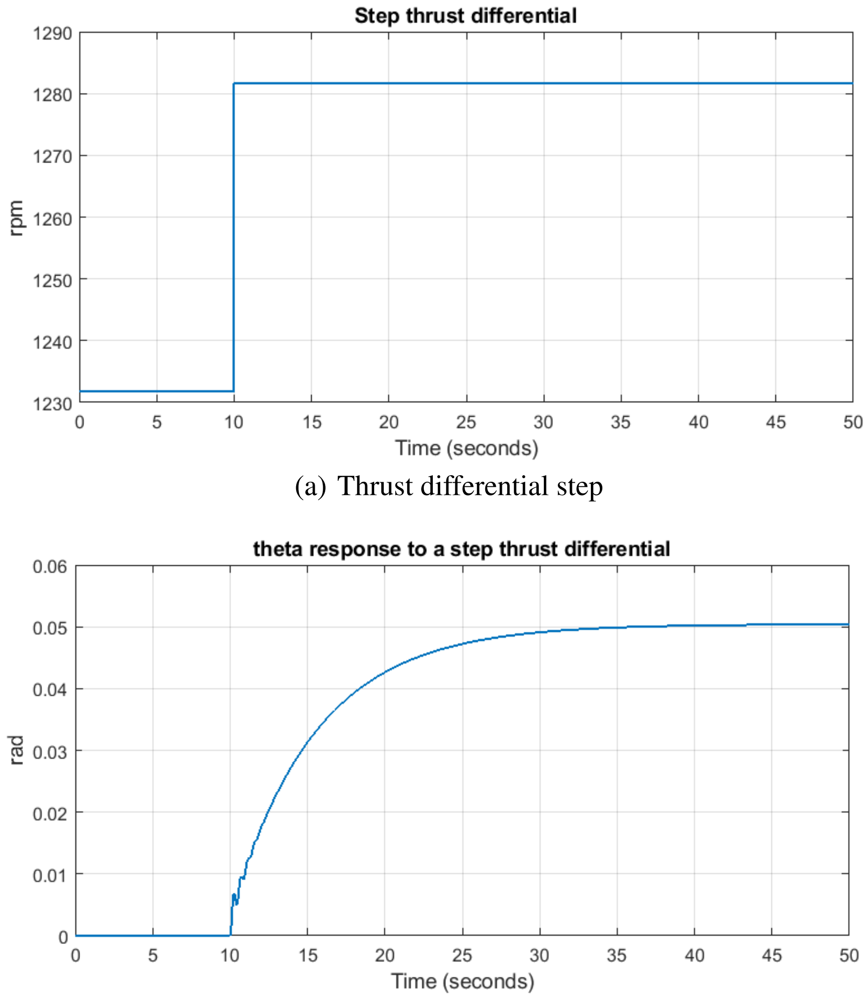


Figure 5.3 α_s for $u = 25\ m/s$ and $\theta = 12.66°$

The previous figures show how as velocity is increased, the difference between the angles o attack of the different parts is smaller. This meets the expectations because the wing-propeller interaction theory explained in Section 2.4.4 is directly dependant on velocity, so as velocity increases the induced angle of attack will be smaller.

This first test has been successful, but it is still necessary to validate the simulator in some way. For this purpose, a test observing the responses and behaviour of the system in case of physical disturbance has been done.

Figure 5.4(b) shows the response of the aircraft when subjected to a step thrust differential disturbance shown in Figure 5.4(a).



(a) Thrust differential step

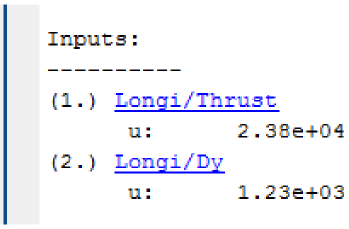
(b) θ response to a step perturbation

Figure 5.4 System response to a step disturbance in pitch

It is easy to see how after the disturbance θ tends to reach the stationary value, as expected.

With respect to the trim, the aircraft is trimmed in a cruise flight, with $\theta = 0\, rad$, $V = 35.6\, m/s$. First, it is worth noticing that the aircraft is asymmetric around $y - axis$ as shown in Figure 2.9(a). Due to this asymmetry, there is a need of a permanent thrust differential between the two wings, in order to cancel the moment created by the wing with the longer arm. This thrust differential can be seen in the trim process of the aircraft, resulting in the inputs being those shown in Figure 5.5. D_y, as shown in Equation 3.2, is the thrust differential needed to cancel the aircraft asymmetric moment and $Thrust$ corresponds to the basis rotational speed for all propellers.

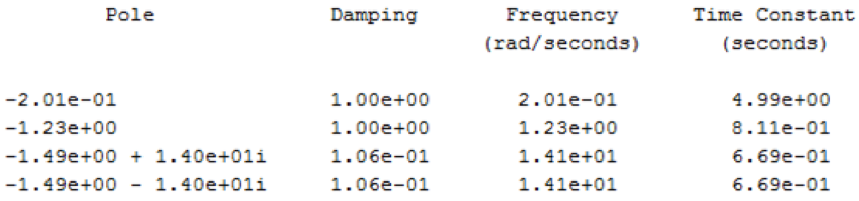
Within a more deeply analysis of the longitudinal model, it can be proven by dynamic analysis of the linear system at this equilibrium point the existence of a poorly damped short period oscillation. Figure 5.6 shows the short period oscillation mode and its characteristics. It is worth noticing the poor damping ratio ($\zeta = 0.1$), which can be easily identified in the simulator q response on



```
Inputs:
----------
(1.) Longi/Thrust
      u:        2.38e+04
(2.) Longi/Dy
      u:        1.23e+03
```

Figure 5.5 Trim Inputs

Figure 5.7 to the disturbance shown in Figure 5.8.



```
         Pole                 Damping       Frequency      Time Constant
                                          (rad/seconds)     (seconds)

 -2.01e-01                   1.00e+00       2.01e-01         4.99e+00
 -1.23e+00                   1.00e+00       1.23e+00         8.11e-01
 -1.49e+00 + 1.40e+01i       1.06e-01       1.41e+01         6.69e-01
 -1.49e+00 - 1.40e+01i       1.06e-01       1.41e+01         6.69e-01
```

Figure 5.6 System dynamics - poorly damped sp oscillation

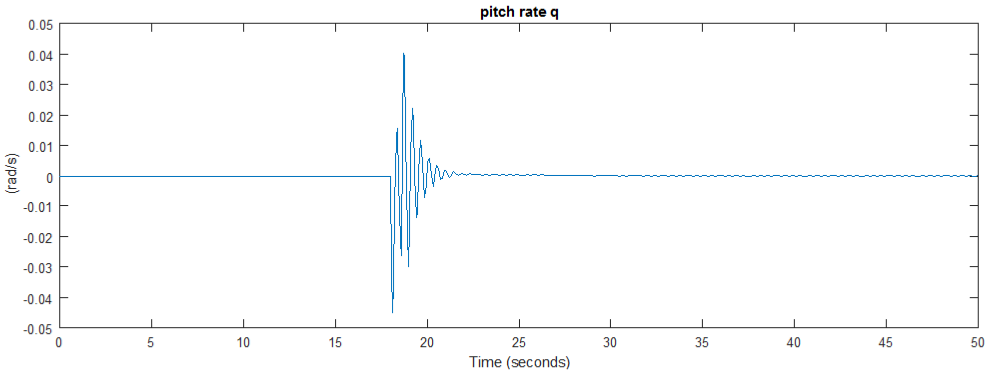


Figure 5.7 q response to a positive disturbance

Figure 5.9 shows a disturbance on the opposite direction. In Figure 5.10, can be noticed the same short period behaviour, but also the angle θ goes in the same direction than the disturbance.

These tests validate the simulator and linearization tool, as it can be seen the physical behaviour of the aircraft on the non-linear model simulation, and also it can be seen through the analysis of the linearized model.

Analysing Figure 5.6 more deeply, there is also a phugoid mode with a period of 4.99 seconds, which makes sense considering the size of the prototype. In fact, for the microlight (ultralight) aircraft, the phugoid period is about $15-25$ seconds, and it has been suggested that model plainnesses show convergence between the phugoid and short period modes.

For the 60 cm wingspan prototype, 4.99 seconds makes sense for this speed. It is difficult to use the known classical formulas as in [28] because they are valid only for bigger conventional aircrafts. At this moment, it is not possible to find data that could be taken as reference for the phugoid analysis.

It is also worth noticing the presence of an additional mode, which is slow enough to have effects on the dynamics of the aircraft, but has a period of 0.8 seconds. A first order mode, slow

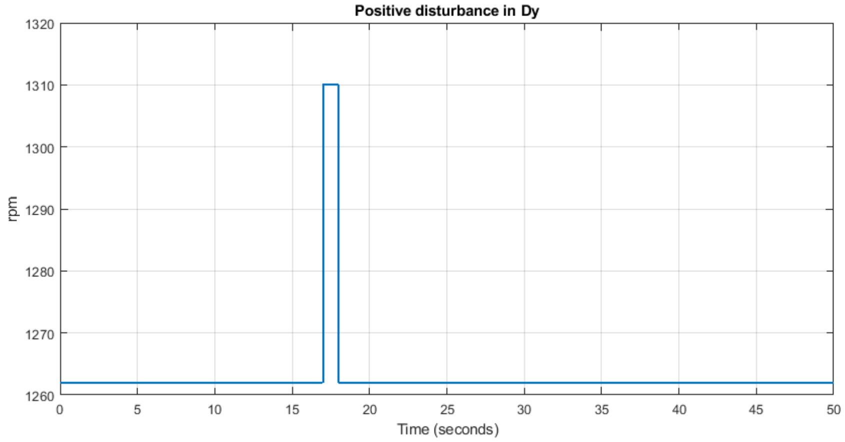


Figure 5.8 Disturbance input in D_y

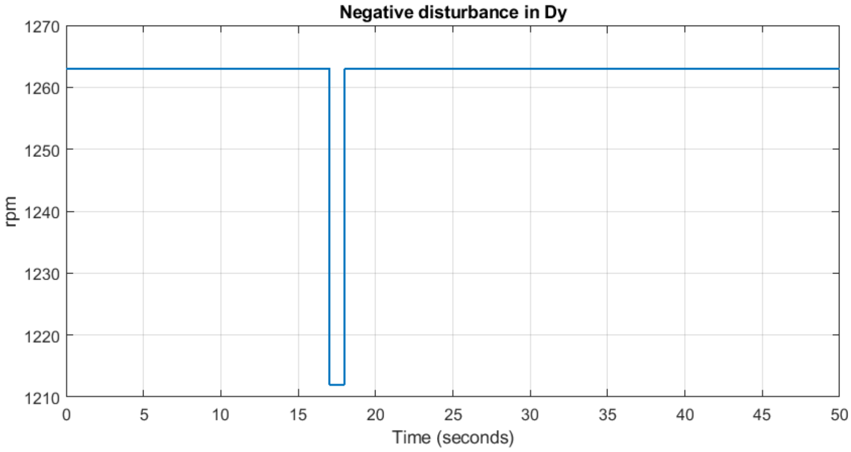


Figure 5.9 Disturbance in the negative direction

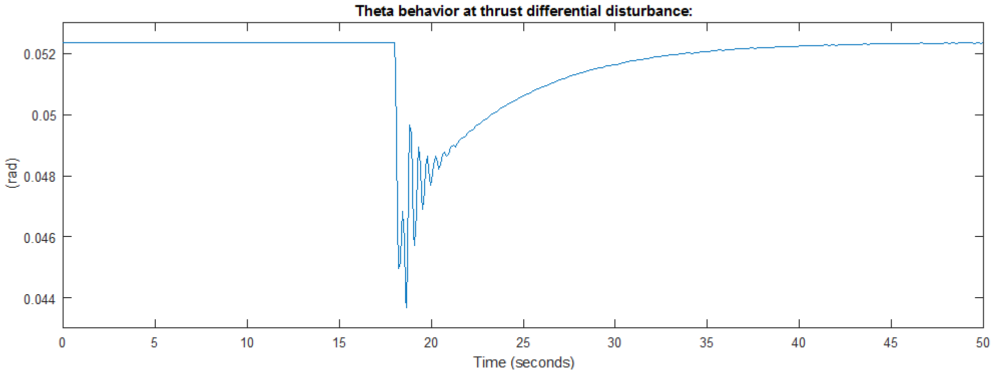


Figure 5.10 θ Response to the same disturbance but in the opposite direction

and of a short period, is some sort of depreciated phugoid or something else. The origin of this pole and its effects still need to be analysed.

5.1.2. Lateral-Directional model

After decoupling the model and getting the lateral-directional model separately from the longitudinal one, some interesting conclusions were obtained.

The model did not give the response expected and it have been considered different options as possible reasons for it:

- The conventional theory for lateral-directional dynamics normally includes a vertical stabilizer, which is missing in the prototype. The lack of this component could be affecting the stability of lateral flight.
- For longitudinal flight Neoptera provided many aerodynamic analysis and information that was not provided for lateral flight. For example, there is no data for aerodynamic lateral coefficients such as C_{l_β} or C_{y_β}. These coefficients among other are crucial for lateral dynamics and a good estimation of them is still to be done.
- The tandem wing configuration of the eOpter prototype is slightly different from the usually used. In Figure 5.11 is shown the Dragofly configuration, where fuselage splits in two parts both wings. This clearly creates two different sections separated by the vertical plane of the aircraft, so there are aerodynamic forces on the left and on the right side of the vehicle. For the eOpter, as seen in Figure 5.12, the wings does not traverse the fuselage, so a new consideration of the applying point or line of the aerodynamic forces may be done.

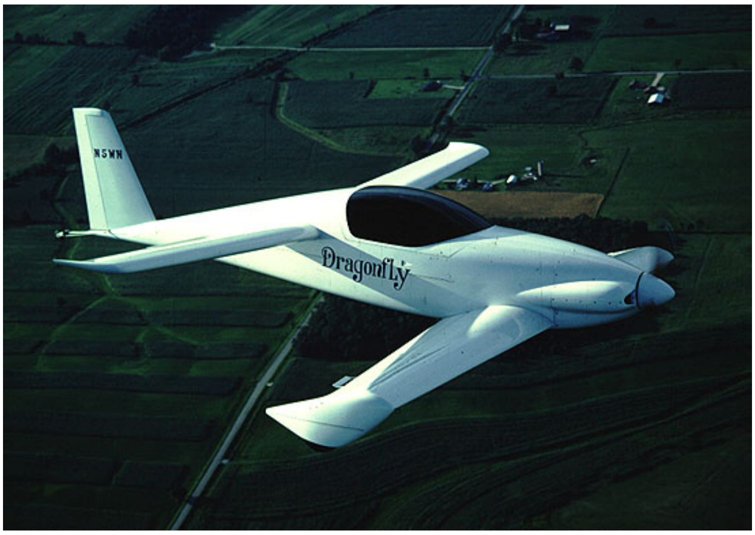


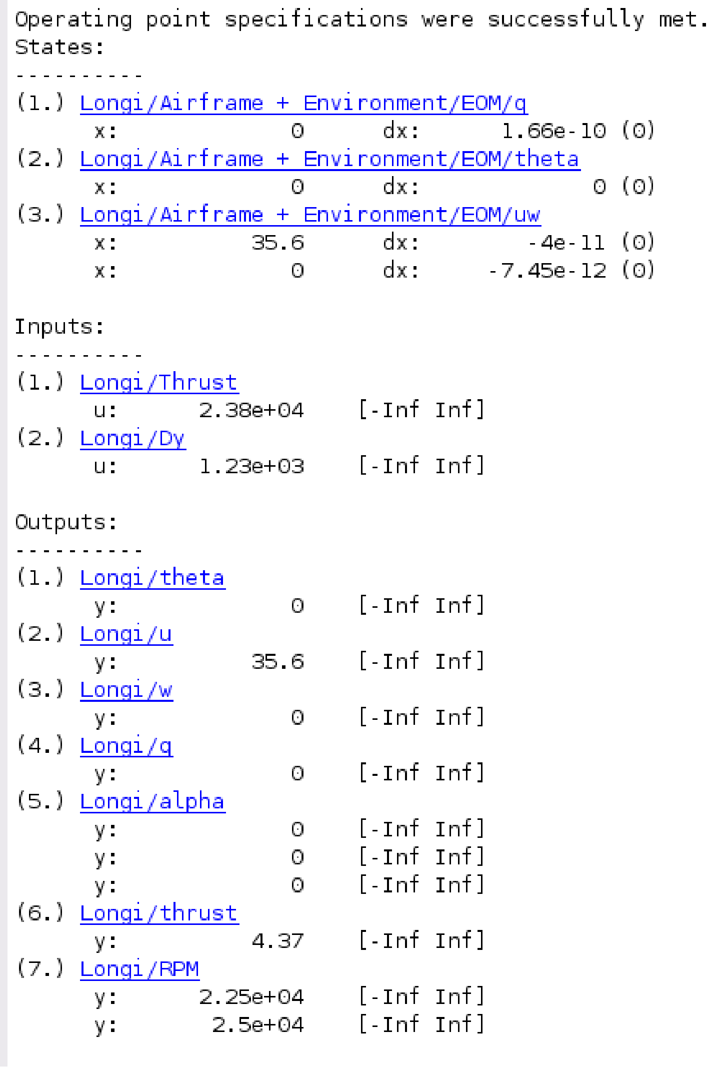
Figure 5.11 Dragonfly Aircraft with tandem wing configuration

5.2. Equilibrium points results

Matlab tools have been used to perform the minimization algorithm which solves the equation $\dot{x} = f(x,u) = 0$. Figure 5.13 shows the trimming process of the longitudinal model under some constraints. The resulting operating point is in steady state ($\dot{x} = 0$), and the couple (x,u) is computed with some constraints on the states such as $\theta = 0$ and $w = 0$.

Different equilibrium points can be found trying to trim the model with different states conditions.

Figure 5.12 eOpter aircraft with tandem wing configuration



```
Operating point specifications were successfully met.
States:
----------
(1.) Longi/Airframe + Environment/EOM/q
      x:            0      dx:      1.66e-10 (0)
(2.) Longi/Airframe + Environment/EOM/theta
      x:            0      dx:             0 (0)
(3.) Longi/Airframe + Environment/EOM/uw
      x:         35.6      dx:         -4e-11 (0)
      x:            0      dx:     -7.45e-12 (0)

Inputs:
----------
(1.) Longi/Thrust
      u:     2.38e+04    [-Inf Inf]
(2.) Longi/Dy
      u:     1.23e+03    [-Inf Inf]

Outputs:
----------
(1.) Longi/theta
      y:            0    [-Inf Inf]
(2.) Longi/u
      y:         35.6    [-Inf Inf]
(3.) Longi/w
      y:            0    [-Inf Inf]
(4.) Longi/q
      y:            0    [-Inf Inf]
(5.) Longi/alpha
      y:            0    [-Inf Inf]
      y:            0    [-Inf Inf]
      y:            0    [-Inf Inf]
(6.) Longi/thrust
      y:         4.37    [-Inf Inf]
(7.) Longi/RPM
      y:     2.25e+04    [-Inf Inf]
      y:      2.5e+04    [-Inf Inf]
```

Figure 5.13 Example of equilibrium point found

Then a comparison is made between equilibrium points obtained with the Matlab program and those previously obtained by Neoptera through an aerodynamic simulation with XFLR5. The simulation consists of an horizontal flight and several equilibrium points have been found for different velocities.

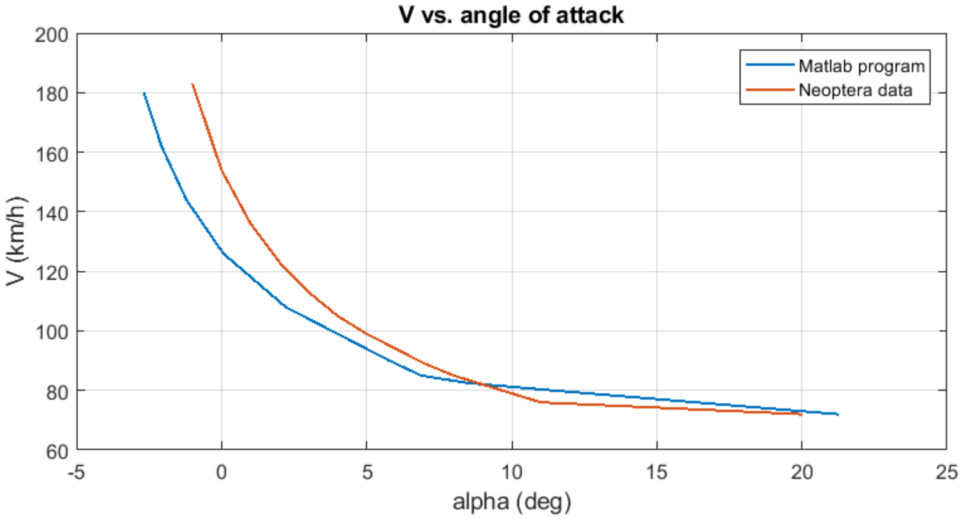


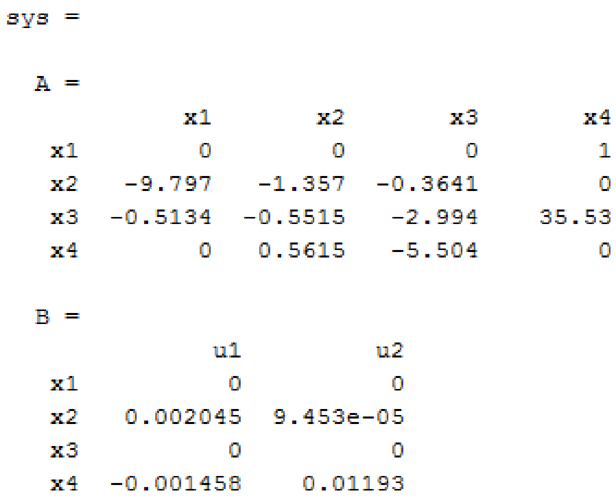
Figure 5.14 Comparison between equilibrium points found with Matlab code and with XFLR5 simulation

As can be seen in Figure 5.14, the equilibrium points found for velocities over 90 km/h by the the Matlab program have lower angles of attack than those found by Neoptera. However, for velocities under 90 km/h the angles obtained by the Matlab routine are greater than those obtained with XFLR5. The main difference between the two models is that in the Matlab program is taken into account the wing-propeller interaction while XFLR5 simulation just considers two tandem wings but does not consider the presence of any propeller. As explained in Section 2.4.4, the presence of the propeller on the leading edge of the wing produces a decrease on the effective angle of attack. This phenomenon is easy to see at high velocities, because the influence of the propeller on the wing aerodynamics is directly related to the airspeed.

For lower velocities, the interaction is not that important, so both simulations tend to coincide.

5.3. Linearization

The Jacobian linearization algorithm is performed in order to obtain the linearized system in state space form around the equilibrium point found as shown in the previous section. Figure 5.15 shows the resulting matrices.

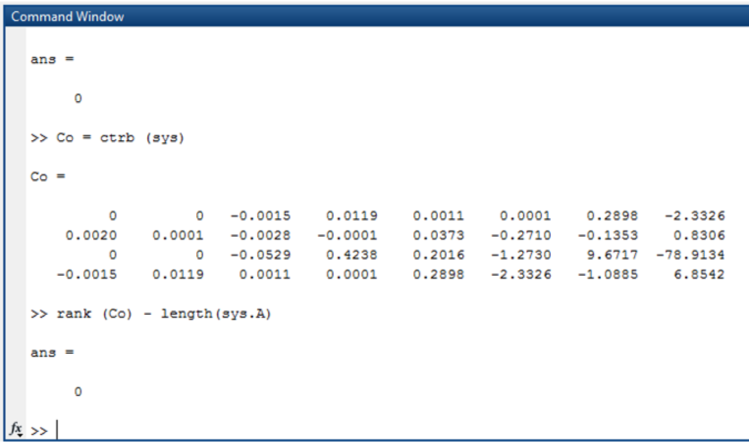


```
sys =

  A =
             x1        x2        x3        x4
   x1         0         0         0         1
   x2    -9.797    -1.357   -0.3641         0
   x3   -0.5134   -0.5515    -2.994     35.53
   x4         0    0.5615    -5.504         0

  B =
                u1          u2
   x1            0           0
   x2     0.002045   9.453e-05
   x3            0           0
   x4    -0.001458     0.01193
```

Figure 5.15 Linearization process results

5.4. Controllability results

In order to use the pole placement method, it is necessary to be sure that system is controllable. Physically, the controllability checks if internal states (attitude, position, orientation) of the aircraft can be controlled using an external input (propellers). After computing the controllability matrix Co is obtained, $R = length(A) - rank(Co)$, where R is the number of uncontrollable states. Figure 5.16 shows the results obtained in Matlab.



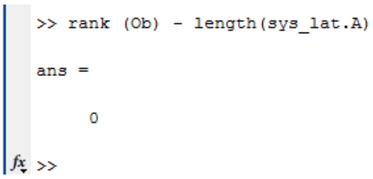
```
ans =

     0

>> Co = ctrb (sys)

Co =

         0         0   -0.0015    0.0119    0.0011    0.0001    0.2898   -2.3326
    0.0020    0.0001   -0.0028   -0.0001    0.0373   -0.2710   -0.1353    0.8306
         0         0   -0.0529    0.4238    0.2016   -1.2730    9.6717  -78.9134
   -0.0015    0.0119    0.0011    0.0001    0.2898   -2.3326   -1.0885    6.8542

>> rank (Co) - length(sys.A)

ans =

     0

fx >> 
```

Figure 5.16 Checking controllability, a condition for pole placement method

Once this was check, there is observability rest to test in order to apply the pole placement method.

5.5. Observability results

eOpter prototype is equipped with multiple sensors which permit to capture all the states of the vehicle. Numerically, this is a requirement in order to use the pole placement method, as it uses full states feedback. Therefore all the states must be observable. Figure 5.17 shows the computation of the observability matrix Ob and $Ro = length(A) - rank(Ob)$, where Ro is the number of non-observable states.



```
>> rank (Ob) - length(sys_lat.A)

ans =

     0

fx >>
```

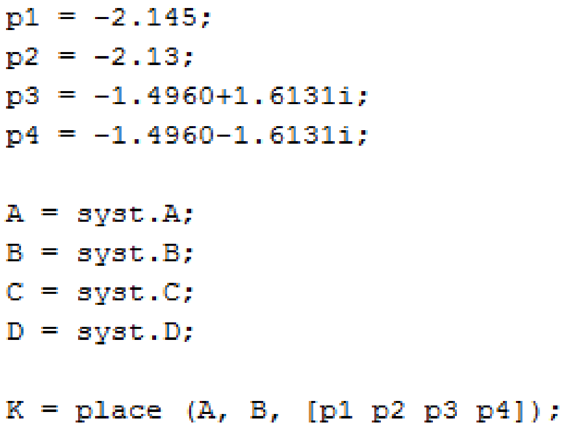
Figure 5.17 Checking observability, a condition for pole placement method

After checking also the observability of the system, it can be performed the pole placement method.

5.6. Pole placement results

As explained in Section 4.3 the pole placement method has been decided to be used by using the formula of Ackermann ([25]). With this formula it is obtained the gain matrix K for given poles locations (Figure 5.18), but it is still necessary to find the exact location of the poles that will be used.

The poles locations are calculated depending on the flying qualities requirements, while keeping the gains small enough not to reach saturation of the actuators. There have not been specified any requirements for the moment, but a damping ratio ξ of around 0.7 and short period frequency of $\omega_n = 2.3\ rad$ are selected for testing purposes. With theses two values, the complex dominant poles for the oscillatory mode are computed. The position of the non-complex poles is found by trial and error.



```
p1 = -2.145;
p2 = -2.13;
p3 = -1.4960+1.6131i;
p4 = -1.4960-1.6131i;

A = syst.A;
B = syst.B;
C = syst.C;
D = syst.D;

K = place (A, B, [p1 p2 p3 p4]);
```

Figure 5.18 Pole placement implementation in Matlab

The intuition is to place them at positions where they are fast enough not to have much effect on the system. But the faster they are, the bigger effort the controller needs to do, and this might result in non-real physical behaviour, or saturation of the actuators. So it is necessary to arrive to an intermediate situation between both limits. Fast enough, but not requiring too much effort. The goal being to get good system response. Figures 5.19 show the impact of the full states feedback with the gain matrix K on the dynamics of the aircraft.

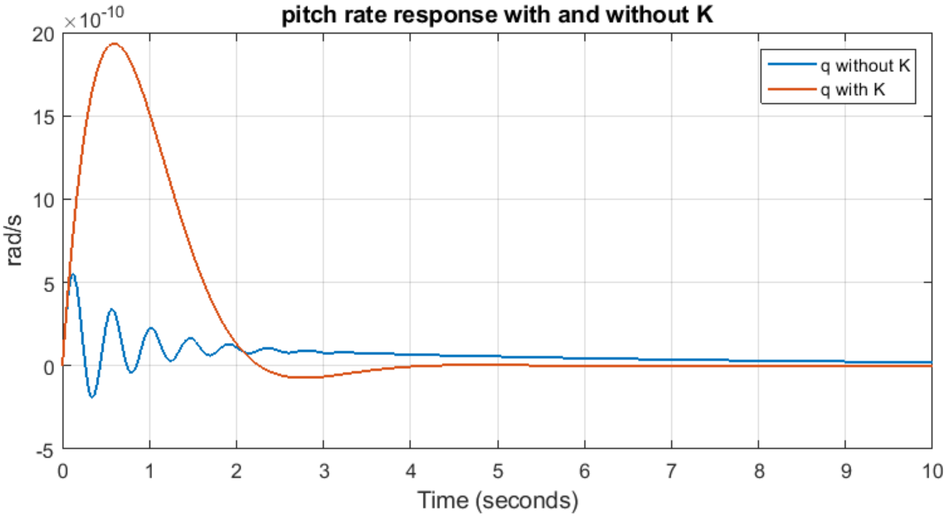


Figure 5.19 Gain matrix K in full states feedback for better performances

As seen in Figure 5.19, the price to pay for getting a signal with a smaller settling time is a big overshoot at the beginning. However, the signal with K does not oscillate like the other and, as mentioned, it arrives to the desired value around 4 s, while the signal without K has not still settled after 10 s. With a deeper tuning of the gains, the overshoot could be reduced.

5.7. Reference Tracking: Integrator Control results

At this point, the states feedback stabilizes the aircraft and enhances performances, but there is still not total control of the aircraft. The integrator control is a robust controller which guarantees asymptotic reference tracking as long as the aircraft is stable. Figure 5.20 shows the aircraft reference tracking abilities with the integrator controller. The mission chosen for this representations is a full mission going from horizontal flight, a transition to vertical flight, some seconds in vertical flight and the transition back to horizontal flight.

It can be assumed that the integrator controller used gives good results and variable q is able to track the reference signal with a negligible error.

5.8. Full mission

Finally, a full mission was performed. This mission followed the θ profile shown in Figure 5.21.

The mission selected starts with a cruise flight with $\theta = 0°$, it transitions to vertical flight $\theta = 90°$ and this is followed by a new transition back to horizontal flight. This flight path was used because it was easier to find equilibrium points with this sequence of phases. It is expected to be possible to simulate a vertical-horizontal-vertical flight after the implementation of the gain surface in the gain scheduling method. Hereafter, the results obtained are analysed.

With respect to the thrust produced by the propellers, all four propellers of each wing have been regrouped and there are just two different thrust values, one for Wing 1 and the other for Wing 2.

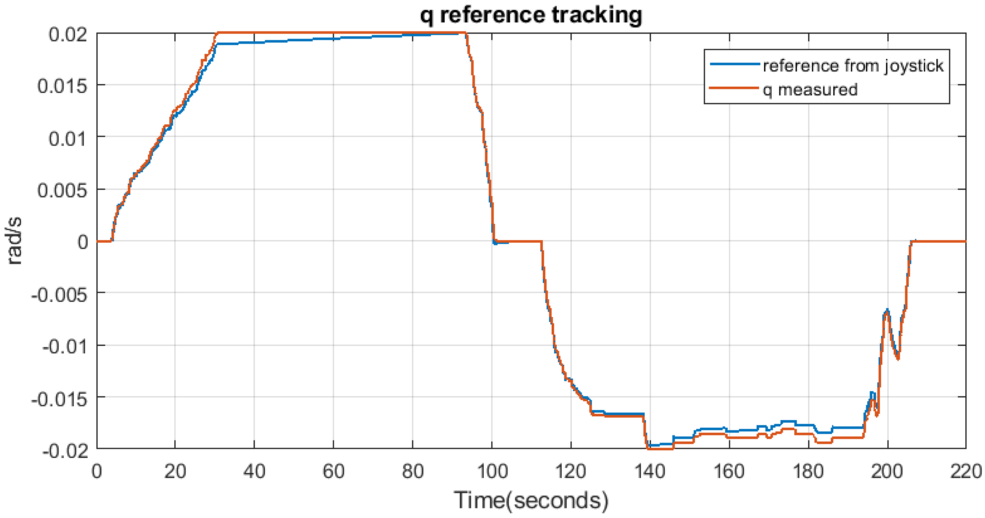


Figure 5.20 Reference tracking on q with the integrator control

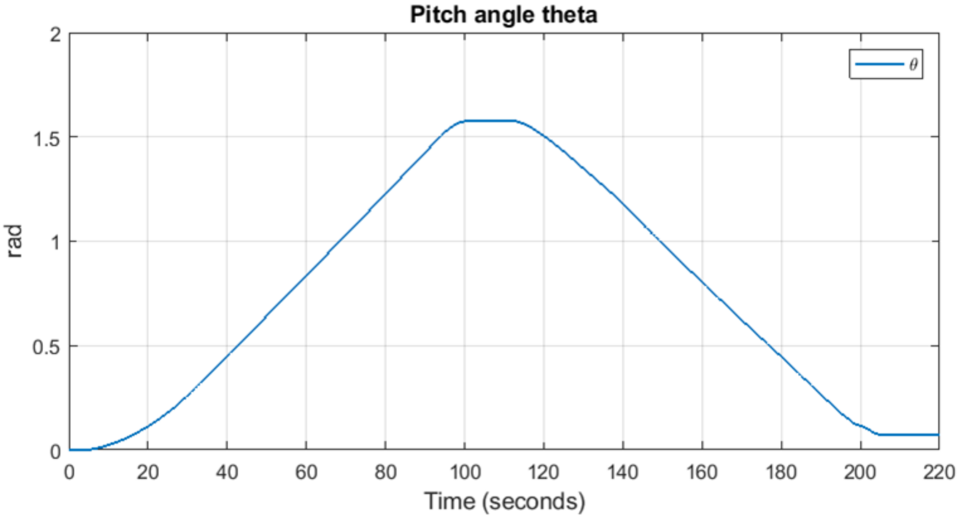


Figure 5.21 Pitch angle mission profile

During the whole mission thrust is different from one wing to the other because of the moment around $y-axis$ that need to be constantly compensated due to the asymmetry of the vehicle. Besides, during transition the thrust of each wing has a different tendency. Since transition is from horizontal to vertical configuration, Wing 2 must give more thrust and Wing 1 less thrust to achieve the moment around $y-axis$ desired to manoeuvre.

About angles of attack, Figure 5.23 shows how the angle of attack seen by each wing and the one of the fuselage are not equal.

This difference is due to the wing-propeller interaction. As explained before, each wing has a different thrust at each moment of the flight and this leads to a different slipstream produced by propellers. Hence, the angle of attack of each wing will be different. Besides, the angle of attack of the fuselage is different from the ones of the wings because it is not under any propeller influence.

Finally, velocities are shown in Figure 5.24.

Velocities, as expected, also change during the mission. Vertical velocity (along $z-axis$) is zero at the beginning and the end of the mission because the vehicle is in horizontal flight and the

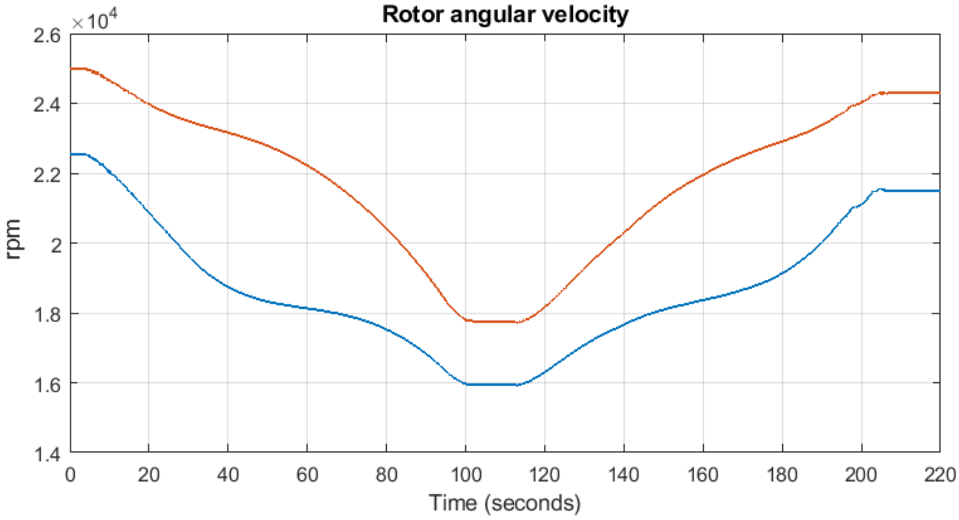


Figure 5.22 Thrust from both wings during the whole mission

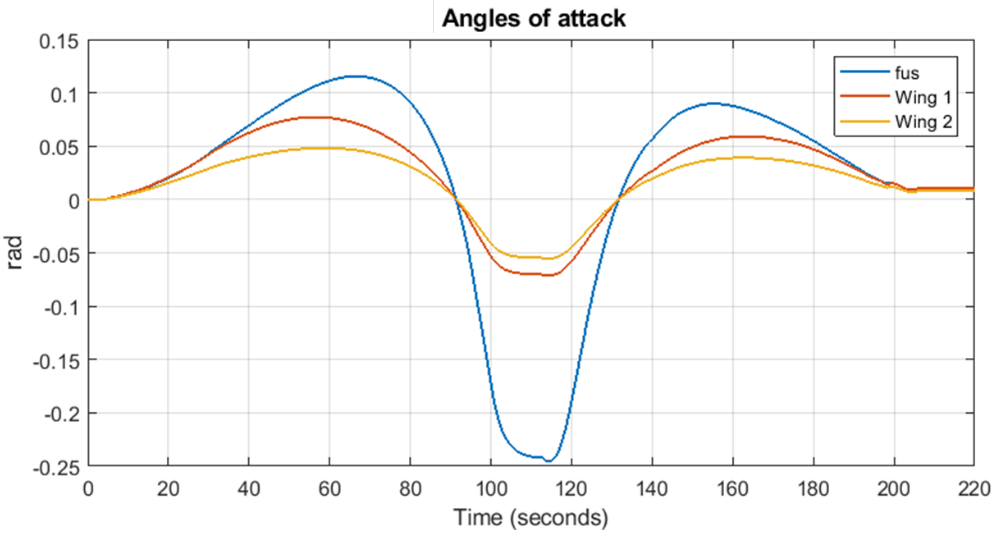


Figure 5.23 Angles of attack during the whole mission

altitude is constant. At these two moments, on the other hand, axial velocity is different from zero because the vehicle is going forward. When transition starts, vertical velocity has a slight increase after a decrease to negative values. This first increase represents a small ascent before arriving to vertical flight and the negative values of velocity implies a forward flight in vertical mode. With respect to axial velocity, it descends during transition, as expected, because to achieve the transition velocity must have smaller values. Also, when it is on vertical mode it has a slight ascend.

5.9. Stability system robustness

In order to check the robustness of the stability system to face a perturbation a test has been done. A perturbation in angular rate q is applied and it is observed the behaviour of pitch angle θ in the event of such a perturbation.

Figure 5.25 shows how the pitch angle stabilizes very quickly after the perturbation. So the

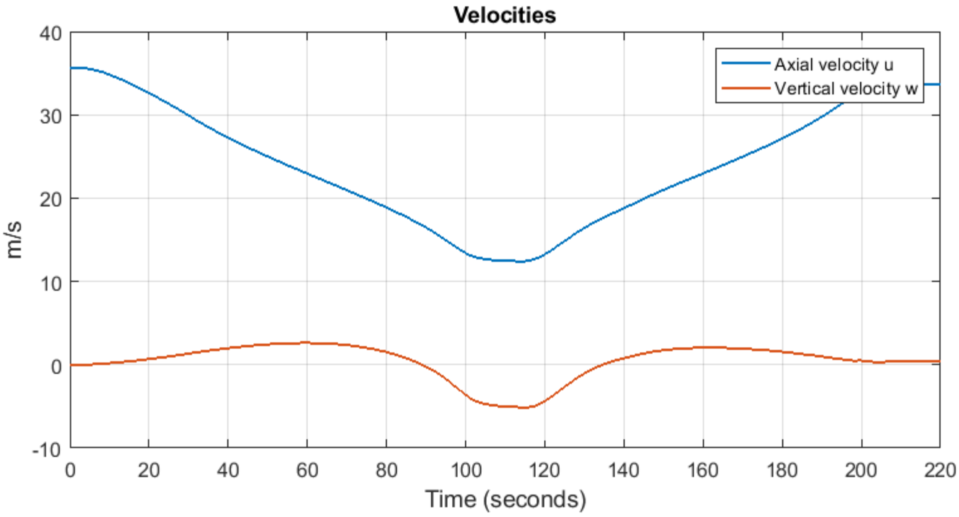
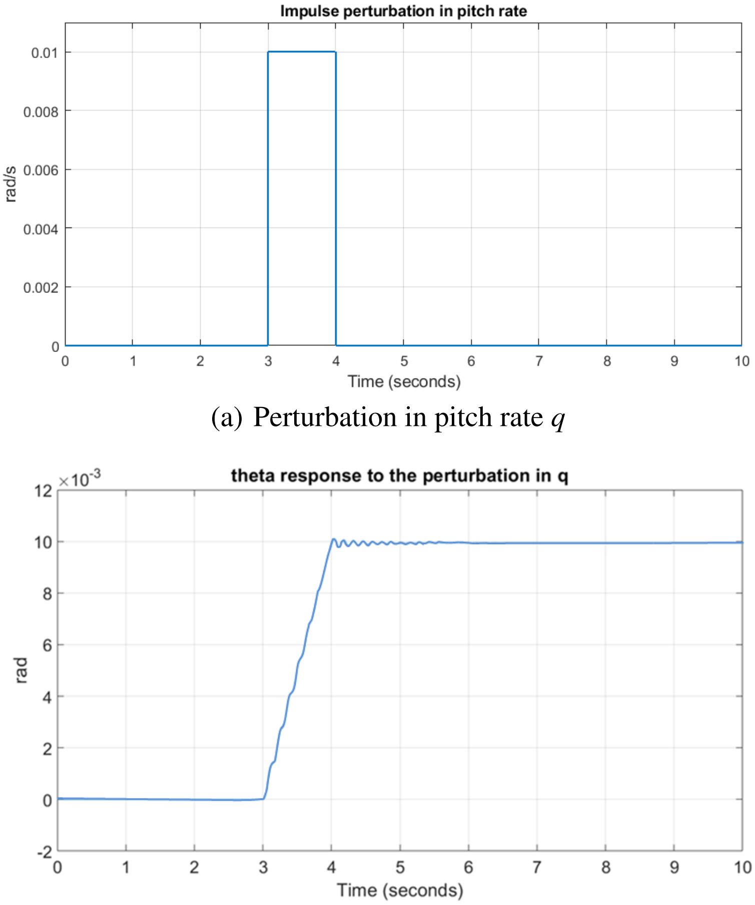


Figure 5.24 Velocities during the whole mission



(a) Perturbation in pitch rate q

(b) Pitch angle θ response to the perturbation

Figure 5.25 Robustness test

stability system can be considered robust enough to face a perturbation in any longitudinal state.

Chapter 6

Conclusions

VTOL vehicles are expected to be a revolution for urban air mobility. Great designs have been created and from a constructive point of view these aircrafts could fly over cities. However, the necessity of a control system makes it harder to launch them into the market. Besides, the airworthiness of VTOL vehicles is still not easy to certificate, which makes it even harder to see them surfing the skies in a short term.

This study establishes a numerically advantageous physical model of the eOpter concept that can be used for control design and dynamic analysis. The concept can almost be treated as a classical airplane in the longitudinal flight. The main difference is the importance of wing-propeller interactions which changes the dynamic of the airplane considerably during transition. This change is due to the fact that the aircraft is subject to different angles of attack α and side-slip angles β depending on the velocity at each specific part of the aircraft. In this way, each wing has different aerodynamic angles and the fuselage also has its own aerodynamic angles. The induced angles produced by the different air flows lead to an increase in the lift coefficient, which makes stall to appear later than it would if propellers were not there. The importance of this phenomenon during transition is obvious because the reached angle of attack in this phase is very big, and it is important to avoid stall as much as possible and give time to propellers to take the lead when lift force starts to weaken.

The lateral-directional model, however, still needs some deeper analysis. The configuration of the eOpter is not very similar to conventional airplanes, mainly the aspects related to lateral dynamics. For this reason, the classical lateral-directional approach may not be exactly the optimal solution and it probably needs some modifications that make it adapt better to the vehicle in study. It is crucial to give and answer to the following issues:

- The lack of a vertical stabilizer and how this impact on the lateral-directional dynamics and stability.
- How the particular tandem wing configuration of eOpter without the wings traversing the fuselage affects the lateral-directional dynamics.
- Where to set the position of the centre of pressure due to the fact explained above.

These facts could be the reason why the behaviour of the lateral-directional model developed by the moment is not as expected and that some modifications need to be done.

The stability augmentation system and controller methods used have been proven to be robust enough to keep the control of the aircraft during the entire flight envelope, as long as the operating point around which the aircraft is linearized are chosen carefully. The Matlab tools created allow to find the equilibrium points with some given constraints, to linearize around these points, to observe the dynamic characteristics of the longitudinal model and to compute the gain matrices for chosen poles, corresponding to certain flying qualities specifications. The equilibrium points obtained depend on the velocity chosen, included in the constraints. In this way, the gain matrix will change as velocity changes, which is the main objective of the gain scheduling method.

From this point, the next step would be to implement the gain scheduling method but considering both, velocity and pitch angle, as scheduling variables. To deal with the two variables it is necessary to create a gain surface of gain matrices so each combination of velocity and pitch angle has a gain matrix associated.

Concerning the prototype, an airframe of the eOpter aircraft which also generates a control mixer has been developed in the PX4 software and is currently being tested. The structure of the PX4 is complex and requires a lot of time to understand. But through flight tests the software it is expected to improve by fixing minor problems and, in this way, reach better flight performances.

References

[1] Vertical Flight Society
www.vtol.org

[2] BCG
www.bcg.com

[3] NEOPTERA
http://neoptera.aero/

[4] Kamal, Ashraf M.; Ramirez-Serrano, Alex (2018)
Design methodology for hybrid (VTOL + Fixed Wing) unmanned aerial vehicles.
Aeronautics and Aerospace Open Access Journal, Volume 2 Issue 3: 165-176

[5] Amritkar, Amit R. (2009)
Aerodynamic analysis of a vertical takeoff and landing vehicle.
University of Utah, Department of Mechanical Engineering

[6] Gómez Tierno, Miguel Ángel; Pérez Cortés, Manuel (2012)
Mecánica del vuelo, 2ª Edición.
Garceta Grupo Editorial

[7] Cook, Michael V. (2007)
Flight Dynamics Principles, 2nd Edition.
Elsevier

[8] Shevell, Richard S. (1989)
Fundamentals of Flight, 2nd Edition.
Prentice-Hall

[9] McCormick, Barnes W. (1979)
Aerodynamics, Aeronautics, and Flight Mechanics, 2nd Edition.
Wiley

[10] McCormick, Barnes W. (1998)
Aerodynamics of V/STOL flight.
Dover publications

[11] Ribeiro Lustosa, Leandro (2017)
The Phi-theory approach to flight control design of hybrid vehicles.
Institut Supérieur de l'Aéronautique et de l'Espace (ISAE)

[12] Stevens, Brian L.; Lewis, Frank L.; Johnson, Eric N. (2015)
Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems, 3rd Edition.
Wiley-Blackwell

[13] Vepa, Ranjan (2014)
Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems, 3rd Edition.
CRC Press

[14] McLean, Donald (1969)
Flight Dynamics, Simulation, and Control: For Rigid and Flexible Aircraft, 1st Edition.
Prentice Hall

[15] Etkin, Bernard (1998)
Dynamics of fight: stability and control.
John Wiley & Sons Australia, Limited

[16] Tang Yeo, Yih; Liu, Hugh H.T. (2018)
Transition Control of a Tilt-Rotor VTOL UAV.
American Institute of Aeronautics and Astronautics. AIAA SciTech Forum

[17] Hankey, W.L.; Miller, L.E.; Scherr, S.J. (1984)
Use of quaternions in flight mechanics.
Technical report, DTIC Document

[18] Filippone, Antonio (2006)
Flight Performance of Fixed and Rotary Wing Aircraft, 1st Edition.
Butterworth-Heinemann

[19] Department of Defense (1984)
Department of Defense World Geodetic System, 3rd Edition.
NIMA Technical Report TR8350.2

[20] US Government (1976)
U.S. Standard Atmosphere.
US government printing office

[21] Packard, Andrew; Poolla, Kameshwar; Horowitz, Roberto (2002)
Dynamic Systems and Feedback. Class Notes for ME 132.
Department of Mechanical Engineering, University of California

[22] Ogata, Katsuhiko (1997)
Modern Control Engineering, 3rd Edition.
Upper Saddle River, NJ: Prentice-Hall

[23] Battipede, Manuela (2017)
Guida e Controllo del Velivolo, appunti.
Politecnico di Torino

[24] Kalman, R.E. (1963)
Mathematical Description of Linear Dynamical System.
SIAM Journal, Volume 1 Issue 2: 152-192

[25] Shinners, Stanley M. (1998)
Modern Control System Theory and Design, 2nd Edition.
Automatica, Volume 29 Issue 6: 1617-1619

[26] Leith, D.J.; Leithead, W.E. (2012)
Survey of Gain-Scheduling Analysis and Design.
Department of Electronic and Electrical Engineering, University of Strathclyde

[27] Kwong, Raymond H. (2008)
Control Systems notes.
Department of Electrical and Computer Engineering, University of Toronto

[28] Bunge, Roberto A. (2015)
Aircraft Flight Dynamics.
Stanford University

Appendix A

Technical Data

A.1. Geometrical Data

The geometrical data of the vehicle has been provided by NEOPTERA by a Technical Data document for this prototype.

Also all the distances expressed in Chapter 2 have been taken from the Technical Data document mentioned.

A.2. Inertial Data

The matrix of inertia of the vehicle is represented in Equation A.1. It is referred to the centre of gravity and it was obtained from a CAD of the vehicle created with CATIA by NEOPTERA.

$$I = \begin{pmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{pmatrix} \tag{A.1}$$

It is also known that the global mass of the vehicle is $m = 2.626\ kg$.

A.3. Aerodynamic Data

Wings aerodynamic polar have been calculated with Javafoil. The airfoil of both wings is CLARK YM-15, represented in Figure A.1.

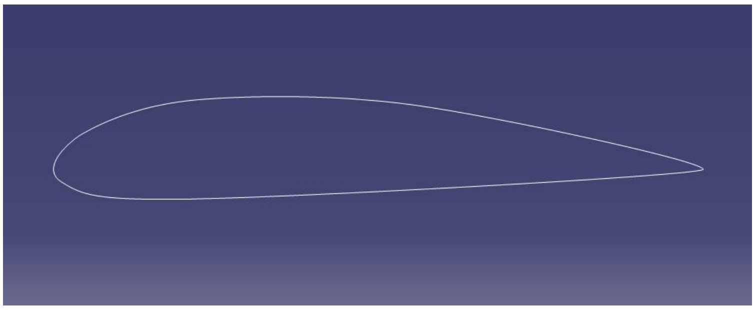


Figure A.1 Airfoil Clark YM-15

C_{m_0}	$C_{L\alpha}$	C_{D_0}	C_{Y_β}	α_0 $[rad]$	e
-0.061	0.0548	0.0171	0	0	0.8065

Table A.1 Aerodynamic Data

The aerodynamic data of both wings is collected in Table A.3.

Where C_{m_0}, $C_{L\alpha}$ and C_{D_0} have been obtained from the Javafoil graphics provided by NEOPTERA (Figure A.2, Figure A.4 and Figure A.3); C_{Y_β} and α_0 are deduced from the Technical Data document; and the Oswald Factor e have been calculated with a Matlab routine based on the Shevell method (R.S. Shevell [8]). This method computes the Oswald Factor from the aspect ratio AR, the sweep of the wing Λ, the zero-lift drag coefficient C_{D_0}, a ratio between fuselage diameter and wing span and planform efficiency.

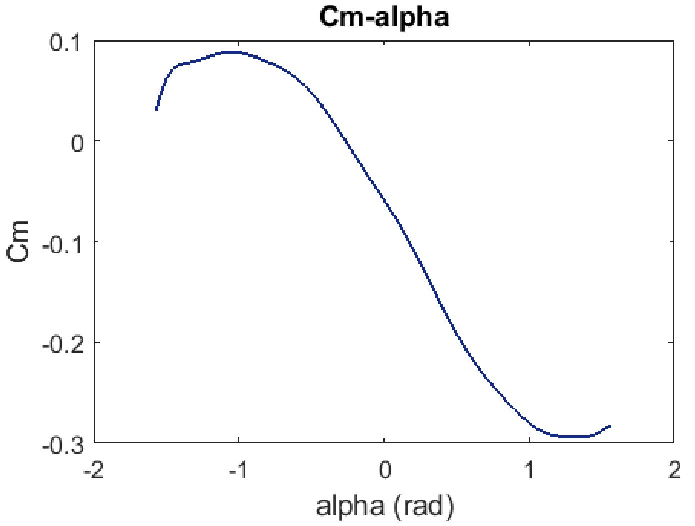


Figure A.2 Graph C_m vs. α (rad)

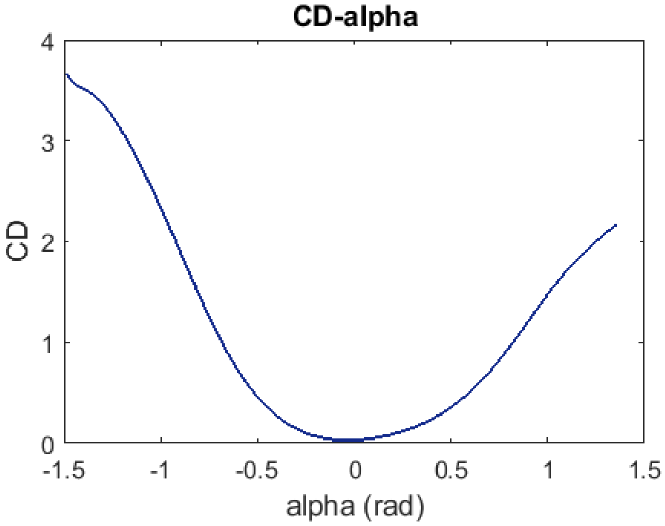


Figure A.3 Graph C_D vs α (rad)

To calculate C_L for each possible angle of attack α, a graphic from the Technical Data document has been imported to Matlab so SIMULINK can use this graphic in each iteration. The graphic of C_L vs. α corresponds to Figure A.4.

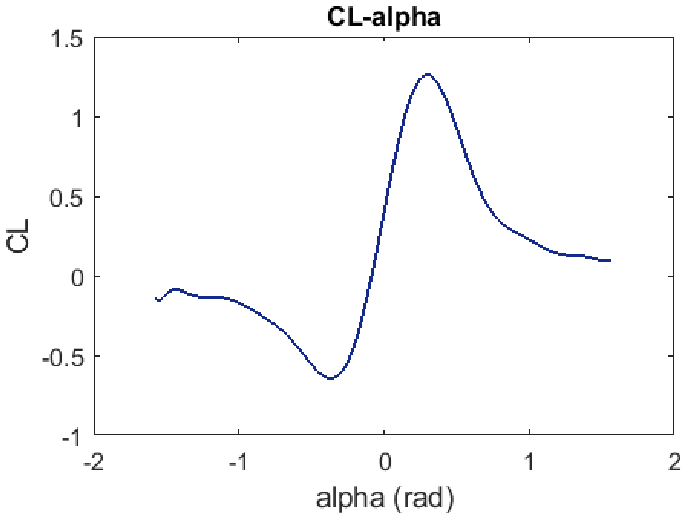


Figure A.4 Graphic C_L vs. α

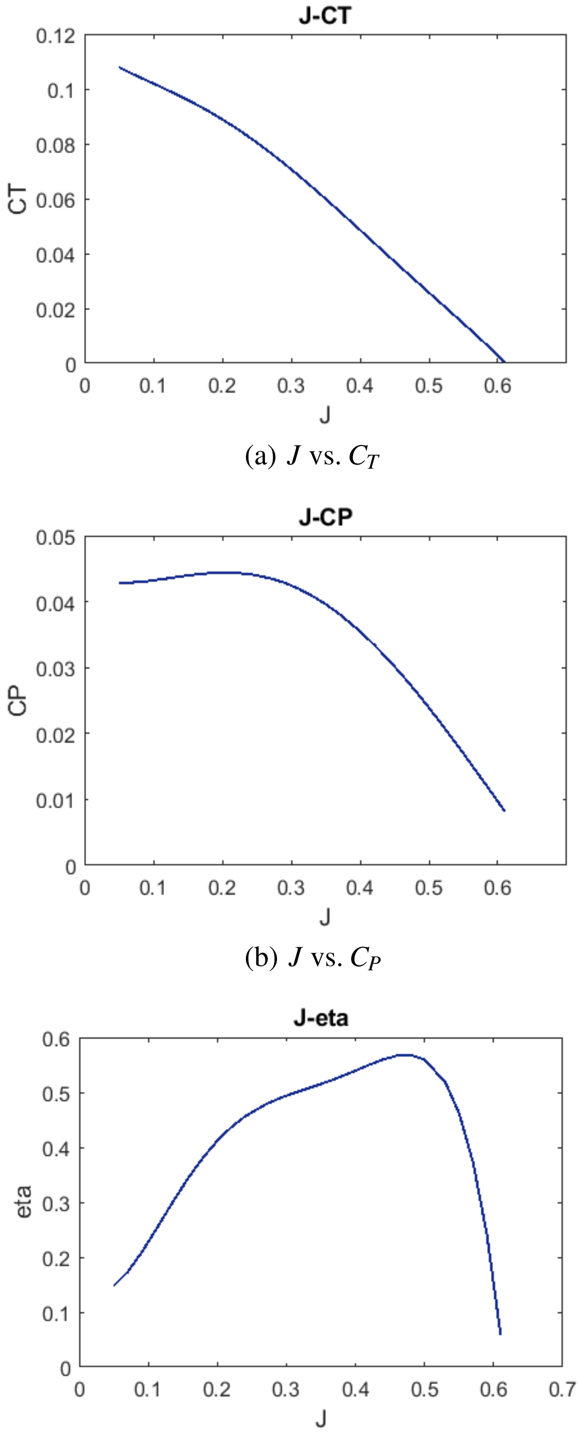
In addition, a value of C_{D_0} and a value for the fuselage surface S have been estimated. It is intended to do a more accurate study for the fuselage aerodynamics.

A.4. Propellers Data

With respect to propellers, they all include two blades of type 6045 (6 *in* of diameter and 4.5 of pitch). However, it was not possible to find technical data of these blades and graphics from PropCal 3.0 of blade Taipan 8" *x* 4" have been used.

The graphics used for propellers data can be seen in Figure A.5.

where C_T is thrust coefficient, C_P is power coefficient, η is the efficiency and $J = \dfrac{V}{nD}$ is the advance ratio.



(a) J vs. C_T

(b) J vs. C_P

(c) J vs. η

Figure A.5 Propellers coefficients and efficiency