Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

University of Padova

School of Engineering

Department of Information Engineering – DEI

Master Degree in Control Systems Engineering

Master thesis

Modeling and differential flatness based control for a convertible aircraft

Candidate:
Monika Pasquali
(Student ID 2055171)

Supervisor:
Prof. Angelo Cenedese
(University of Padova)

Co-Supervisor:
Prof. Tudor-Bogdan Airimitoaie
(University of Bordeaux)

Co-Supervisor:
Prof. Patrick Lanusse
(INP Bordeaux)

Padova, 5th September 2023

Academic Year 2022/2023

Once you have tasted flight,
you will forever walk the earth
with your eyes turned skyward.
For there you have been and there
you will always long to return.

Leonardo da Vinci

Acknowledgments

Firstly, I would like to thank Professor Tudor-Bogdan Airimitoaie and Professor Patrick Lanusse for having offered me the possibility to work on such an interesting topic at the *Laboratoire de l'Intégration du Matériau au Système* in Bordeaux, France.

I am sincerely thankful to the constant support and availability I received from Professor Tudor, who patiently followed me during this entire research project, giving me precious suggestions and sharing his knowledge regarding the aeronautical field, of which I have always been interested in, but never had the chance to explore deeply before now.

A special thanks goes to Professor Angelo Cenedese, since without his support as a Supervisor, this project abroad would not have been possible.

Moreover, I would like to thank my family for always being present and my friends, Matteo, Giacomo, Nicolas, Ikram, Yara and all the others, with whom I shared all the challenges I faced along the road, they were really important.

I am also very thankful to Andrew, for always being ready to remind me that despite all the difficulties this project has presented me with and the hard work it required, I would have always been capable of overcoming them.

Abstract

One of the latest challenges in the aeronautical field is the research and development of convertible Unmanned Aerial Vehicles (UAV). This kind of aircrafts offers interesting alternatives to fixed-wing and Vertical Take-Off and Landing (VTOL) UAVs. Fixed-wing UAVs are very efficient in the forward flight and they require runways for take-off and landing, VTOL UAVs, in contrast, have the ability to hover but are not very efficient in forward flight.

Especially for missions like observation and inspection of structures, in which, for instance, the vehicle has to inspect a wind turbine and then to fly rapidly towards another one, both fixed-wing and VTOL UAVs are unsuitable. Convertible Unmanned Aerial Vehicles (UAV), instead, guarantee a good performance in these kind of missions, as they provide efficient capabilities in both vertical take-off and landing, hovering and forward flight.

This research project is initially focused on the accurate dynamic modeling of a bi-rotor, tilt-body, tail-sitter, delta-wing convertible aircraft. Then, the obtained system is proved to satisfy the differential flatness property, for which two different analyses, for zero/small and high aerodynamic velocities, are considered. Differentially flat systems have the property that all the states and control inputs can be expressed as a function of a set of flat outputs and a finite number of their derivatives. This study allows to compute the inverse dynamic model of the system and it is used for the feedforward control.

Finally, the linearized system is decoupled with the Singular Value Decomposition (SVD) method, and PID controllers are designed.

It is worth to remark that this research project is simulation based, but considerable effort was made in order to keep this work as close as possible to reality.

Sommario

Una delle sfide più recenti in campo aeronautico consiste nella ricerca e nello sviluppo di veicoli aerei senza pilota (UAV) convertibili. Questo tipo di velivoli offre interessanti alternative agli UAV ad ala fissa e a decollo e atterraggio verticale (VTOL). Gli UAV ad ala fissa sono molto efficienti nel volo orizzontale e richiedono piste per il decollo e l'atterraggio, mentre gli UAV VTOL hanno la capacità di volo stazionario ma non sono molto efficienti nel volo orizzontale.

Specialmente per missioni come l'osservazione e l'ispezione di strutture, in cui, ad esempio, il veicolo deve ispezionare una turbina eolica e poi volare rapidamente verso un'altra turbina, sia gli UAV ad ala fissa che quelli VTOL non risultano adatti. I veicoli aerei senza pilota convertibili (UAV), invece, garantiscono una buona performance in questo tipo di missioni, in quanto forniscono capacità efficienti sia nel decollo e nell'atterraggio verticale che nel volo orizzontale.

Questo progetto di ricerca è inizialmente incentrato sulla modellazione dinamica accurata di un velivolo convertibile con ala a delta, bi-rotore, tilt-body e tail-sitter. In seguito, si è dimostrato che il sistema ottenuto soddisfa la proprietà di differential flatness, per la quale sono state effettuate due analisi diverse, per nulle/basse ed elevate velocità aerodinamiche. I sistemi differentially flat hanno proprietà che tutti gli stati e gli input di controllo possono essere espressi in funzione di un insieme di flat outputs e di un numero finito delle loro derivate. Questo studio permette di calcolare il modello dinamico inverso del sistema e viene utilizzato per il controllo feedforward.

Infine, il sistema linearizzato viene disaccoppiato con il metodo di Singular Value Decomposition (SVD) e sono progettati i controllori PID.

Vale la pena sottolineare che questo progetto di ricerca è simulativo, ma sono stati compiuti notevoli sforzi per mantenere questo lavoro il più vicino possibile alla realtà.

Contents

List of Figures

Chapter 1

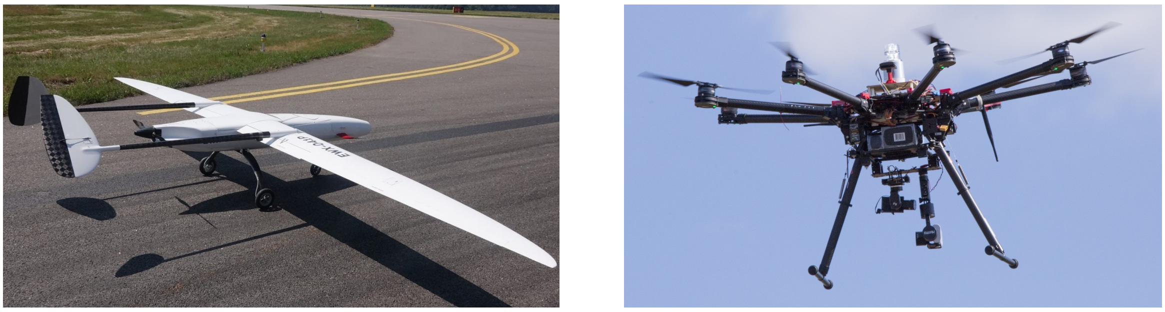
Introduction

Unmanned aerial vehicles (UAVs) made their initial appearance in the military as remote-control aircrafts, but nowadays are widely used for a variety of purposes, for instance in the surveillance, air-to-ground and air-to-air strikes, wildlife monitoring, aerial photography, etc.

As [21] reports, UAVs were traditionally classified into: Fixed-Wing and Vertical Take-Off Landing (VTOL). Concerning Fixed-Wing UAVs, the flight is mainly based on the thrust to cancel the drag induced by the air movement and the aerodynamic lift on the wings to compensate for the weight of the vehicle. These vehicles are very efficient in the forward flight and they require runways for take-off and landing. VTOL UAVs, in contrast, have the ability to hover, but are not very efficient in forward flight. These two configurations are illustrated in Figure 1.1a and Figure 1.1b, respectively.

In applications like observation and inspection of structures, where, for instance, the UAV has to inspect a wind turbine and then to fly rapidly to another one away from its initial position, it is required efficient capabilities in both vertical take-off, hovering and forward flight.

Interesting alternatives to fixed-wing and VTOL UAVs, that are clearly unsuitable for these kind of missions, are the later emerged convertible UAVs.



(a) Example of a fixed-wing UAV. Photo by [15] (b) Example of a VTOL UAV. Photo by [14]

Figure 1.1: UAVs configurations.

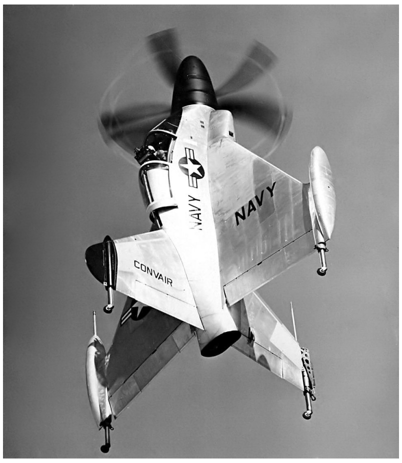


Figure 1.2: Convair XFY Pogo, 1954. Photo by [12]

Among the possible diverse configurations for a convertible aircraft, this research project is focused on the tilt-body aerial vehicles, which are increasingly popular in unmanned applications. Tilt-body convertible UAVs require the body of the aircraft to rotate during the transition flight and, additionally, wings and rotor hubs are rigidly attached to the aircraft body. Also, a special kind of tilt-body aircraft, which is considered here, is the so-called tail-sitter due to the ability to take off and land on its tail.

In the history of aeronautics, these ideas have been already followed in the development of manned aircraft and not only UAVs. For instance, an early prototype of a tilt-body, tail-sitter aircraft, was the Convair XFY Pogo, illustrated in Figure 1.2. The thrust generation was effectuated by a pair of three-bladed contra-rotating propellers, and it was mainly exploited for its ability to perform VTOL on limited surfaces, for instance on a small warship. Nevertheless, one of the disadvantages of this model was the difficulty for the pilot, in the uncomfortable sitting position, to perform landing.

The Laboratoire de l'Intégration du Matériau au Système (IMS), at the Polytechnic Institute of Bordeaux (Bordeaux INP), where this research project was conducted, is doing research about this topic. The considered aircraft, whose model is shown in Figure 2.1, is a delta-wing with two propellers, which is characterized by four control inputs, namely the deflections of the left and right elevons and the rotation speeds of the left and right counter-rotating propeller-engines, explained in detail in Section 2.1.

1.0.1 Aim and challenges of the thesis

The aim of this master thesis is to initially study the aerodynamic forces and moments acting on the aircraft, in order to design a high accurate model which is as close as possible to reality. The critical aspects during the different flight modes, as, for instance, an increase of the angle of attack which exceeds the stall angle, resulting in a sudden drop in the lift produced by the wing, and the gimbal-lock issue, are taken into account and modeled carefully and precisely.

Afterwards, the design of a feedforward controller based on the differential flatness theory and the design of a feedback controller, are carried out. The approach based on flatness, which is a property of some nonlinear systems, has numerous advantages: by identifying a set of flat-outputs, the complete system's state variables and control inputs, can be defined as functions of this subset of flat-outputs and a finite number of their time derivatives. This allows to implement the feedforward control without the need to solve any differential equations but only algebraic ones.

The mathematical proof of the flatness property for this kind of aircraft is challenging, not only due to the high complexity of the resulting nonlinear dynamic model, but also due to the presence of the different flight modes. In this research project, they are treated separately, and the reasons behind this choice are deeply investigated in Section 3.2.

In addition, the nonlinear model exhibits a high coupling between different motions, as for instance when the aircraft rolls, it can induce a pitch motion, and vice versa. This behaviour requires an input-decoupling, where the chosen method is the Singular Value Decomposition (SVD) approach, before proceeding with the design of the PID controllers.

The practical work was performed using the Softwares `MATLAB` & `Simulink` and the mathematical analyses for the differential flatness property were carried out with *Maple*.

1.0.2 Thesis overview

A brief overview of each chapter is presented below.

Chapter 2 provides the accurate description of the system, the assumptions and notations adopted for the whole analysis and the study of the nonlinear dynamic model. The resulting equations are reported in Section 2.63.

Chapter 3 is initially focused on the differential flatness theory, which is briefly illustrated for a general nonlinear system in Section 3.1. Afterwards the chosen analysis approach is explained deeply, in Section 3.2, analyzing the reasons that led to the final solution and the choice to proceed separately for zero/small and for higher aerodynamic velocities. Finally, in Sections 3.3 and Section 3.4, the representation of the mathematical model adopted in the different flight modes, the simplifying assumptions, the choice of the flat-outputs and the resulting solutions are provided.

Chapter 4 illustrates the implementation of the open-loop control based on the differential flatness approach. Firstly, in Section 4.1, it is introduced the concept of trim, and an algorithm in order to find the control inputs, which guarantees the aircraft to be in equilibrium, is implemented. Afterwards, in Section 4.2, based on the results obtained from the trim algorithm, the trajectories are generated both for vertical take-off/landing and hover flight, and for horizontal flight. In particular they are illustrated the role of the differential flatness property and the advantages it provides. Finally, the results of the open-loop control are shown: the control inputs obtained from the differential flatness analyses perfectly match the ones obtained from the trim for all the flight modes, and in particular a very high accuracy is achieved in the final results, with the exception of the variables $z(t)$ and $\theta(t)$ during horizontal flight, which present some imprecision in following the generated trajectories.

Chapter 5 explains, in Section 5.1, the procedure followed for the design of PID controllers, for the linearized and decoupled system via Singular Value Decomposition, during the different flight modes. Afterwards, in Section 5.1.2, the obtained results achieved with the addition of the feedback control, are compared to the results obtained with only the feedforward action. The usefulness of the feedback control part is evident comparing the results obtained with only the feedforward action, with respect to the generated trajectory. It allows to correct the inaccuracy of the feedforward action alone, which is particularly visible for the angle $\theta(t)$ in horizontal flight, and to keep the errors of the controlled variables close to zero.

Chapter 6 shows the reached conclusions and the possible further developments.

Chapter 2

Nonlinear dynamic model of the aircraft

2.1 System description

The UAV convertible aircraft considered is a tail-sitter, tilt-body, with two counter-rotating rotors, see Figure 2.1. The wings are symmetric triangular shaped, called delta-wing, and the airfoil is chosen to be symmetric, as illustrated in Figure 2.2.

In order to maneuver the aircraft, by modifying the aerodynamic forces and moments, the control surfaces have to be defined. Generally, when standard aircraft configurations are considered, the control surfaces include the elevator, the aileron and the rudder. The aileron deflection is denoted as δ_a, the rudder deflection as δ_r and the elevator deflection as δ_e, as represented in Figure 2.3. In the case of a delta-wing configuration, the control surfaces are called elevons. The left angular deflection is denoted as δ_{el} and the right angular deflection as δ_{er}. The mathematical relation between elevons and aileron-elevator

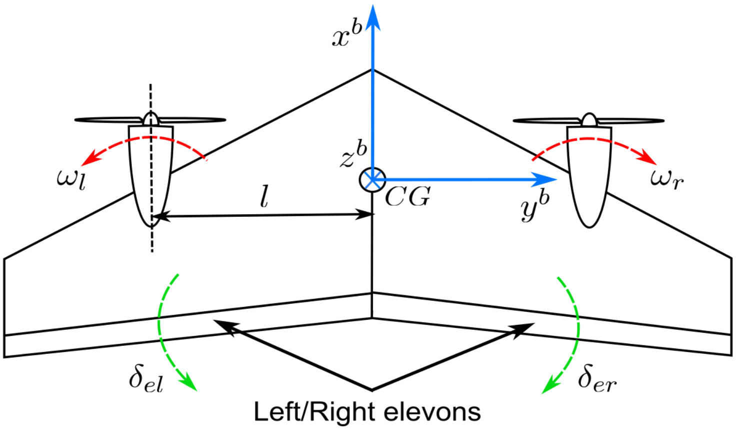


Figure 2.1: Simplified model of the delta-wing aircraft.

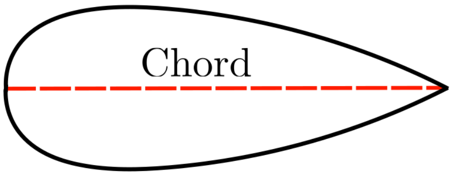


Figure 2.2: Symmetric profile of the airfoil, where the chord represents an imaginary straight line joining the leading edge and the trailing edge of the aerofoil.

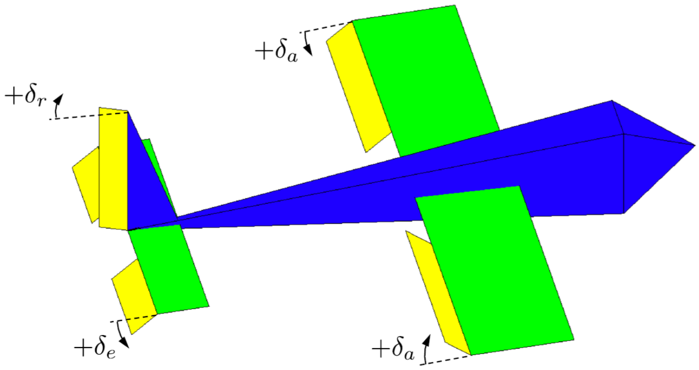


Figure 2.3: Control surfaces for a standard aircraft configuration. Image from [4].

signals is defined as follows:

$$\begin{bmatrix} \delta_e \\ \delta_a \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \delta_{er} \\ \delta_{el} \end{bmatrix} \tag{2.1}$$

In fact by driving the elevons differentially $\delta_{el} = -\delta_{er}$ has the same effect as ailerons, providing a torque about the x axis of the aircraft (roll control), instead driving the elevons together $\delta_{el} = \delta_{er}$ has the same effect as an elevator, producing a torque about the y axis of the aircraft (pitch control). The yaw movement is controlled with different spinning of the propellers $\omega_l \neq \omega_r$, called “Differential Propeller Rotation”.

In synthesis, the four control inputs are defined in this way:

- δ_{el}, δ_{er} which are the deflections of the left and right elevons, are expressed in [rad] and positive deflections are trailing edge down, as shown in Figure 2.1.
- ω_l, ω_r which are the rotation speeds of the left and right counter-rotating propeller-engines, are expressed in Pulse Width Modulation [PWM].

and so the control input vector is:

$$u = \begin{bmatrix} \delta_{el} \\ \delta_{er} \\ \omega_l \\ \omega_r \end{bmatrix} \tag{2.2}$$

The flight of this kind of aircraft can be differentiated into the following flight modes:

- Take-off, landing and hover: in this modes, the aircraft is positioned vertically on its tail, the aerodynamic velocity is small (or zero during hover) and the thrust from its propellers allows it to take off, hover and land like a helicopter.

- Horizontal flight (cruise): in this mode, the aircraft flies horizontally with higher aerodynamic velocities, like a conventional fixed-wing airplane and the control surfaces become the primary means of maneuvering the aircraft.

2.2 Assumptions and notations

The trigonometric functions sin, cos, tan are abbreviated into c, s, t respectively, and in some of the following equations the dependence on time (t) is avoided to save space.

The following assumptions are considered for the whole analysis:

- A constant gravity acceleration is assumed, equal to $g = 9.81\,\mathrm{m\,s^{-2}}$, hence the Center of Mass (CoM) of the aircraft coincides with its Center of Gravity (CG).

- Zero wind condition.

- Flat and fixed earth.

- Absence of external disturbances.

2.2.1 Skew-symmetric operator

Given a vector $P = [p_1, p_2, p_3]^T$ in $\mathbb{R}^3$, its skew-symmetric operator $[.]_\times$ is defined as:

$$[P]_\times = \begin{bmatrix} 0 & -p_3 & p_2 \\ p_3 & 0 & -p_1 \\ -p_2 & p_1 & 0 \end{bmatrix} \tag{2.3}$$

2.2.2 Reference frames

In this section it is explained what reference systems are taken into account throughout the whole analysis and what are the relations between them.

In this research project, the reference frames are indicated with upper-case calligraphic letters and the lower-case superscripts denote the projection frames, and are expressed with the right-hand rule.

Inertial-frame $\mathcal{I}$

The inertial-frame $\mathcal{I} = (O; x^i, y^i, z^i)$ has the origin O at the surface of the earth and the North-East-Down (NED) convention is used to define its axis x^i, y^i, z^i, which point respectively towards the north, the east and the center of the earth.

Vehicle-carried normal Earth frame $\mathcal{O}$

The vehicle-carried normal Earth $\mathcal{O} = (CG; x^o, y^o, z^o)$, has the origin at the CG of the aircraft and its axes are parallel to those of the inertial-frame $\mathcal{I}$.

The gravitational force vector is defined in this frame as:

$$G^o = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{2.4}$$

with $m = 1.56$ kg being the mass of the aircraft, as proposed in the Zagi flying wing by [4].

Body-frame $\mathcal{B}$

The body-frame $\mathcal{B} = (CG; x^b, y^b, z^b)$ is centered in the aircraft's Center of Gravity, with the axis x^b pointing towards the nose of the aircraft, y^b out of the right wing and z^b pointing into the belly of the aircraft, as illustrated in Figure 2.1. The axes x^b and z^b define the symmetry plane of the vehicle, to which y^b is perpendicular.

Defining three elementary rotation matrices in terms of three successive rotations ϕ, θ, ψ, called Euler-angles, around the x,y,z axes respectively:

$$\begin{gathered} R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi) & c(\phi) \end{bmatrix} \quad R_y(\theta) = \begin{bmatrix} c(\theta) & 0 & s(\theta) \\ 0 & 1 & 0 \\ -s(\theta) & 0 & c(\theta) \end{bmatrix} \\ R_z(\psi) = \begin{bmatrix} c(\psi) & -s(\psi) & 0 \\ s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{gathered} \tag{2.5}$$

then the relation between the body-frame $\mathcal{B}$ and the normal Earth-frame $\mathcal{O}$, shown in Figure 2.4, is defined as follows:

$$R_b^o = R_z(\psi(t))R_y(\theta(t))R_x(\phi(t)) = \tag{2.6}$$

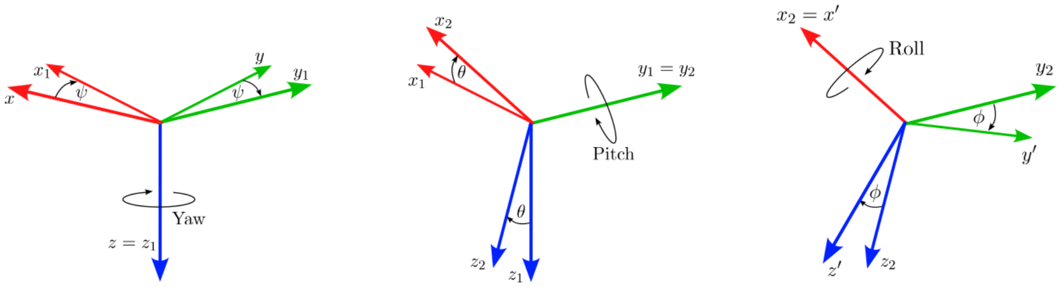


Figure 2.4: Three rotations from frame $\mathcal{B}$ to frame $\mathcal{O}$

$$= \begin{bmatrix} c(\psi)c(\theta) & -s(\psi)c(\phi) + c(\psi)s(\theta)s(\phi) & s(\psi)s(\phi) + c(\psi)s(\theta)c(\phi) \\ s(\psi)c(\theta) & c(\psi)c(\phi) + s(\psi)s(\theta)s(\phi) & -c(\psi)s(\phi) + s(\psi)s(\theta)c(\phi) \\ -s(\theta) & c(\theta)s(\psi) & c(\theta)c(\phi) \end{bmatrix} \tag{2.7}$$

where the subscript b denotes a vector defined in the frame $\mathcal{B}$, which is transformed into a vector in the frame $\mathcal{O}$, denoted with the superscript o.

Kinematic frame $\mathcal{K}$

The kinematic frame $\mathcal{K} = (CG; x^k, y^k, z^k)$ has the origin at the aircraft CG, and the axis x^k points towards the direction of the kinematic speed of the aircraft. The axis z^k is given by a first rotation of an angle $-\alpha_k$ (kinematic angle of attack) about y^b, and the axis y^k is the result of a second rotation of an angle β_k (kinematic sideslip angle) about z^k.

Remark *The kinematic velocity, also called ground speed, represents the velocity of the aircraft relative to the earth. Thus,*

$$V_k(t) = \frac{dOCG^{\mathcal{I}}}{dt} = \begin{bmatrix} v_k(t) \\ 0 \\ 0 \end{bmatrix} \tag{2.8}$$

where OCG is the vector connecting the center of the Inertial-frame, O, with the aircraft's CG.

Aerodynamic-frame $\mathcal{A}$

The aerodynamic-frame $\mathcal{A} = (CG; x^a, y^a, z^a)$, with the origin at the aircraft CG, has the x^a axis pointing towards the aircraft's aerodynamic speed. $\mathcal{A}$ is given by two rotations: z^a is the result of a first rotation of an angle $-\alpha_a = -\alpha$ (aerodynamic angle of attack) about the axis y^b, and the axis y^a and x^a are obtained with a second rotation of an angle $\beta_a = \beta$ (aerodynamic sideslip angle), about the axis z^a.

Remark *The aerodynamic velocity, also called air speed, represents the velocity of the aircraft with respect to the air. In other words, the aerodynamic velocity of the aircraft*

CG represents the distance, generated by the aircraft per second, between the aircraft's CG and a particle of air W, pushed to a new position by CG. Thus,

$$V_a(t) = \frac{dWCG^{\mathcal{I}}}{dt} = \begin{bmatrix} v_a(t) \\ 0 \\ 0 \end{bmatrix} \tag{2.9}$$

The normal Earth-frame $\mathcal{O}$ is related to the aerodynamic-frame $\mathcal{A}$ as:

$$R_a^o = R_z(\chi(t))R_y(\gamma(t))R_x(\mu(t)) = \tag{2.10}$$

$$= \begin{bmatrix} c(\chi)c(\gamma) & -s(\chi)c(\mu) + c(\chi)s(\gamma)s(\mu) & s(\chi)s(\mu) + c(\chi)s(\gamma)c(\mu) \\ s(\chi)c(\gamma) & c(\chi)c(\mu) + s(\chi)s(\gamma)s(\mu) & -c(\chi)s(\mu) + s(\chi)s(\gamma)c(\mu) \\ -s(\gamma) & c(\gamma)s(\mu) & c(\gamma)c(\mu) \end{bmatrix} \tag{2.11}$$

Through the angle of attack $\alpha(t)$ and the side-slip angle $\beta(t)$, the relation between the aerodynamic-frame $\mathcal{A}$ and the body-frame $\mathcal{B}$ can be defined as:

$$R_a^b = R_y(-\alpha(t))R_z(\beta(t)) = \begin{bmatrix} c(\alpha)c(\beta) & -c(\alpha)s(\beta) & -s(\alpha) \\ s(\beta) & c(\beta) & 0 \\ s(\alpha)c(\beta) & -s(\alpha)s(\beta) & c(\alpha) \end{bmatrix} \tag{2.12}$$

Remark *When the aerodynamic velocity $v_a(t) = 0$, the axis of the frame $\mathcal{A}$ and those of the frame $\mathcal{B}$ are aligned.*

2.2.3 Velocity notations

The relation between the kinematic speed, aerodynamic speed and $V_w(t)$, the velocity of the air mass relative to the ground, namely the wind velocity is given by:

$$V_k(t) = V_a(t) + V_w(t) \tag{2.13}$$

Assuming zero wind condition, $V_w(t) = 0$, then $V_k(t) = V_a(t)$, namely $\mathcal{A}$ coincides with $\mathcal{K}$. The velocity $V_a(t)$ can be expressed in the body-frame as:

$$V_a^b(t) = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \tag{2.14}$$

which allows to write, as [4] shows, the aerodynamic velocity and the angle of attack as:

$$v_a(t) = \sqrt{v_x^2 + v_y^2 + v_z^2} \tag{2.15}$$

$$\alpha(t) = \arctan\left(\frac{v_z}{v_x}\right) \tag{2.16}$$

2.2.4 Angular velocities notation

The attitude of the aircraft is represented by the Euler angles roll ϕ, pitch θ, and yaw ψ. Its time derivative is defined in the inertial frame as:

$$\Omega = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2.17}$$

The angular velocities of a general frame a relative to frame b, and projected in frame c, is defined as Ω_{ab}^c. For instance the angular velocity of the body frame $\mathcal{B}$ relative to $\mathcal{O}$ and projected in the body-frame $\mathcal{B}$, is:

$$\Omega_{bo}^b = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.18}$$

2.2.5 Inertia matrix

The inertia matrix is defined as:

$$I = \begin{bmatrix} \int (y^2 + z^2)dm & -\int xydm & -\int xzdm \\ -\int xydm & \int (x^2 + z^2)dm & -\int yzdm \\ -\int xzdm & -\int yzdm & \int (x^2 + y^2)dm \end{bmatrix} = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{xy} & I_y & -I_{yz} \\ -I_{xz} & -I_{yz} & I_z \end{bmatrix} \tag{2.19}$$

Where the diagonal terms of I are called moments of inertia, which are measures of the aircraft's tendency to oppose acceleration about a specific axis of rotation, and the off-diagonal terms are the products of inertia. In practice this matrix is numerically computed from mass properties using, for instance, CAD models. The integrals in (2.19) are calculated with respect to the axis x^b, y^b, z^b in frame $\mathcal{B}$, hence I is constant when viewed from the body-frame.

Due to the symmetry of the aircraft, with respect to the plane defined by x^b and z^b, then $I_{xy} = I_{yz} = 0$, which implies:

$$I = \begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{xz} & 0 & I_z \end{bmatrix} \tag{2.20}$$

where its values are taken from Zagi flying wing in [4] and they are $I_x = 0.1147$ kg m^2, $I_y = 0.0576$ kg m^2, $I_z = 0.1712$ kg m^2 and $I_{xz} = 0.0015$ kg m^2.

2.3 Aerodynamic forces and moments

A pressure distribution is generated around the UAV body, when it passes through the air. The strengh and the distribution of the pressure acting on the UAV is a function of the air density $\rho = 1.2682$ kg m^{-3}, the airspeed V_a, and the shape and attitude of the vehicle. A typical approach, in order to avoid characterizing the pressure distribution around the wing, is to capture the effect of the pressure with a combination of forces and moments. A common procedure, proposed in [4] , is to decompose the aerodynamic forces and moments into longitudinal and lateral ones.

2.3.1 Longitudinal aerodynamics

The longitudinal forces and moments act in the x^b-z^b plane, called pitch plane. They include the lift and drag forces in the x^b and z^b axis, F_{lift} and F_{drag}, and the pitch moment in the y^b axis, m_{pitch}. These are influenced by the angle of attack α, the pitch rate q and the elevator deflection δ_e.

$$F_{\text{lift}} = \frac{1}{2}\rho V_a^2 S C_L(\alpha, q, \delta_e) \tag{2.21}$$

$$F_{\text{drag}} = \frac{1}{2}\rho V_a^2 S C_D(\alpha, q, \delta_e) \tag{2.22}$$

$$m_{\text{pitch}} = \frac{1}{2}\rho V_a^2 S C_m(\alpha, q, \delta_e) \tag{2.23}$$

C_L, C_D and C_m are nondimensional aerodynamic coefficients, $S = 0.2589$ m^2 is the planform area of the UAV wing, and $c = 0.3302$ m is the mean chord of the UAV wing, as illustrated in Figure 2.2. These values are taken from Zagi flying wing in [4].

In order to have an accurate model of the aircraft, considering linear aerodynamic models in α for the lift and drag forces,

$$C_L = C_{L_0} + C_{L_\alpha}\alpha \tag{2.24}$$

$$C_D = C_{D_0} + C_{D_\alpha}\alpha \tag{2.25}$$

does not allow to include the stall effect, which happens for high angles of attack. When the aircraft is flying at low or moderate angles of attack, the flow over the wing is laminar and as it flows over the wing, it remains attached. In contrast, when the aircraft is flying at high angles of attack, exceeding the critical stall angle, the flow starts to separate from the top surface of the wing causing turbulent flow and a sudden drop in the lift produced

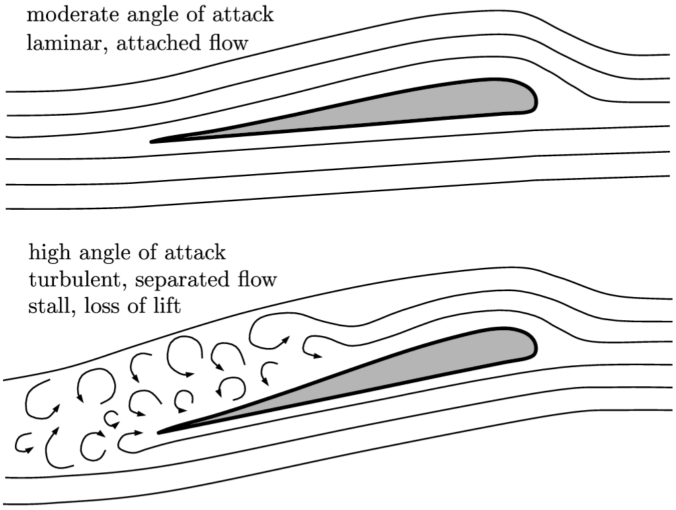


Figure 2.5: The upper drawing represents a wing under a laminar flow. The lower drawing represents the wing under stall conditions due to a high angle of attack. Image from [4].

by the wing, see Figure 2.5.

Hence linear aerodynamic models in α, erroneously predict that the lift force continues to increase as the angle of attack increases to physically unrealistic flight conditions. The proposed solution in [21], consists in interpolating the lift coefficients from two models: one for small angles of attack and one for large angles of attack.

Consider the following lift models:

$$\begin{cases} c_{L1}(\alpha, Re) = c_1 \sin(2\alpha) & \text{for } 0 \leq \alpha \leq \alpha_0(Re) \text{ or } 180^o - \alpha_o(Re) \leq \alpha \leq 180^o \\ c_{L2}(\alpha, Re) = c_2 \sin(2\alpha) & \text{otherwise} \end{cases} \tag{2.26}$$

where Re is the Reynolds number, which in fluid mechanics, is a dimensionless quantity that helps predict fluid flow patterns in different situations by measuring the ratio between inertial and viscous forces:

$$Re = \frac{\rho v L}{\mu} = \frac{v L}{\nu} \tag{2.27}$$

with v the velocity of the fluid in $(\mathrm{m\,s^{-1}})$, ρ the fluid density in $(\mathrm{kg\,m^{-3}})$, μ the dynamic viscosity in $(\mathrm{kg\,m^{-1}\,s^{-1}})$, L the length of the chord in (m), and $\nu = \mu\rho^{-1}$ is the kinematic viscosity in $(\mathrm{m^2\,s^{-1}})$.

In particular, when this number is low, the flows tend to be dominated by laminar flow, instead at high Reynolds numbers, flows tend to be turbulent.

c_1 is the lift constant for small angles of attack α, c_2 is an “average” lift constant and $\alpha_0(Re)$ is the angle at which the stall zone starts. In order to combine $c_{L1}(\alpha, Re)$ and $c_{L2}(\alpha, Re)$ into a continuous and differentiable curve, a function $\sigma(\alpha, \alpha_o(Re))$, which combines two sigmoid curves is exploited:

$$\sigma(\alpha, \alpha_o(Re)) = \frac{1}{1+e^{\alpha-\alpha_0(Re)}} + \frac{1}{1+e^{180-\alpha-\alpha_0(Re)}} \tag{2.28}$$

The angle $\alpha_0(Re)$ is scaled for the change of the Reynolds number:

$$\alpha_0(Re) = \alpha_{0\text{ref}} \left(\frac{Re}{Re_{\text{ref}}}\right)^{Re_{\text{exp}}} \tag{2.29}$$

where Re_{ref} is the reference Reynolds number, $\alpha_{0\text{ref}}$ is the angle at which the stall zone starts for the airfoil at the reference Reynolds number and Re_{exp} is the exponential scaling constant. Hence, the lift coefficient is modeled as:

$$C_L(\alpha, Re) = c_{L1}(\alpha, Re)\sigma(\alpha, \alpha_o(Re)) + c_{L2}(\alpha, Re)(1 - \sigma(\alpha, \alpha_o(Re))) \tag{2.30}$$

Regarding the drag coefficient, it is almost independent of the Reynolds number, hence it is modeled as:

$$C_D(\alpha, Re) \approx C_D(\alpha) = c_1 + 2c_2 \sin^2(\alpha) \tag{2.31}$$

For simulation purposes, as proposed in [16], a symmetric profile NACA0012 is considered, with $c_1 = 5.3344$ and $c_2 = 1.0150$, the reference Reynolds number $Re_{\text{ref}} = 160000$, $\alpha_{0\text{ref}} = 9$ deg and $Re_{\text{exp}} = 0.15$.

A $Re_{\text{ref}} = 160000$ corresponds to an aerodynamic velocity of 7.1595 $\mathrm{m\,s^{-1}}$, for the chord $c = 0.3302$ m and a kinematic viscosity $\nu = 1.4776e^{-05}$ $\mathrm{m^2\,s^{-1}}$, for the air at 15^o.

The obtained lift aerodynamic coefficient is shown in Figure 2.6.

Thus, the lift, drag and pitching moment are given, respectively, by the following relations:

$$F_{\text{lift}} = \frac{1}{2}\rho V_a^2 S(c_{L1}(\alpha, Re)\sigma(\alpha, \alpha_o(Re)) + c_{L2}(\alpha, Re)(1 - \sigma(\alpha, \alpha_o(Re))) \tag{2.32}$$

$$+ C_{L_q}\frac{c}{2V_a}q + C_{L_{\delta_e}}\delta_e) \tag{2.33}$$

$$F_{\text{drag}} = \frac{1}{2}\rho V_a^2 S(c_1 + 2c_2\sin^2(\alpha) + C_{D_q}\frac{c}{2V_a}q + C_{D_{\delta_e}}\delta_e) \tag{2.34}$$

$$m_{\text{pitch}} = \frac{1}{2}\rho V_a^2 S(C_{m_0} + C_{m_\alpha}\alpha + \frac{C_{m_q}cq}{2V_a} + C_{m_{\delta_e}}\delta_e) \tag{2.35}$$

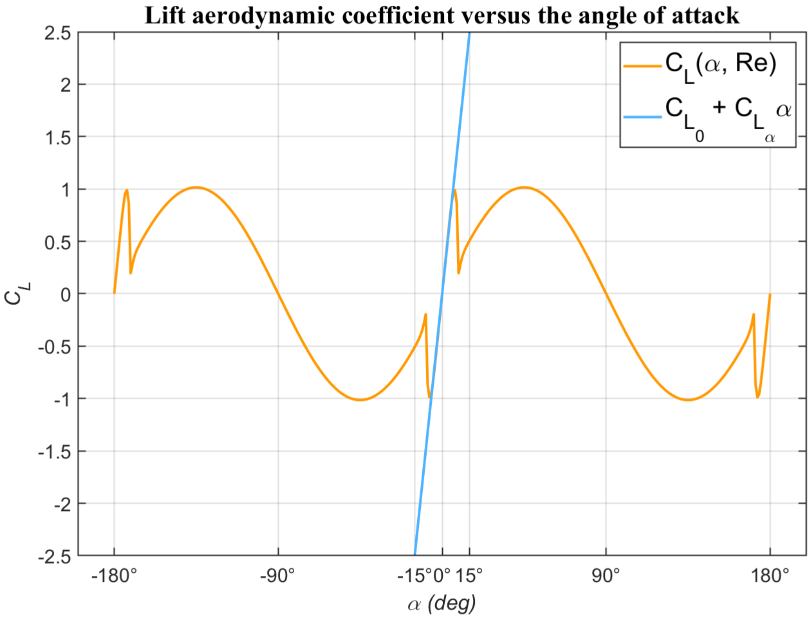


Figure 2.6: Lift aerodynamic coefficient C_L versus the angle of attack α, for $Re_{\text{ref}} = 160000$. The linear behaviour $C_L = C_{L_0} + C_{L_\alpha}\alpha$ is maintained for small α.

2.3.2 Lateral Aerodynamics

The forces and moments for the lateral aerodynamics, cause both rotational motions in roll and yaw, and translation motion in the lateral direction along the y^b axis, which will result in directional changes in the flight path of the UAV.

The lateral aerodynamics is mainly influenced by the sideslip angle β, the roll rate p, the yaw rate r and the deflection of the aileron δ_a.

Thus, the lateral force F_y, the roll moment l and the yaw moment n, are respectively:

$$F_y = \frac{1}{2}\rho V_a^2 S C_Y(\beta, p, r, \delta_{el}, \delta_{er}) \tag{2.36}$$

$$l = \frac{1}{2}\rho V_a^2 S b C_l(\beta, p, r, \delta_{el}, \delta_{er}) \tag{2.37}$$

$$n = \frac{1}{2}\rho V_a^2 S b C_n(\beta, p, r, \delta_{el}, \delta_{er}) \tag{2.38}$$

with $b = 1.4224$ m the wingspan of the aircraft, namely the distance from one wingtip to the opposite wingtip, and C_Y, C_L and C_n nondimensional aerodynamic coefficients. Accurate models can be obtained with linear aerodynamic coefficients, proposed in [4], as follows:

$$F_y \approx \frac{1}{2}\rho V_a^2 S(C_{Y_0} + C_{Y_\beta}\beta + \frac{C_{Y_p} bp}{2v_a} + \frac{C_{Y_r} br}{2v_a} + C_{Y_{\delta_a}}\delta_a) \tag{2.39}$$

$$l \approx \frac{1}{2}\rho V_a^2 Sb(C_{l_0} + C_{l_\beta}\beta + \frac{C_{l_p} bp}{2v_a} + \frac{C_{l_r} br}{2v_a} + C_{l_{\delta_a}}\delta_a) \tag{2.40}$$

$$n \approx \frac{1}{2}\rho V_a^2 Sb(C_{n_0} + C_{n_\beta}\beta + \frac{C_{n_p} bp}{2v_a} + \frac{C_{n_r} bp}{2v_a} + C_{n_{\delta_a}}\delta_a) \tag{2.41}$$

2.3.3 Aerodynamic coefficients

The same notation as [4] is maintained for the following aerodynamic coefficients.

C_{Y_0} is the value of the lateral force coefficient C_Y, when $\beta = p = r = \delta_a = 0$, which is equal to zero, as C_{l_0} and C_{n_0}, since the aircraft is symmetric about the plane $x^b - z^b$.

C_{m_α}, C_{l_β}, C_{n_β}, C_{l_p} and C_{n_r} are referred to as stability derivatives, since their values determine the static and dynamic stability of the aircraft. In particular C_{m_α}, C_{l_β} and C_{n_β} determine its static stability, representing the change in the moment coefficients due to the changes in the direction of the relative wind.

C_{m_α} is referred to as the longitudinal static stability derivative and it must be negative, such that an increase in α would cause the aircraft to nose down in order to maintain the nominal angle of attack.

C_{l_β} is called roll static stability derivative, which must be negative, such that it results in a rolling moment that roll the aircraft away from the direction of the sideslip, hence leading the angle β to zero.

C_{n_β} is the yaw static stability derivative, it must be positive to guarantee stability. This implies that for a positive sideslip angle, a positive yawing moment is induced. This yaws the aircraft into the direction of the relative wind, leading the angle β to zero.

C_{m_q}, C_{l_p} and C_{n_r} refer, respectively, to the pitch, roll and yaw damping derivative. These are negative, namely they produce a moment that opposes the direction of motion, meaning damping the motion.

$C_{m_{\delta_e}}$, $C_{l_{\delta_a}}$ are referred to as the primary control derivatives, associated with the deflection of the control surfaces. They are defined as primary, since the moments they produced are the mean result of the specific control surface deflection. Since a positive elevator deflection causes a nose-down pitching moment, negative about y^b, and a positive aileron deflection causes a right-wing-down rolling moment, positive about x^b, then $C_{m_{\delta_e}}$ is negative and $C_{l_{\delta_a}}$ is positive.

In particular, to provide positive and negative actions for the pitch and roll control, the deflection of the elevons should be:

- Deflected downward simultaneously, to generate a nose-up pitching moment, causing the aircraft's nose to pitch up (positive pitch control). This action is used for pulling the aircraft into a climb or reducing the angle of descent during landing.

- Deflected upward simultaneously, to generate a nose-down pitching moment, causing the aircraft's nose to pitch down (negative pitch control). This action is used for pushing the aircraft into a descent or reducing the angle of ascent during climb-out.

- The left elevon deflected upward while the right elevon is deflected downward, causing the aircraft to roll to the right (positive roll control).

- The left elevon is deflected downward while the right elevon is deflected upward, causing the aircraft to roll to the left (negative roll control).

The values of the aerodynamic coefficients are reported in Table 2.1.

Table 2.1: Aerodynamic coefficients for Zagi flying wing by [4]

Longitudinal Coef.	**Value**	**Lateral Coef.**	**Value**
C_{L_0}	0.09167	C_{Y_0}	0
C_{D_0}	0.01631	C_{l_0}	0
C_{m_0}	-0.02338	C_{n_0}	0
C_{L_α}	3.5016	C_{Y_β}	-0.07359
C_{D_α}	0.2108	C_{l_β}	-0.02854
C_{m_α}	-0.5675	C_{n_β}	-0.00040
C_{L_q}	2.8932	C_{Y_P}	0
C_{D_q}	0	C_{l_P}	-0.3209
C_{m_q}	-1.3990	C_{n_P}	-0.01297
$C_{L_{\delta_e}}$	0.2724	C_{Y_r}	0
$C_{D_{\delta_e}}$	0.3045	C_{l_r}	0.03066
$C_{m_{\delta_e}}$	-0.3254	C_{n_r}	-0.00434
C_{prop}	1.0	$C_{Y_{\delta_a}}$	0
M	50	$C_{l_{\delta_a}}$	0.1682
α_0	0.4712	$C_{n_{\delta_a}}$	-0.00328

2.3.4 Propulsion Forces and Moments

Propeller thrust

The thrust generated by a propeller can be modeled using the Bernoulli's principle, namely to compute the pressure ahead of and behind the propeller, and then to apply this pressure difference to the area of the propeller. The total pressure upstream and downstream of the propeller are defined, respectively, as:

$$P_{\text{upstream}} = P_0 + \frac{1}{2}\rho V_a^2 \tag{2.42}$$

$$P_{\text{downstream}} = P_0 + \frac{1}{2}\rho V_{\text{out}}^2 \tag{2.43}$$

with ρ the density of the air, P_0 the static pressure, V_{out} the speed of the air when it leaves the propeller, which is defined by the linear relationship:

$$V_{\text{out}} = k_{\text{motor}}\omega_i \tag{2.44}$$

with $i \in \{l, r\}$ and $k_{\text{motor}} = 40$, which specifies the efficiency of the motor.

Thus, the thrust produced by the motor, is defined in the body frame $\mathcal{B}$ as:

$$f_{p_i}^b = S_{\text{prop}}C_{\text{prop}}(P_{\text{downstream}} - P_{\text{upstream}}) = \tag{2.45}$$

$$= \frac{1}{2}\rho S_{\text{prop}}C_{\text{prop}}\left((k_{\text{motor}}\omega_i)^2 - V_a^2\right) \tag{2.46}$$

Acting along the x^b axis:

$$F_{p_i}^b = \frac{1}{2}\rho S_{\text{prop}}C_{\text{prop}} \begin{bmatrix} (k_{\text{motor}}\omega_i)^2 - V_a^2 \\ 0 \\ 0 \end{bmatrix} \tag{2.47}$$

Propeller Torque

When a propeller of the aircraft spins, it generates a force to the air that goes through the propeller. This increases the momentum of the air while generating a thrust force on the aircraft. The result of these equal and opposite forces applied by the air on the propeller, is a torque about the propeller axis of rotation. The propeller applies an equal and opposite torque to the motor, in response to the torque the motor applies to the propeller, and therefore to the air. This torque is proportional to the square of the propeller's angular velocity and opposed to the propeller's direction of rotation, as shown by:

$$T_P = -K_{T_P}(k_\Omega\omega_i)^2 \tag{2.48}$$

with $K_{T_P} = 1e^{-6}$ a constant obtained by experiment and $\Omega = (k_\Omega\omega_i)$, $i \in \{l, r\}$, the left and right propeller speed, with $k_\Omega = 1e^3$, where both values are taken from Zagi flying wing by [4].

Hence, the moments induced by the propulsion system, in the body frame $\mathcal{B}$, are given by:

$$m_t = \begin{bmatrix} -K_{T_P}(k_\Omega \omega_i)^2 \\ 0 \\ 0 \end{bmatrix} \tag{2.49}$$

Even if the effects of this propeller torque are minor, if they are not considered it generates a slow rolling motion opposed to the propeller rotation. This could be easily counteracted by generating a rolling moment, due to a small differential elevon deflections.

As explained in [2], each propulsion force induces also a moment, in the body-frame, as follows:

$$m_{p_i} = \overrightarrow{(CG, O_i)}|_{\mathcal{B}} \wedge \begin{bmatrix} F_{p_i}^b \\ 0 \\ 0 \end{bmatrix} \tag{2.50}$$

where $\overrightarrow{(CG, O_i)}|_{\mathcal{B}}$ gives the coordinates of the application point for each of the two propulsion forces in the body-frame, $i \in \{l, r\}$.

Since the two propellers are positioned in a symmetric way in the plane formed by x^b and y^b, with a distance of x_p on the axis x^b and $y_p = \frac{b}{4}$ on y^b, then the total moment from the propulsion forces, in the body-frame $\mathcal{B}$, can be defined as:

$$m_p = \begin{bmatrix} 0 \\ 0 \\ y_p(f_{p_l}^b - f_{p_r}^b) \end{bmatrix} \tag{2.51}$$

2.4 Wing-propeller interaction

In order to allow attitude control in hover flight, it is necessary to introduce the slip-stream air induced velocity, due to the spinning of the two propellers. The section of the wing affected is called wet area S_w, illustrated in Figure 2.7, and its effect on the aerodynamic equations is modeled by taking into account a induced free-stream velocity v_i.

For this simulation purpose, S_w is modeled as half of the total area of the wing S.

The induced velocity of the slip-stream, at the downstream side of the propeller disk, as proposed by [17], is a solution of:

$$v_i^2 + v_a \cos(\alpha) v_i - \frac{F_{\text{thrust}}}{2\rho S_{\text{prop}}} = 0 \tag{2.52}$$

where $S_{\text{prop}} = 0.0314\ \text{m}^2$ is the area of the propeller disk.

Thus,

$$v_i = \frac{1}{2}\left[\sqrt{(v_a \cos(\alpha))^2 + \frac{2F_{\text{thrust}}}{\rho S_{\text{prop}}}} - v_a \cos(\alpha)\right] \tag{2.53}$$

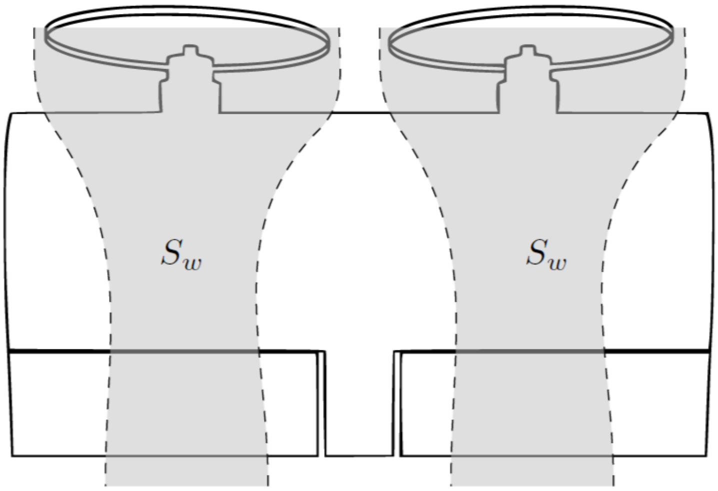


Figure 2.7: Wet and dry areas of the wing, due to the wing-propeller interaction. Image from [17].

The aerodynamic forces and moments in the slip-stream area are considered to act in the body frame $\mathcal{B}$ and are given by the lift force, the drag force and the pitch moment respectively:

$$L_w = \frac{1}{2}\rho v_i^2 S_w C_{L_{\delta_e}} \delta_e \tag{2.54}$$

$$D_w = \frac{1}{2}\rho v_i^2 S_w C_{D_{\delta_e}} \delta_e \tag{2.55}$$

$$m_w = \frac{1}{2}\rho v_i^2 c S_w C_{m_{\delta_e}} \delta_e \tag{2.56}$$

It is also considered the roll moment induced by the propeller wing interaction in the body-frame, which is not proposed in [17], and it is defined as:

$$m_e = y_p \frac{1}{2}\rho v_i^2 S_w C_{L_{\delta_a}} (\delta_a) \tag{2.57}$$

2.5 Total forces and moments

The total force vector acting in the aerodynamic frame $\mathcal{A}$ is given by:

$$F^a = \begin{bmatrix} X^a(t) \\ Y^a(t) \\ Z^a(t) \end{bmatrix} = F_a^a + R_b^a F_p^b + R_b^a F_w^b \tag{2.58}$$

with F_a^a defined as the contributions of the lift force, the lateral force and the drag force in frame $\mathcal{A}$:

$$F_a^a = \begin{bmatrix} -F_{\text{drag}} \\ F_y \\ -F_{\text{lift}} \end{bmatrix} \tag{2.59}$$

and F_w^b as the contribution of the aerodynamic forces due to the wing-propeller interaction:

$$F_w^b = \begin{bmatrix} -D_w \\ 0 \\ -L_w \end{bmatrix} \tag{2.60}$$

The total torque vector, is expressed in the body frame $\mathcal{B}$ and it's given by:

$$\tau^b = \begin{bmatrix} L^b(t) \\ M^b(t) \\ N^b(t) \end{bmatrix} = \tau_a^b + m_p + m_t + m_w + m_e \tag{2.61}$$

with the vector τ_a^b defined as follows:

$$\tau_a^b = \begin{bmatrix} L_a^b(t) \\ M_a^b(t) \\ N_a^b(t) \end{bmatrix} = \begin{bmatrix} l \\ m_{\text{pitch}} \\ n \end{bmatrix} \tag{2.62}$$

2.6 Nonlinear dynamic model

The nonlinear model for the convertible aircraft is described using the Newton-Euler formalism.

It is worth to remark that there are various ways on the choice of the frames to write the equations for the translational and rotational motion.

In this research project both the equations for the translational and rotational motion are written in the body frame $\mathcal{B}$, as proposed in [2]:

$$\dot{\xi} = R_b^o V_a^b \tag{2.63}$$

$$m\frac{dV_a^b}{dt} + \Omega_{bo}^b \wedge mV_a^b = R_a^b F^a + (R_b^o)^T G^o \tag{2.64}$$

$$\dot{R}_b^o = R_b^o[\Omega_{bo}^b]_\times \tag{2.65}$$

$$\frac{d(I\Omega_{bo}^b)}{dt} + \Omega_{bo}^b \wedge I\Omega_{bo}^b = \tau^b \tag{2.66}$$

Where $\xi = [x, y, z]^T$ and (x, y, z) is the position of the CG of the aircraft relative to frame $\mathcal{I}$.

$\Omega_{bo}^b = [p, q, r]^T$, with (p, q, r) being the rotational speeds, and $[.]_\times$ represents the skew-symmetric operator, defined in (2.3).

2.6.1 Gimbal lock and Modified Rodrigues Parameters (MRP)

The primary objective of this research project is to model and control a convertible aircraft, which is capable of taking-off and landing vertically.

Since the aircraft is a tail-sitter, it has to perform a rotation of 90^o around the pitch axis from the vertical take-off to the forward flight, and from the forward-flight to the vertical landing.

This requires particular attention to the choice of the most suitable description for the orientation of the body, indeed, choosing a good attitude coordinates allows not only to simplify mathematical calculations and analysis, but in particular to avoid geometrical and mathematical singularities.

This leads to a deep investigation on the most suitable description for the rigid body orientation, which is presented here.

The first subject of investigation concerns the Euler angles, which include three angles with respect to a fixed coordinate system. The angles are uniquely determined except for the singular case, which is called Gimbal lock.

To better understand this issue consider an analogy using physical gimbals, which are mechanical devices with nested rings that allow an object to rotate in three axes: yaw, pitch and roll.

Imagine an aircraft with three gimbals representing each rotational axis, in such a system, any orientation can be achieved by independently rotating each gimbal.

However, the problem occurs when the pitch angle approaches 90 degrees (or -90 degrees), where the roll and yaw axes become aligned, resulting in a loss of independence between the rotations. Effectively, one degree of freedom is lost, and now there are only two independent axes for rotation instead of three, which is shown in Figure 2.8.

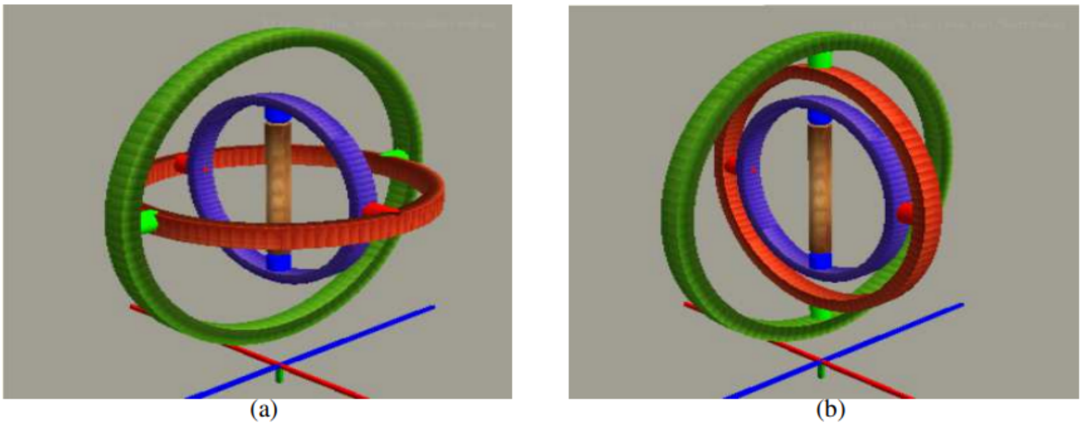


Figure 2.8: In (a) the rotor of the gyroscope is maintained in balance: The red gimbal ring performs a rotation along the x-axis (red line) to produce the pitch motion; the blue gimbal ring performs a rotation along the y-axis (green line) to produce the yaw motion; the green gimbal ring performs a rotation along the z-axis (blue line) to produce the roll motion. In (b) the rotor of the gyroscope is instead out of balance: Gimbal lock, namely two out of the three gimbals are in the same plane and one degree of freedom is lost. Photo by [28].

It becomes evident that Euler angles are not suitable for the specific objectives of this research project and an alternative orientation description, which allows to overcome the gimbal lock issue, must be explored.

A possible approach, as proposed in [28], could be the use of the rotation matrices R for the attitude representation, which can globally and uniquely denote any orientation. Nevertheless, they include six extra parameters due to the orthogonality condition $R^T R = I_{3\times3}$, resulting in a 9-parameter representation.

This high-redundancy for describing a relative orientation makes the rotation matrices less desirable, leading to further investigations into alternative and more efficient representations.

Two possible solutions, which also overcome the gimbal lock issue, are the quaternions and the Modified Rodrigues Parameters (MRP).

Quaternions, also called Euler parameters, provide a nonsingular attitude description and are well suited to describe arbitrary and large rotations.

The quaternion vector β is described in terms of the principal rotation elements as:

$$\beta_0 = \cos(\frac{\Phi}{2}) \tag{2.67}$$

$$\beta_1 = e_1 \sin(\frac{\Phi}{2}) \tag{2.68}$$

$$\beta_2 = e_2 \sin(\frac{\Phi}{2}) \tag{2.69}$$

$$\beta_3 = e_3 \sin(\frac{\Phi}{2}) \tag{2.70}$$

Since $e_1^2 + e_2^2 + e_3^2 = 1$, then the parameters β_i, $i = 0, 1, 2, 3$, have to satisfy the holonomic constraint:

$$\beta_0^2 + \beta_1^2 + \beta_2^2 + \beta_3^2 = 1 \tag{2.71}$$

which describes a four-dimensional unit sphere, on which any rotation has a trajectory on this constrained sphere. In this notation, due to the non-uniqueness of the principal rotation elements, given a certain attitude, the same orientation will be described by two sets of Euler parameters. The same quaternion vector β is obtained switching between the sets $(\hat{e}, \Phi)$ and $(-\hat{e}, -\Phi)$, when $\hat{e} = [e_1, e_2, e_3]^T$.

It is worth to notice that any point on the unit constraint sphere surface and its anti-pole represent the same exact orientation, with the only difference that one specifies the orientation through the shortest single axis rotation and the other through the longest. To choose the Euler parameter vector corresponding to the shortest rotation, namely $\Phi \leq 180$ deg, the coordinate β_0 must be nonnegative.

The second solution, which is adopted to describe the nonlinear model of the aircraft, consists in the use of Modified Rodrigues Parameters. As described in [26], the MRP vector σ is defined in terms of Euler parameters as the transformation:

$$\sigma_i = \frac{\beta_i}{1 + \beta_0} \quad i = 1, 2, 3 \tag{2.72}$$

This equation is well-behaved, except near the singularity at $\beta_0 = -1$, where $\Phi \to \pm 360$ deg. It describes a stereographic projection of the Euler parameter unit sphere onto the MRP hyperplane normal to the β_0 axis at $\beta_0 = 0$, where the projection point is at $\beta = (-1, 0, 0, 0)$. This is illustrated in Figure 2.9.

It needs to be taken into account the fact that the projection of the alternate Euler parameter vector $-\beta$ results in a distinct set of shadow (or image) MRPs. Hence, one can arbitrarily switch between the two vectors through the mapping:

$$\sigma_i^S = \frac{-\beta_i}{1 - \beta_0} = \frac{-\sigma_i}{\sigma^2} \quad i = 1, 2, 3 \tag{2.73}$$

where the choice as to which vector is the original and which is the shadow vector is arbitrary. It is typically referred to σ to represent the mapping point interior to the unit sphere and σ^S the point exterior to the unit sphere.

Considering the non-uniqueness of the principal rotation vector γ and the Euler parameter vector β, one set of MRPs always corresponds to a principal rotation $\phi \leq 180$ deg and the other to $\Phi \geq 180$ deg. This behaviour can be seen in Figure 2.9. Of particular importance is the unit sphere $|\sigma| = 1$, which corresponds to all principal rotations of 180 deg from the origin. When one set of MRPs exits the unit sphere, the other (shadow) set enters.

It is worth to notice that the definitions of the two sets allows to avoid MRP singularities altogether, by switching between original and shadow MRP sets as one MRP vector approaches a singular orientation. In particular, comparing to the original MRPs which are singular at $\Phi = \pm 360$ deg, the shadow MRPs have a singular orientation at $\Phi = 0$ deg. The switch between the two MRPs is chosen here when the vector σ enters the surface $\sigma^T\sigma = 1$. This is due to the fact that the mapping between the two MRP vectors simplifies on this surface to $\sigma^S = \sigma$ and that the magnitude of σ will remain bounded above by 1. This last property is fundamental, as it reflects the fact that two orientations can differ only by a finite rotation, and also the current MRP attitude description will always describe the shortest principal rotation.

Hence, this choice provides for a nonsingular, bounded and minimal attitude description, which is suited to describe large and arbitrary motions.

Substituting the inverse transformation:

$$\beta_0 = \frac{1-\sigma^2}{1+\sigma^2} \quad \beta_i = \frac{2\sigma_i}{1+\sigma^2} \quad i = 1,2,3 \tag{2.74}$$

defining $\sigma^{2n} = (\sigma^T\sigma)^n$, into the holonomic constraint (2.71), the direction cosine matrix is obtained:

$$[C] = \frac{1}{(1+\sigma^2)^2} \times \begin{bmatrix} 4(\sigma_1^2 - \sigma_2^2 - \sigma_3^2) + (1-\sigma^2)^2 & 8\sigma_1\sigma_2 + 4\sigma_3(1-\sigma^2) & \cdots \\ 8\sigma_2\sigma_1 - 4\sigma_3(1-\sigma^2) & 4(-\sigma_1^2 + \sigma_2^2 - \sigma_3^2) + (1-\sigma^2)^2 & \cdots \\ 8\sigma_3\sigma_1 + 4\sigma_2(1-\sigma^2) & 8\sigma_3\sigma_2 - 4\sigma_1(1-\sigma^2) & \end{bmatrix}$$

$$\begin{bmatrix} \cdots & 8\sigma_1\sigma_3 - 4\sigma_2(1-\sigma^2) \\ \cdots & 8\sigma_2\sigma_3 + 4\sigma_1(1-\sigma^2) \\ & 4(-\sigma_1^2 - \sigma_2^2 + \sigma_3^2) + (1-\sigma^2)^2 \end{bmatrix} \tag{2.75}$$

Hence, the orientation of the aircraft from frame $\mathcal{O}$ to frame $\mathcal{B}$, R_o^b, is defined with this notation, that in compact vector form is parametrized in terms of the MRP as:

$$R_o^b = [I_{3\times 3}] + \frac{8[\tilde{\sigma}]^2 - 4(1-\sigma^2)[\tilde{\sigma}]}{(1+\sigma^2)^2} \tag{2.76}$$

where $[\tilde{\sigma}]$ is the MRP skew-symmetric operator.

The state vector is defined as $x = [x, y, z, v_x, v_y, v_z, \sigma_1, \sigma_2, \sigma_3, p, q, r]^T$, with σ_1, σ_2, and σ_3 the MRP vector components explained in (2.72).

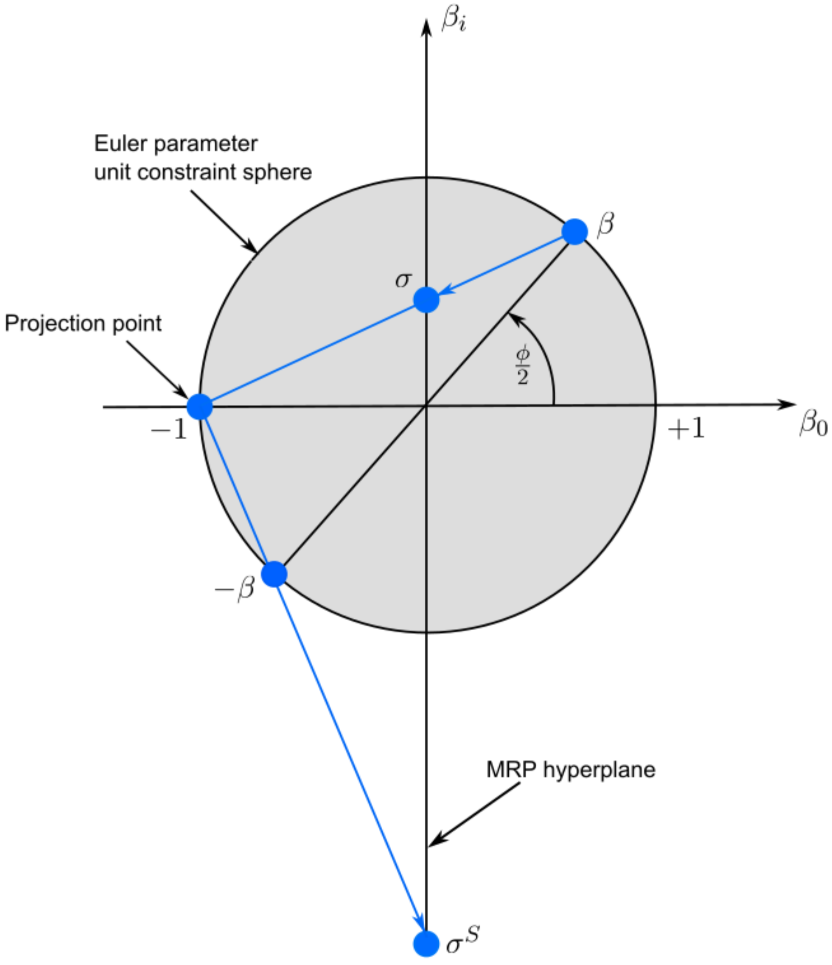


Figure 2.9: Singular behaviour of the MRPs as they describe a complete revolution.

Chapter 3

Differential flatness

This chapter is initially focused on the differential flatness theory, which is briefly illustrated for a general nonlinear system in Section 3.1. Afterwards the chosen analysis approach is explained deeply, in Section 3.2, analyzing the reasons that led to the final solution and the choice to proceed separately for zero/small and for higher aerodynamic velocities. Finally, in Sections 3.3 and Section 3.4, the representation of the mathematical model adopted in the different flight modes, the simplifying assumptions, the choice of the flat-outputs and the resulting solutions are provided.

3.1 Theoretical background

The flatness theory can be a very useful approach to the control problem of a nonlinear system. It was initially developed by Jean Levine, see [11], but many researchers summarized in their works and papers its main aspects and the possible applications to several systems. The purpose of this section is to explain which is the concept of flatness, seen from an engineering point of view, and which are the most important advantages of this property.

Flatness, as [13] and [20] propose, is a system property that extends the notion of controllability from linear systems to nonlinear dynamical systems. If one system proves this property is called flat system and it admits a set of outputs (flat outputs), which can be used to express all its states and inputs, in terms of the flat outputs and a finite number of their time derivatives, see Figure 3.1.

Definition A nonlinear system

$$\dot{x}(t) = f(x(t), u(t)), \quad u(t) \in \mathbb{R}^m, \quad x(t) \in \mathbb{R}^n, \quad m \leq n \tag{3.1}$$

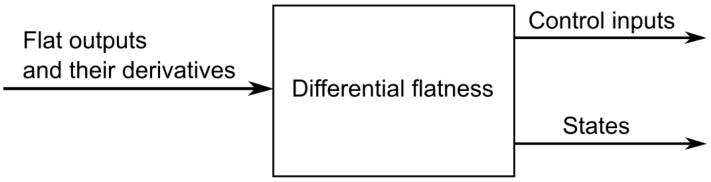


Figure 3.1: Differential flatness block.

is differentially flat if, and only if, there exists an m-dimensional vector

$$y = \phi(x, u, \dot{u}, ..., u^{(\alpha)}) \tag{3.2}$$

such that x and u can be expressed as functions of the components of y and a finite number of their time derivatives.

Applications of flatness to problems of engineering interest have grown persistently in recent years: when a system is flat, as [18] suggests, it is an indication that the nonlinear structure of the system is well characterized and one can exploit that structure in designing control algorithms for motion planning, trajectory generation, stabilization and for the feedforward control. The application to the trajectory generation and to the control problem is evident for several reasons. In particular, the outputs of the block of trajectory generation are only the flat ones, which can be defined as different kind of splines (polynomial, trigonometric, exponential, etc), also the feedforward control is realized only knowing algebraic relationships between control inputs and flat outputs. In the feedback the states are built knowing the measurements and so having previously defined the relationships between the flat outputs and the states, it is simple to get the actual flat outputs and calculate the error for the reference values. As a result, there are no differential equations to solve.

In this research project the differential flatness property is exploited for the feedforward control.

3.2 Introducing the chosen analysis approach

A first approach to the differential flatness analysis consisted in considering a unique non-linear dynamic model for the convertible aircraft, with the equations in the translational motion written in frame $\mathcal{A}$, and those for the rotational motion in frame $\mathcal{B}$. In this model the orientation from frame $\mathcal{O}$ to frame $\mathcal{A}$ was expressed in terms of quaternions, as:

$$R_o^a = I_3 - 2\beta_0[\beta]_\times + 2[\beta]_\times[\beta]_\times \tag{3.3}$$

where $[\beta]_\times$ is the skew-symmetric operator of the quaternion vector β. As explained in Section 2.6.1, quaternions allow to avoid the gimbal lock issue present during the vertical take-off/landing and hover flight.

The chosen set of flat-outputs was defined as

$$y(t) = (\beta_0, \beta_1, \beta_2, \alpha) \tag{3.4}$$

with $\beta_0, \beta_1, \beta_2$ the first three components of the quaternion vector, defined as (2.67), and the simplyfing assumption of the side-slip angle equal to zero, which implies $Y^a(t) = 0$, was considered.

The mathematical analysis was carried out successfully, allowing to express all the states and control inputs as a function of the flat-outputs and a finite number of their derivatives, but resulting in a singularity issue related to the fact that several equations presented a division by the aerodynamic velocity, which could not be simplified. As already explained, this is not an issue during the horizontal flight, since v_a is always greater than zero, instead, it represents a problem when it comes to the hover flight, in which the aerodynamic velocity is equal to zero. This reason led to the impossibility of using the same differential-flatness analysis both for the horizontal flight and for the hover flight.

Having to distinguish into two different analyses, and possibly two different representations of the model of the system, led to another possible solution. In this proposed solution, two different representations of the mathematical model of the aircraft are considered:

- In the first model the equations for the translational motion are expressed in the aerodynamic frame $\mathcal{A}$ and those for the rotational motion in the body frame $\mathcal{B}$, and it is used for the differential flatness analysis for high aerodynamic velocities (horizontal flight).

- In the second model the equations for the translational motion are expressed in frame $\mathcal{I}$ and those for the rotational motion in the body frame $\mathcal{B}$, and it is used for the differential flatness analysis for zero and small aerodynamic velocities (vertical take-off/landing and hover flight).

Dealing with two different mathematical models offers the advantage of dealing with the gimbal lock issue only in the second one, for vertical take-off/landing and hover flight, since during the horizontal flight this issue is not encountered. To overcome this problem a different solution from the one proposed in Section 2.6.1, is adopted, which consists in the use of vertical Euler angles and it will be explained in details later in this chapter, in Section 3.4.1.

The choice of expressing both models in terms of the Euler angles and vertical Euler angles, abandoning the first approach in terms of quaternions also for the horizontal flight, derives from a practical reason which was explored afterwards by searching for this other approach. Indeed, the use of the quaternions or the MRPs, as proposed in (2.63), offers a good solution to define a unique nonlinear dynamic model for the simulator while avoiding the gimbal lock issue in the vertical take-off/landing and in hover flight, but it does not offer an intuitive and straightforward representation of the behaviour of the aircraft during flight, which is instead provided by the use of Euler angles.

3.3 Differential flatness analysis for horizontal flight

3.3.1 Mathematical model

The differential flatness analysis is performed on the nonlinear dynamic model of the aircraft, where the equations for the translational motion are expressed in the aerodynamic frame $\mathcal{A}$ and those for the rotational motion in the body frame $\mathcal{B}$ in order to simplify the equations, as [1] proposes:

$$\dot{\xi} = R_a^o V_a^a \tag{3.5}$$

$$m\frac{dV_a^a}{dt} + \Omega_{ao}^a \wedge mV_a^a = F^a + (R_a^o)^T G^o \tag{3.6}$$

$$\dot{R}_a^o = R_a^o[\Omega_{ao}^a]_\times \tag{3.7}$$

$$\frac{d(I\Omega_{bo}^b)}{dt} + \Omega_{bo}^b \wedge I\Omega_{bo}^b = \tau^b \tag{3.8}$$

Where the same notation as (2.63) is maintained. The relation between Ω_{ao}^a and Ω_{bo}^b derives from the following property:

$$\Omega_{ao} = \Omega_{ab} + \Omega_{bo} \tag{3.9}$$

Projecting this equation in frame $\mathcal{A}$:

$$\Omega_{ao}^a = \Omega_{ab}^a + R_b^a \Omega_{bo}^b \tag{3.10}$$

and using:

$$\dot{R}_a^b = R_a^b[\Omega_{ab}^a]_\times \qquad \Longrightarrow \qquad [\Omega_{ab}^a]_\times = (R_a^b)^T \dot{R}_a^b \tag{3.11}$$

it becomes possible to determine the elements of Ω_{ab}^a as functions of the angle of attack $\alpha(t)$ and the side-slip angle $\beta(t)$, and their derivatives:

$$\Omega_{ab}^a = \begin{bmatrix} -\dot{\alpha}s(\beta) \\ -\dot{\alpha}c(\beta) \\ \dot{\beta} \end{bmatrix} \tag{3.12}$$

Substituting (3.12) in (3.10), the expression of Ω_{ao}^a as a function of Ω_{bo}^b is obtained.

The complete nonlinear dynamic equations of the model are developed as follows:

$$\dot{x}(t) = c\chi c\gamma v_a \tag{3.13}$$

$$\dot{y}(t) = s\chi c\gamma v_a \tag{3.14}$$

$$\dot{z}(t) = -s\gamma v_a \tag{3.15}$$

$$\dot{v}_a(t) = \frac{X^a}{m} - s\gamma g \tag{3.16}$$

$$\dot{\beta}(t) = s\alpha p - c\alpha r + \frac{c\gamma s\mu m g + Y^a}{m v_a} \tag{3.17}$$

$$\dot{\alpha}(t) = q - (c\alpha p + s\alpha r)t\beta + \frac{c\gamma c\mu}{c\beta}\frac{g}{V_a^a} + \frac{Z^a}{c\beta m v_a} \tag{3.18}$$

$$\dot{\chi}(t) = \frac{-Z^a s\mu + Y^a c\mu}{v_a m c\mu} \tag{3.19}$$

$$\dot{\gamma}(t) = \frac{-c\gamma g m - Y^a s\mu - Z^a c\mu}{v_a m} \tag{3.20}$$

$$\dot{\mu}(t) = \frac{-c\mu c\gamma s\beta g}{v_a c\beta} + \frac{p c\alpha + r s\alpha}{c\beta} - \frac{Z^a s\beta}{v_a m c\beta} + \frac{s\gamma(Y^a c\mu - Z^a s\mu)}{v_a m c\gamma} \tag{3.21}$$

$$\dot{p}(t) = \frac{(I_{xz}(I_{xx} - I_{yy} + I_{zz})p - (I_{xz}^2 - I_{zz}(I_{yy} - I_{zz}))r)q + I_{xz}N^b + I_{zz}L^b}{I_{xx}I_{zz} - I_{xz}^2} \tag{3.22}$$

$$\dot{q}(t) = \frac{-I_{xz}p^2 - r(I_{xx} - I_{zz})p + I_{xz}r^2 + M^b}{I_{yy}} \tag{3.23}$$

$$\dot{r}(t) = \frac{((I_{xz}^2 + I_{xx}(I_{xx} - I_{yy}))p - I_{xz}(I_{xx} - I_{yy} + I_{zz})r)q + I_{xx}N^b + I_{xz}L^b}{I_{xx}I_{zz} - I_{xz}^2} \tag{3.24}$$

where $x = [x, y, z, v_a, \beta, \alpha, \chi, \gamma, \mu, p, q, r]^T$ is the state vector and $u = [\delta_{el}, \delta_{er}, \omega_l, \omega_r]^T$ is the control input vector, as defined in (2.1).

In order to simplify the analysis, it is assumed that the side-slip angle β is equal to zero, which implies $Y^a(t) = 0$, see (2.58). Also, the velocity induced by the propeller-wing interaction is assumed to be null and the aerodynamic coefficients are assumed to be linear in the angle of attack $\alpha(t)$.

3.3.2 Choice of the flat-outputs

As [6] explains, some remarks need to be taken into account in the choice of the flat outputs, in particular:

- The number of the flat outputs need to coincide to the number of the control inputs.
- It is not required for the flat outputs to have particular physical meaning, however it is more intuitive to consider first the physical variables of the system as flat-outputs candidates.

Since the number of control inputs is four, then the number of flat-outputs has to be the same. A set of flat-outputs for this model can be given by:

$$y(t) = (x(t), y(t), z(t), \alpha(t)) \tag{3.25}$$

where $x(t), y(t), z(t)$ is the position of the CG of the aircraft relative to frame $\mathcal{I}$ and $\alpha(t)$ is the angle of attack. In this way the 12 system variables and the input variables can be expressed, both, as functions of these flat outputs and a finite number of their time derivatives. In other words, 16 system variables and control inputs, minus 4 flat outputs that can be expressed as functions of themselves, implies that in total 12 variables need to be expressed as a function of this set of flat outputs.

3.3.3 Mathematical proof of the differential flatness property

The first step consists in deriving from the equations (3.13), (3.14), (3.15), the aerodynamic velocity $v_a(t)$ and the angles $\gamma(t)$ and $\chi(t)$:

$$v_a(t) = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} \tag{3.26}$$

$$\gamma(t) = \arcsin\left(-\frac{\dot{z}}{v_a}\right) \tag{3.27}$$

$$\chi(t) = \arctan\left(\frac{\dot{y}}{\dot{x}}\right) \tag{3.28}$$

Recall that no singularities are present in (3.27), as in all the following equations, since the aerodynamic velocity is always greater than zero, and it is exploited the atan2 instead of the arctan in (3.28).

The derivatives of these states are necessary for the following analysis, and are defined as:

$$\dot{v}_a(t) = \frac{\dot{x}\ddot{x} + \dot{y}\ddot{y} + \dot{z}\ddot{z}}{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} \tag{3.29}$$

$$\ddot{v}_a(t) = -\frac{(2\dot{x}\ddot{x} + 2\dot{y}\ddot{y} + 2\dot{z}\ddot{z})^2}{4(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)^{\frac{3}{2}}} + \frac{2\ddot{x}^2 + 2\dot{x}\,\dddot{x} + 2\ddot{y}^2 + 2\dot{y}\,\dddot{y} + 2\ddot{z}^2 + 2\dot{z}\,\dddot{z}}{2\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}} \tag{3.30}$$

$$\dot{\gamma}(t) = -\frac{1}{v_a\sqrt{1 - \frac{\dot{z}^2}{v_a^2}}}\ddot{z} + \frac{\dot{z}}{v_a^2\sqrt{1 - \frac{\dot{z}^2}{v_a^2}}}\dot{v}_a \tag{3.31}$$

$$\dot{\chi}(t) = \frac{\ddot{y}\dot{x} - \dot{y}\ddot{x}}{\dot{x}^2 + \dot{y}^2} \tag{3.32}$$

$$\ddot{\chi}(t) = \frac{(-\dot{x}^2\dot{y} - \dot{y}^3)\,\dddot{x} + (\dot{x}^3 + \dot{y}^2\dot{x})\,\dddot{y} - 2(\dot{y}\ddot{y} + \dot{x}\ddot{x})(\ddot{y}\dot{x} - \dot{y}\ddot{x})}{(\dot{x}^2 + \dot{y}^2)^2} \tag{3.33}$$

It is worth to remark that equations (3.32) and (3.33) can be singular in straight vertical flight, since $v_a(t) = \dot{z}$. Recalling that $c(.)$ and $s(.)$ are respectively sin and cos, then, dividing (3.32) by (3.20), the angle $\mu(t)$ is obtained:

$$\mu(t) = \arctan\left(\frac{v_a m c(\gamma)\dot{\chi}}{v_a m\dot{\gamma} + c(\gamma)gm}\right) \tag{3.34}$$

Where it's reported here only its first derivative, while its second derivative which is required in (3.42) is not written for sake of simplicity:

$$\dot{\mu}(t) = \frac{m\left(\dot{\chi}c(\gamma)^2\dot{v}_a gm + c(\gamma)^2 v_a \ddot{\chi} gm + c(\gamma)v_a^2 s(\gamma)\ddot{\chi} - \dot{\chi}v_a^2\dot{\gamma}\right)}{v_a^2 mc(\gamma)^2\dot{\chi}^2 + c(\gamma)^2 g^2 + m^2 + 2c(\gamma)v_a s(\gamma)gm - c(\gamma)^2 v_a^4} \tag{3.35}$$

$Z^a(t)$ is computed from (3.20):

$$Z^a(t) = -\frac{v_a m\dot{\gamma} + c(\gamma)gm}{c(\mu)} \tag{3.36}$$

Its derivative is given by:

$$\dot{Z}^a(t) = -\frac{(m\dot{\gamma}\dot{v}_a + mv_a\ddot{\gamma} - gms(\gamma)\dot{\gamma}}{c(\mu)} - \frac{s(\mu)(mv_a\dot{\gamma} + gmc(\gamma))\dot{\mu}}{c(\mu)^2} \tag{3.37}$$

From (3.16), $X^a(t)$ is computed:

$$X^a(t) = m\dot{v}_a + s(\gamma)g \tag{3.38}$$

The pitch angular rate $q(t)$ is obtained from equation (3.18) as:

$$q(t) = \dot{\alpha} - \frac{c(\gamma)c(\mu)g}{v_a} - \frac{Z^a}{mv_a} \tag{3.39}$$

Then the roll and yaw angular rates, $p(t)$ and $r(t)$, are obtained from (3.21) and (3.17) respectively:

$$p(t) = \frac{1}{(1+\tan^2(\alpha))}\left(\frac{\dot{\mu}}{\cos(\alpha)} - \tan(\alpha)\left(\frac{\cos(\gamma)\sin(\mu)g}{\cos(\alpha)v_a}\right) + \frac{Z^a\sin(\mu)\sin(\gamma)}{m\cos(\gamma)v_a\cos(\alpha)}\right) \tag{3.40}$$

$$r(t) = \frac{v_a\sin(\alpha)p + \cos(\gamma)\sin(\mu)g}{\cos(\alpha)v_a} \tag{3.41}$$

Their derivatives are given by:

$$
\begin{aligned}
\dot{p}(t) = {} & \frac{1}{v_a^2 m\cos^2(\gamma)}\bigl(-v_a\cos(\gamma)\bigl(g\cos^2(\gamma)\sin(\mu)\cos(\alpha)m+\sin(\alpha)mv_a\dot{\mu}\cos(\gamma) \\
& +\sin(\gamma)\sin(\mu)\sin(\alpha)Z_a\bigr)\,\dot{\alpha}-\cos(\gamma)\cos(\mu)v_a\bigl(g\cos^2(\gamma)\sin(\alpha)m-\sin(\gamma)\cos(\alpha)Z_a\bigr)\,\dot{\mu} \\
& +\cos(\gamma)\sin(\mu)\bigl(g\cos^2(\gamma)\sin(\alpha)m-\sin(\gamma)\cos(\alpha)Z_a\bigr)\,\dot{v_a}+\bigl(\sin(\mu)\bigl(g\cos^2(\gamma) \\
& \cdot\sin(\gamma)\sin(\alpha)m+\cos(\alpha)Z_a\bigr)\,\dot{\gamma}+\cos(\gamma)\cos(\alpha)\Bigl(\cos(\gamma)m\ddot{\mu}v_a+\sin(\mu)\sin(\gamma)\dot{Z}_a\Bigr)\Bigr)\,v_a\Bigr) \\
\dot{q}(t) = {} & \frac{(\cos(\gamma)\cos(\mu)gm+Z_a)\dot{v_a}+v_a(g\sin(\gamma)\cos(\mu)\dot{\gamma}m+g\sin(\mu)m\cos(\gamma)\dot{\mu}+mv_a\ddot{\alpha}-\dot{Z}_a)}{mv_a^2} \\
\dot{r}(t) = {} & \frac{1}{\cos^2(\alpha)v_a^2}v_a(g\sin(\mu)\sin(\alpha)\cos(\gamma)+v_ap)\dot{\alpha}-\cos(\alpha)((v_a\sin(\alpha)p+\cos(\gamma)\sin(\mu)g)\dot{v_a} \\
& +v_a(\dot{\gamma}\sin(\gamma)\sin(\mu)g-\cos(\gamma)\dot{\mu}\cos(\mu)g-v_a\sin(\alpha)\dot{p}))
\end{aligned}
\tag{3.42}
$$

Finally, the components $L^b(t), M^b(t), N^b(t)$ of the total torque vector τ^b, see (2.61), are computed from (3.42), (3.68) and (3.69):

$$L^b(t) = I_{xx}\dot{p} - I_{xz}\dot{r} - q((I_{yy} - I_{zz})rI_{xz}p \tag{3.43}$$

$$M^b(t) = \dot{q}I_{yy} + I_{xz}p^2 + r(I_{xx} - I_{zz})p - I_{xz}r^2 \tag{3.44}$$

$$N^b(t) = -I_{xz}\dot{p} + I_{zz}\dot{r} - q((I_{xx} - I_{yy})p - I_{xz}r) \tag{3.45}$$

The total state vector x is now defined as a functions of only the flat-outputs and a finite number of their time derivatives, of which the maximum order is four, included in the derivatives of the states: $\dot{v}_a$, $\ddot{v}_a$, $\dot{\gamma}$, $\ddot{\gamma}$, $\dot{\chi}$, $\ddot{\chi}$, $\dot{\mu}$, $\ddot{\mu}$, $\dot{Z}^a$, $\dot{p}$, $\dot{q}$ and $\dot{r}$, which are needed to express all the states found above.

The control inputs can now be derived from τ^b and F^a, since they depend on $L^b(t)$, $M^b(t)$, $N^b(t)$ and $X^a(t)$, $Y^a(t)$ assumed to be equal to zero and $Z^a(t)$, obtained from the flatness analysis. Due to the complexity of the equations, the computations were performed with the software *Maple* and the resulting control inputs are reported here:

$$
\begin{aligned}
w_l = {} & \Bigg(\frac{C_{m_q} q v_a S c^2 \rho C_{D_{\delta_e}} l_{\text{motor}} + 4\cos(\alpha) C_{\text{prop}} v_a^2 S_{\text{prop}} \rho c C_{m_{\delta_e}} l_{\text{motor}}}{4 C_{\text{prop}} \cos(\alpha) S_{\text{prop}} c k_{\text{motor}}^2 \rho C_{m_{\delta_e}} l_{\text{motor}}} + \dots \\
& \dots + \frac{2\rho v_a^2 S C_{D_\alpha} \alpha c C_{m_{\delta_e}} l_{\text{motor}} + 2 v_a^2 \alpha S c \rho C_{D_{\delta_e}} C_{m_\alpha} l_{\text{motor}} + 4\cos(\alpha) N^b c C_{m_{\delta_e}}}{4 C_{\text{prop}} \cos(\alpha) S_{\text{prop}} c k_{\text{motor}}^2 \rho C_{m_{\delta_e}} l_{\text{motor}}} + \dots \\
& \dots + \frac{4 X_a c C_{m_{\delta_e}} l_{\text{motor}} - 4 M^b C_{D_{\delta_e}} l_{\text{motor}}}{4 C_{\text{prop}} \cos(\alpha) S_{\text{prop}} c k_{\text{motor}}^2 \rho C_{m_{\delta_e}} l_{\text{motor}}} \Bigg)^{\frac{1}{2}} \\
w_r = {} & \Bigg(\frac{C_{m_q} q v_a S c^2 \rho C_{D_{\delta_e}} l_{\text{motor}} + 4\cos(\alpha) C_{\text{prop}} v_a^2 S_{\text{prop}} \rho c C_{m_{\delta_e}} l_{\text{motor}}}{4 C_{\text{prop}} \cos(\alpha) S_{\text{prop}} c k_{\text{motor}}^2 \rho C_{m_{\delta_e}} l_{\text{motor}}} + \dots \\
& \dots + \frac{2\rho v_a^2 S C_{D_\alpha} \alpha c C_{m_{\delta_e}} l_{\text{motor}} + 2 v_a^2 \alpha S c \rho C_{D_{\delta_e}} C_{m_\alpha} l_{\text{motor}} - 4\cos(\alpha) N^b c C_{m_{\delta_e}}}{4 C_{\text{prop}} \cos(\alpha) S_{\text{prop}} c k_{\text{motor}}^2 \rho C_{m_{\delta_e}} l_{\text{motor}}} + \dots \\
& \dots + \frac{4 X_a c C_{m_{\delta_e}} l_{\text{motor}} - 4 M^b C_{D_{\delta_e}} l_{\text{motor}}}{4 C_{\text{prop}} \cos(\alpha) S_{\text{prop}} c k_{\text{motor}}^2 \rho C_{m_{\delta_e}} l_{\text{motor}}} \Bigg)^{\frac{1}{2}} \\
\delta_{\text{el}} = {} & \frac{1}{4 v_a^2 S b \rho^2 C_{l_{\delta_a}} C_{m_{\delta_e}} C_{\text{prop}} S_{\text{prop}} k_{\text{motor}} l_{\text{motor}}} (-2b S_{\text{prop}} \rho^2 (C_{m_\alpha} C_{l_{\delta_a}} \alpha + C_{m_{\delta_e}}(C_{l_0})) \dots \\
& \cdot l_{\text{motor}} k_{\text{motor}} C_{\text{prop}} S v_a^2 - S C_{\text{prop}} S_{\text{prop}} b k_{\text{motor}} \rho^2 l_{\text{motor}} (C_{m_q} q C_{l_{\delta_a}} c + b C_{m_{\delta_e}} C_{l_p} p + \dots \\
& \dots + r b C_{m_{\delta_e}} C_{l_r}) v_a + 4 L^b C_{m_{\delta_e}} C_{\text{prop}} S_{\text{prop}} k_{\text{motor}} \rho l_{\text{motor}} + 4 M^b C_{l_{\delta_a}} C_{\text{prop}} S_{\text{prop}} \dots \\
& \cdot k_{\text{motor}} \rho l_{\text{motor}} - 8 k_\omega C_{m_{\delta_e}} k_{T_p} N^b) \\
\delta_{\text{er}} = {} & \frac{1}{4 v_a^2 S b \rho^2 C_{l_{\delta_a}} C_{m_{\delta_e}} C_{\text{prop}} S_{\text{prop}} k_{\text{motor}} l_{\text{motor}}} \big(2b S_{\text{prop}} \rho^2 (-C_{m_\alpha} C_{l_{\delta_a}} \alpha + C_{m_{\delta_e}}(C_{l_0})) \dots \\
& \cdot l_{\text{motor}} k_{\text{motor}} C_{\text{prop}} S v_a^2 + S C_{\text{prop}} S_{\text{prop}} b k_{\text{motor}} \rho^2 l_{\text{motor}} (b C_{m_{\delta_e}} C_{l_p} p + r b C_{m_{\delta_e}} C_{l_r} - \dots \\
& \dots - C_{m_q} q C_{l_{\delta_a}} c) v_a - 4 L^b C_{m_{\delta_e}} C_{\text{prop}} S_{\text{prop}} k_{\text{motor}} \rho l_{\text{motor}} + 4 M^b C_{l_{\delta_a}} C_{\text{prop}} S_{\text{prop}} \dots \\
& \cdot k_{\text{motor}} \rho l_{\text{motor}} + 8 k_\omega C_{m_{\delta_e}} k_{T_p} N^b \big)
\end{aligned}
\tag{3.46}
$$

3.4 Differential flatness analysis for vertical take-off, vertical landing and hover flight

3.4.1 Mathematical model

The nonlinear dynamic model of the aircraft on which the differential flatness analysis is performed for zero/small aerodynamic velocities is given by:

$$\dot{\xi} = V_o^o \tag{3.47}$$

$$m\frac{dV_o^o}{dt} = R_b^o F^b + G^o \tag{3.48}$$

$$\dot{R}_b^o = R_b^o [\Omega_{bo}^b]_\times \tag{3.49}$$

$$\frac{d(I\Omega_{bo}^b)}{dt} + \Omega_{bo}^b \wedge I\Omega_{bo}^b = \tau^b \tag{3.50}$$

Where the same notation as (2.63) is maintained.

$V_o^o = [v_{x_o}, v_{y_o}, v_{z_o}]^T$ is the vector of the components of the velocity along the x, y, z directions in the frame $\mathcal{I}$. In this model the orientation of the aircraft from frame $\mathcal{B}$ to frame $\mathcal{O}$, is no longer expressed in terms of the MRP, as explained in Section 3.2, but in terms of vertical Euler angles.

The state vector is $x = [x, y, z, v_{x_o}, v_{y_o}, v_{z_o}, \phi, \theta, \psi, p, q, r]^T$ and the control input vector is $u = [\delta_{el}, \delta_{er}, \omega_l, \omega_r]^T$, as defined in (2.1).

Some simplifying assumptions are considered for the model. Since the aircraft is flying with a null or small aerodynamic speed, then the aerodyanamic forces, F_{lift}, F_y and F_{drag}, are neglected. It is also neglected the lift force due to the propeller wing interaction, L_w. The propulsion force F_p^b and the drag force due to the propeller wing interaction, D_w, are considered:

$$F^b = \begin{bmatrix} F_p^b - D_w \\ 0 \\ 0 \end{bmatrix} \tag{3.51}$$

The pitch moment due to the propeller wing interaction is also taken into account. Finally, the aerodynamic coefficients are considered to be linear in the angle of attack $\alpha(t)$.

Vertical Euler angles

This set of Euler angles is constructed to provide unambiguous attitudes in the vertical flight orientation. They are shown in Figure 3.2 and are defined as follows:

- Starting from a vertical attitude in which the vehicle x^b axis points upwards and with the z^b axis pointing North.

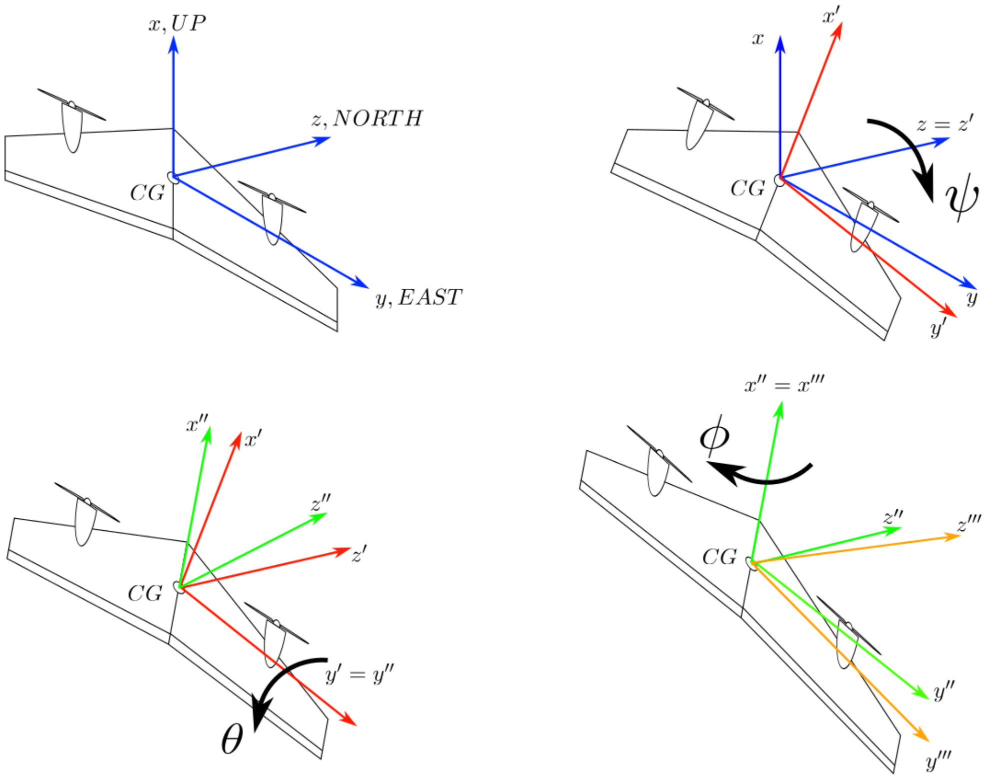


Figure 3.2: Definition of vertical Euler angles.

- Apply a yaw rotation ψ about the z^b axis.
- Apply a pitch rotation θ about the new position of the y^b axis.
- Finally perform a roll rotation ϕ about the axis x^b moved twice.

It should be noted that in this system the order of rotations is yaw, pitch, roll ψ, θ, ϕ as in the standard Euler angle rotations, differently from [7], where this concept is derived from, and the proposed order of rotations is roll, pitch and yaw ϕ, θ, ψ. This is due to the fact that the initial rotation of 90 deg around the y^b axis, was introduced to the already defined matrix R_b^o with the standard Euler angles notation. Also, this system gives singularities when the vertical pitch angle θ is ±90 deg, namely when the aircraft is horizontal.

The orientation of the body from frame $\mathcal{B}$ to frame $\mathcal{O}$, is now defined as:

$$R_b^o = R_y\left(\frac{\pi}{2}\right) R_z(\psi) R_y(\theta) R_x(\phi) = \tag{3.52}$$

$$= \begin{bmatrix} -s(\theta) & s(\psi)c(\theta) & c(\theta)c(\phi) \\ s(\psi)c(\theta) & -s(\psi)s(\theta)s(\phi) + c(\psi)c(\phi) & -c(\psi)s(\phi) + s(\psi)s(\theta)c(\phi) \\ -c(\psi)c(\theta) & s(\psi)c(\phi) - c(\psi)s(\theta)s(\phi) & -s(\psi)s(\phi) - c(\psi)s(\theta)c(\phi) \end{bmatrix} \tag{3.53}$$

3.4.2 Choice of the flat-outputs

The same considerations explained in Section 3.3.2, are taken into account. In particular the chosen flat-outputs are:

$$y(t) = (x(t), y(t), z(t), \phi(t)) \tag{3.54}$$

where $x(t), y(t), z(t)$ is the position of the CG of the aircraft relative to frame $\mathcal{I}$ and $\phi(t)$ is the roll angle.

3.4.3 Mathematical proof of the differential flatness property

The first step consists in deriving from equation (3.47) the following relations:

$$\dot{x} = v_{x_o} \tag{3.55}$$

$$\dot{y} = v_{y_o} \tag{3.56}$$

$$\dot{z} = v_{z_o} \tag{3.57}$$

Then, the aerodynamic velocity $v_a(t)$ is given by:

$$v_a(t) = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} \tag{3.58}$$

From (3.48) defined as:

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} -s(\theta)(F_p^b - D_w) \\ s(\psi)c(\theta)(F_p^b - D_w) \\ -c(\psi)c(\theta)(F_p^b - D_w) \end{bmatrix} \tag{3.59}$$

The total force $F^b(1) = F_p^b - D_w$ is obtained:

$$F_p^b - D_w = m\sqrt{\ddot{x}^2 + \ddot{y}^2 + (\ddot{z} - g)^2} \tag{3.60}$$

From (3.48) they are also obtained the angles $\psi(t)$ and $\theta(t)$:

$$\psi(t) = \arctan\left(\frac{\ddot{y}}{g - \ddot{z}}\right) \tag{3.61}$$

$$\theta(t) = -\arcsin\left(\frac{m\ddot{x}}{F^b}\right) \tag{3.62}$$

And their derivatives are given by:

$$\begin{aligned}
\dot{\psi}(t) &= \frac{\dddot{y}(g-\ddot{z})}{\ddot{y}^2+(g-\ddot{z})^2} + \frac{\ddot{y}\dddot{z}}{\ddot{y}^2+(g-\ddot{z})^2} \\
\ddot{\psi}(t) &= \frac{\ddddot{y}g^3 + \ddddot{z}g^2\ddot{y} + 2g^2\dddot{y}\dddot{z} - 3\ddddot{y}g^2\ddot{z} + \ddddot{y}g\ddot{y}^2 - 2g\dot{y}\ddot{y}^2 - 2\ddddot{z}g\ddot{y}\ddot{z} + 2g\ddot{y}\dddot{z}^2}{(g^2-2g\ddot{z}+\ddot{y}^2+\ddot{z}^2)^2} \\
&\quad - \frac{4g\dddot{y}\dddot{z}\ddot{z} + 3\ddddot{y}g\ddot{z}^2 + \ddddot{z}\ddot{y}^3 - 2\ddot{y}^2\dddot{y}\dddot{z} - \ddddot{y}\ddot{y}^2\ddot{z} + 2\ddot{y}\dddot{y}^2\ddot{z} + \ddddot{z}\ddot{y}\ddot{z}^2 - 2\ddot{y}\ddot{z}\dddot{z}^2}{(g^2-2g\ddot{z}+\ddot{y}^2+\ddot{z}^2)^2} \\
&\quad + \frac{2\dddot{y}\ddot{z}^2\dddot{z} - \ddddot{y}\ddot{z}^3}{(g^2-2g\ddot{z}+\ddot{y}^2+\ddot{z}^2)^2} \\
\dot{\theta}(t) &= \frac{\ddot{x}^2\dddot{x} - \dddot{x}(\ddot{x}^2+\ddot{y}^2+(g-\ddot{z})^2) - g\ddot{x}\dddot{z} + \ddot{x}\ddot{y}\dddot{y} + \ddot{x}\ddot{z}\dddot{z}}{\left(1-\frac{\ddot{x}^2}{\ddot{x}^2+\ddot{y}^2+(g-\ddot{z})^2}\right)^{\frac{1}{2}}(\ddot{x}^2+\ddot{y}^2+(g-\ddot{z})^2)^{\frac{3}{2}}}
\end{aligned} \tag{3.63}$$

The second derivative of $\theta(t)$ is not reported here for sake of simplicity, but it is needed in (3.68) and (3.69). From (3.49) the roll, pitch and yaw angular rates, $p(t)$, $q(t)$ and $r(t)$ are derived:

$$p(t) = -\dot{\psi}s(\theta) + \dot{\phi} \tag{3.64}$$

$$q(t) = s(\phi)\dot{\psi}c(\theta) + \dot{\theta}c(\phi) \tag{3.65}$$

$$r(t) = \dot{\psi}c(\phi)c(\theta) - \dot{\theta}s(\phi) \tag{3.66}$$

With their derivatives defined as:

$$\dot{p}(t) = -\ddot{\psi}s(\theta) - \dot{\psi}\dot{\theta}c(\theta) + \ddot{\phi} \tag{3.67}$$

$$\dot{q}(t) = \dot{\phi}c(\phi)\dot{\psi}c(\theta) + s(\phi)\ddot{\psi}c(\theta) - s(\phi)\dot{\psi}\dot{\theta}s(\theta) + \ddot{\theta}c(\phi) - \dot{\theta}\dot{\phi}s(\phi) \tag{3.68}$$

$$\dot{r}(t) = \ddot{\psi}c(\phi)c(\theta) - \dot{\psi}\dot{\phi}s(\phi)c(\theta) - \dot{\psi}c(\phi)\dot{\theta}s(\theta) - \ddot{\theta}s(\phi) - \dot{\theta}\dot{\phi}c(\phi) \tag{3.69}$$

From (3.50) the components of the total torque vector τ^b are computed:

$$L^b(t) = I_{xx}\dot{p} - I_{xz}\dot{r} - q((I_{yy}-I_{zz})rI_{xz}p) \tag{3.70}$$

$$M^b(t) = \dot{q}I_{yy} + I_{xz}p^2 + r(I_{xx}-I_{zz})p - I_{xz}r^2 \tag{3.71}$$

$$N^b(t) = -I_{xz}\dot{p} + I_{zz}\dot{r} - q((I_{xx}-I_{yy})p - I_{xz}r) \tag{3.72}$$

The total state vector x is now defined as a functions of only the flat-outputs and a finite number of their time derivatives, of which the maximum order is four, included in the derivatives of the states: $\dot{\psi}$, $\ddot{\psi}$, $\dot{\theta}$, $\ddot{\theta}$, $\dot{p}$, $\dot{q}$ and $\dot{r}$, which are needed to express all the states found above.

The control inputs can be now computed from the total torque vector τ^b and the total force vector F^b, of which its components $L^b(t)$, $M^b(t)$, $N^b(t)$ and $(F_p^b - D_w)$ are derived from the flatness analysis. Also in this case, due to the complexity of the equations, the computations were performed with the software *Maple* and the resulting control inputs are reported here:

$$\delta_{el} = \frac{1}{2v_i^2}\left(\frac{2M^b}{\rho S_w c C_{m_{\delta_e}}} + \frac{8(L^b - k_{T_p} k_\omega^2 \frac{2N^b}{\rho S_{\text{prop}} C_{\text{prop}} k_{\text{motor}}^2 l_{\text{motor}}})}{b \rho S_w C_{L_{\delta_e}}}\right) \tag{3.73}$$

$$\delta_{er} = \frac{1}{2v_i^2}\left(\frac{2M^b}{\rho S_w c C_{m_{\delta_e}}} - \frac{8(L^b - k_{T_p} k_\omega^2 \frac{2N^b}{\rho S_{\text{prop}} C_{\text{prop}} k_{\text{motor}}^2 l_{\text{motor}}})}{b \rho S_w C_{L_{\delta_e}}}\right) \tag{3.74}$$

$$w_l = \frac{1}{\sqrt{2}}\sqrt{\left(\frac{2F_p}{\rho S_{\text{prop}} C_{\text{prop}}} + 2v_a^2\right)\frac{1}{k_{\text{motor}}^2} + \frac{2N^b}{\rho S_{\text{prop}} C_{\text{prop}} k_{\text{motor}}^2 l_{\text{motor}}}} \tag{3.75}$$

$$w_r = \frac{1}{\sqrt{2}}\sqrt{\left(\frac{2F_p}{\rho S_{\text{prop}} C_{\text{prop}}} + 2v_a^2\right)\frac{1}{k_{\text{motor}}^2} - \frac{2N^b}{\rho S_{\text{prop}} C_{\text{prop}} k_{\text{motor}}^2 l_{\text{motor}}}} \tag{3.76}$$

Chapter 4

Open-loop control implementation and results

The aim of this chapter is to illustrate the implementation of the open-loop control based on the differential flatness approach, see Figure 4.1.

Firstly, in Section 4.1, it is introduced the concept of trim, and an algorithm in order to find the control inputs, which guarantees the aircraft to be in equilibrium, is implemented.

Afterwards, in Section 4.2, based on the results obtained from the trim algorithm, the trajectories are generated both for vertical take-off/landing and hover flight, and for horizontal flight. In particular they are illustrated the role of the differential flatness property and the advantages it provides.

Finally, the results of the open-loop control are shown.

4.1 Trim

Based on the theory illustrated in [4] and in [8], given a nonlinear system described by differential equations

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

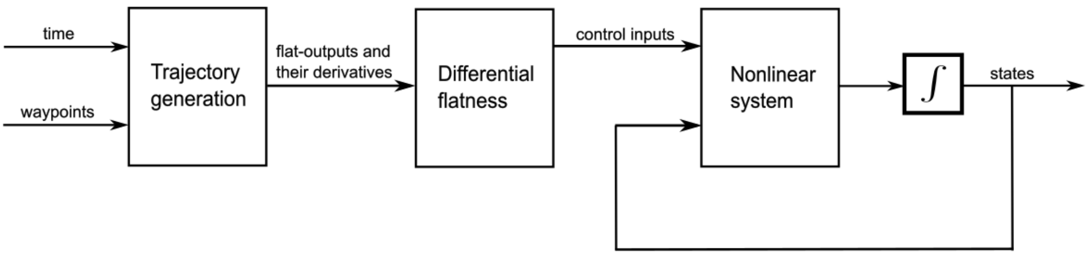


Figure 4.1: Schematic representation of the open-loop control.

where $f : \mathbb{R}^n \times \mathbb{R}^m \longrightarrow \mathbb{R}^n$ is given by the equations (2.63), x is called the state of the system, and u is the input. The system is said to be in equilibrium at the state x_{trim} and input u_{trim} if

$$f(x_{\text{trim}}, u_{\text{trim}}) = 0 \tag{4.2}$$

In the aerodynamic literature, an aircraft in equilibrium is said to be in trim, namely the forces and moments acting on it are balanced, resulting in steady and level flight. Since trim conditions may include states that are not constant, it is more appropriate to define the conditions for trim in a general way, as:

$$\dot{x}_{\text{trim}} = f(x_{\text{trim}}, u_{\text{trim}}) \tag{4.3}$$

The trim points vary depending on the flight mode and are treated differently for when the aircraft is in hover flight and when it is flying horizontally. For the vertical take-off and landing it is considered that the aircraft is flying from one equilibrium point to an other one, found for the hover flight.

The equilibrium point in hover flight given by x_{trim} and u_{trim} is unique, it has to satisfy (4.2), and it depends only on the propellers angular rates expressed in PWM, that need to produce a total thrust force capable of equalizing the weight of the body. Hence, the aircraft remains at a constant altitude z^*, with a constant orientation given in terms of MRP, $\sigma_1^* = 0$, $\sigma_2^* = 0.4142$ and $\sigma_3^* = 0$, which corresponds to a rotation of 90 deg around the y^b axis, namely the aircraft's nose is pointing up. The remaining states are all equal to zero.

$$x_{\text{trim}} = \begin{bmatrix} x^* = 0 \\ y^* = 0 \\ z^* \\ v_x^* = 0 \\ v_y^* = 0 \\ v_z^* = 0 \\ \sigma_1^* \\ \sigma_2^* \\ \sigma_3^* \\ p^* = 0 \\ q^* = 0 \\ r^* = 0 \end{bmatrix} \tag{4.4}$$

Recalling that the aerodynamic velocity during hover, is equal to zero, then the values of the propeller angular rates which guarantee this condition are derived from (2.45) which is equalized to the weight of the aircraft mg:

$$\omega_i = \frac{1}{k_{\text{motor}}}\sqrt{\frac{mg}{\rho S_{\text{prop}} C_{\text{prop}}}} \quad i = l, r \tag{4.5}$$

Hence, the control input vector is given by:

$$u_{\text{trim}} = \begin{bmatrix} \delta_{el}^* \\ \delta_{er}^* \\ \omega_l^* \\ \omega_r^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0.4986 \\ 0.4986 \end{bmatrix} \tag{4.6}$$

These results are shown in Figure 4.2.

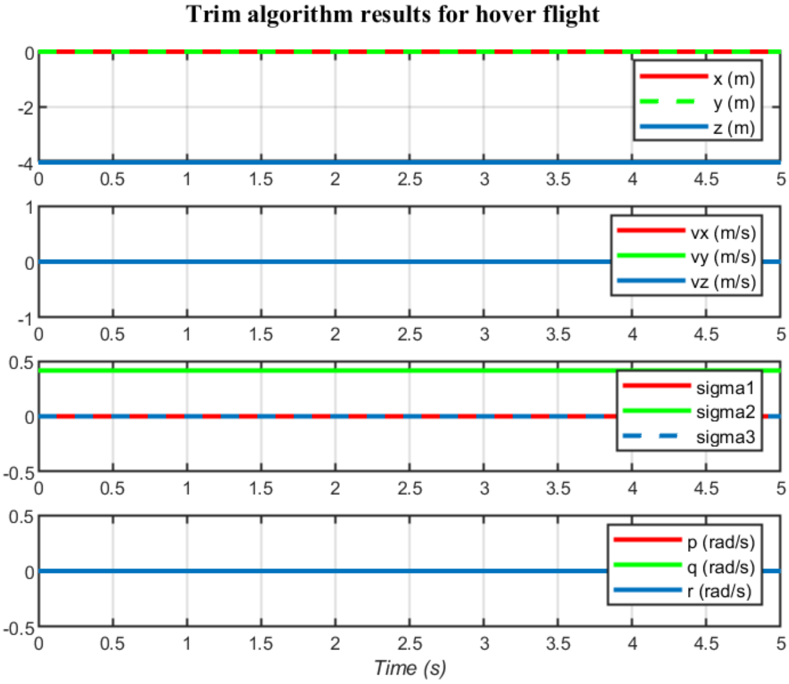


Figure 4.2: Results of the trim algorithm for the hover flight, with a constant altitude of 4 m.

For sake of completeness, they are reported in Figure 4.3 also the resulting forces and moments, which as expected are all equal to zero.

During horizontal flight, instead, the equilibrium is not unique, since the aircraft can fly at different velocities and directions. The objective is to find the states x_{trim} and control inputs u_{trim}, that satisfy (4.3). The aircraft is flying at a given altitude, with an

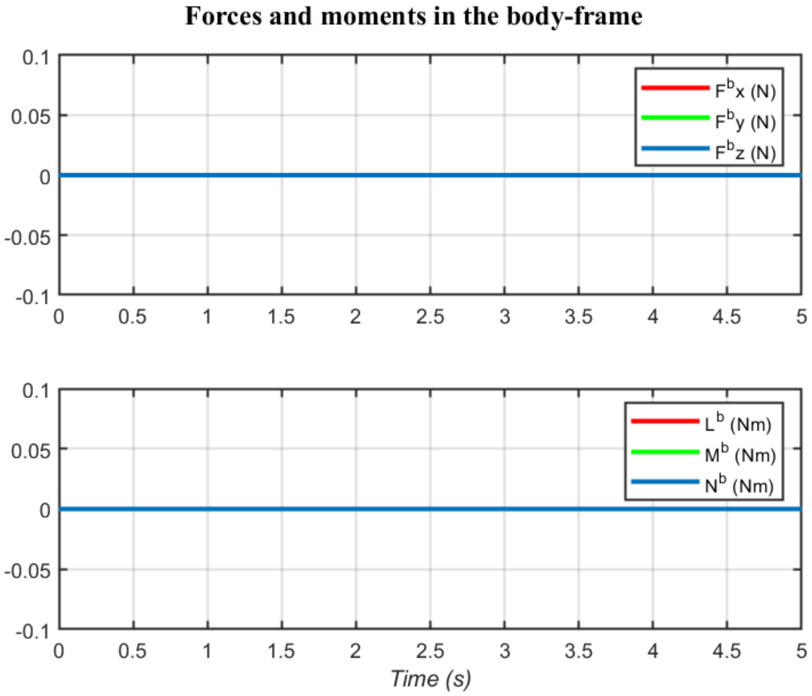


Figure 4.3: Resulting forces and moments in the body-frame of the trim algorithm for the hover flight, with a constant altitude of 4 m.

assigned aerodynamic velocity. Since, $x(t) = v_a(t)t$, the state vector x_{trim} is given by:

$$x_{\text{trim}} = \begin{bmatrix} x^* = v_a(t)t \\ y^* = 0 \\ z^* \\ v_x^* \\ v_y^* \\ v_z^* \\ \sigma_1^* \\ \sigma_2^* \\ \sigma_3^* \\ p^* = 0 \\ q^* = 0 \\ r^* = 0 \end{bmatrix} \tag{4.7}$$

where v_x^*, v_y^* and v_z^* are the components of the aerodynamic velocity in the body frame, see (2.14), and the MRP components are given by the specified orientation at which it is desired the aircraft to fly. The value of $\alpha_{\text{trim}}(t)$ is retrieved from (2.16):

$$\alpha_{\text{trim}}(t) = \arctan\left(\frac{v_z^*}{v_x^*}\right) \tag{4.8}$$

Consequently, the vector of the time derivatives of the states is defined as:

$$\dot{x}_{\text{trim}} = \begin{bmatrix} \dot{x}^* = v_a \\ \dot{y}^* = 0 \\ \dot{z}^* = 0 \\ \dot{v}_x^* = 0 \\ \dot{v}_y^* = 0 \\ \dot{v}_z^* = 0 \\ \dot{\sigma}_1^* = 0 \\ \dot{\sigma}_2^* = 0 \\ \dot{\sigma}_3^* = 0 \\ \dot{p}^* = 0 \\ \dot{q}^* = 0 \\ \dot{r}^* = 0 \end{bmatrix} \tag{4.9}$$

The optimization of the cost function $||f(x_{\text{trim}}, u_{\text{trim}}) - \dot{x}_{\text{trim}}||$, in the five unknowns α_{trim}, δ_{el}, δ_{er}, ω_l and ω_r, is performed by the function `trim` [1] in Matlab, which uses a sequential quadratic programming algorithm. For instance, if the aircraft is required to fly with an aerodynamic velocity $v_a(t) = 20\ \mathrm{m\,s^{-1}}$ at an altitude of 4 m, the corresponding state and input vectors are given by:

$$x_{\text{trim}} = \begin{bmatrix} v_a(t)t \\ 0 \\ -4 \\ 19.9935 \\ 0 \\ 0.5080 \\ 0 \\ 0.0064 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad u_{\text{trim}} = \begin{bmatrix} -0.0142 \\ -0.0142 \\ 0.6404 \\ 0.6404 \end{bmatrix} \tag{4.10}$$

These results are shown in Figure 4.4.

[1] https://it.mathworks.com/help/simulink/slref/trim.html

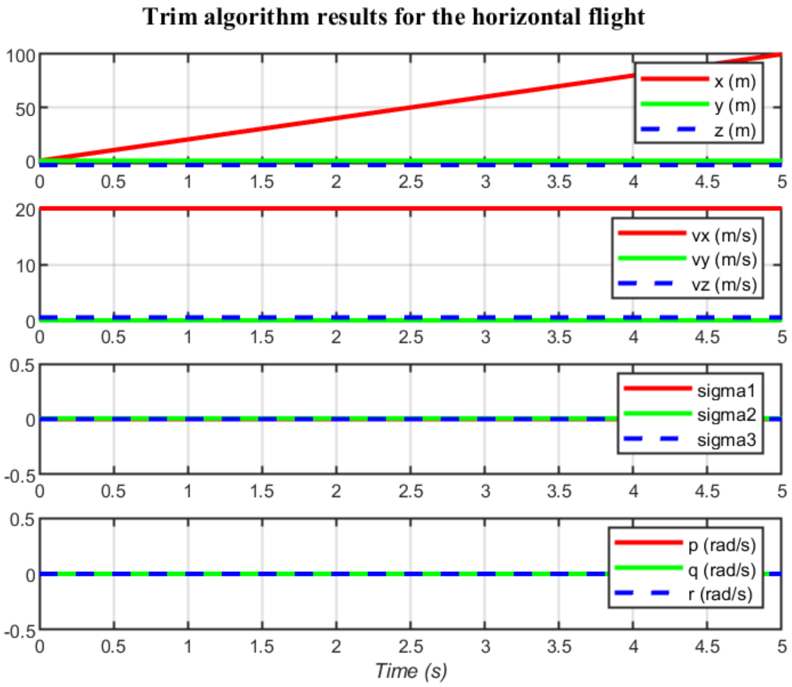


Figure 4.4: Results of the trim algorithm for the horizontal flight, with an aerodyanamic velocity of $v_a(t) = 20\ \mathrm{m\,s^{-1}}$, an initial angle of attack of $\pi/100$ and an altitude of $4\,\mathrm{m}$.

4.2 Trajectory generation

In the context of trajectory planning, a system which is proved to be differentially flat, has the advantage that the trajectory can be specified directly in the flat-output space, and the control inputs needed to execute that trajectory can be obtained as an algebraic function of the trajectory and its derivatives.

As explained in Section 3.3.2, the flat-output variables do not need to correspond to convenient physical quantities, but often they do. For the convertible aircraft, the four flat-outputs, both for zero/small and high aerodynamic velocities, are $x(t), y(t), z(t)$ which is the position of the CG of the aircraft relative to frame $\mathcal{I}$ along with the angle of attack $\alpha(t)$ for the horizontal flight, and the roll angle $\phi(t)$ for the vertical take-off/landing and hover flight. Therefore, it becomes easy to specify start and goal locations in the flat-output space.

Note that in order to recover the states and the inputs to the vehicle along the trajectory, they are needed not only the values of the flat-outputs along the trajectory, but also their derivatives.

It is convenient to define the trajectories using a differentiable basis function, as, intuitively, the states and inputs of the vehicle at some point along a trajectory must depend not only on the position, but also on the velocity, acceleration and higher order derivatives of the trajectory at that point. Among the various choices that would work in this setting, it is decided to adopt the idea proposed in [5] and [24], where the use of polynomials, due to their analytical and computational tractability, is chosen.

Consider the evolution of a flat-output variable, over a time interval $t \in [t_1, t_2]$, for instance a position coordinate, prescribed by a polynomial $P(t)$ between two points in the space of flat-outputs. The objective is to select the coefficients of the polynomial, in order to match the endpoints at t_1 and t_2 (and the derivatives at those endpoints), with those that have been specified. Hence, it is desired to optimize some cost function of the derivatives of the polynomial, by using the minimum-snap (4^{th} derivative) cost function, which was proposed in the trajectory generation for quadrotors. This method allows to discourage abrupt changes in the motor command to the quadrotor, resulting in graceful trajectories, and it is explained briefly in the following.

Denote p_n the coefficients of a polynomial P of degree N such that:

$$P(t) = p_n t^N + p_{n-1} t^{n-1} + ... + p_0 = \sum_{n=0}^{N} p_n(t) t^n \tag{4.11}$$

The coefficients of P has to be optimized, in order to minimize cost functions in this form:

$$J = \int_{t=t_1}^{t=t_2} c_0 P(t)^2 + c_1 P'(t)^2 + c_2 P''(t)^2 + ... + c_N P^{(N)}(t)^2 dt \tag{4.12}$$

which can be written in quadratic form as:

$$J = p^T Q p \tag{4.13}$$

where $p \in (R)^{N+1}$ is the vector of the polynomial coefficients and Q is the cost matrix, corresponding to the desired penalty on each of the polynomial derivatives. In addition, it should be also constrained the solution to enforce specific conditions on the value and derivatives of the initial and final enpoints of the polynomial. These constraints are defined as:

$$A_{t_1} p = b_{t_1} \tag{4.14}$$

$$A_{t_2} p = b_{t_2} \tag{4.15}$$

with A_{t_1} and A_{t_2} the matrices which map the coefficients p to the derivatives of the polynomial at the beginning and end, respectively. The values of the derivatives to constrain are indicated by the vectors b_{t_1} and b_{t_2}. Every row of (4.14) and (4.15) corresponds to a particular derivative to be constrained, including the 0^{th} derivative, that allows to constrain the value of the polynomial, namely the position. Hence, the standard quadratic programming problem can be obtained by concatenating the desired constraints on the initial and final derivatives of the polynomial:

$$\begin{aligned} \min_{p} \quad & p^T Q p \\ \text{s.t.} \quad & Ap - b = 0 \end{aligned} \tag{4.16}$$

The trajectory followed by the convertible aircraft consists in four independent polynomials that describe the evolution of the four flat output variables through time, and it starts and end at rest as the higher order derivatives begin and end at zero.

The function used in Matlab, which performs what has been explained up to now, is the `minsnappolytraj` [2], which generates a minimum snap polynomial trajectory that achieves a given set of input waypoints with their corresponding time points. In particular, the function returns positions, velocities, accelerations, jerks, and snaps at the given number of samples. It also returns the piecewise polynomial form of the polynomial trajectory with respect to time, as well as the time points, and the sample times.

4.2.1 Trajectory generation for horizontal flight

For generating the trajectory in the horizontal flight it is considered that the aircraft is flying at a given arbitrary altitude with respect to frame $\mathcal{I}$ of 4 m, namely $z(t) = -4$ m, with a $y(t)$ coordinate equal to zero. Instead the $x(t)$ coordinate in frame $\mathcal{I}$ is obtained from:

$$x(t) = v_a t \tag{4.17}$$

where the value of the aerodynamic velocity can be selected arbitrarily. In this way the vector of waypoints for the trajectory generation for high aerodynamic velocities is defined in terms of the aerodynamic velocity and the remaining flat-output $\alpha(t)$ as:

$$\text{wPts} = \begin{bmatrix} v_a(t) \\ \alpha(t) \end{bmatrix} \tag{4.18}$$

Since the aircraft has to fly from one equilibrium point to the next one, explained in Section 4.1, the values for the angle of attack and for the final aerodynamic velocity are defined by the trim, respectively as α_{trim_1}, α_{trim_2} and v_{a_1}, v_{a_2}. Considering that the aircraft has to fly horizontally, for instance in a time period $t \in [t_1, t_2] = [10, 40]$s, then the vectors of waypoints at time t_1 and t_2 are given respectively by:

$$\text{wPts}_1 = \begin{bmatrix} v_{a_1} \\ \alpha_{\text{trim}_1} \end{bmatrix} \quad \text{wPts}_2 = \begin{bmatrix} v_{a_2} \\ \alpha_{\text{trim}_2} \end{bmatrix} \tag{4.19}$$

[2] https://it.mathworks.com/help/robotics/ref/minsnappolytraj.html

where, the aerodynamic speeds are $v_{a_1} = 20\ \mathrm{m\,s^{-1}}$ at the first trim point, and $v_{a_2} = 25\ \mathrm{m\,s^{-1}}$ at the second, and the angles of attack are obtained as $\alpha_{\mathrm{trim}_1} = 0.0254$ rad, $\alpha_{\mathrm{trim}_2} = 0.0163$ rad, as shown in Figure 4.5.

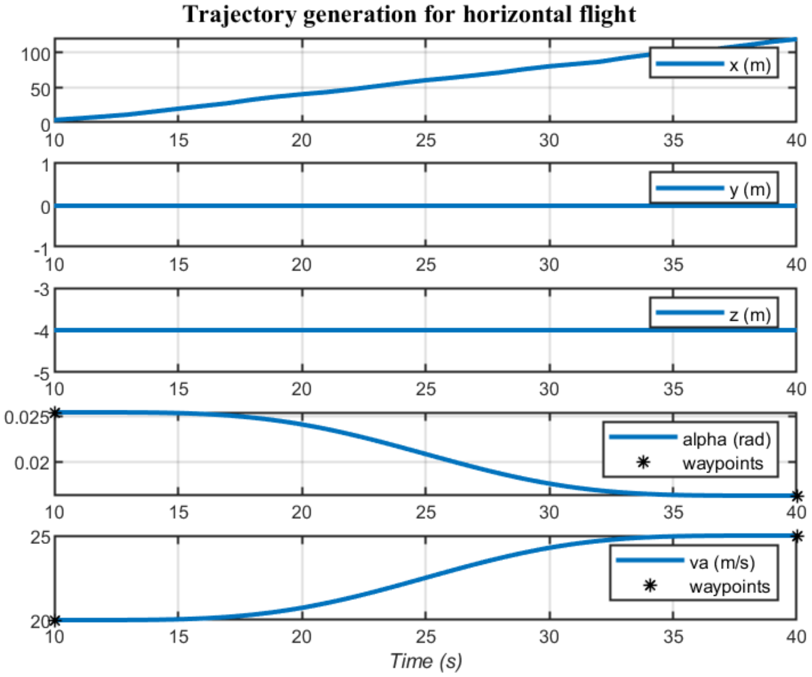


Figure 4.5: Examples of a polynomial trajectory for an initial aerodynamic velocity of $20\ \mathrm{m\,s^{-1}}$, showing the evolution of the flat-outputs and $v_a(t)$ during time.

4.2.2 Trajectory generation for vertical take-off/landing and hover flight

The vector of waypoints for the trajectory generation for zero/small aerodynamic velocities is defined in terms of the flat-outputs as:

$$\mathrm{wPts} = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \\ \phi(t) \end{bmatrix} \tag{4.20}$$

An example of a trajectory for the hover flight consists in requiring the aircraft to perform a rotation around the roll angle $\phi(t)$ from $0\,\mathrm{rad\,s^{-1}}$ at time t_1, to $\pi/4\,\mathrm{rad\,s^{-1}}$ at time t_2, in an arbitrary time interval, $t \in [t_1, t_2] = [10, 40]$s, at a constant altitude of 4 m, namely:

$$\text{wPts}_1 = \begin{bmatrix} 0 \\ 0 \\ -4 \\ 0 \end{bmatrix} \quad \text{wPts}_2 = \begin{bmatrix} 0 \\ 0 \\ -4 \\ \frac{\pi}{4} \end{bmatrix} \tag{4.21}$$

Another movement the aircraft is required to perform during the vertical flight modes is to move from an equilibrium point to another one, see Section 4.1, namely from a certain altitude position, for instance an height of 4m, up to another altitude position, as 8m, defined in frame $\mathcal{I}$ as -4m and -8m, namely:

$$\text{wPts}_1 = \begin{bmatrix} 0 \\ 0 \\ -4 \\ 0 \end{bmatrix} \quad \text{wPts}_2 = \begin{bmatrix} 0 \\ 0 \\ -8 \\ 0 \end{bmatrix} \tag{4.22}$$

These trajectories are shown in Figure 4.6 and 4.7.

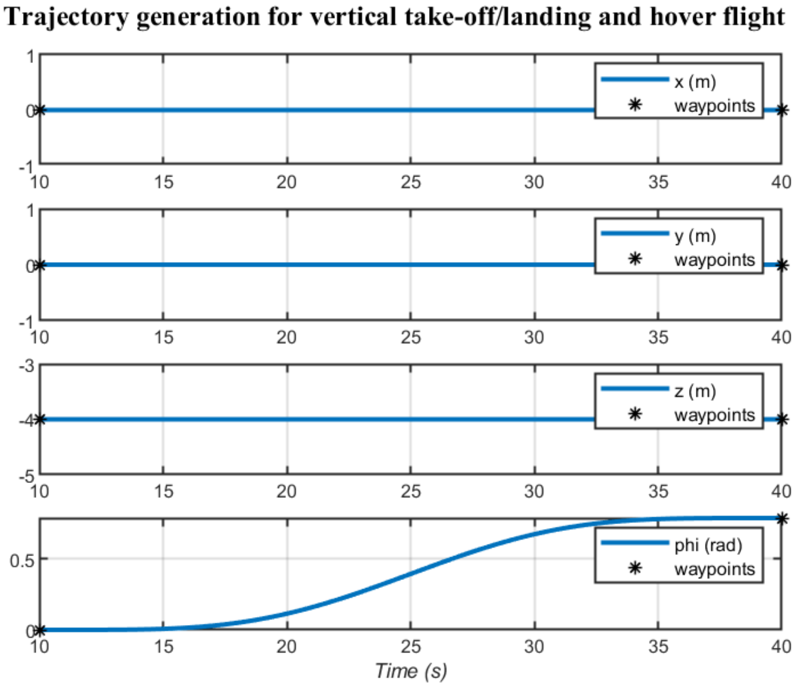


Figure 4.6: Example of a polynomial trajectory for hover flight showing the evolution of the flat-outputs during time, where the aircraft is required to remain at a constant altitude of 4 m and to perform a rotation of $\phi(t) = \frac{\pi}{4}$ in the time interval $t \in [10, 40]$s.

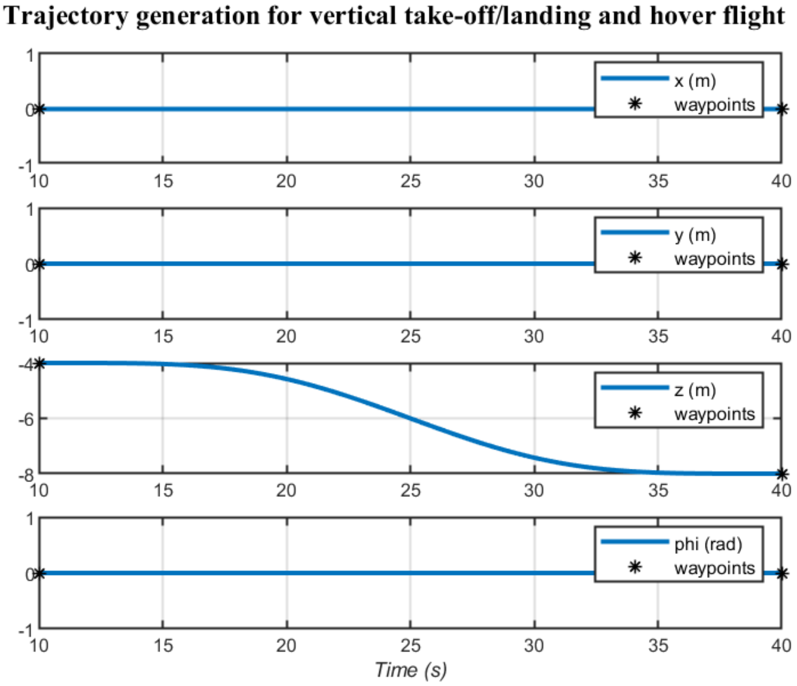


Figure 4.7: Example of a polynomial trajectory for vertical flight showing the evolution of the flat-outputs during time, when the aircraft is required to fly from an altitude of 4m up to an altitude of 8m.

4.3 Results of the open-loop control for horizontal flight

The obtained results of the open-loop control, for an initial aerodynamic velocity of 20 $\mathrm{m\,s^{-1}}$, an altitude of 4m and an initial angle of attack of 0.0254 rad are reported in this section and compared with the generated ones, in Figures 4.8, 4.9, 4.10, 4.11 and 4.12.

It is worth to notice that the control inputs obtained from the differential flatness analysis perfectly match the ones obtained by the trim algorithm, ensuring equilibrium during flight:

$$u_{\text{trim}} = \begin{bmatrix} -0.014 \\ -0.014 \\ 0.6404 \\ 0.6404 \end{bmatrix} \qquad u = \begin{bmatrix} -0.014 \\ -0.014 \\ 0.6404 \\ 0.6404 \end{bmatrix} \tag{4.23}$$

and the total forces and moments acting on the aircraft are zero, as expected.

An interesting remark is that the generated angle $\theta(t)$ is close to the angle of attack, as desired during horizontal flight, for the no wind condition and the absence of other external factors, by contrast the simulated result for $\theta(t)$ presents some imprecision, see Figure 4.11. Also, the altitude $z(t)$ is not staying constant at the required altitude due to the parametric uncertainty in the lift aerodynamic coefficient, see Figure 4.8.

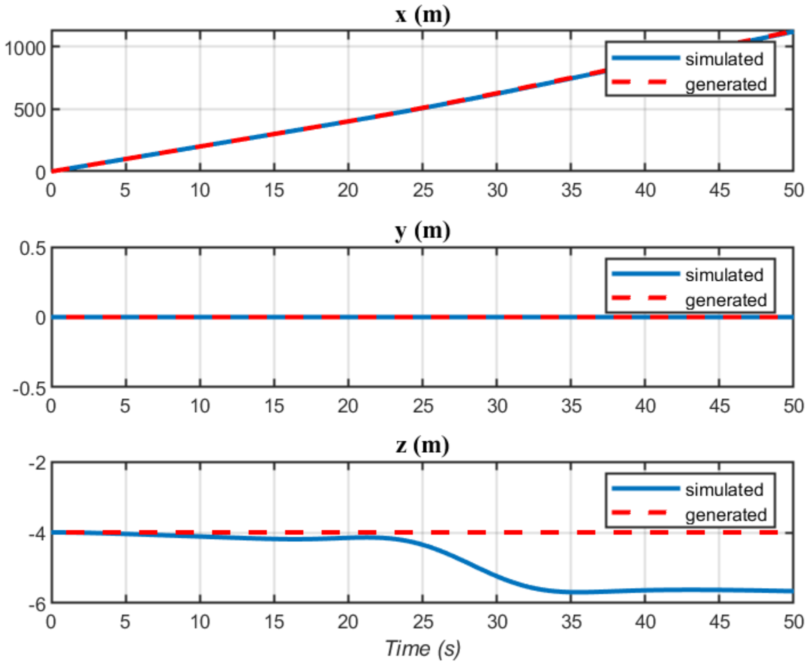


Figure 4.8: Result of the open-loop control for the position of the aircraft in horizontal flight with respect to frame $\mathcal{I}$.

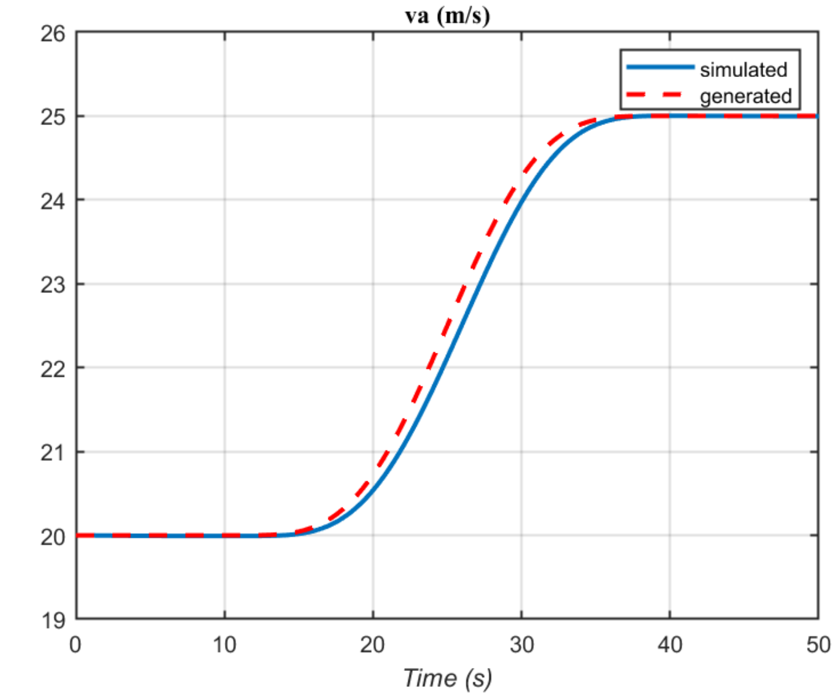


Figure 4.9: Result of the open-loop control for the aerodynamic velocity of the aircraft in horizontal flight, the aerodynamic velocity has been obtained from the norm of the components of V_a^b, namely v_x, v_y and v_z.

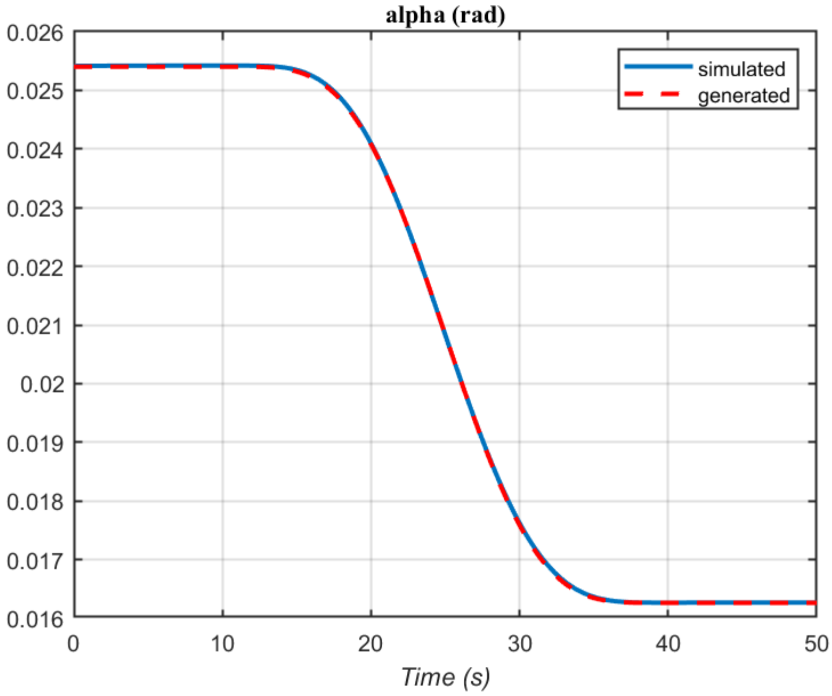


Figure 4.10: Result of the open-loop control for the angle of attack of the aircraft in horizontal flight. The value of α is obtained from (2.16).

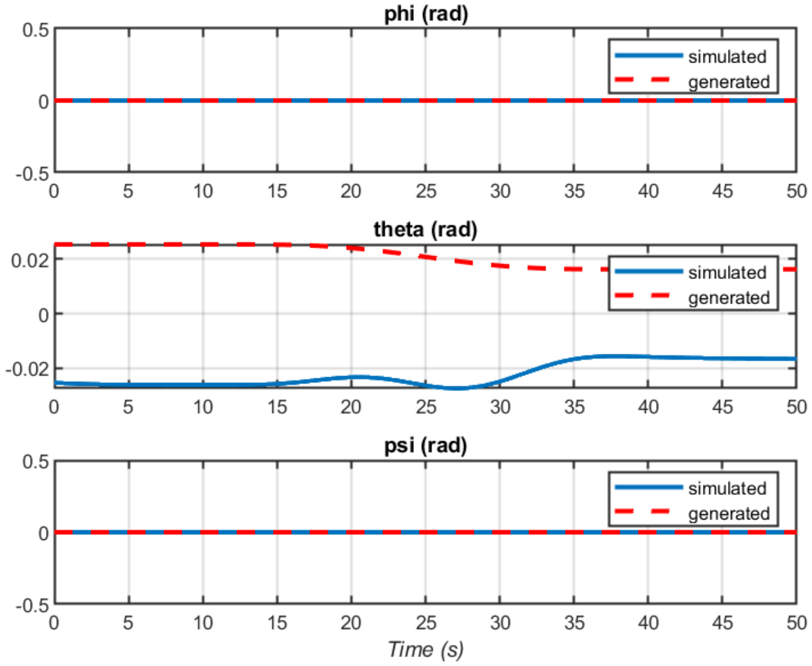


Figure 4.11: Resulting roll, pitch and yaw angles of the open-loop control, derived from the angles $\chi(t)$, $\gamma(t)$ and $\mu(t)$, for sake of clarity.

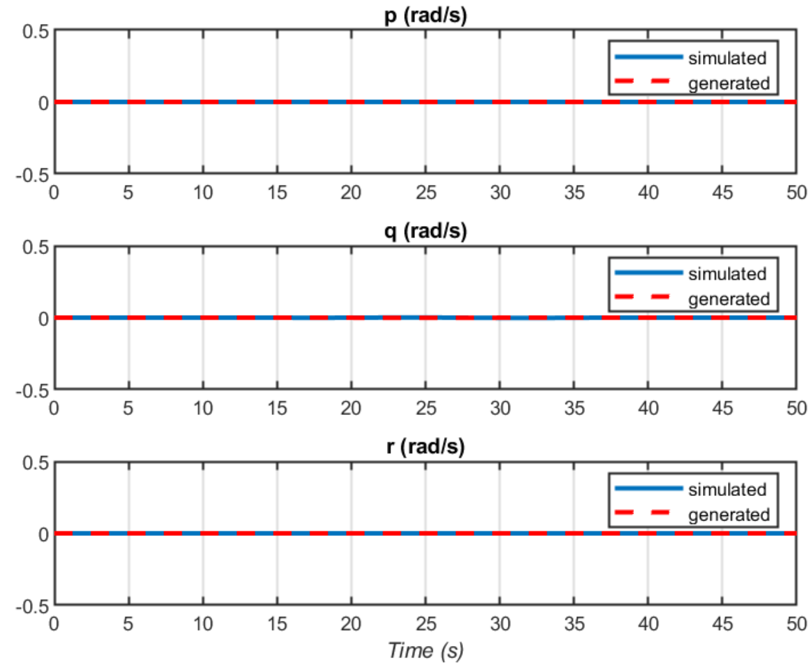


Figure 4.12: Result of the open-loop control for the angular rates $p(t)$, $q(t)$ and $r(t)$ in horizontal flight.

4.4 Results of the open-loop control for vertical take-off/landing and hover flight

The results for zero/small aerodynamic velocities are illustrated in this Section, in Figures 4.13, 4.14, 4.15 and 4.16. It is worth to remark that the control inputs obtained from the differential-flatness analysis perfectly match the ones obtained by the trim algorithm, ensuring equilibrium during flight, and that the total forces and moments acting on the aircraft are zero.

4.4.1 Results for vertical flight

The obtained results of the open-loop control for the vertical flight are reported here. The simulation shown here, illustrates the vertical movement during the take-off, from an altitude of 4m up to 8m in the time interval $t \in [10, 40]$s.

All the simulated results are highly accurate in comparison with the generated ones.

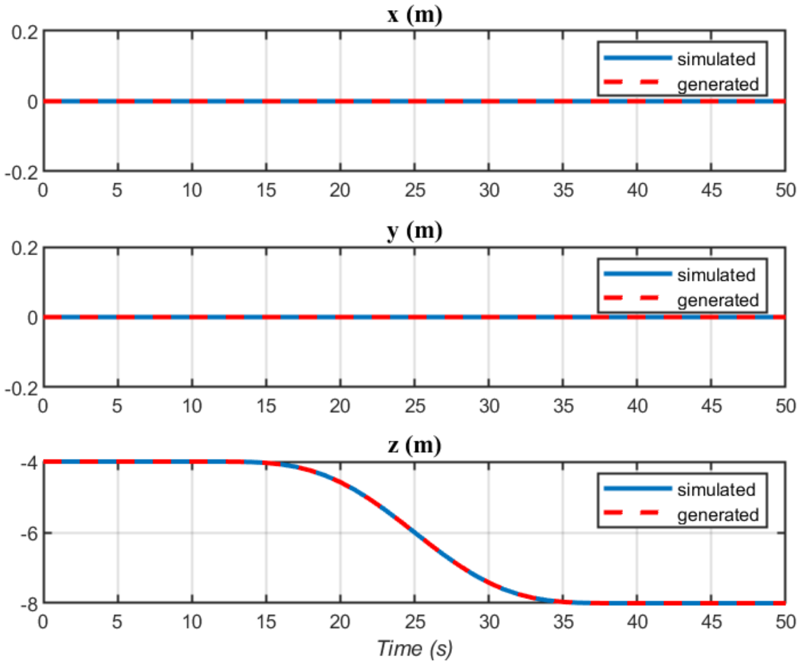


Figure 4.13: Result of the open-loop control, where the aircraft is required to vertically move from an altitude of 4m up to 8m in the time interval $t \in [10, 40]$s.

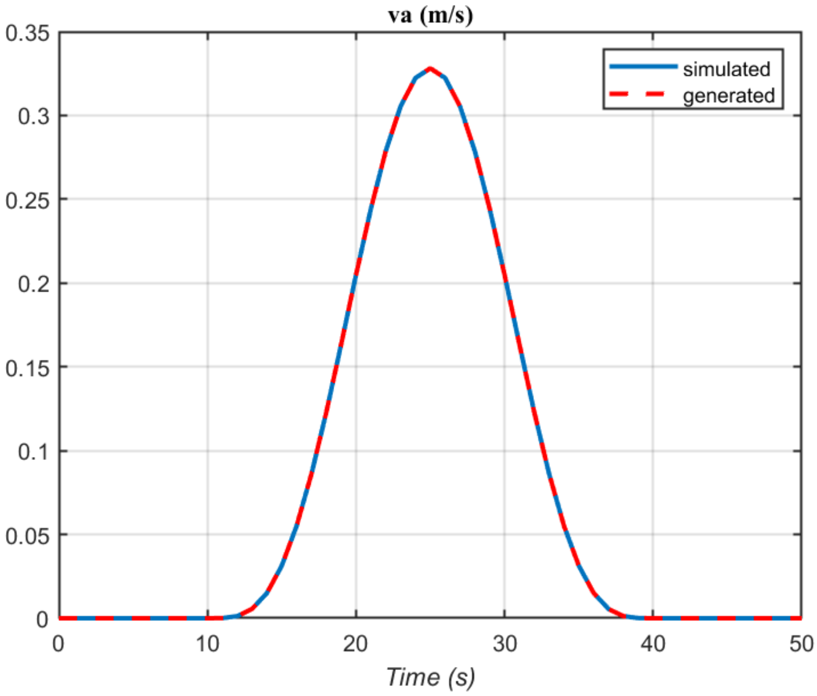


Figure 4.14: Resulting aerodynamic velocity of the open-loop control, as expected its value increases in the time interval $t \in [10, 40]$s, due to the variation in altitude of the aircraft.

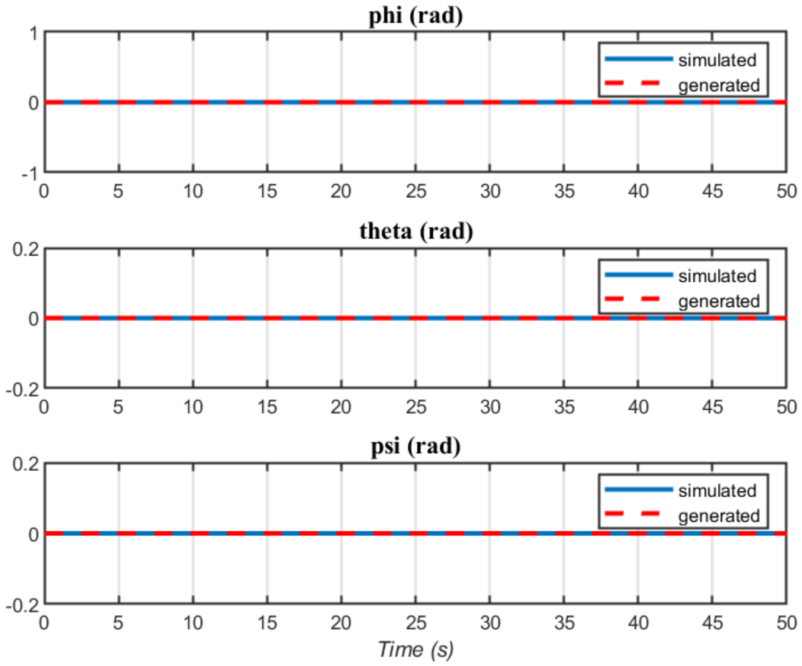


Figure 4.15: Resulting roll, pitch and yaw angles of the open-loop control, derived from the MRP components, for sake of clarity.

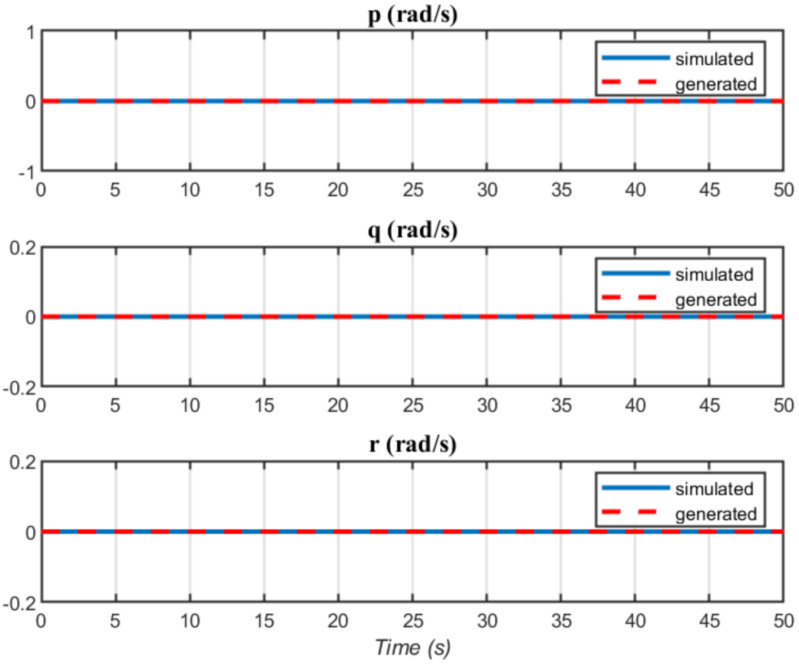


Figure 4.16: Resulting angular rates of the open-loop control.

4.4.2 Results for hover flight

The obtained results of the open-loop control, when the aircraft is required to fly at a fixed altitude of 4m and to perform a rotation from $\phi(t) = 0$ to $\phi(t) = \frac{\pi}{4}$ in the time interval $t \in [10, 40]$s, are shown in Figures 4.17, 4.18, 4.19 and 4.20.

All the simulated results are highly accurate in comparison with the generated ones, with a slight imprecision after 45s of simulation for the variables $x(t)$, $y(t)$ and $v_a(t)$.

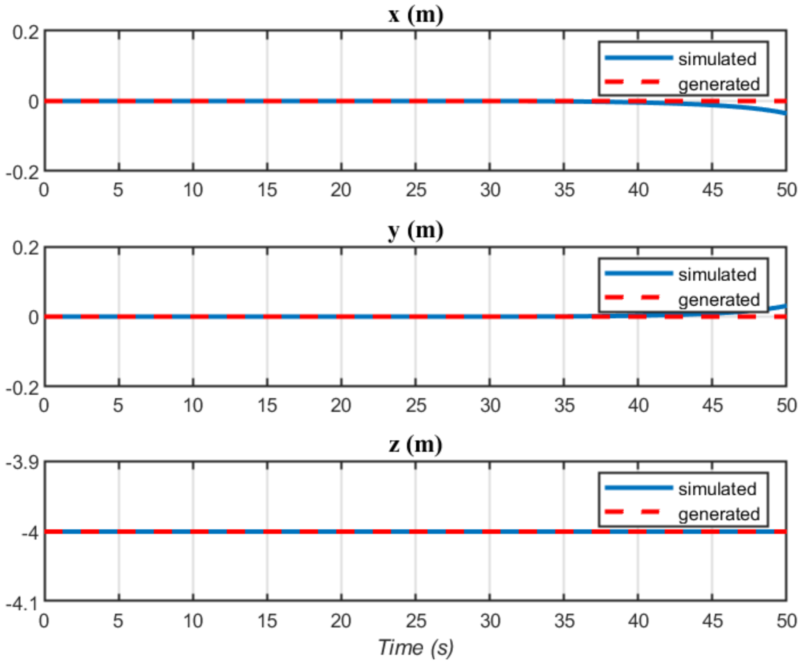


Figure 4.17: Result of the open-loop control for the position of the aircraft with respect to frame $\mathcal{I}$, where the aircraft is required to remain at an altitude of 4m.

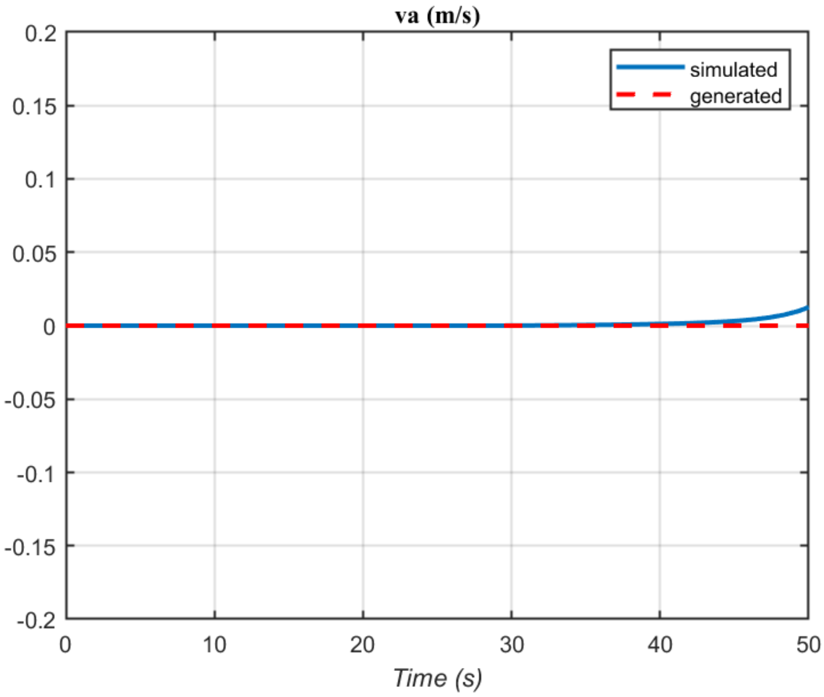


Figure 4.18: Resulting aerodynamic velocity of the open-loop control. As expected, since the aircraft is in hover flight, it is equal to zero. It has been obtained from the norm of the components of V_a^b, namely v_x, v_y and v_z.

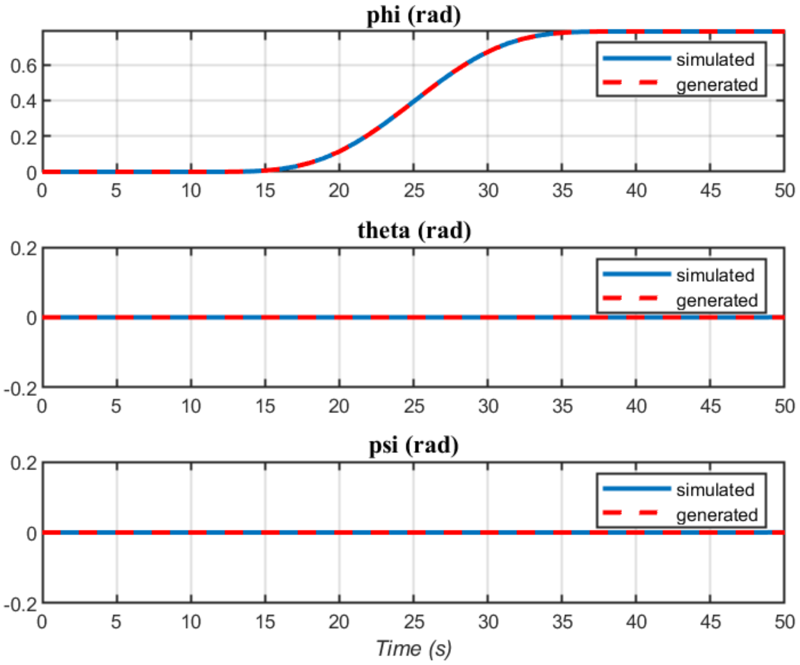


Figure 4.19: Resulting roll, pitch and yaw angles of the open-loop control, derived from the MRP components, for sake of clarity. As expected, the aircraft performs the requested rotation ϕ in the time interval $t \in [10, 40]$s.

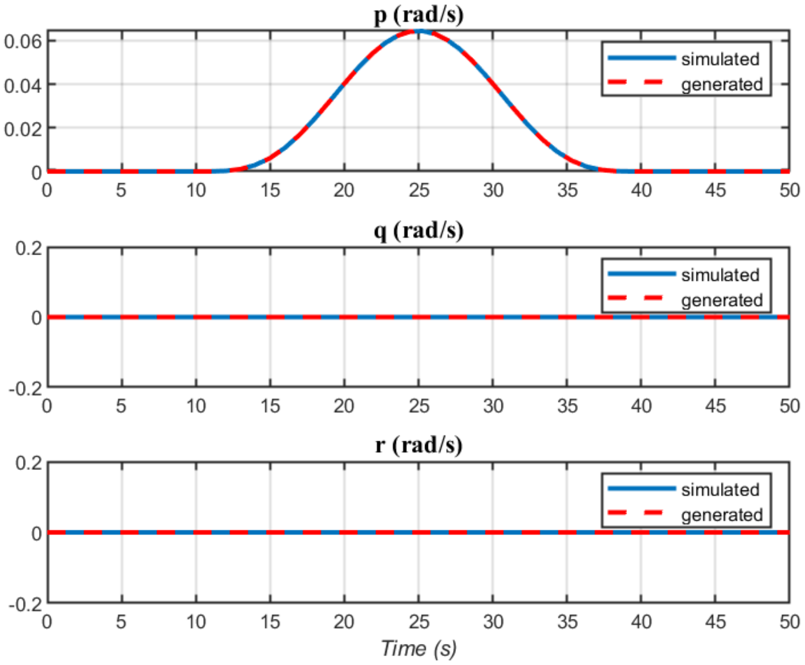


Figure 4.20: Resulting angular rates of the open-loop control. As expected, to perform the requested rotation ϕ, an effort from $p(t)$ is required in the time interval $t \in [10, 40]$s.

Chapter 5

Closed-loop control implementation and results

In Section 5.1 it is explained the procedure followed for the design of PID controllers , for the linearized and decoupled system via Singular Value Decomposition, during the different flight modes. Afterwards, in Section 5.2 and 5.3, the obtained results achieved with the addition of the feedback control, are compared to the results obtained with only the feedforward action.

5.1 Input decoupling and design of PID controllers

5.1.1 Horizontal flight

During horizontal flight the variables to control are the roll, pitch and yaw angles, $\phi(t)$, $\theta(t)$, $\psi(t)$, and the aerodynamic velocity $v_a(t)$. These movements are intrinsically coupled because actions taken to control one of them can affect the others. For instance, when the aircraft rolls, it can induce a pitch motion, and vice versa.

Hence, four new control inputs are defined as functions of the true ones $\delta_{el}, \delta_{er}, \omega_l, \omega_r$:

$$u_{va} = \omega_l + \omega_r \tag{5.1}$$

$$u_\theta = \delta_{el} + \delta_{er} \tag{5.2}$$

$$u_\phi = \delta_{el} - \delta_{er} \tag{5.3}$$

$$u_\psi = \omega_l - \omega_r \tag{5.4}$$

It is of interest to isolate the effect of each input, starting with the linear time-invariant

(LTI) system represented by the state-space equations:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{5.5}$$

$$y(t) = Cx(t) + Du(t) \tag{5.6}$$

where A, B, C, D are the matrices that describe the system dynamics. Among the various methods suitable for this purpose, the chosen one is the Singular Value Decomposition (SVD), which is applied at the decoupling frequency of $2\pi 10$ (rad/s). This allows to obtain four transfer functions for the variables $\phi(t)$, $\theta(t)$, $\psi(t)$ and $v_a(t)$ and for each of these, a PID controller is designed, whose transfer function, ideally, is defined as follows:

$$C(s) = K_P + \frac{K_I}{s} + K_D s = K_P \left(1 + \frac{1}{T_I s} + T_D s\right) \tag{5.7}$$

Since this is an improper transfer function due to the term $T_D s$, for physical realizability, the ideal derivative is replaced by the high-pass filter

$$H(s) = \frac{s}{T_L s + 1} \tag{5.8}$$

namely, a pole at high frequency $-1/T_L$ is introduced. So, the actual transfer function of the PID controller is

$$C(s) = K_P + \frac{K_I}{s} + K_D \frac{s}{T_L s + 1} = K_P \left(1 + \frac{1}{T_I s} + T_D \frac{s}{T_L s + 1}\right) \tag{5.9}$$

and its block diagram implementation is shown in Fig. 5.1.

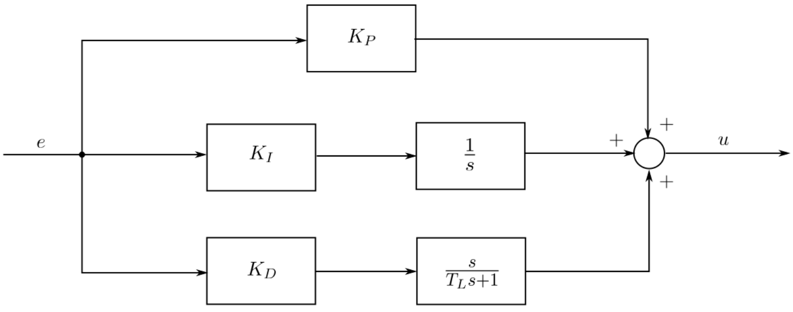


Figure 5.1: PID scheme.

In order to design the controller, then, the three gains K_P, K_I and K_D, related to the proportional, integral and derivative action respectively, must be chosen. Also, the high-frequency pole for the real derivative implementation must be placed.

The chosen tuning technique for the determination of the three gains is the *Bode's method* (frequency response method) proposed in [23], in order to satisfy the required control design specifications on the step response of the system. In particular, once that are given the requested settling time $t_{s,5\%}$ and the maximum admissible overshoot M_P (in this case they are chosen as $t_{s,5\%} \leq 0.15s$ and $M_P \leq 10\%$), assuming that the closed loop system admits a second order dominant pole approximation

$$P(s) = \frac{\omega_n^2}{s^2 + 2\delta\omega_n s + \omega_n^2}, \qquad 0 \leq \delta < 1 \tag{5.10}$$

the relationship between the time domain specifications and the parameters of the dominant poles (the natural frequency ω_n and the damping factor δ) is

$$t_{s,5\%} = \frac{3}{\delta\omega_n} \qquad \delta = \log \frac{\frac{1}{M_P}}{\sqrt{\pi^2 + \log^2 \frac{1}{M_P}}} \tag{5.11}$$

And, for a closed-loop system that admits a second order dominant pole approximation, the following relations hold between the parameters of the dominant poles and the target gain cross over frequency ω_{gc} and phase margin φ_m:

$$\omega_{gc} \approx \frac{3}{\delta t_{s,5\%}} \qquad \varphi_m = \operatorname{atan} \frac{2\delta}{\sqrt{\sqrt{1 + 4\delta^4} - 2\delta^2}} \tag{5.12}$$

Then, according to Bode's method, the PID gains can be computed as follows:

$$\begin{cases} K_P &= \Delta K \cos \Delta_\varphi \\ K_I &= \dfrac{K_P}{T_I} \\ K_D &= K_P T_D \end{cases} \tag{5.13}$$

where

$$\Delta K = |(P(j\omega_{gc}))|^{-1} \tag{5.14}$$

$$T_D = \frac{\tan(\Delta\varphi) + \sqrt{(\tan(\Delta\varphi))^2 + 4/\alpha}}{2\omega_{gc}} \tag{5.15}$$

$$T_I = \alpha T_D, \qquad \alpha = 4 \tag{5.16}$$

And finally, the frequency of the pole for the derivative implementation, is chosen 5 times larger than the gain cross over frequency:

$$T_L = \frac{1}{5\omega_{gc}} \tag{5.17}$$

The obtained PID transfer functions are defined as C_{va}, C_ϕ, C_θ and C_ψ, and the true control inputs are obtained as functions of u_{va}, u_ϕ, u_θ and $u\psi$:

$$\delta_{el} = \frac{u_\theta + u_\phi}{2} \tag{5.18}$$

$$\delta_{er} = \frac{u_\theta - u_\phi}{2} \tag{5.19}$$

$$\omega_l = \frac{u_{va} + u_\psi}{2} \tag{5.20}$$

$$\omega_r = \frac{u_{va} - u_\psi}{2} \tag{5.21}$$

The Simulink scheme is reported in Figure 5.2.

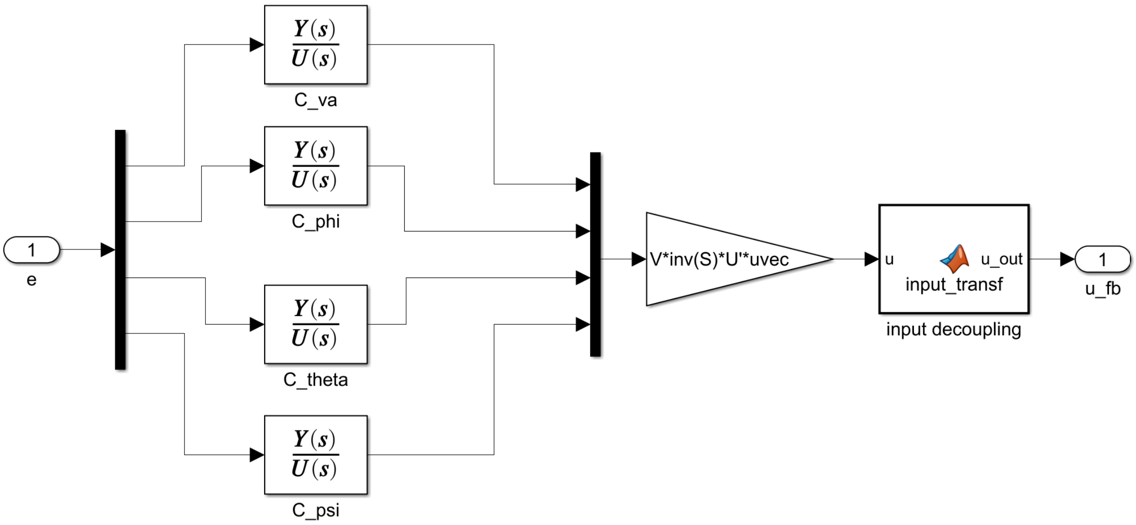


Figure 5.2: Simulink scheme of the feedback for the horizontal flight. S, V and U represent the matrices of the Singular Value Decomposition. V and U are the orthogonal matrices and S is the diagonal matrix containing singular values.

The obtained results are presented and commented in Section 5.2.

5.1.2 Vertical take-off/landing and hover flight

During the vertical and hover flight, the variables to control are the roll, pitch and yaw angles, $\phi(t)$, $\theta(t)$, $\psi(t)$, and the altitude $z(t)$. As in the horizontal flight mode, also these variables are intrinsically coupled and the four new control inputs are defined:

$$u_z = \omega_l + \omega_r \tag{5.22}$$

$$u_\theta = \delta_{el} + \delta_{er} \tag{5.23}$$

$$u_\phi = \delta_{el} - \delta_{er} \tag{5.24}$$

$$u_\psi = \omega_l - \omega_r \tag{5.25}$$

The same procedure used in horizontal flight which leads to the design of the four PID controllers is followed, and it is not reported here for sake of simplicity.

The definition of the true control inputs as function of u_z, u_ϕ, u_θ and u_ψ, is given by:

$$\delta_{el} = \frac{u_\theta + u_\phi}{2} \tag{5.26}$$

$$\delta_{er} = \frac{u_\theta - u_\phi}{2} \tag{5.27}$$

$$\omega_l = \frac{u_z + u_\psi}{2} \tag{5.28}$$

$$\omega_r = \frac{u_z - u_\psi}{2} \tag{5.29}$$

The obtained results are presented and commented in Section 5.3.

5.2 Results of the closed-loop control for horizontal flight

The obtained results with the feedback control, for an initial aerodynamic velocity of 20 $\mathrm{m\,s^{-1}}$, an altitude of 4m and an initial angle of attack of 0.0254 rad are reported in this Section and compared with the ones obtained with the feedforward control and the generated trajectory that should be followed, in Figures 5.3, 5.4, 5.5, 5.6, 5.7 and 5.8.

The usefulness of the feedback control part is evident comparing the results obtained with only the feedforward action, with respect to the generated ones. This can be seen for the aerodynamic velocity in Figure 5.3 and for the angle $\theta(t)$ in Figure 5.5, where the feedback corrects the inaccuracy of the feedforward action.

In Figures 5.7 and 5.8 it is shown how the feedback action keeps the errors between the generated and simulated trajectory of the variables to control close to zero. Clearly, due to the imprecision of the aerodynamic velocity and the angle $\theta(t)$ in following the desired trajectory even with the introduction of the feedback, the initial errors e_{va} and e_{theta} are higher.

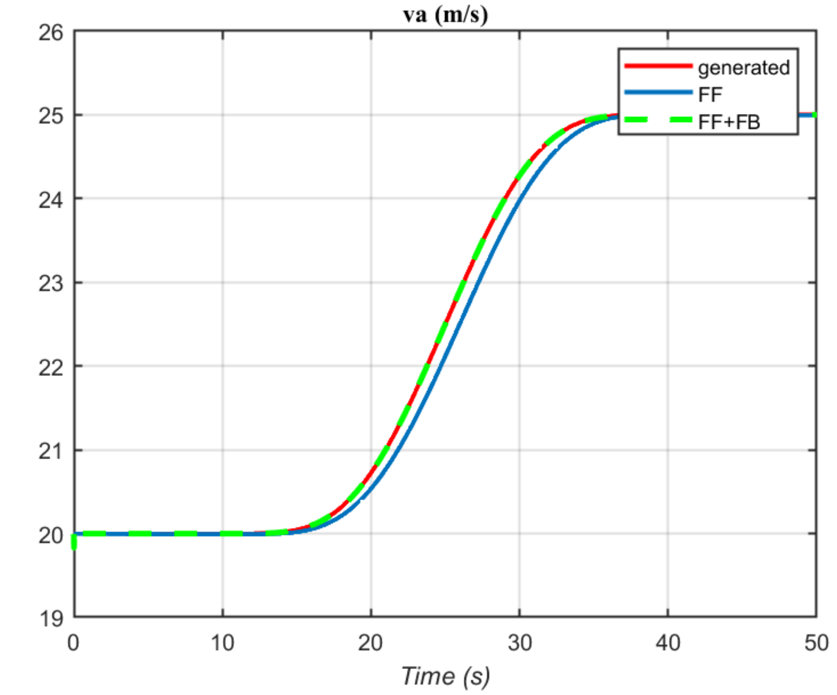


Figure 5.3: Comparison between the resulting aerodynamic velocity with the feedback control versus with the feedforward control, for the horizontal flight with an initial $v_a(t) = 20\ \mathrm{m\,s^{-1}}$.

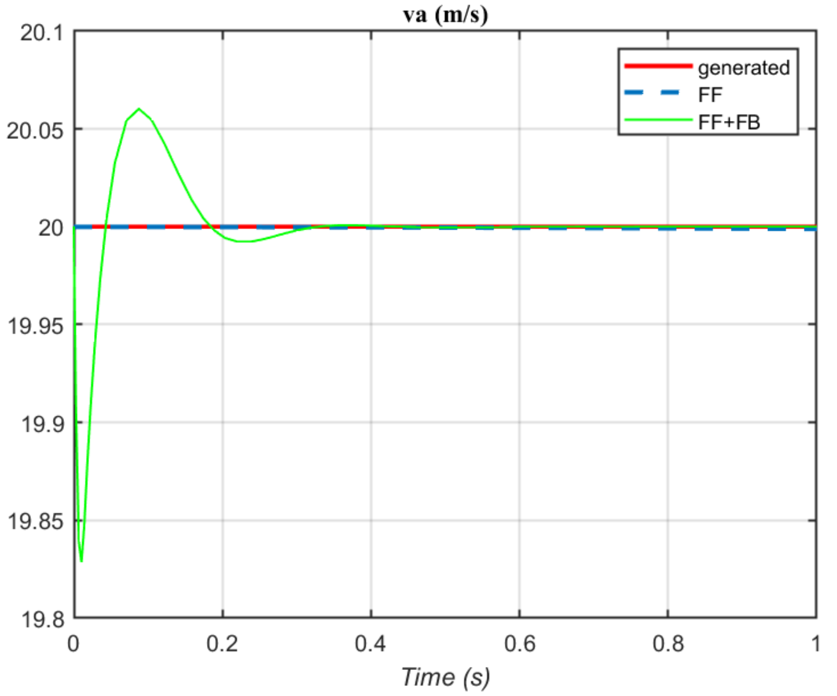


Figure 5.4: Zoomed view of Figure 5.3.

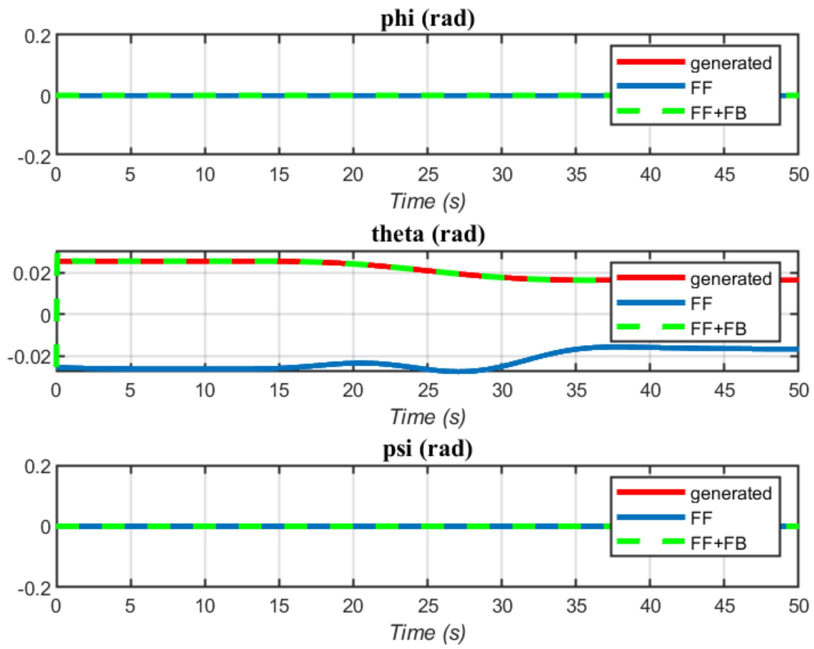


Figure 5.5: Comparison between the resulting roll, pitch and yaw angles with the feedback control versus with the feedforward control, for the horizontal flight with an initial $v_a(t) = 20\ \mathrm{m\,s^{-1}}$.

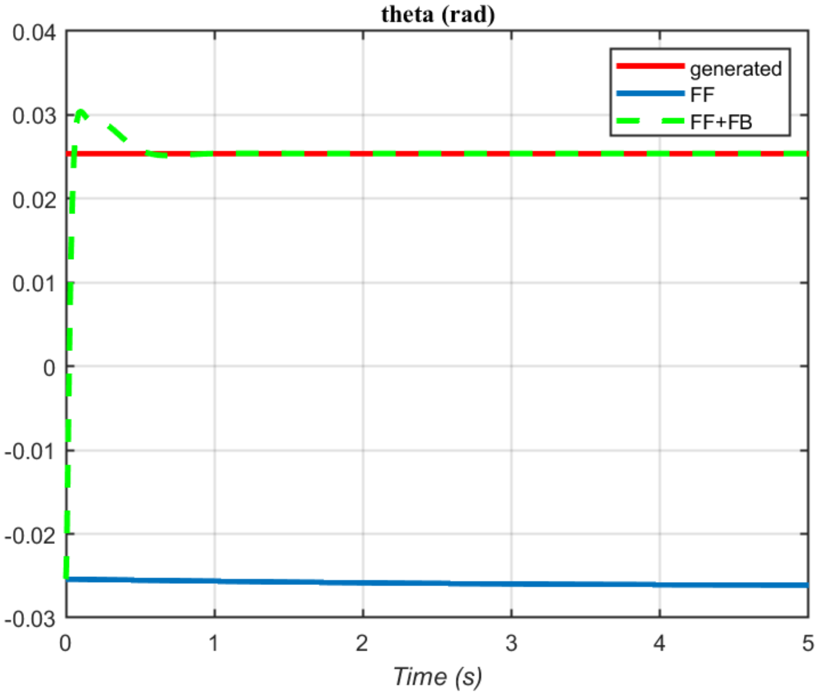


Figure 5.6: Zoomed view of the angle $\theta(t)$ in Figure 5.5.

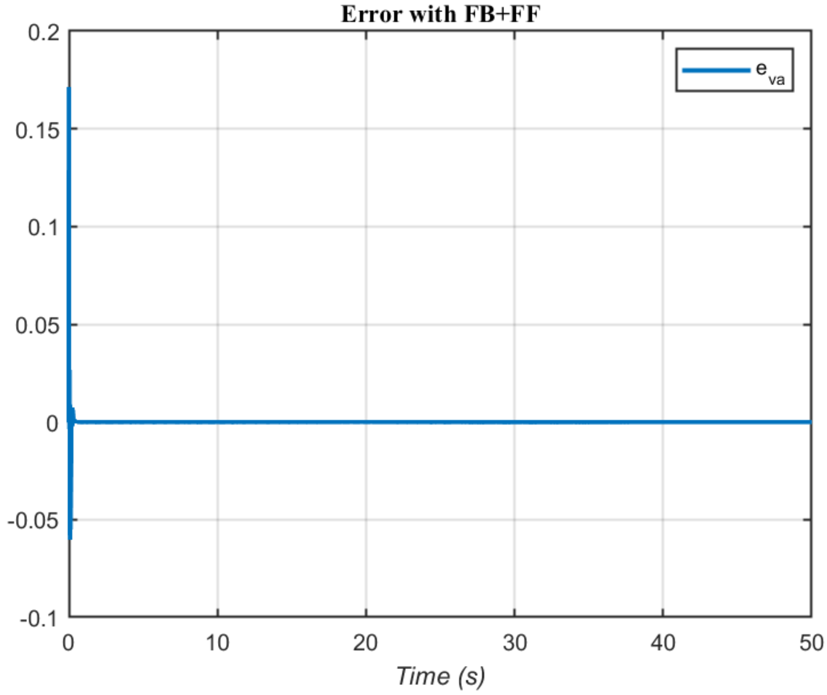


Figure 5.7: Error for the aerodynamic velocity adding the feedback to the feedforward action, for the horizontal flight with an initial $v_a(t) = 20\ \mathrm{m\,s^{-1}}$.

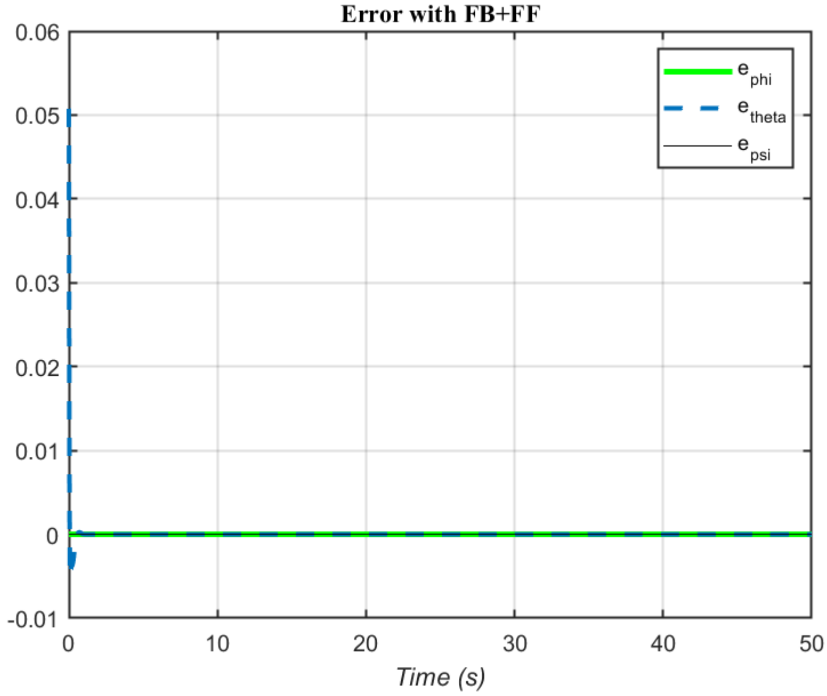


Figure 5.8: Error for the roll, pitch and yaw angles adding the feedback to the feedforward action, for the horizontal flight with an initial $v_a(t) = 20\ \mathrm{m\,s^{-1}}$.

5.3 Results of the closed-loop control for vertical take-off/landing and hover flight

5.3.1 Results for vertical flight

The obtained results with the feedback control for an initial altitude of 4m are reported in this Section and compared with the ones obtained with the feedforward control and the generated trajectory, in Figures 5.9 and 5.10.

The feedback action maintains an accurate tracking of the generated trajectory, and it can be observed in Figures 5.11 and 5.12, that the errors for the controlled variables are kept to zero.

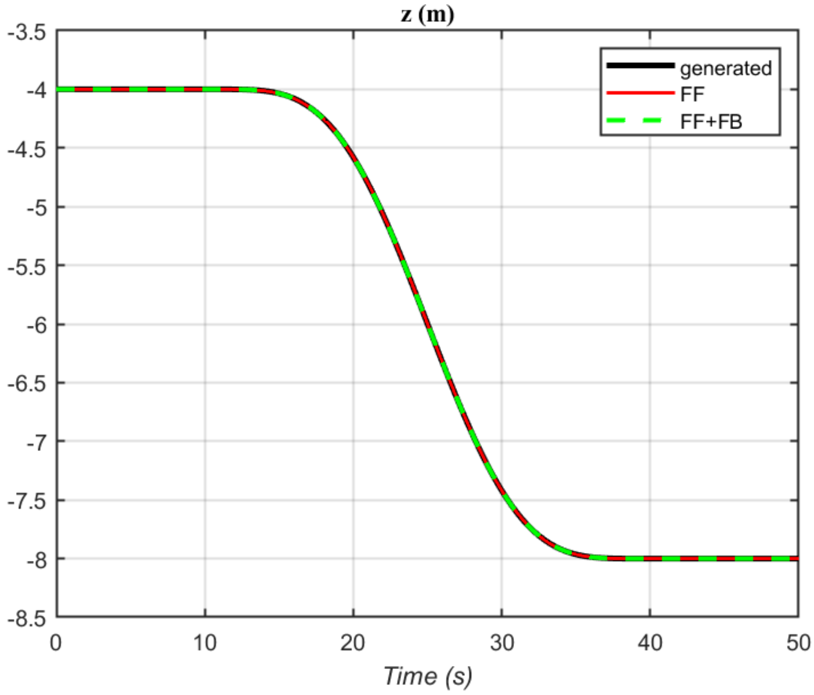


Figure 5.9: Comparison between the resulting altitude with the feedback control versus with the feedforward control, for the vertical flight with an initial altitude of 4m.

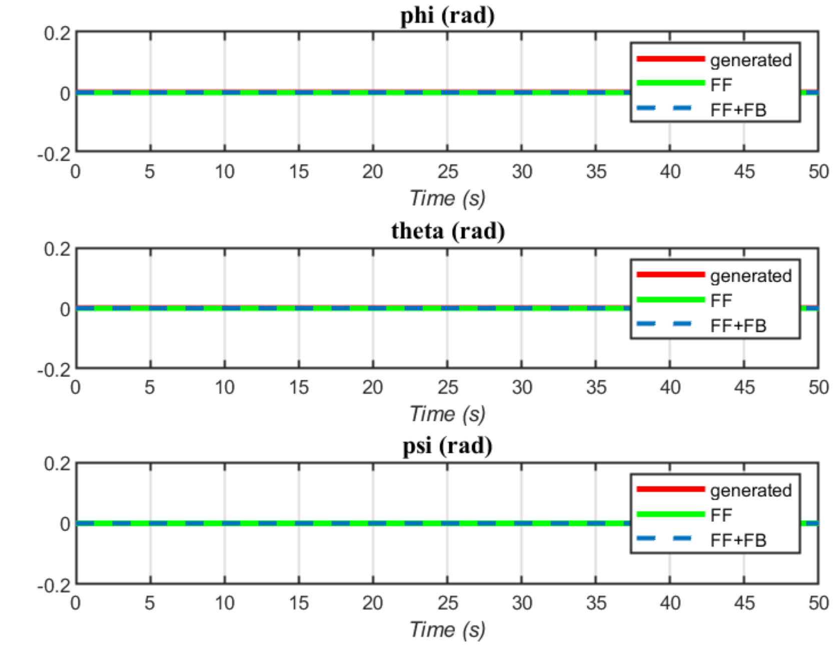


Figure 5.10: Comparison between the resulting roll, pitch and yaw angles with the feedback control versus with the feedforward control, for the vertical flight.

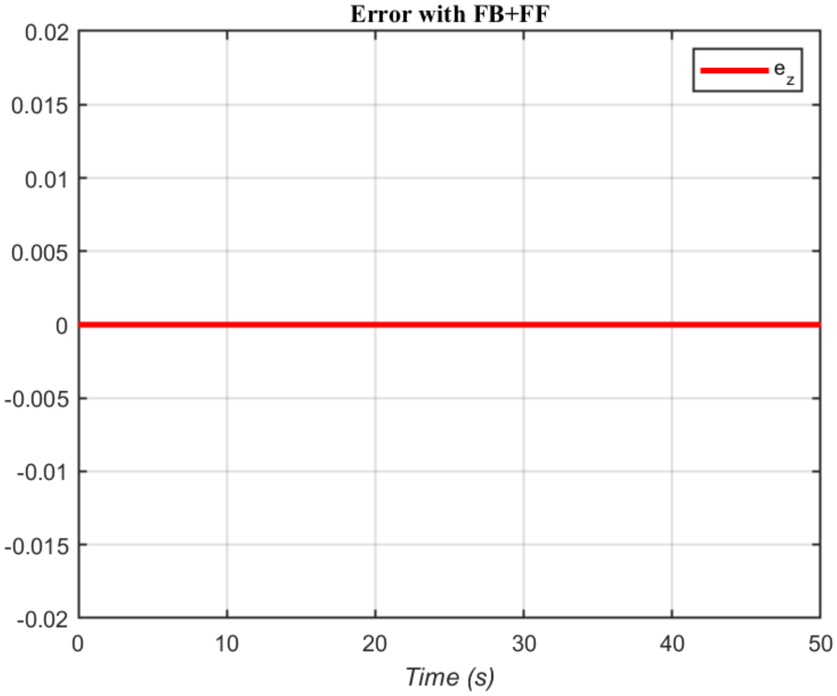


Figure 5.11: Error for the altitude $z(t)$ adding the feedback to the feedforward action, for the vertical flight.

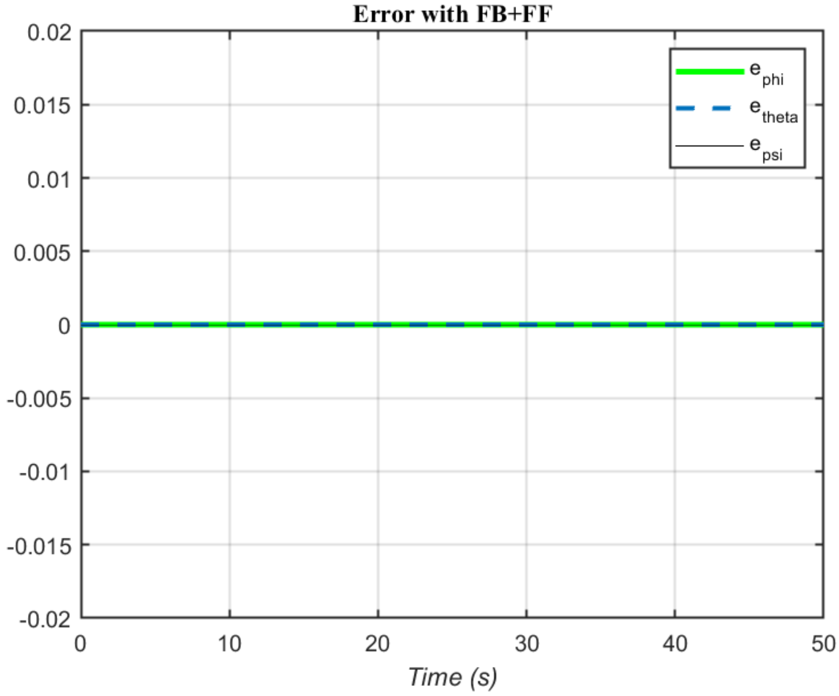


Figure 5.12: Error for the roll, pitch and yaw angles adding the feedback to the feedforward action, for the vertical flight.

5.3.2 Results for hover flight

The obtained results with the feedback control, when the aircraft is required to fly at a fixed altitude of 4m and to perform a rotation from $\phi(t) = 0$ to $\phi(t) = \frac{\pi}{4}$ in the time interval $t \in [10, 40]$s, are reported in this Section and compared with the ones obtained with the feedforward control and the generated trajectory, in Figures 5.13 and 5.14.

The feedback action maintains an accurate tracking of the generated trajectory, and it can be observed in Figures 5.15 and 5.16, that the errors for the controlled variables are kept to zero.

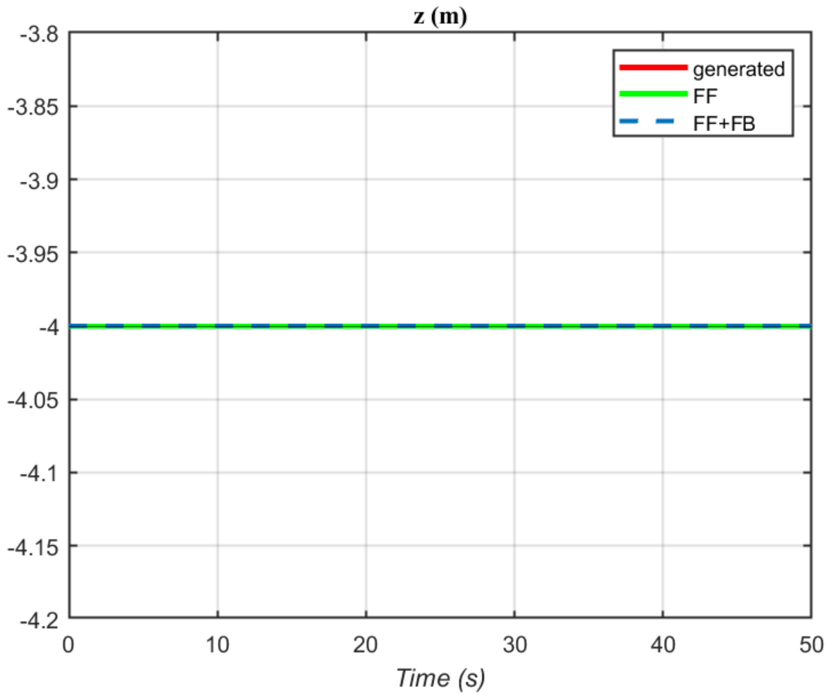


Figure 5.13: Comparison between the resulting altitude with the feedback control versus with the feedforward control, for the hover flight with an altitude of 4m.

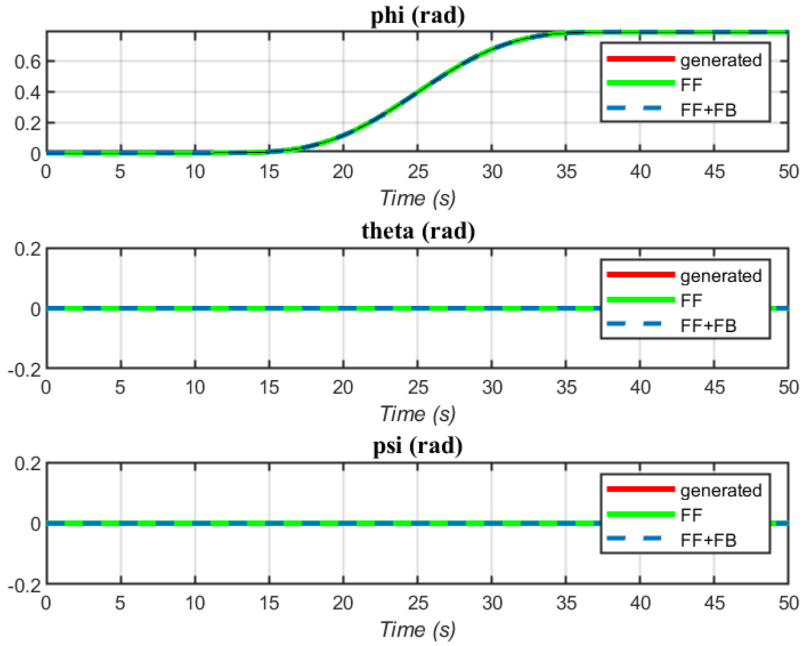


Figure 5.14: Comparison between the resulting roll, pitch and yaw angles with the feedback control versus with the feedforward control, for the hover flight.

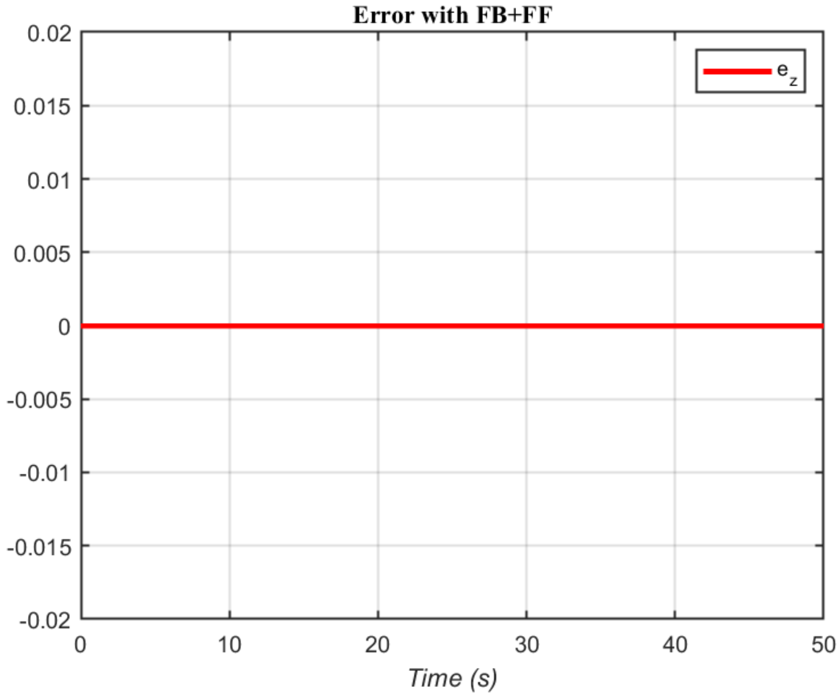


Figure 5.15: Error for the altitude $z(t)$ adding the feedback to the feedforward action, for the hover flight.

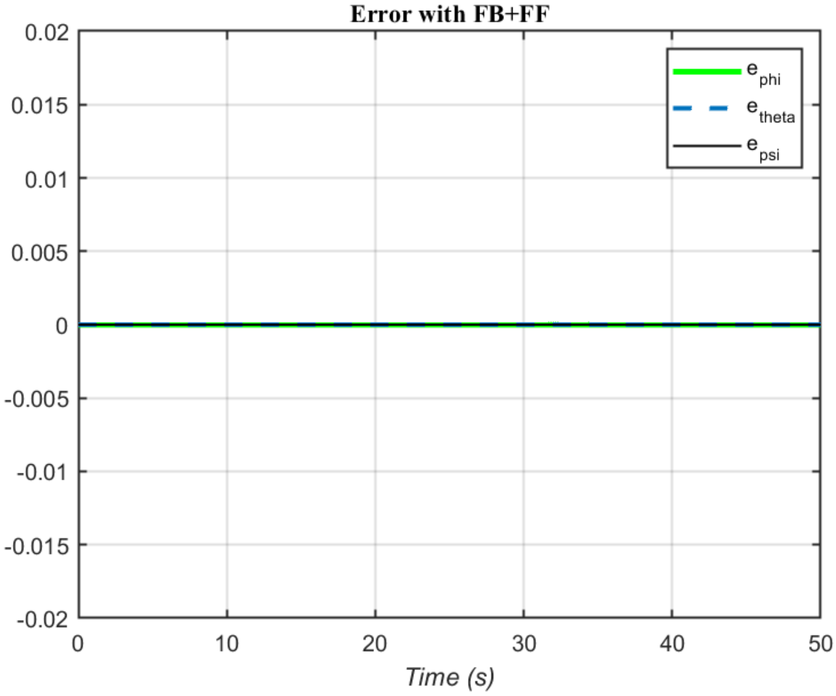


Figure 5.16: Error for the roll, pitch and yaw angles adding the feedback to the feedforward action, for the hover flight.

Chapter 6

Conclusion and further developments

The derived model includes all the significant phenomena for all the flight modes of this kind of aircraft, providing a very precise starting point for subsequent analyses and investigations.

The overall system has been proven to be flat, namely all the system's state and input variables can be expressed as functions of the flat-outputs and a finite number of their time derivatives. The flat-outputs that satisfy this property are $x(t)$, $y(t)$ and $z(t)$ for all the flight modes, with the addition of $\alpha(t)$ for the horizontal flight and of $\phi(t)$ for the vertical and hover flight.

The trajectory has been generated by four independent polynomials for the horizontal flight and vertical/hover flight respectively, that describe the evolution of their four flat output variables through time. This allows to define all the flight phases and their related maneuvers.

The results prove that a feedforward control, able to steer the system along the generated trajectory, can be built using the flatness relationships, which are only algebraic. Additionally, the control inputs obtained from this analysis perfectly match the ones derived from the trim algorithm, thus ensuring equilibrium during flight.

It is also proved that the feedforward action alone is not sufficient to guarantee high accuracy in tracking the generated trajectory, but a simple feedback law can be introduced, allowing to follow the trajectory in a good way.

Further developments may consist in implementing a switch based on the aerodynamic velocity, in order to automatically pass from the feedforward control in vertical and hover flight, to the horizontal flight. Nevertheless, the current result of the differential flatness analysis in horizontal flight at low aerodynamic speeds appears to be inaccurate and needs to be improved.

Appendix A

Matlab codes

A.1 Lift aerodynamic coefficient C_L versus the angle of attack α

```
function [coeffL] = coeffLf(alpha_deg)

alpha_rad = alpha_deg/180*pi;
va = 5;
c = 0.3302;
nu = 1.4776e-05;
Re = va*c/nu;

% Lift sigmoid function parameters
alpha0_deg = 9;
Re0 = 16e4;
pow_coef = 0.15;

% Lift coefficients
c1_lift = 4.9781;
c2_lift = 1.015;

alpha_0_Re = alpha0_deg*(Re/Re0)^pow_coef;

if (abs(alpha_deg)<=alpha_0_Re) || (abs(alpha_deg)>=180-alpha_0_Re)
   CL1 = c1_lift*sin(2*abs(alpha_rad));
   CL2 = 0;
elseif (abs(alpha_deg)>alpha_0_Re) && (abs(alpha_deg)<180-alpha_0_Re)
   CL1 = 0;
   CL2 = c2_lift*sin(2*abs(alpha_rad));
else
   CL1 = 0;
   CL2 = 0;
```

```
end
sigmoide = 1/(1+exp(abs(alpha_deg) - alpha_0_Re)) + 1/(1+exp(180-abs(alpha_deg) - alpha_0_Re));
coeffL = sign(alpha_deg) * (CL1*sigmoide+CL2*(1 - sigmoide));
end
```

A.2 Forces and moments

```
function[F,M,IMat,m] = forces_moments(x,u)

delta_el = u(1); % left elevon deflection angle
delta_er = u(2); % right elevon deflection angle
w_l = u(3); % left propeller angular rate (PWM)
w_r = u(4); % right propeller angular rate (PWM)

po_x = x(1); % x coordinate position in Earth frame
po_y = x(2); % y coordinate position in Earth frame
po_z = x(3); % z coordinate position in Earth frame
vb_x = x(4); % vx coordinate speed in Body frame
vb_y = x(5); % vy coordinate speed in Body frame
vb_z = x(6); % vz coordinate speed in Body frame

%% Compute air data
va = sqrt(vb_x^2+vb_y^2+vb_z^2); % aerodynamic speed
alpha_rad = atan2(vb_z,vb_x); % angle of attack
alpha_deg = alpha_rad/pi*180;
beta_rad = atan2(vb_y,va); % side-slip angle
beta_deg = beta_rad/pi*180;

bRa = Ry(-alpha_rad)*Rz(beta_rad); % rotation matrix from Aerodynamic to Body frame

%% Modified Rodrigues Parameters
sig_vec = x(7:9);
sig1 = sig_vec(1);
sig2 = sig_vec(2);
sig3 = sig_vec(3);

%% Rotation matrix with MRP
norm_sig_squared = sig_vec'*sig_vec;
sig_sqew = [0 -sig3 sig2; sig3 0 -sig1; -sig2 sig1 0];
bRo = eye(3)+1/(1+norm_sig_squared)^2*(8*sig_sqew*sig_sqew-4*(1-norm_sig_squared)*sig_sqew);

```

```
%% Roll, pitch and yaw angular rates
p = x(10);
q = x(11);
r = x(12);

%% Parameters
m = 1.56; % (kg) full weight of the delta-wing
g = 9.80665; % (m/s^2) gravitational acceleration
Ixx=.1147; % (kg*m^2) (Zagi flying wing)
Iyy=.0576; % (kg*m^2) (Zagi flying wing)
Izz=.1712; % (kg*m^2) (Zagi flying wing)
Ixz=.0015; % (kg*m^2) (Zagi flying wing)
IMat = [Ixx 0 -Ixz;0 Iyy 0;-Ixz 0 Izz];

rho = 1.225; % (kg/m^3) (density of the air)
mu = 1.81e-5; % (kg/m/s) (dynamic viscosity of the air at 15  )
nu = mu/rho; % (m^2/s) (kinematic viscosity of the air at 15  )

S = 0.2589; % (m^2) surface of the delta-wing
S_w = 1/2*S; % wet area
c = 0.3302; % (m) mean chord
b = 1.4224; % (m) tip to tip length of the wings
l_motor = b/4; % distance from motor to Oxb axis
S_prop = 0.0314; % (m^2)
k_motor = 40;
k_T_p = 1e-6; % maximum 1 Nm torque
k_omega = 1e3; % maximum 2000 rad/s

%% Lift sigmoid function parameters
Re0 = 16e4;
pow_coef = .15;
e = 0.8; % Oswald efficiency factor

%% Aerodynamic coefficients for Zagi flying wing
% Longitudinal aerodynamics
C_L_0 = 0;
C_D_0 = 0;
C_m_0 = 0;
C_L_alpha = 3.5016;
C_D_alpha = .2108;
C_m_alpha = -.5675;
C_L_q = 2.8932;
C_D_q = 0;
C_m_q = -1.3990;
C_L_delta_e = .2724;
C_D_delta_e = .3045;
```

```
C_m_delta_e = -.3254;
C_prop = 1;

% Lateral aerodynamics
C_Y_0 = 0;
C_l_0 = 0;
C_n_0 = 0;
C_Y_beta = -.07359;
C_l_beta = -.02854;
C_n_beta = -.00040;
C_Y_p = 0;
C_l_p = -.3209;
C_n_p = -.01297;
C_Y_r = 0;
C_l_r = .03066;
C_n_r = -.00434;
C_Y_delta_a = 0;
C_l_delta_a = .1682;
C_n_delta_a = -.00328;

%% Forces
% coeffL defined in "Lift aerodynamic coefficient versus the angle of attack"
coeffD = C_D_0+C_D_alpha*abs(alpha_rad);
coeffM = C_m_0+C_m_alpha*alpha_rad;

Fa_ra_x = ((rho*S)/2)*(coeffD*va^2+(C_D_q*c*q*va)/(2)+C_D_delta_e*(abs(delta_el)+abs(delta_er))*va^2); % drag force due to va
Fa_ra_z =  ((rho*S)/2)*(coeffL*va^2+C_L_q*c*q*va/(2)+C_L_delta_e*(delta_el+delta_er)*va^2);  % lift force due to va
Fa_ra_y = ((rho*S)/2)*(C_Y_0*va^2+C_Y_beta*beta_rad*va^2+C_Y_p*b*r*va*p/(2)+C_Y_delta_a*(delta_el-delta_el)*va^2); % lateral force due to va

Fthl_rb_x = rho*S_prop*C_prop*((k_motor*w_l)^2-va^2)/2; % thrust force due to the left propeller
Fthr_rb_x = rho*S_prop*C_prop*((k_motor*w_r)^2-va^2)/2; % thrust force due to the right propeller
F_thrust_tot = Fthl_rb_x+Fthr_rb_x; % total thrust force

if (va*cos(alpha_rad))^2+(2*(F_thrust_tot))/(rho*S_prop) >= 0
    vi = 1/2*(sqrt((va*cos(alpha_rad))^2+(2*(F_thrust_tot))/(rho*S_prop))-va*cos(alpha_rad)); % due to wing propeller interaction
else
    vi = 0;
end

```

```
Fa_ra_x_vi = ((rho*S_w)/2)*(C_D_delta_e*(abs(delta_el)+abs(delta_er))*vi^2); % drag force due to vi
Fa_ra_z_vi = ((rho*S_w)/2)*(C_L_delta_e*(delta_el+delta_er)*vi^2); % lift force due to vi

F_aero_A_va = [-Fa_ra_x Fa_ra_y -Fa_ra_z]';
F_aero_vi = [-Fa_ra_x_vi 0 -Fa_ra_z_vi]';
F = F_aero_vi +bRa*F_aero_A_va;
F = F + [Fthl_rb_x+Fthr_rb_x 0 0]'+bRo*[0;0;m*g]; % total force vector

%% Moments
C_L_simplified_va = C_l_0*va^2+C_l_beta*beta_rad*va^2+(C_l_p*b*p*va)/(2)+(C_l_r*b*r*va)/(2)+C_l_delta_a*(delta_el-delta_er)*va^2;
C_M_simplified_va = coeffM*va^2+(C_m_q*c*q*va)/(2)+(C_m_delta_e)*(delta_el+delta_er)*va^2;
C_N_simplified_va = C_n_0*va^2+C_n_beta*beta_rad*va^2+C_n_p*b*va/(2)*p+C_n_r*b*va/(2)*p+C_n_delta_a*(delta_el-delta_er)*va^2;

coeff_vector = [C_L_simplified_va C_M_simplified_va C_N_simplified_va]';
tau_ab = 1/2*rho*S*c*coeff_vector; % moment due to the aerodynamic effects

M_t_b = [k_T_p*(k_omega*w_l)^2-k_T_p*(k_omega*w_r)^2; 0; 0]; % moment induced by the propellers

M_p_b = [0;0;l_motor*(Fthl_rb_x-Fthr_rb_x)]; % moment induced by propulsion force

C_M_vi = (C_m_delta_e)*(delta_el+delta_er)*vi^2; % torques generated by the propeller wing interaction
M_prop = 1/2*rho*S_w*c*[0;C_M_vi;0];
M_elevon_lift = b/4 * [rho*S_w/2*(C_L_delta_e*(delta_el-delta_er)*vi^2);0;0];

M = M_t_b + M_prop + M_p_b + tau_ab + M_elevon_lift; % total torque vector

end
```

A.3 Trajectory generation and trim for vertical take-off/landing, hover and horizontal flight

```
trim_sel = input('Select horizontal (0, default) or hover trim (1): ');

if isempty(trim_sel)
        trim_sel = 0;
```

```
end


if trim_sel==1
    %% Trim function for vertical take-off/landing and hover flight

    % Initial altitude of the aircraft to be chosen
    z_t1 = -4;

    q0 = [cos(pi/4);0;sin(pi/4);0];
    sig0 = q0(2:4)/(1+q0(1));
    x_trim = [0;0;z_t1;zeros(3,1);sig0;zeros(3,1)]

    % Parameters
    m = 1.56;
    g = 9.80665;
    k_motor = 40;
    rho = 1.225;
    S_prop = 0.0314;
    C_prop = 1;

    w_lr = sqrt(m*g/rho/S_prop/C_prop)/k_motor;
    u_trim = [0;0;w_lr;w_lr]

    % Trajectory generation with waypoints and time interval to be chosen
    wPts1 = [0;0;-4;0];
    wPts2 = [0;0;-4;pi/4];
    tPts = [10,40];
    [q,qd,qdd,qddd,qdddd,pp,tPoints,tSamples] = minsnappolytraj([wPts1,wPts2],tPts,100);


elseif trim_sel==0
    %% Trim function for horizontal flight

    % Initial parameters to be chosen
    z_t1 = -4;
    alpha0 = pi/100;
    pos0 = [0;0;z_t1];

    % Selection of the velocity in [m/s] for the aircraft in horizontal flight
    v_selection = input('Select the velocity for the horizontal flight (m/s): ');
    if isempty(v_selection)
              v_selection = 20;
```

```
end
v0 = [v_selection;0;0];

q0 = [cos(alpha0/2);0;sin(alpha0/2);0];
sig0 = q0(2:4)/(1+q0(1));
norm_sig0_squared = sig0'*sig0;
sig_sqew = [0 -sig0(3) sig0(2); sig0(3) 0 -sig0(1); -sig0(2) sig0(1) 0];
bRo = eye(3)+1/(1+norm_sig0_squared)^2*(8*sig_sqew*sig_sqew-4*(1-norm_sig0_squared)*sig_sqew);
vB = bRo*v0;
x0 = [pos0;vB;sig0;zeros(3,1)];
x_trim = 0*x0;

u0_trim = [-0.2;-0.2;0.4;0.4];
u_trim = 0*u0_trim;
ix_trim = [2,3,5,7,9:12]; %fixed states
dx0_trim = [v0;zeros(9,1)];
idx_trim = [1:12]'; % fixed derivatives
disp('trim ...')
[x_trim,u_trim,~,~,options] = trim('trim_model_MRP',x0,u0_trim,[],ix_trim,[],[],dx0_trim,idx_trim);
disp('These are x_0 and x_trim')
[x0, x_trim]
disp('These are u0 and u_trim')
[u0_trim, u_trim]
disp('done !')
disp('These are alpha_0 and alpha_trim (in deg)')
[atan2(x0(6),x0(4)), atan2(x_trim(6),x_trim(4))]
alpha_trim = atan2(x_trim(6),x_trim(4));

% Second trim point
v0 = [v_selection+5;0;0];
q0 = [cos(alpha0/2);0;sin(alpha0/2);0];
vB = bRo*v0;
x0 = [pos0;vB;sig0;zeros(3,1)];
u0_trim = [-0.2;-0.2;0.4;0.4];
u_trim1 = u_trim;
u_trim = 0*u0_trim;
ix_trim = [2,3,5,7,9:12]; %fixed states
dx0_trim = [v0;zeros(9,1)];
idx_trim = [1:12]'; % fixed derivatives
disp('trim ...')
[x_trim2,u_trim2,~,~,options] = trim('trim_model_MRP',x0,u0_trim,[],ix_trim,[],[],dx0_trim,idx_trim);
disp('These are x_0 and x_trim')
```

```
    [x0, x_trim2]
    disp('These are u0 and u_trim')
    [u0_trim, u_trim2]
    disp('done !')
    disp('These are alpha_0 and alpha_trim')
    [atan2(x0(6),x0(4)), atan2(x_trim2(6),x_trim2(4))]
    alpha_trim2 = atan2(x_trim2(6),x_trim2(4));

    u_trim = u_trim1;

    % Trajectory generation with waypoints and time interval to be chosen
    tPts = [10,50];
    wPts1 = [v_selection;alpha_trim]
    wPts2 = [v_selection+5;alpha_trim2]
    [q,qd,qdd,qddd,qdddd,pp,tPoints,tSamples] = minsnappolytraj([wPts1,
       wPts2],tPts,100);
    tInterp = minsnappolytraj([wPts1,wPts2],tPts,100);

else
    disp('Error: bad trim select value!')
end
```

A.4 SVD and PID design

It is reported here only the code for the vertical and hover flight, since for the horizontal flight the same procedure is mentained.

```
%% State_space representation
[A,B,C,D] = linmod('lin_hover',x_trim,zeros(size(u_trim)));
sys_struc = linmod('lin_hover',x_trim,zeros(size(u_trim)));
sys = ss(A,B,C,D)

%% SVD decoupling
% Decoupling frequency (rad/s)
wc = 2*pi*10;

% System's response at the decoupling frequency
H1 = evalfr(sys,1j*wc);

% Real approximation of G(j*wc)
D = pinv(real(H1'*H1));
H1 = pinv(D*real(H1'*diag(exp(1j*angle(diag(H1*D*H1.'))/2))))

% Singular value decomposition
[U,S,V] = svd(H1);
```

```

% Decoupled system
G_svd = tf(sys*V*inv(S)*U');

G_z = G_svd(1,1)
G_phi = G_svd(2,2)
G_theta = G_svd(3,3)
G_psi = G_svd(4,4)

%% PID parameters Bode method

% Specifications
ts5 = 0.15; % settling time
Mp = 10/100; % maximum peak

% Specification translation
delta = (log(1/Mp)/sqrt(pi^2 + (log(1/Mp))^2)); % damping factor
wgc = 3/(delta*ts5); % gain crossover frequency
phi_m = atan(2*delta/sqrt(sqrt(1+4*delta^4)-2*delta^2)); % phase margin

% Parameter for real derivative
Tl = 1/(5*wgc);
rd_num = [1 0];
rd_den = [Tl 1];

% Bode method for altitude control
alpha_z = 4;
Delta_K_z = 1/abs(evalfr(G_z,1i*wgc));
Delta_phi_z = -pi + phi_m - angle(evalfr(G_z,1i*wgc));
Kp_z = Delta_K_z*cos(Delta_phi_z);
TD_z = (tan(Delta_phi_z)+sqrt((tan(Delta_phi_z))^2+4/alpha_z))/(2*wgc);
TI_z = alpha_z*TD_z;
Kd_z = Kp_z*TD_z;
Ki_z = Kp_z/TI_z;

% Bode method for roll control
alpha_phi = 4;
Delta_K_phi = 1/abs(evalfr(G_phi,1i*wgc));
Delta_phi_phi = -pi + phi_m - angle(evalfr(G_phi,1i*wgc));
Kp_phi = Delta_K_phi*cos(Delta_phi_phi);
TD_phi = (tan(Delta_phi_phi)+sqrt((tan(Delta_phi_phi))^2+4/alpha_phi))/(2*wgc);
TI_phi = alpha_phi*TD_phi;
Kd_phi = Kp_phi*TD_phi;
Ki_phi = Kp_phi/TI_phi;

```

```matlab
% Bode method for pitch control
alpha_theta = 4;
Delta_K_theta = 1/abs(evalfr(G_phi,1i*wgc));
Delta_phi_phi = -pi + phi_m - angle(evalfr(G_phi,1i*wgc));
Kp_theta = Delta_K_theta*cos(Delta_phi_phi);
TD_theta = (tan(Delta_phi_phi)+sqrt((tan(Delta_phi_phi))^2+4/alpha_theta))/(2*wgc);
TI_theta = alpha_theta*TD_theta;
Kd_theta = Kp_theta*TD_theta;
Ki_theta = Kp_theta/TI_theta;

% Bode method for yaw control
alpha_psi = 4;
Delta_K_psi = 1/abs(evalfr(G_phi,1i*wgc));
Delta_phi_phi = -pi + phi_m - angle(evalfr(G_phi,1i*wgc));
Kp_psi = Delta_K_psi*cos(Delta_phi_phi);
TD_psi = (tan(Delta_phi_phi)+sqrt((tan(Delta_phi_phi))^2+4/alpha_psi))/(2*wgc);
TI_psi = alpha_psi*TD_psi;
Kd_psi = Kp_psi*TD_psi;
Ki_psi = Kp_psi/TI_psi;


%% PID for z (altitude)
C_num_z = [Kp_z*(TI_z*Tl+TD_z*TI_z) Kp_z*TI_z+Tl Kp_z*1];
C_den_z = [TI_z*Tl TI_z 0];
C_z = tf(C_num_z,C_den_z);

%% PID for phi (roll angle)
C_num_phi = [Kp_phi*(TI_phi*Tl+TD_phi*TI_phi) Kp_phi*TI_phi+Tl Kp_phi*1];
C_den_phi = [TI_phi*Tl TI_phi 0];
C_phi = tf(C_num_phi,C_den_phi);

%% PID for theta (pitch angle)
C_num_theta = [Kp_theta*(TI_theta*Tl+TD_theta*TI_theta) Kp_theta*TI_theta+Tl Kp_theta*1];
C_den_theta = [TI_theta*Tl TI_theta 0];
C_theta = tf(C_num_theta,C_den_theta);

%% PID for psi (yaw angle)
C_num_psi = [Kp_psi*(TI_psi*Tl+TD_psi*TI_psi) Kp_psi*TI_psi+Tl Kp_psi*1];
C_den_psi = [TI_psi*Tl TI_psi 0];
C_psi = tf(C_num_psi,C_den_psi);
```

Bibliography

[1] Tudor-Bogdan Airimitoaie et al. “Convertible aircraft dynamic modelling and flatness analysis”. In: *IFAC-PapersOnLine* 51.2 (2018), pp. 25–30.

[2] Tudor-Bogdan Airimitoaie et al. “Convertible delta-wing aircraft for teaching and research”. In: *IFAC-PapersOnLine* 52.12 (2019), pp. 478–483.

[3] Ahmed Alshahir et al. “Quadrotor UAV Dynamic Visual Servoing Based on Differential Flatness Theory”. In: *Applied Sciences* 13.12 (2023), p. 7005.

[4] RW Beard and TW Mclain. “Navigation, Guidance, and Control of Small Unmanned Aircraft”. In: *Guidance, and Control of Small Unmanned Aircraft, Brigham Young University* (2009).

[5] Adam Bry et al. “Aggressive flight of fixed-wing and quadrotor aircraft in dense indoor environments”. In: *The International Journal of Robotics Research* 34.7 (2015), pp. 969–1002.

[6] Marco Capodiferro and Giovanni Castelli Dezza. “Flatness-based trajectory generator for VTOL transition aircraft”. In: (2020).

[7] Pedro Castillo, Rogelio Lozano, and Alejandro Enrique Dzul. *Modelling and control of mini-flying machines*. Springer Science & Business Media, 2005.

[8] Agostino De Marco, Eugene Duke, and Jon Berndt. “A general solution to the aircraft trim problem”. In: *AIAA Modeling and Simulation Technologies Conference and Exhibit*. 2007, p. 6703.

[9] Matthias Faessler, Antonio Franchi, and Davide Scaramuzza. “Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories”. In: *IEEE Robotics and Automation Letters* 3.2 (2017), pp. 620–626.

[10] Jeff Ferrin et al. “Differential flatness based control of a rotorcraft for aggressive maneuvers”. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Ieee. 2011, pp. 2688–2693.

[11] Michel Fliess et al. “Flatness and defect of non-linear systems: introductory theory and examples”. In: *International journal of control* 61.6 (1995), pp. 1327–1361.

[12] *https://en.wikipedia.org/wiki/Convair_XFY_Pogo.*

[13] *https://en.wikipedia.org/wiki/Flatness_(systems_theory).*

[14] *https://www.nasa.gov/feature/langley/avoidance-is-the-name-of-the-game-at-uas-safety-testing.*

[15] *https://www.uavos.com/uavos-fixed-wing-uav-sitaria-completed-flight-tests/.*

[16] Jihène Kraiem. *Modélisation et détection de défauts: Application à deux systèmes plats.* Master thesis, 2018.

[17] Leandro R Lustosa, François Defay, and Jean-Marc Moschetta. "Longitudinal study of a tilt-body vehicle: modeling, control and stability analysis". In: *2015 International Conference on Unmanned Aircraft Systems (ICUAS).* IEEE. 2015, pp. 816–824.

[18] Philippe Martin, Pierre Rouchon, and Richard M Murray. "Flat systems, equivalence and trajectory generation". PhD thesis. Optimization and Control, 2006.

[19] Richard M Murray, Muruhan Rathinam, and Willem Sluis. "Differential flatness of mechanical control systems: A catalog of prototype systems". In: *ASME international mechanical engineering congress and exposition.* Citeseer. 1995.

[20] Michiel van Nieuwstadt, Muruhan Rathinam, and Richard M Murray. "Differential flatness and absolute equivalence of nonlinear control systems". In: *SIAM Journal on Control and Optimization* 36.4 (1998), pp. 1225–1239.

[21] Duc Kien Phung. "Conception, modeling, and control of a convertible mini-drone". PhD thesis. Université Pierre et Marie Curie-Paris VI, 2015.

[22] Daniele Pucci. "Towards a unified approach for the control of aerial vehicles". PhD thesis. Université Nice Sophia Antipolis, 2013.

[23] Francesco Ticozzi Riccardo Antonello. "Laboratory activity HANDOUT 0: Position PID–control of a DC servomotor". In: (2021).

[24] Charles Richter, Adam Bry, and Nicholas Roy. "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments". In: *Robotics Research: The 16th International Symposium ISRR.* Springer. 2016, pp. 649–666.

[25] Jocelyn Sabatier et al. "Fractional order differentiation and robust control design". In: *Intelligent systems, control and automation: science and engineering* 77 (2015), pp. 13–18.

[26] Hanspeter Schaub and John L Junkins. *Analytical mechanics of space systems.* Aiaa, 2003.

[27] Ezra Tal and Sertac Karaman. “Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness”. In: *IEEE Transactions on Control Systems Technology* 29.3 (2020), pp. 1203–1218.

[28] Dandan Zhang, Xin Jin, and Hongye Su. “A perspective on Attitude Control Issues and Techniques”. In: *arXiv preprint arXiv:2206.15077* (2022).