# POLITECNICO DI TORINO

**Master's Degree in Aerospace Engineering**

**Master's Degree Thesis**

# Control System Design of a H-shape configuration Fixed Wing VTOL based on a Patented Vertical Thrust System

**Supervisors**

**Prof. Angelo LERRO**

**Prof. Piero GILI**

**Candidate**

**Giorgio Antonio ORLANDO**

**October 2023**

# Summary

Urban air mobility (UAM) can become an integral part of worldwide urban mobility for last-mile connections (up to 50 km range) or sub-regional routes (from 100 to 300 km range). To work in future urban scenarios, environmentally friendly air vehicles are designed to be able to perform vertical take-off and landing (VTOL). Although many challenges are expected to be faced for enabling technologies, infrastructures and air space management, many prototypes of UAM vehicles are proposed. VTOL characteristics can be easily achieved by a multirotor configuration in order to produce a vertical force to balance the weight of the system. However, such a configuration limits the aerodynamic efficiency and poses noticeable endurance restrictions, thus limiting the operability of the UAVs to local missions only. A first approach to improve the flight efficiency is to design a hybrid configuration with both Fixed Wing (FW) and MultiRotor (MR) capabilities in order to exploit lift generation in horizontal flight. The hybrid nature of the vehicle implies several design complications in terms of transition phase from MR to FW mode during which the system is extremely over-actuated. Moreover, for multiple people carrying applications, the vertical thrust required for vertical operations involves large rotors. This could significantly penalize the aerodynamic performance of the aircraft in FW configuration. In order to overcome the aforementioned limitations, a novel solution based on the patented system ThrustPod is proposed. A conventional FW aircraft is equipped with rotors that are activated only during vertical flight phases. On the contrary, during the FW cruise phase, the rotors are retracted inside the fuselage, and the horizontal thrust is provided by dedicated tail propellers. The typical mission of the vehicle would include MR vertical take-off and landing and FW cruising with a transition to mildly transfer the control functions from the vertical rotors to the aerodynamic surfaces. The transition regime causes additional difficulties in the control system design as control inputs from MR and FW configurations are simultaneously active. Therefore, a control allocation scheme is needed to adequately distribute the control load on both MR and FW control variable sets for the best exploitation of the aerodynamic forces. The proposed idea is to gradually prioritize the aerodynamic control surfaces over MR actuators as the speed of the vehicle is increased. The vehicle is represented as a state machine in which each phase corresponds to a state. First MR state is activated to perform vertical flight and hovering. As horizontal speed increases, the second state is initiated and a control system for the transition phase comes into play. A control load placement over the multiple actuators is investigated in order to provide a controlled altitude transition. The control inputs are weighted with a continuous function of velocity and an Incremental Nonlinear Dynamic Inversion (INDI) technique is adopted. At this stage an INDI approach is preferred over the classical NDI as it is less reliant on the mathematical model of the plant, consequently it allows to use a less accurate aerodynamic model. The third state of the state machine (i.e. Fixed Wing flight) is activated at around stall velocity. However the same control logic as the transition phase can be retained, with exception of providing all the required forces and torques through aerodynamic surfaces only.

# Acknowledgements

I would like to express my deepest gratitude to Professors Angelo Lerro and Piero Gili for their guidance, assistance, and unwavering support throughout my academic journey. I also extend my thanks to doctoral student Luca Nanu for the ongoing exchange of ideas and invaluable advice that has enriched my knowledge.

To my mother Irene, father Roberto, and brother Giacomo, I am profoundly grateful for their love and constant encouragement. Without them, I could not have achieved my goals.

I offer my heartfelt thanks to Aurora, who supports and patiently tolerates me every day.

A special appreciation goes to all the people I have crossed paths with on my journey, each of whom, in their own way, has contributed to my presence here today.

*“Where the world ceases to be the scene of our personal hopes and wishes,*
*where we face it as free beings admiring, asking and observing,*
*there we enter the realm of Art and Science.”*
[A. Einstein: The Human Side (1979), p.37-27 January 1921 ]

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

In recent years, significant progress in energy storage, particularly in battery capacities, coupled with the development of Distributed Electric Propulsion (DEP), has paved the way for a new era in transportation known as Urban Air Mobility (UAM). UAM represents a pivotal component of the broader concept of Advanced Air Mobility (AAM), which extends beyond urban areas to encompass rural, regional, and subregional routes. DEP technology is distinct from traditional urban aircraft like helicopters in that it replaces a single large rotor with multiple smaller propellers, offering versatility in design and layout.

The concept of UAM itself has historical roots dating back to the 1980s in Europe, exemplified by services like the Airlink shuttle, which shuttled passengers between Gatwick and Heathrow airports. However, concerns related to noise, pollution, safety, and stringent low-altitude flight regulations led UAM to languish in obscurity for years. Only recently has a renewed interest and optimism been cast upon the field [1].

According to the European Union, the urban population has more than doubled over the past four decades and is projected to reach a staggering 5 billion by 2050 [2]. These statistics raise concerns about the challenges that future cities will face, with traffic congestion and associated emissions being at the forefront. As such, it is imperative to explore innovative transportation solutions to address these impending urban challenges.

The new wave of Advanced Air Mobility (AAM) proposes to use electric vertical take-off and landing (eVTOL) vehicles to exploit a third dimension, i.e. the relatively unused low-altitude airspace for transportation purposes. This could help to solve some of the challenges facing current transportation systems, such as traffic congestion and pollution.

These eVTOLs are designed to be low-noise, low-emission, safe, and cost-effective, positioning them as promising alternatives.

Despite skepticism, the UAM industry is anticipated to reach significant growth by 2050, although technology refinement and regulatory framework development are ongoing challenges [3]. EASA has taken steps to develop a regulatory framework for UAM, including airworthiness regulations for VTOL aircraft, preparations for pilot and operator rules, and the creation of U-Space/UTM regulations for safe UAS integration in urban environments [4].

Currently, there is ongoing research into various configurations of eVTOLs, and most of these configurations can be categorized as follows [5]:

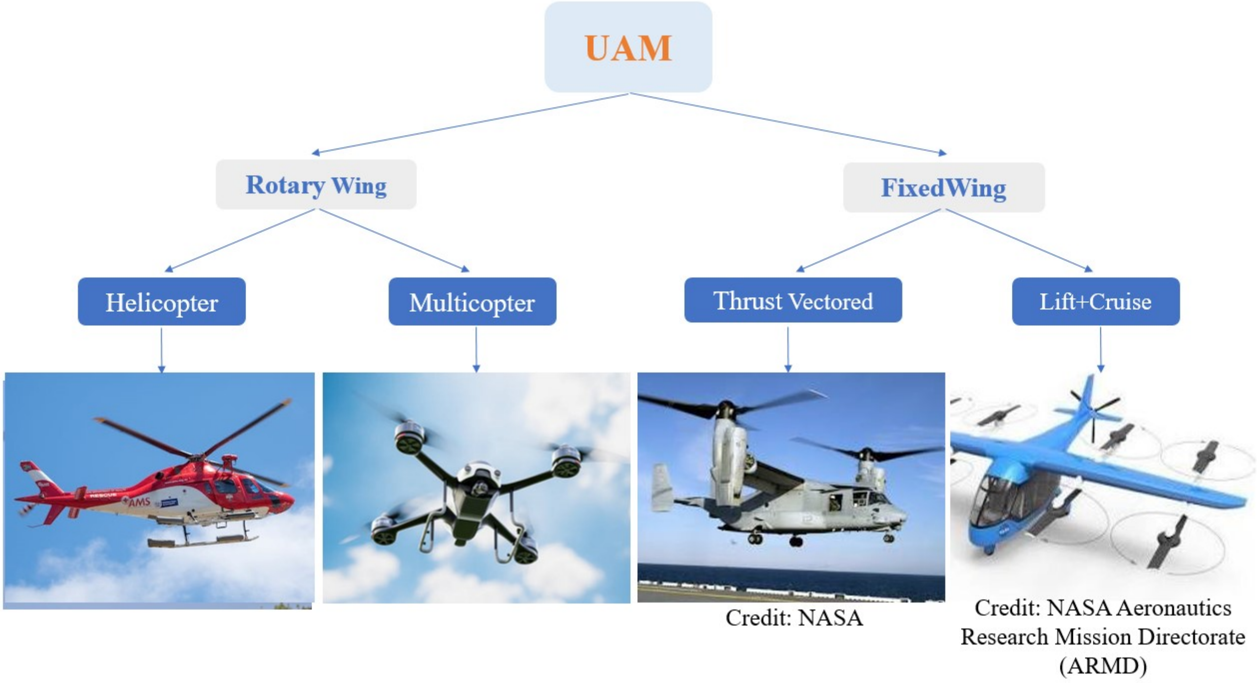


**Figure 1.1:** eVTOLs classification

Multicopters represent an advancement in the world of helicopters, thanks to the application of Distributed Electric Propulsion technology. Their ability to hover and perform vertical flight makes them well-suited for urban environments. Unlike traditional helicopters, DEP technology allows for optimized designs without the need for a large main rotorcraft. However, their primary limitation lies in flight efficiency, affecting their endurance. The absence of lifting surfaces makes them less suitable for extended operations.

This limitation is addressed by Fixed Wing eVTOLs, a category of vehicles that combines vertical flight capabilities with fixed-wing cruising. This enables eVTOLs to cover longer distances. However, the flight dynamics become more complex, particularly in the thrust vectoring subcategory, where the same propeller is used for both thrust and vertical flight, making it challenging to individually optimize each phase of flight.

Independently optimizing each phase becomes achievable in the Lift+Cruise subcategory, where thrust and lift motors are separate. Nevertheless, traditional Lift+Cruise vehicles tend to experience reduced aerodynamic efficiency due to additional aerodynamic drag caused by lift propellers.
To tackle this particular challenge, the forthcoming chapter introduces an innovative solution built upon the patented ThrustPod (TP) technology.

The aim of this thesis project is to develop a control system for an innovative eVTOL configuration that relies on the previously mentioned TP technology. The project unfolds as follows:

- **Chapter 2**: This chapter begins with an overview of the vehicle and delves into its primary control challenges.
- **Chapter 3**: Here, we provide an introduction to the mathematical tools essential for this project's analysis and design.
- **Chapter 4**: We conduct a linear stability analysis of the innovative vehicle, considering various trim velocities.
- **Chapter 5**: This chapter offers a concise explanation of the control theory concepts that will be applied in the design of the control system, a process detailed in Chapter 6.
- **Chapter 6**: In this chapter, we comprehensively design the control system, leveraging the control theory concepts discussed in Chapter 5.
- **Chapter 7/8**: The project culminates in the simulation of two extensive missions, where the developed control system is put to the test.
- **Chapter 9**: The project concludes with a concise project summary and highlighting potential avenues for future developments and research studies in the field.

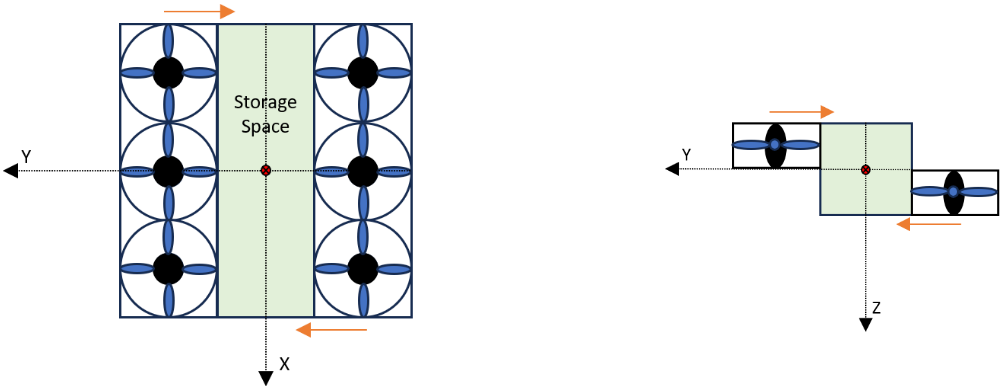
# Chapter 2

# ThrustPod

In this chapter, we introduce the patented solution known as ThrustPod (TP). This innovative aerial platform features a unique design that merges the VTOL capabilities of Multirotor (MR) vehicles with the aerodynamic efficiency of fixed-wing (FW) flight. An essential aspect of this design is to maintain minimal interdependence between the two modes of operation, allowing for independent optimization of each state. To achieve this goal, the design eliminates the mechanical complexity associated with tilting surfaces [6].

The primary objective is to develop a green, fully electric aircraft capable of autonomously completing sub-regional routes within urban environments, thereby contributing to the expansion of Urban Air Mobility (UAM).

The platform's hovering and vertical flight capabilities are provided by six ducted vertical lift propellers, which are exclusively used during VTOL phases. These lift propellers can be fully retracted within the fuselage during other operational phases to prevent aerodynamic performance degradation. The storage of lift motors within the main body is made possible by offsetting along the Z-axis the motors on one side of the longitudinal plane, enabling overlapping when retracted. The rotor extraction and retraction mechanism employs rails, ensuring extremely low energy consumption.

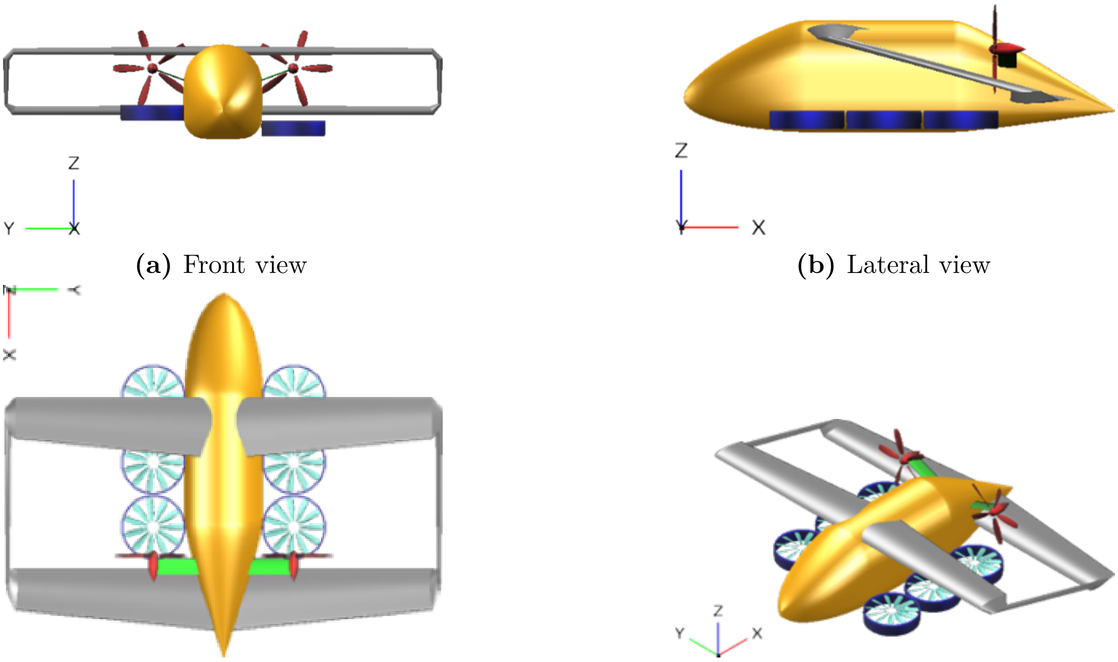


**(a)** Top view

**(b)** Front view

**Figure 2.1:** TP technology

The platform's wing-tail architecture is achieved through a boxed wing design aimed at reducing induced drag [6]. Aerodynamic control surfaces include elevators for pitch control and ailerons for roll control. Additionally, each wing features a horizontal thrust propeller, allowing for yaw control through differential thrust. This design choice eliminates the need for a vertical empennage.



**(a)** Front view

**(b)** Lateral view

**(c)** Top view

**(d)** Isometric view

**Figure 2.2:** eVTOL views

In summary, ThrustPod (TP) presents a pioneering approach to aerial platform design, combining VTOL and fixed-wing capabilities while prioritizing independent optimization and aerodynamic efficiency. The integration of lift propellers and the retraction mechanism contributes to the realization of a green and autonomous aircraft for Urban Air Mobility (UAM) purposes.

## 2.1 Mission overview

One of the primary challenges encountered within the domain of Urban Air Mobility (UAM) revolves around the safe navigation within constricted and confined spaces. This scenario gives rise to two pivotal consequences: the inevitable requirement for a Multirotor (MR) mode during phases involving vertical flight and the imposition of stringent size constraints. As a result of these prescribed limitations, the platform's capacity is constrained to accommodating a maximum of five passengers throughout the subsequent phases:

1. Vertical Takeoff and Hovering: The aircraft must execute a vertical takeoff and maintain a hovering posture until reaching a suitable speed for the transition to fixed-wing (FW) mode.

2. Climb: The aircraft is mandated to accelerate and ascend up to an altitude of 3000 meters, where pressurization is unnecessary.

3. Cruise: The aircraft is obliged to cover the stipulated distance while maintaining a cruising speed of 75 m/s.

4. Descent: The aircraft is instructed to descend to the altitude required for executing a vertical landing.

5. Hovering and Vertical Landing: Upon transitioning back to the Multirotor (MR) mode, the aircraft is obligated to conduct hovering and achieve a vertical landing.

Notably, the transition from Multirotor (MR) to fixed-wing (FW) flight and viceversa entails a shift in control variables. Hence, an intermediary transition phase is indispensable to facilitate a gradual change in the aircraft's state. While the ideal scenario envisions a transition occurring during steady level flight, safety considerations mandate that the aircraft must possess the capability to execute maneuvers during the transition phase as well. Overall, the lift motors represent the most energy-intensive control component, highlighting the need for a quick and smooth transition.

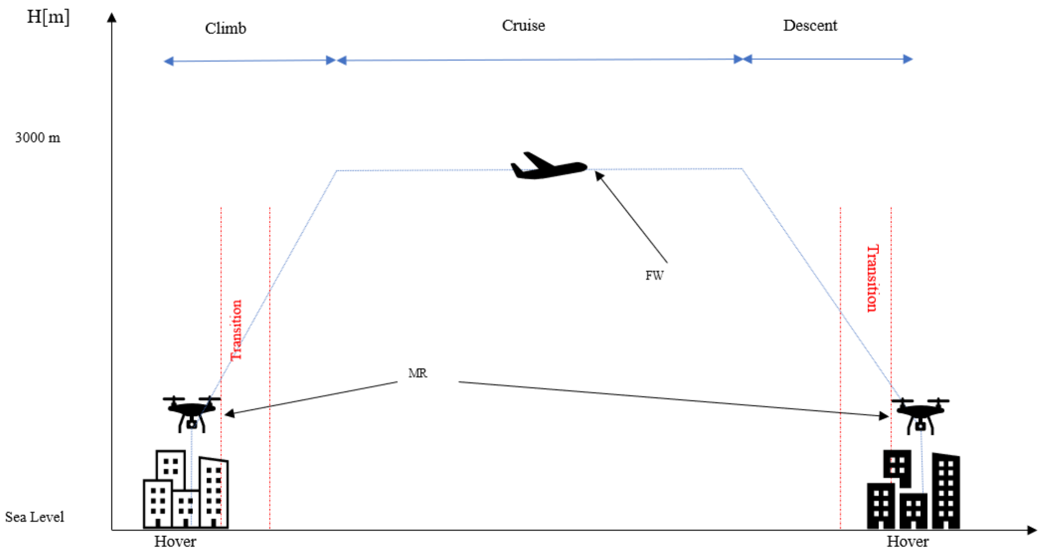


**Figure 2.3:** Mission Profile

## 2.2 Descent and Energy Recovery System

One of the fundamental strengths of the patented system lies in its environmentally friendly design, grounded in a fully electric concept that eliminates emissions. In order to maintain energy sustainability, a high degree of aerodynamic efficiency is essential. Despite being crucial to achieving high efficiency, excessive minimization of intrinsic drag could prevent the system from reaching a sufficiently high flight path angle ($\gamma$) during descent without facing uncontrolled acceleration due to gravitational forces.

In the basic scenario, in absence of additional drag by means of aerobrakes or substitutes, a simplified 2D dynamical model can be employed to assess the maximum permissible $\gamma$ under the assumption of pure longitudinal motion:

$$\begin{cases} Tsin(\alpha) + L - mgcos(\gamma) = 0 & (2.1) \\ Tcos(\alpha) - D - mgsin(\gamma) = 0 & (2.2) \end{cases}$$

A visual representation of the longitudinal flight's free body diagram is depicted in Figure 2.4.

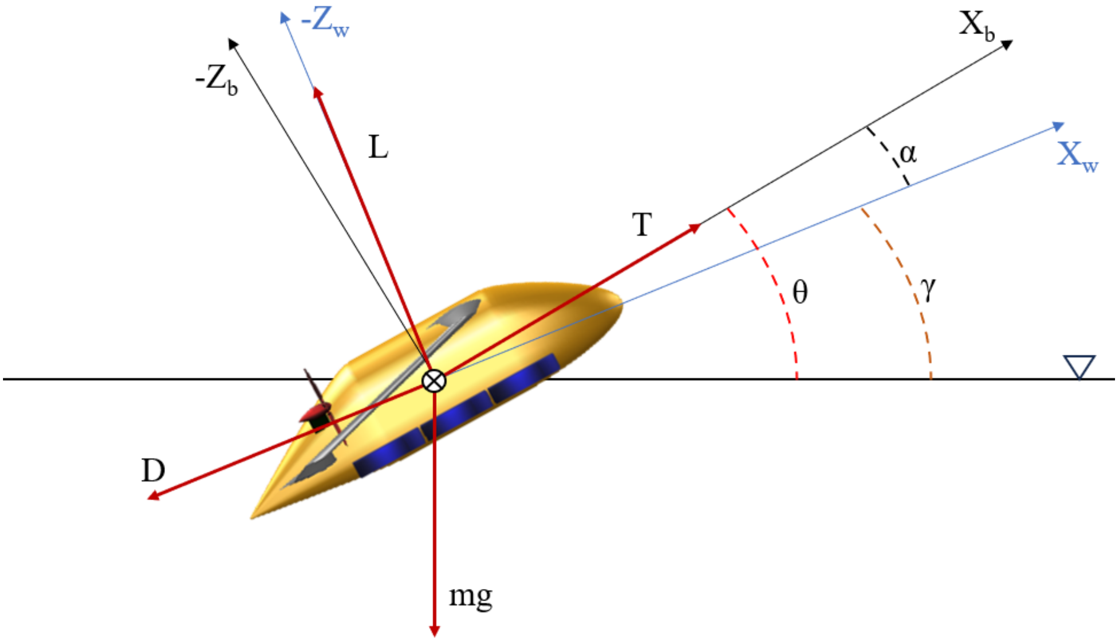


**Figure 2.4:** Free Body diagram for longitudinal flight

The highest descending flight path angle can be determined by setting the thrust term $T$ to zero. Rearranging the equations, the analytical solution for the limiting value of $\gamma$ is given as: :

$$\gamma_{lim} = tan^{-1}(-\frac{D}{L}) = tan^{-1}(-\frac{1}{E}) \tag{2.3}$$

It is crucial to note that this value heavily relies on the vehicle's aerodynamic efficiency. In the nominal case, this value was found to be unacceptably low (less than 1.8°). Any value exceeding this threshold would result in uncontrolled acceleration due to gravity.

To mitigate an excessively prolonged descent phase, the platform must be equipped with a system to increase drag and consequently enhance the descent angle. Two options are available:

1. **Aerobrakes**: This conventional choice enables the aircraft to decelerate and generate the necessary drag for maintaining a steeper descent at a constant speed. However, the energy is dissipated and cannot be reclaimed.

2. **Energy Recovery System**: An alternative approach involves deploying the lift motors to function as aerogenerators, recharging the batteries through the influence of ram air. This approach provides a dual benefit as it partially

converts potential energy loss during descent into electrical energy, enhancing the platform's sustainability, while creating a force opposing gravity.

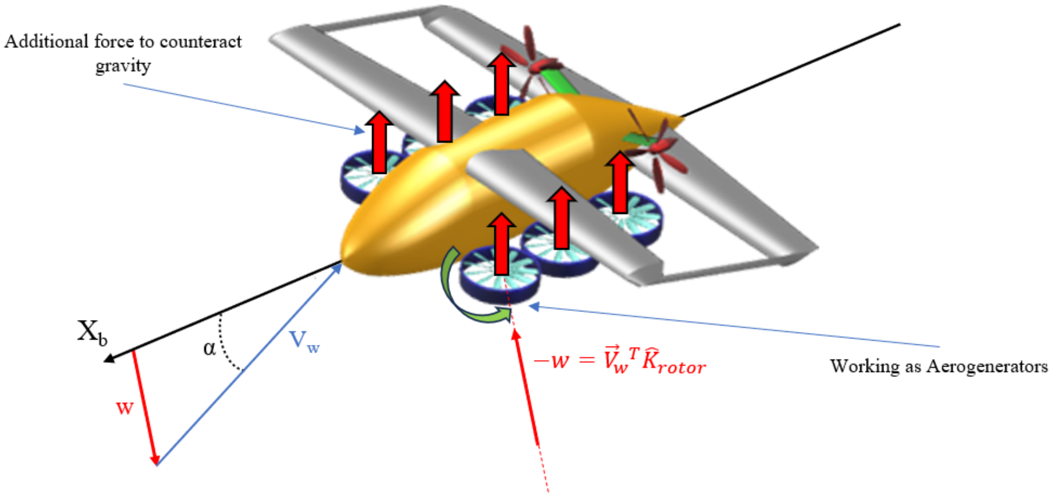

Additional force to counteract gravity
$X_b$
$\alpha$
$V_w$
w
$-w = \vec{V}_w{}^T \hat{R}_{rotor}$
Working as Aerogenerators

**Figure 2.5:** Energy Recovery System

The design of the energy recovery system is beyond the scope of this work and it will require further investigation in the future.

### 2.2.1 State Machine and Controls

The nominal mission profile consists of three main phases, and it is beneficial to view the platform from a discrete state machine perspective in terms of control. Each phase of flight can be associated with a different state of the aircraft, with transition logic governing the progression from one state to the next.

In this context, the system can be discretized based on its forward speed. However, it's essential to consider that lift propellers are highly energy-intensive. Therefore, the goal is to transition to forward fixed-wing flight as early as possible to reduce the weight of the battery. This transition should ideally occur at very low speeds.

One challenge with transitioning early is that below the stall velocity, aerodynamic lift alone is insufficient to keep the vehicle airborne. To address this, an additional state is introduced between FW (Fixed-Wing) and MR (Multicopter) modes, known as Transition or Hybrid mode. This state is essential for facilitating a seamless transition from the VTOL (Vertical Takeoff and Landing) phase to cruising by appropriately allocating control actions based on the flight velocity [7].

The resulting state machine is:

1. Vertical flight phase → Multi-Rotor mode (MR) $\iff u < V_{tr_i}$.

   Active actuators: lift ducted propellers.

2. Transition phase → Transition mode (TR) $\iff V_{tr_i} < u < K_S V_{st} = V_{tr_f}[\frac{m}{s}]$.

   Active actuators: lift ducted propellers, aerodynamic surfaces and thrust propellers.

3. Fixed Wing phase → Fixed-Wing mode (FW) $\iff u > V_{tr_f}$.

   Active actuators: aerodynamic surfaces and thrust propellers.

Where $K_S V_{st}$ is the stall speed of the aircraft multiplied by a safety factor and $V_{tr_i}$, $V_{tr_f}$ are the transition starting and ending velocities, respectively.
The stall speed can be computed as:

$$V_{st} = \sqrt{\frac{2mg}{\rho S C_{L_{Max}}}} \tag{2.4}$$

The Hybrid mode is characterized by the coexistence of control variables from both FW and MR modes. The problem of redundancy is addressed by splitting control actions between each mode based on the current velocity. Assuming that aerodynamic control becomes more effective as the speed relative to the wind increases, we can allocate any requested control action (denoted as $A_C$) as follows:

$$f(V) = \frac{V - V_{tr_i}}{V_{tr_f} - V_{tr_i}} \rightarrow A_C = \underbrace{A_C \cdot f(V)}_{\text{FW actuators}} + \underbrace{A_C \cdot (1 - f(V))}_{\text{MR actuators}} \tag{2.5}$$

The transition function $f(V)$ is constructed to ensure that boundary conditions between contiguous machine states are respected [7]. Indeed, if $V = V_{tri}$, all the control load is placed on MR actuators. Conversely, if $V = Vtr_f$, the control action is entirely generated by FW actuators.
These boundary conditions are crucial for achieving a smooth transition between different operational states of the aircraft.

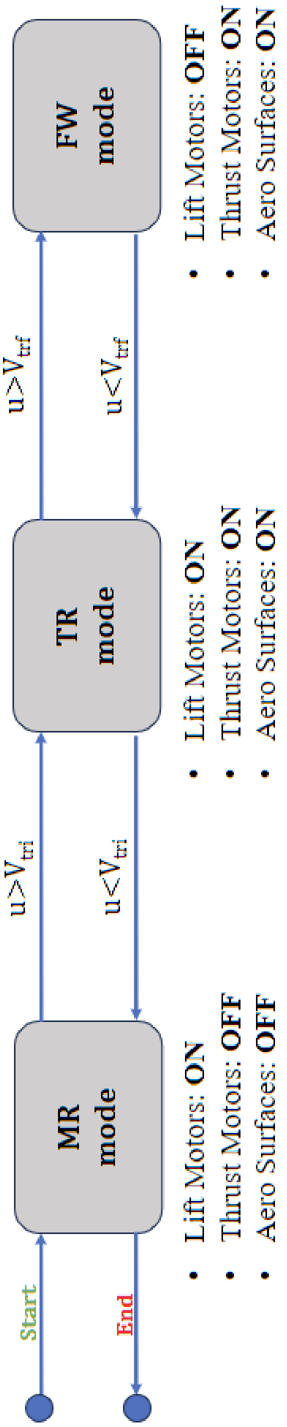


**Figure 2.6:** State machine

# Chapter 3

# Mathematical model

This chapter serves as an introductory exploration into the mathematical model of the aircraft. First a description of the essential reference frames is provided, followed by the derivation of the comprehensive non-linear state-space model. The latter encompasses a detailed description of forces and moments exerted on the system.

## 3.1 Reference Frames and Rotations

To describe the orientation and position of one rigid body relative to another, reference frames are essential. Each reference frame encompasses an origin point and three mutually orthogonal base vectors.

For this discussion, it is beneficial to introduce the following:

1. Inertial frame
2. Body attached frame
3. Wind frame
4. Flight-path frame

### 3.1.1 Inertial Frame

When assuming a flat, non-rotating Earth, the North-East-Down (NED) reference frame can serve as an inertial frame, i.e. Newton's laws hold. Its origin is located at the projection of the vehicle's center of mass (CoM) at time $t_0$ onto the Earth's surface, and its axes are as follows:

- $X_N$: Lies within the plane tangential to the Earth's surface and points toward the North pole.

- $Z_D$: Perpendicular to the tangent plane and parallels the gravity vector; if Earth is considered to be perfectly spherical, the D-axis consistently points to its center.

- $Y_E$: Resides within the tangent plane and follows the right-hand rule.

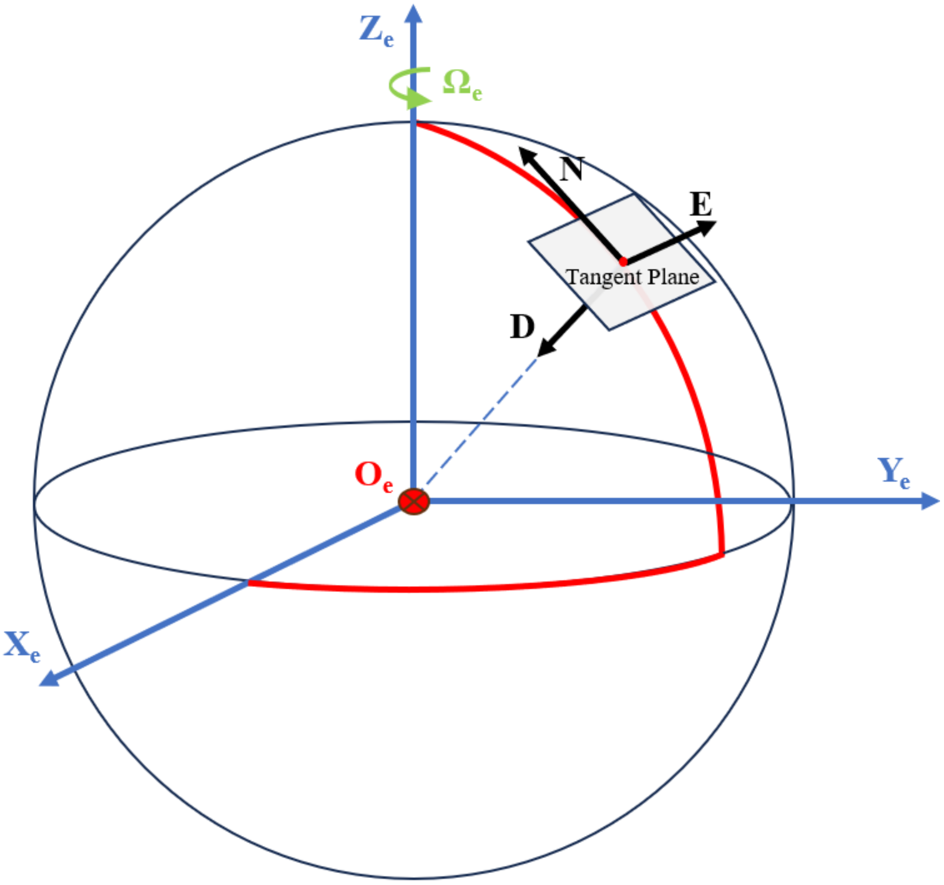


**Figure 3.1:** NED reference frame

It's important to note that the NED frame isn't ideal for accurately describing the inertial position of the vehicle.

### 3.1.2 Body Frame

The body frame remains fixed with the vehicle and is particularly useful for describing the system's mass and geometric attributes as constants over time. It is typically centered at the aircraft's center of mass, and its axes are defined as:

- $X_B$: Exists within the aircraft's longitudinal plane, aligning with the forward direction of motion.;
- $Z_B$: Resides within the aircraft's longitudinal plane, extending toward the underside of the fuselage.;
- $Y_B$: Conventional orientation assumes it extends to the right of an imaginary pilot situated within the vehicle.

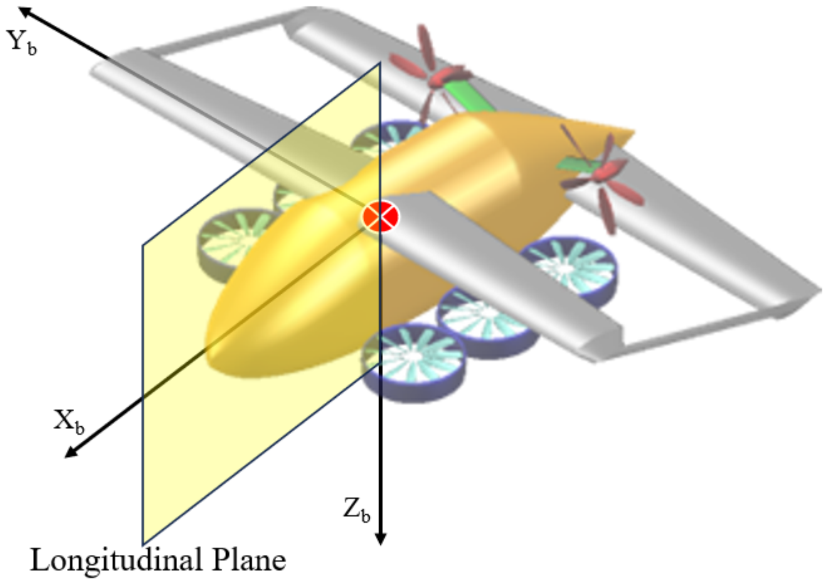


**Figure 3.2:** Body reference frame

### 3.1.3 Wind Frame

The wind frame provides a convenient framework for describing aerodynamic actions on the vehicle:

- $X_W$: Aligned with the direction of relative wind, i.e. the wind vector perceived by the aircraft due to its own motion relative to the surrounding air mass.

- $Z_W$: Runs parallel to the Lift force but in the opposite direction, pointing downwards.
- $Y_W$: Completes the right-handed coordinate system.

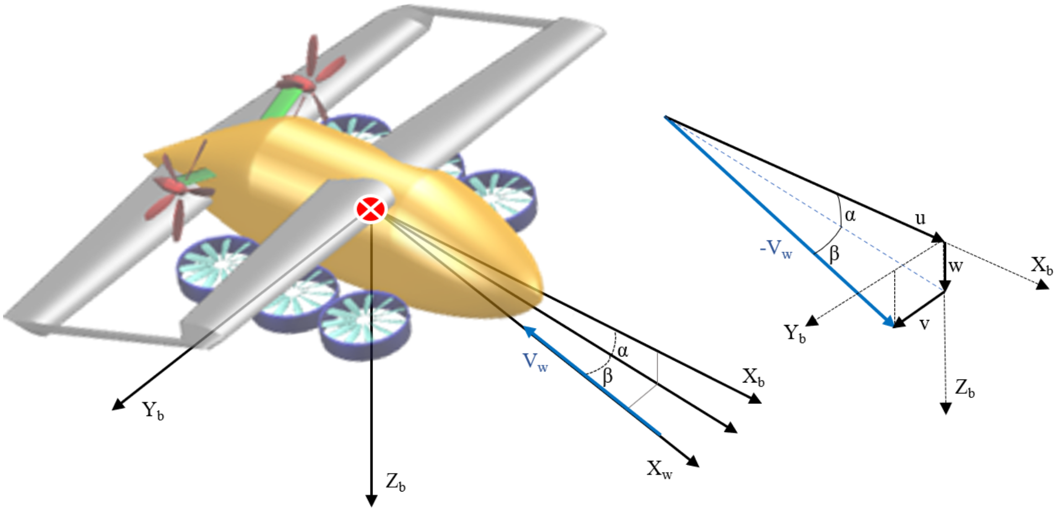


**Figure 3.3:** Wind reference frame

The velocity vector can be written in terms of body velocity components $[u, v, w]^T$ which allow us to define the aerodynamic angles of sideslip ($\beta$) and attack ($\alpha$) as:

$$\alpha = atan(\frac{w}{u}) \tag{3.1}$$

$$\beta = asin(\frac{v}{V_\infty}) \tag{3.2}$$

where $V_\infty = |\boldsymbol{V}_w| = \sqrt{u^2 + v^2 + w^2}$.

### 3.1.4 Flight-path Frame

The flight-path axes are aligned with the velocity vector of the aircraft with respect to the ground. They are used to identify the inertial displacement over time of the CoM of the system.

- $X_{FP}$: Aligned with the direction of the aircraft's velocity vector and pointing in the direction of the aircraft's motion through the air.
- $Y_{FP}$: Runs perpendicular to both the local vertical and the velocity vector.

- $Z_{FP}$: Completes the right-handed coordinate system.

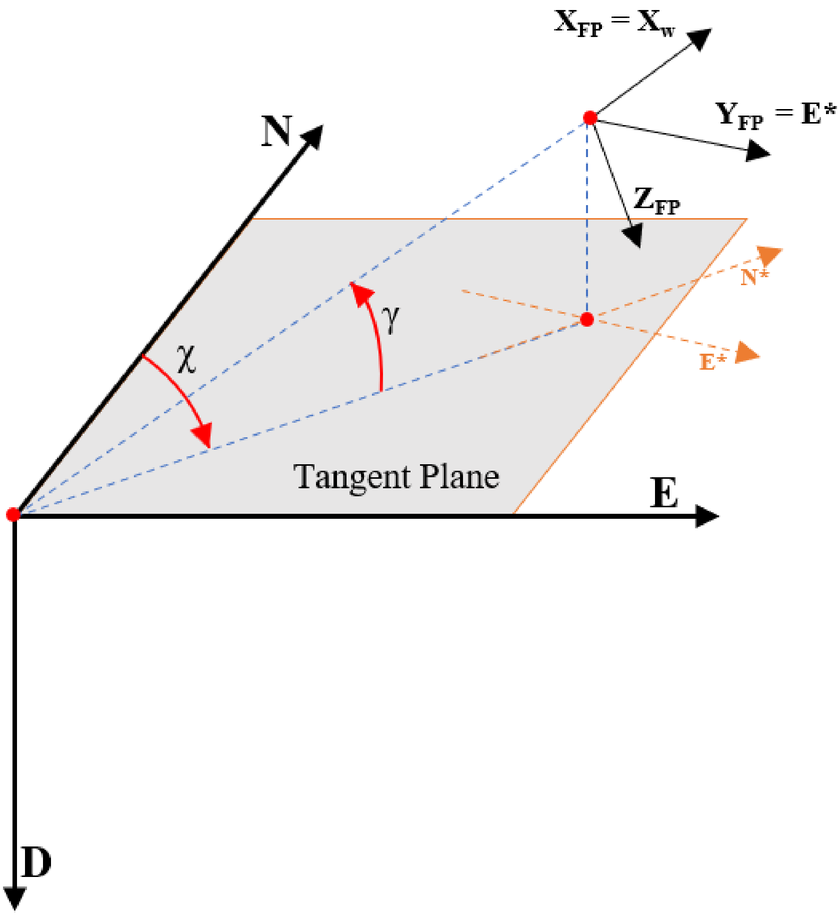


**Figure 3.4:** Flight-path reference frame

This reference frame allows us to introduce two fundamental angles. Defining the inertial velocity vector of the CoM as $[\dot{X}, \dot{Y}, \dot{Z}]^T$ we can define the flight-path angle $\gamma$ and the azimuth angle $\chi$ as:

$$\gamma = acos(\frac{\sqrt{\dot{X}^2 + \dot{Y}^2}}{V_\infty}) \tag{3.3}$$

$$\chi = atan(\frac{\dot{Y}}{\dot{X}}) \tag{3.4}$$

### 3.1.5 Rotations

It is possible to align any two reference frames by applying a sequence of no more than three planar rotations. The order in which these rotations are performed is significant, and according to the Euler convention, the sequence $Z \rightarrow Y \rightarrow X$ is adopted. In this context, a passive perspective of rotation is embraced, wherein Direction Cosine Matrices (DCMs) are employed to describe the same vector in various coordinate systems. Notably, the vector remains fixed in space throughout, and it is only the coordinate systems that change orientation.
When converting a vector's coordinates from NED (North-East-Down) reference frame to Body reference frame, a transformation matrix needs to be applied. This Direction Cosine Matrix (DCM) results from a sequence of rotations aligning the NED frame with the Body frame:

1. $(N, E, D) \rightarrow (N^*, E^*, D)$ through a rotation of $\psi$ about the $D$ axis.
2. $(N^*, E^*, D) \rightarrow (X_B, E^*, D^*)$ through a rotation of $\theta$ about the $E^*$ axis.
3. $(X_B, E^*, D^*) \rightarrow (X_B, Y_B, Z_B)$ through a rotation of $\phi$ about the $X_B$ axis.

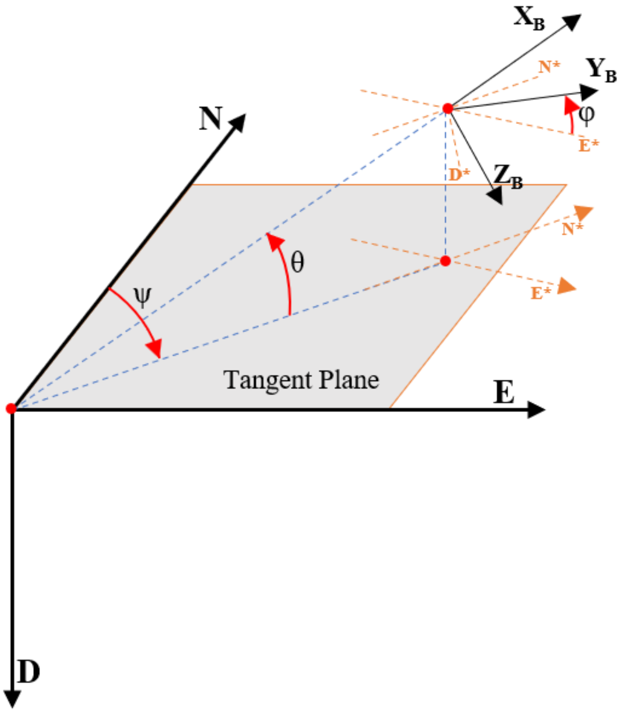


**Figure 3.5:** NED to Body transformation

Each elementary rotation's DCM can be computed through scalar products of the vector bases between the involved frames. The first DCM, about the D axis, is computed as:

$$DCM = \begin{bmatrix} \hat{e}_{N^*} \cdot \hat{e}_N & \hat{e}_{N^*} \cdot \hat{e}_E & \hat{e}_{N^*} \cdot \hat{e}_D \\ \hat{e}_{E^*} \cdot \hat{e}_N & \hat{e}_{E^*} \cdot \hat{e}_E & \hat{e}_{E^*} \cdot \hat{e}_D \\ \hat{e}_D \cdot \hat{e}_N & \hat{e}_D \cdot \hat{e}_E & \hat{e}_D \cdot \hat{e}_D \end{bmatrix} = \begin{bmatrix} c(\psi) & s(\psi) & 0 \\ -s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

where $c(\cdot)$ and $s(\cdot)$ denote the cosine and sine functions respectively. Similarly, the other two elementary DCMs can be computed, resulting in the final transformation from NED to Body:

$$C_{B/N} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & s(\phi) \\ 0 & -s(\phi) & c(\phi) \end{bmatrix} \begin{bmatrix} c(\theta) & 0 & -s(\theta) \\ 0 & 1 & 0 \\ s(\theta) & 0 & c(\theta) \end{bmatrix} \begin{bmatrix} c(\psi) & s(\psi) & 0 \\ -s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.6}$$

The Body coordinates of any vector $\mathbf{V}$ can now be expressed in terms of its NED coordinates and the transformation matrix as follows:

$$\mathbf{V}^B = C_{B/N}(\phi, \theta, \psi)\mathbf{V}^N \tag{3.7}$$

A similar process can be applied to derive the transformation matrices from Body to Wind and from NED to Flight-path axes:

$$C_{W/B} = \begin{bmatrix} c(\beta) & s(\beta) & 0 \\ -s(\beta) & c(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c(\alpha) & 0 & s(\alpha) \\ 0 & 1 & 0 \\ -s(\alpha) & 0 & c(\alpha) \end{bmatrix} \tag{3.8}$$

$$C_{FP/N} = \begin{bmatrix} c(\gamma) & 0 & -s(\gamma) \\ 0 & 1 & 0 \\ s(\gamma) & 0 & c(\gamma) \end{bmatrix} \begin{bmatrix} c(\chi) & s(\chi) & 0 \\ -s(\chi) & c(\chi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.9}$$

The following properties for DCMs hold and will be used later on:

- $C_{A/B}^{-1} = C_{A/B}^T = C_{B/A}$
- $C_{A/C} = C_{A/B}C_{B/C}$

## 3.2 Kinematic and Dynamic equations

The motion of a rigid body can be seen as the combination of the CoM translation and the attitude change due to its rotation about the CoM itself. The complete set of equations will therefore include both rotational and translational kinematics and dynamics.

### 3.2.1 Kinematics

The linear kinematics emerges directly from the mathematical connection between the NED and Body reference frames. The initial set of equations, commonly referred to as *navigation equations*, is a direct result of this relationship:

$$\dot{\mathbf{P}}^N = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix}^N = C_{N/B}(\phi, \theta, \psi)\mathbf{V}^B = C_{N/B}(\phi, \theta, \psi) \begin{bmatrix} u \\ v \\ w \end{bmatrix}^B \tag{3.10}$$

The kinematics governing rotational motion can be derived by employing the chain rule for angular velocities. According to this rule, when dealing with N distinct reference frames, the angular velocity of frame N with respect to frame 1 can be obtained through the summation of angular velocities as follows:

$$\boldsymbol{\omega}_{N/1} = \boldsymbol{\omega}_{N/N-1} + ... + \boldsymbol{\omega}_{j/j-1} + ... + \boldsymbol{\omega}_{2/1} \tag{3.11}$$

In this equation, each term $\boldsymbol{\omega}_{j/j-1}$ represents the angular velocity of frame $j$ with respect to frame $j-1$. Applying such rule, the link between body angular rates and Euler rates can be found as:

$$\dot{\mathbf{\Phi}} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s(\phi)t(\theta) & c(\phi)t(\theta) \\ 0 & c(\phi) & -s(\phi) \\ 0 & \frac{s(\phi)}{c(\theta)} & \frac{c(\phi)}{c(\theta)} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = T(\mathbf{\Phi})\boldsymbol{\omega}^B_{B/N} \tag{3.12}$$

where $t(\cdot)$ stands for tangent function and $\boldsymbol{\omega}^B_{B/N}$ is the angular velocity vector of the Body frame w.r.t. the inertial one in body components.

### 3.2.2 Dynamics

The effect of moments and forces acting on the system is mathematically described by the *flight dynamics equations.* The linear equations of motion relate the inertial acceleration of the system CoM to the external forces exerted onto it and can be derived by *Newton's second law*:

$$\frac{d(m\boldsymbol{V})}{dt} = \sum \boldsymbol{F}_{ext} \tag{3.13}$$

It is convenient to take the time derivative of the velocity vector in the same frame that we intend to express the equation in. Leveraging the time derivative transport theorem for vectors, the resulting matrix equation in Body coordinates is:

$$^B\frac{d}{dt}\boldsymbol{V}^B + \boldsymbol{\omega}^B_{B/N} \times \boldsymbol{V}^B = \frac{1}{m}\sum \boldsymbol{F}^B_{ext} \tag{3.14}$$

Likewise, the rotational dynamics equations establish a connection between the rate of change of angular momentum of the system with respect to the CoM and the external moments exerted on the system:

$$\frac{d(\boldsymbol{H})}{dt} = \sum \boldsymbol{M}_{ext} \tag{3.15}$$

Here, the angular momentum incorporates the inertia tensor of the rigid body. The description of rotational dynamics is also presented in Body coordinates, which conveniently allows us to assume that the inertia matrix remains constant over time. Expanding equation (3.15) we obtain:

$$^{B}\frac{d}{dt}\boldsymbol{H}^{B} + \boldsymbol{\omega}^{B}_{B/N} \times \boldsymbol{H}^{B} = \sum \boldsymbol{M}^{B}_{ext} \tag{3.16}$$

If rotors are included in the analysis, then the angular momentum of the whole system becomes:

$$\boldsymbol{H}^{B} = J^{B}\boldsymbol{\omega}^{B}_{B/N} + J^{B}_{R}\boldsymbol{\omega}^{B}_{R/B} \tag{3.17}$$

where $J^B$ is the inertia matrix of the whole system including rotors at rest, whereas $J^B_R$ is the inertia matrix of the rotors. The term $\boldsymbol{\omega}^B_{R/B}$ represents the angular velocity of the rotors w.r.t. the body fixed frame. Everything is expressed in body coordinates. Substituting (3.17) into (3.16) the resulting expression for the gyroscopic effect is: $\boldsymbol{G}^B = \boldsymbol{\omega}^B_{B/N} \times J^B_R\boldsymbol{\omega}^B_{R/B}$ [8].

## 3.3 Forces and Moments

In order for the set of equations previously derived to be useful, a model of forces and moments is needed. The actions exerted on the system can be categorized into two main groups:

1. Aerodynamic
2. Propulsive

The second category can be further subdivided based on whether they originate from thrust or lift propellers.

### 3.3.1 Aerodynamics

In this dissertation a linear model of the vehicle aerodynamics is included. This means that actions on the aircraft caused by its interaction with the airflow are assumed to be first order polynomials in control and state variables.

Conveniently, forces are given in wind axes components, whereas moments are described in the body frame.

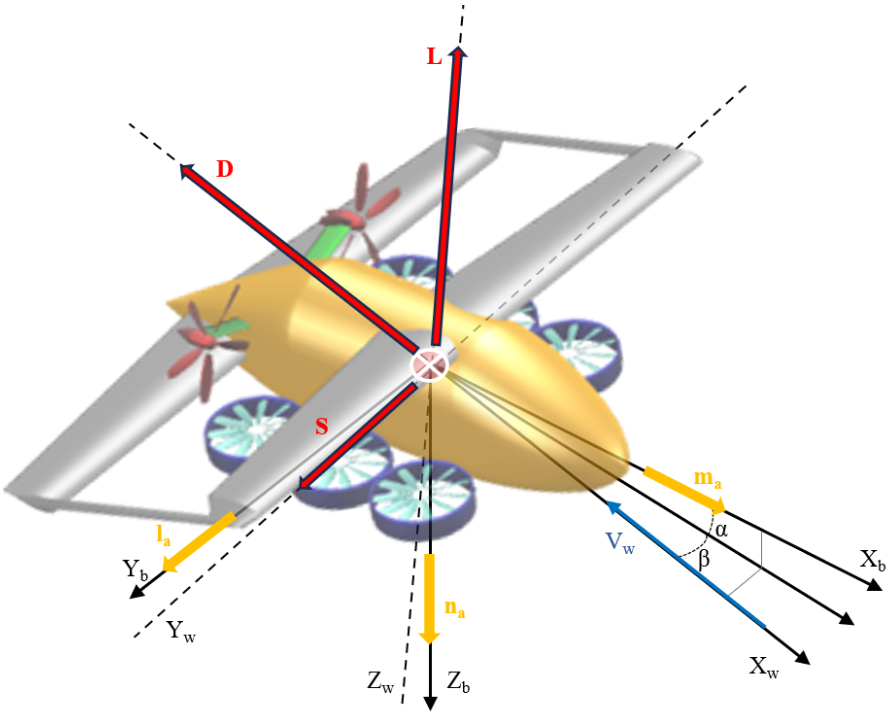


**Figure 3.6:** Aerodynamic actions

Defining the ram air dynamic pressure as $q = \frac{1}{2}\rho V^2$, each aerodynamic force component can be expressed using dimensionless coefficients that encompass both the influence of state ($\boldsymbol{x}$) and control variables ($\boldsymbol{u}$). Additionally, in the linear case, the effect of $\boldsymbol{u}$ can be entirely distinguished from the dependence from $\boldsymbol{x}$, resulting in:

$$\boldsymbol{F}_A^W = \begin{bmatrix} -D \\ S \\ -L \end{bmatrix} = \boldsymbol{F}_{A_x}^W(\boldsymbol{x}) + \boldsymbol{F}_{A_u}^W(\boldsymbol{u}) \tag{3.18}$$

Here, for simplicity's sake, the steady-state coefficients are incorporated into the state dependency, leading to:

$$\boldsymbol{F}^W_{A_x} = qS \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} C_{D_{ss}} & C_{D_\alpha} & C_{D_\beta} & C_{D_p} & C_{D_q} & C_{D_r} & C_{D_u} \\ C_{S_{ss}} & C_{S_\alpha} & C_{S_\beta} & C_{S_p} & C_{S_q} & C_{S_r} & C_{S_u} \\ C_{L_{ss}} & C_{L_\alpha} & C_{L_\beta} & C_{L_p} & C_{L_q} & C_{L_r} & C_{L_u} \end{bmatrix} \begin{bmatrix} 1 \\ \alpha \\ \beta \\ p \\ q \\ r \\ V \end{bmatrix} \tag{3.19}$$

$$\boldsymbol{F}^W_{A_u} = qS \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -C_{L_{\delta_e}} & 0 \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_a \end{bmatrix} \tag{3.20}$$

The same applies to aerodynamic moments that can be written as:

$$\boldsymbol{M}^B_A = \begin{bmatrix} l_a \\ m_a \\ n_a \end{bmatrix} = \boldsymbol{M}^B_{A_x}(\boldsymbol{x}) + \boldsymbol{M}^B_{A_u}(\boldsymbol{u}) \tag{3.21}$$

where:

$$\boldsymbol{M}^B_{A_x} = qS \begin{bmatrix} b & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & b \end{bmatrix} \begin{bmatrix} C_{l_{ss}} & C_{l_\alpha} & C_{l_\beta} & C_{l_p} & C_{l_q} & C_{l_r} & C_{l_u} \\ C_{m_{ss}} & C_{m_\alpha} & C_{m_\beta} & C_{m_p} & C_{m_q} & C_{m_r} & C_{m_u} \\ C_{n_{ss}} & C_{n_\alpha} & C_{n_\beta} & C_{n_p} & C_{n_q} & C_{n_r} & C_{n_u} \end{bmatrix} \begin{bmatrix} 1 \\ \alpha \\ \beta \\ p \\ q \\ r \\ V \end{bmatrix} \tag{3.22}$$

$$\boldsymbol{M}^W_{A_u} = qS \begin{bmatrix} b & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & b \end{bmatrix} \begin{bmatrix} 0 & C_{l_{\delta_a}} \\ C_{m_{\delta_e}} & 0 \\ 0 & C_{n_{\delta_a}} \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_a \end{bmatrix} \tag{3.23}$$

The terms $b, c, S$ correspond, in order, to wingspan, aerodynamic mean chord and wing planform area.

### 3.3.2 Propulsion

Forces and torques generated from a propeller can be approximately modelled using Renard's formulas. Defining $\Gamma = \frac{V}{\omega R_e}$ as the grading ratio, the coventional form of the equations is given as:

$$T = \rho R_e^4 \tau(\Gamma)\omega^2 \tag{3.24}$$

$$Q = \rho R_e^5 K(\Gamma)\omega^2 \tag{3.25}$$

In these equations, $T$ and $Q$ represent the thrust and reaction moment generated by the propeller, respectively. They depend on air density $\rho$, blade radius $R_e$, propeller angular speed $\omega$ and the Renard's coefficients $\tau$ and $K$. A common approach is to determine constant values for these coefficients using a least squares fit from experimental data [9],[10]. A simpler formulation is then achieved as:

$$T = K_T\omega^2 \tag{3.26}$$

$$Q = K_Q\omega^2 \tag{3.27}$$

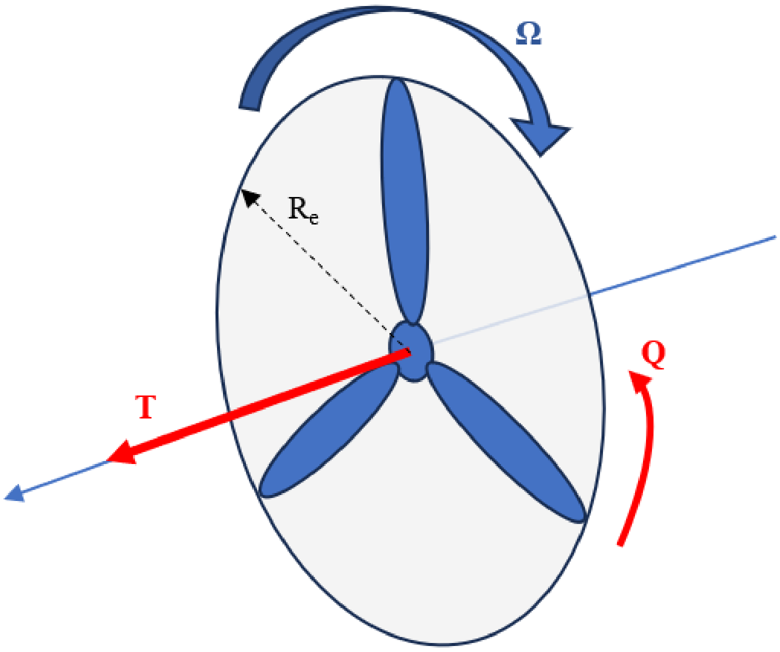


**Figure 3.7:** Propeller actions

### Lifting Propellers

Each propeller acts on the system with a torque and thrust as described in the previous subsection. If we combine all the effects together we can obtain a mathematical equation for the overall effect of the lift propellers on the aircraft.

For the sake of simplicity we can make the following assumptions:

1. The thrust vectors are anti-parallel w.r.t. the $Z_b$ axis.
2. The distance between the rotation axes of the middle motors along $X_b$ and the CoM is equal to zero.

As a result of 1, the net force of the motors will act along $Z_b$ only. Additionally, no moment is produced along the same axis due to differential thrusts and the only rotational effect on $Z_b$ is caused by the reaction torques.

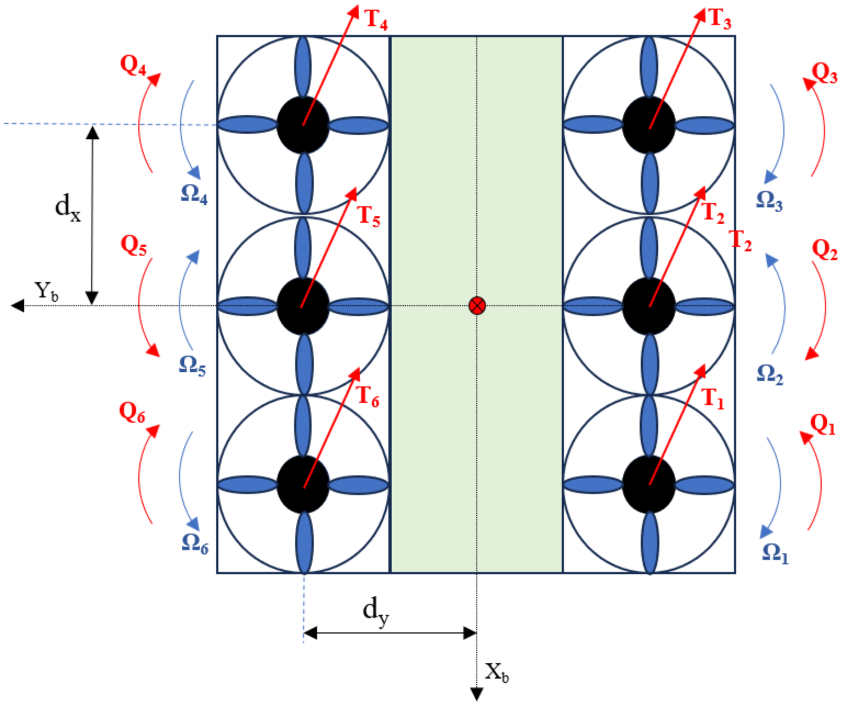


**Figure 3.8:** Vertical propellers' actions

The resulting expression for the net force in body components is given by:

$$\boldsymbol{F}_{T_v}^{B} = \begin{bmatrix} 0 \\ 0 \\ -\sum_{i=1}^{6} T_i \end{bmatrix} = -K_{T_v} \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^{6} \Omega_{v_i}^2 \end{bmatrix} \tag{3.28}$$

Taking into consideration the moment components generated by differential thrust, we can derive the following expression for the net moment exerted on the system:

$$\boldsymbol{M}_{T_v}^{B} = \begin{bmatrix} K_{T_v} d_{y_v} (\sum_{i=1}^{3} \Omega_{v_i}^2 - \sum_{i=4}^{6} \Omega_{v_i}^2) \\ K_{T_v} d_{x_v} (\Omega_{v1}^2 + \Omega_{v6}^2 - \Omega_{v3}^2 - \Omega_{v4}^2) \\ K_{Q_v} \sum_{i=1}^{6} (-1)^{2i-1} \Omega_{v_i}^2 \end{bmatrix} \tag{3.29}$$

When merging the expressions for forces and moments into a single matrix equation, we obtain the following result [10]:

$$\begin{bmatrix} T_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} K_{T_v} & 0 & 0 & 0 \\ 0 & K_{T_v} d_{y_v} & 0 & 0 \\ 0 & 0 & K_{T_v} d_{x_v} & 0 \\ 0 & 0 & 0 & K_{Q_v} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & -1 & -1 & 0 & 1 \\ 1 & 1 & 1 & -1 & -1 & -1 \\ -1 & 1 & -1 & 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} \Omega_{v1}^2 \\ \Omega_{v2}^2 \\ \Omega_{v3}^2 \\ \Omega_{v4}^2 \\ \Omega_{v5}^2 \\ \Omega_{v6}^2 \end{bmatrix} \tag{3.30}$$

### Thrust Propellers

Everything that was stated for the lifting propellers also holds for the thrust ones. In this case, however, the net force is assumed to be aligned with $X_b$.

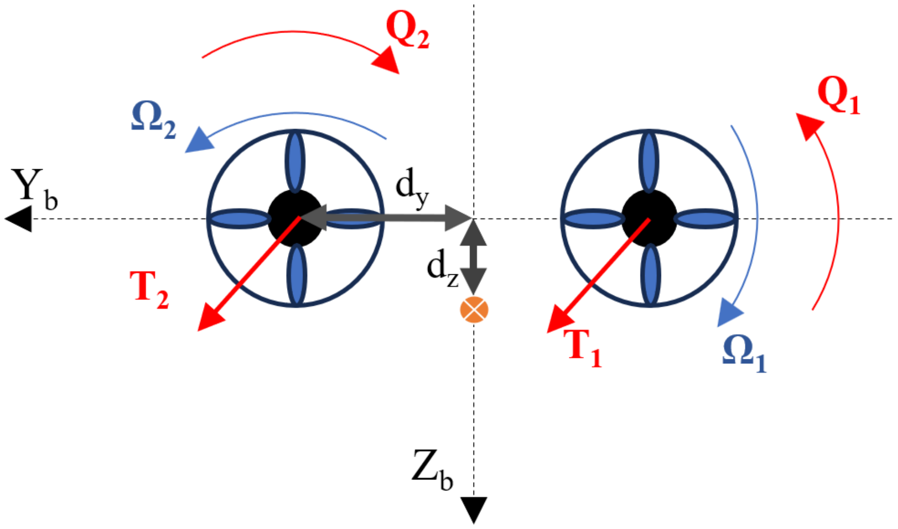


**Figure 3.9:** Horizontal propellers' actions

The resulting matrix equation for forces and moments is:

$$\begin{bmatrix} T_x \\ l_T \\ m_T \\ n_T \end{bmatrix} = \begin{bmatrix} K_{T_h} & 0 & 0 & 0 \\ 0 & K_{Q_h} & 0 & 0 \\ 0 & 0 & K_{T_h} d_{z_h} & 0 \\ 0 & 0 & 0 & K_{T_h} d_{y_h} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ -1 & -1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \Omega_{h_1}^2 \\ \Omega_{h_2}^2 \end{bmatrix} \tag{3.31}$$

The overall propulsive effect on the system will be a function of the control vector $\boldsymbol{u}$ only and can be expressed in body components as:

$$\boldsymbol{F}_T^B(\boldsymbol{u}) = \begin{bmatrix} T_x(\boldsymbol{u}) \\ 0 \\ -T_z(\boldsymbol{u}) \end{bmatrix} \tag{3.32}$$

$$\boldsymbol{M}_T^B(\boldsymbol{u}) = \begin{bmatrix} \tau_x(\boldsymbol{u}) \\ \tau_y(\boldsymbol{u}) \\ \tau_z(\boldsymbol{u}) \end{bmatrix} + \begin{bmatrix} l_T(\boldsymbol{u}) \\ m_T(\boldsymbol{u}) \\ n_T(\boldsymbol{u}) \end{bmatrix} \tag{3.33}$$

In addition, the internal gyroscopic moment due to rotating parts will appear as:

$$\boldsymbol{G}^B(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} ( \begin{bmatrix} J_{R_h} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & J_{R_v} \end{bmatrix} \begin{bmatrix} \Omega_{h_2} - \Omega_{h_1} \\ 0 \\ \sum_{i=1}^{6} (-1)^{i-1} \Omega_{v_i} \end{bmatrix} ) \tag{3.34}$$

### 3.3.3 Non Linear State Space Model

The equations obtained earlier can be integrated into a system of non-linear first-order ordinary differential equations involving both the state ($\boldsymbol{x}$) and control ($\boldsymbol{u}$) variables. With the initial conditions $\boldsymbol{x}(t_0)$ and the time history of inputs $\boldsymbol{u}$, we can determine the complete state of the system for any $t > t_0$. This mathematical framework is not only useful for predicting system behavior but also lends itself well to control applications.

$$\begin{cases} \dot{\boldsymbol{V}}^B = -\boldsymbol{\omega}^B_{B/N} \times \boldsymbol{V}^B + \dfrac{1}{m}(\boldsymbol{F}^B_T + C_{B/N}(m\boldsymbol{g}^N) + C_{B/W}\boldsymbol{F}^W_A) \\ \dot{\boldsymbol{\Phi}} = T(\boldsymbol{\Phi})\boldsymbol{\omega}^B_{B/N} \\ \dot{\boldsymbol{\omega}}^B_{B/N} = (\mathbf{J}^B)^{-1}(-\boldsymbol{\omega}^B_{B/N} \times \mathbf{J}^B\boldsymbol{\omega}^B_{B/N} + \boldsymbol{M}^B_A + \boldsymbol{M}^B_T - \boldsymbol{G}^B) \\ \dot{\boldsymbol{P}}^N = C_{N/B}\boldsymbol{V}^B \end{cases} \tag{3.35}$$

This system consists of 12 first-order non-linear ordinary differential equations involving the following state and control variables:

$$\boldsymbol{x}^T = [\underbrace{u, v, w}_{\boldsymbol{x}_1^T}, \underbrace{\phi, \theta, \psi}_{\boldsymbol{x}_2^T}, \underbrace{p, q, r}_{\boldsymbol{x}_3^T}, \underbrace{P_N, P_E, P_D}_{\boldsymbol{x}_4^T}] \in \mathbb{R}^{12\times 1} \tag{3.36}$$

$$\boldsymbol{u}^T = [\underbrace{\delta_e, \delta_a}_{\boldsymbol{u}_{FW}^T}, \underbrace{\Omega_{h_1}, \Omega_{h_2}}_{\boldsymbol{u}_{HT}^T}, \underbrace{\Omega_{v_1}, \Omega_{v_2}, \Omega_{v_3}, \Omega_{v_4}, \Omega_{v_5}, \Omega_{v_6}}_{\boldsymbol{u}_{VT}^T}] \in \mathbb{R}^{10\times 1} \tag{3.37}$$

For conciseness of notation we may define the following subsets for both $\boldsymbol{x}$ and $\boldsymbol{u}$ as:

$$\boldsymbol{x}^T = [\boldsymbol{x}_1^T, \boldsymbol{x}_2^T, \boldsymbol{x}_3^T, \boldsymbol{x}_4^T] \tag{3.38}$$

$$\boldsymbol{u}^T = [\boldsymbol{u}_{FW}^T, \boldsymbol{u}_{HT}^T, \boldsymbol{u}_{VT}^T] \tag{3.39}$$

With this notation, we can rewrite the state equations as follows:

$$\begin{cases} \dot{\boldsymbol{x}}_1 = -\boldsymbol{x}_3 \times \boldsymbol{x}_1 + C_{B/N}(\boldsymbol{x}_2)\boldsymbol{g}^N + \dfrac{1}{m}(C_{B/W}(\boldsymbol{x}_1)\boldsymbol{F}^W_A(\boldsymbol{x}_1, \boldsymbol{x}_3, \boldsymbol{u}_{FW}) + \boldsymbol{F}^B_T(\boldsymbol{u}_{HT}, \boldsymbol{u}_{VT})) \\ \dot{\boldsymbol{x}}_2 = T(\boldsymbol{x}_2)\boldsymbol{x}_3 \\ \dot{\boldsymbol{x}}_3 = (\mathbf{J}^B)^{-1}(-\boldsymbol{x}_3 \times \mathbf{J}^B\boldsymbol{x}_3 + \boldsymbol{M}^B_A(\boldsymbol{x}_1, \boldsymbol{x}_3, \boldsymbol{u}_{FW}) + \boldsymbol{M}^B_T(\boldsymbol{u}_{HT}, \boldsymbol{u}_{VT}) - \boldsymbol{G}^B(\boldsymbol{x}_3, \boldsymbol{u}_{HT}, \boldsymbol{u}_{VT})) \\ \dot{\boldsymbol{x}}_4 = C_{N/B}(\boldsymbol{x}_2)\boldsymbol{x}_1 \end{cases} \tag{3.40}$$

Finally, the system can be represented in the standard form as:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{3.41}$$

It's worth noting that due to the presence of gyroscopic effects, even with a simplified linear aerodynamic model, the system is not in control-affine form, which means it's not linear in the control variables. If we were to disregard the gyroscopic term and use the squared rotor speeds as inputs, the system could be rewritten in a control-affine form:

$$\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} \tag{3.42}$$

Typically, the state equation is accompanied by an output equation denoted as $\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u})$. Here, $\boldsymbol{y}$ denotes the system's output, which includes a collection of variables that can be measured and used for feedback purposes. In the context of this dissertation, we consider the entire state vector to be accessible for feedback. However, in practical scenarios, an estimation algorithm becomes essential to compute state variables that are not directly measurable.

### 3.3.4 Linear State Space Model

Most of control theory and stability analysis apply to linear systems only. Usually, non linear systems are linearized about an equilibrium point and the stability of the system is assessed for small perturbations only.

In order to linearize the system, we first need to find an equilibrium point by setting the derivative of the state to zero and solving a set of non linear algebraic equations:

$$(\boldsymbol{x}^*, \boldsymbol{u}^*) = (\boldsymbol{x}, \boldsymbol{u})_{eq} \Leftrightarrow \dot{\boldsymbol{x}}^* = \boldsymbol{f}(\boldsymbol{x}^*, \boldsymbol{u}^*) = 0 \tag{3.43}$$

After an equilibrium point is found, we can take the first order Taylor expansion of the state equations as $\boldsymbol{x}$ and $\boldsymbol{u}$ approach their equilibrium values:

$$\dot{\boldsymbol{x}} = \nabla_x \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})_{(x,u)_{eq}} \delta\boldsymbol{x} + \nabla_u \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})_{(x,u)_{eq}} \delta\boldsymbol{u} \quad as \quad \dot{\boldsymbol{x}} \to \dot{\boldsymbol{x}}_{eq} = 0 \tag{3.44}$$

The state and control Jacobian matrices are commonly written as $\mathbb{A}$ and $\mathbb{B}$, whereas $\delta(\cdot)$ indicates a small perturbation from equilibrium.
The standard form for the linear state space representation is:

$$\delta\dot{\boldsymbol{x}} = \mathbb{A}\delta\boldsymbol{x} + \mathbb{B}\delta\boldsymbol{u} \tag{3.45}$$

For a system to exhibit linear stability around an equilibrium point, it's necessary for the real components of all eigenvalues in the matrix $\mathbb{A}$ to be negative. This principle is particularly applicable when analyzing an aircraft's dynamics, which can be divided into two separate planes of motion: longitudinal and lateral-directional. When considering linear stability,indeed, these two planes can be assessed independently. In the context of longitudinal dynamics, an additional factor comes into play due to the inclusion of vertical lift propellers as opposed to conventional FW aircrafts.

For the longitudinal dynamics the $\mathbb{A}$ matrix becomes:

$$
\begin{aligned}
A_{11} &= -\rho \cdot V_{eq} \cdot S \cdot (C_{D_{eq}} + 0.5 \cdot V_{eq} \cdot C_{D_V}) \\
A_{12} &= -T_{x_{eq}} \cdot \sin(\alpha_{eq}) + m \cdot g \cdot \cos(\gamma_{eq}) - q_{eq} \cdot S \cdot C_{D_\alpha} - T_{z_{eq}} \cdot \cos(\alpha_{eq}) \\
A_{13} &= -q_{eq} \cdot S \cdot (C_{D_q} \cdot c/(2 \cdot V_{eq})) \\
A_{14} &= -m \cdot g \cdot \cos(\gamma_{eq}) \\
A_{21} &= -\rho \cdot V_{eq} \cdot S \cdot (C_{L_{eq}} + 0.5 \cdot V_{eq} \cdot C_{L_V}) \\
A_{22} &= -T_{x_{eq}} \cdot \cos(\alpha_{eq}) + m \cdot g \cdot \sin(\gamma_{eq}) - q_{eq} \cdot S \cdot C_{L_\alpha} + T_{z_{eq}} \cdot \sin(\alpha_{eq}) \\
A_{23} &= m \cdot V_{eq} - q_{eq} \cdot S \cdot (C_{L_q} \cdot c/(2 \cdot V_{eq})) \\
A_{24} &= -m \cdot g \cdot \sin(\gamma_{eq}) \\
A_{31} &= \rho \cdot V_{eq} \cdot S \cdot c \cdot (C_{m_{eq}} + 0.5 \cdot V_{eq} \cdot C_{m_V}) \\
A_{32} &= q_{eq} \cdot S \cdot c \cdot C_{m_\alpha} \\
A_{33} &= q_{eq} \cdot S \cdot c \cdot (C_{m_q} \cdot c/(2 \cdot V_{eq})) \\
A_{34} &= 0 \\
A_{4:} &= [0, 0, 1, 0]
\end{aligned}
$$

The $A_{ij}$ components are derived from the linearization of the state equations in Wind Axes, where the state vector becomes:

$$\boldsymbol{x}_{long}^T = [V, \alpha, q, \theta] \tag{3.46}$$

As vertical thrust solely aligns with $-\hat{\boldsymbol{k}}_b$, the supplementary term remains absent in the lateral-directional state matrix. Consequently, the standard configuration commonly found in literature for $\mathbb{A}_{lat}$ can be maintained, and the state vector is defined as:

$$\boldsymbol{x}_{lat}^T = [\beta, \phi, p, r] \tag{3.47}$$

It's essential to acknowledge that Wind Axes might not be the most suitable framework for analyzing the dynamics of the MR (Multi Rotor) mode. However, this dissertation will not include a linear stability analysis of such mode.

### 3.3.5 Numerical Data

In the tables below all the coefficients needed for a numerical model of the aircraft are listed:

| | ss | $\alpha$ | $\beta$ | $\frac{pb}{2V}$ | $\frac{qc}{2V}$ | $\frac{rb}{2V}$ | $Mach$ | $\delta_e$ | $\delta_a$ |
|---|---|---|---|---|---|---|---|---|---|
| $C_D$ | 1.7e-2 | 9.76e-2 | 1.99e-3 | -3.93e-2 | -2.28e-1 | -1.31e-2 | 7.1e-5 | 0 | |
| $C_S$ | 0 | -3e-6 | -3.12e-1 | -1.28e-1 | 4e-6 | 1.41e-1 | 0 | 0 | 0 |
| $C_L$ | 1.51e-1 | 5.62 | 3.89e-3 | -9.78e-4 | 11.7 | 3.14e-3 | 5.32e-3 | 0.745 | 0 |
| $C_l$ | 0 | 3e-6 | -4e-2 | -6.21e-1 | -1.2e-5 | 3.87e-2 | 0 | 0 | -0.127 |
| $C_m$ | -8.12e-2 | -1.79 | -4.69e-3 | 4.35e-3 | -17.12 | -2.96e-3 | 2.75e-3 | -1.34 | 0 |
| $C_n$ | 0 | 1e-6 | 2.7e-2 | -3.24e-3 | 0 | -2.72e-2 | 0 | 0 | -6.7e-3 |

**Table 3.1:** Aerodynamic Coefficients

| Property | | Unit |
|---|---|---|
| $I_{xx}$ | 1.2387e3 | $Kg \cdot m^2$ |
| $I_{yy}$ | 5.4933e3 | $Kg \cdot m^2$ |
| $I_{zz}$ | 6.3186e3 | $Kg \cdot m^2$ |
| $I_{xz}$ | -300 | $Kg \cdot m^2$ |
| $m$ | 2100 | $Kg$ |
| $S$ | 14 | $m^2$ |
| $b$ | 8 | $m$ |
| $c$ | 1 | $m$ |

**Table 3.2:** Aircraft geometrical and mass properties

| Property | | Unit |
|---|---|---|
| $K_{T_h}$ | 3.56e-2 | $N \cdot s^2$ |
| $K_{Q_h}$ | 2e-3 | $N \cdot m \cdot s^2$ |
| $d_{x_h}$ | | $m$ |
| $d_{y_h}$ | 1.4 | $m$ |
| $d_{z_h}$ | 0.4 | $m$ |
| $J_{R_h}$ | 0.073 | $Kg \cdot m^2$ |

**Table 3.3:** Thrust Propellers data [11]

| Property | | Unit |
|---|---|---|
| $K_{T_v}$ | 7.39e-2 | $N \cdot s^2$ |
| $K_{Q_v}$ | 5.1e-3 | $N \cdot m \cdot s^2$ |
| $d_{x_v}$ | 1.25 | $m$ |
| $d_{y_v}$ | 1.35 | $m$ |
| $d_{z_v}$ | | $m$ |
| $J_{R_v}$ | 0.126 | $Kg \cdot m^2$ |

**Table 3.4:** Lift Propellers data [11]

The estimation of propeller inertias involved combining the inertia of the blades with that of the DC motors. As an initial approximation, we can simplify each blade's shape to a rectangular form. This relationship can be expressed as follows:

$$J_R = J_{DC} + N_{blades} \cdot \frac{1}{3} m_{blade} L_{blade}^2 \tag{3.48}$$

Here, $J_{DC}$ denotes the moment of inertia of the DC motor's rotor, which can be extracted from the motor's datasheet. In order to encompass the dynamic characteristics of the propeller, we can deduce an equivalent moment of inertia, as outlined in equation (3.48), under the assumptions of no gear in between the motor and the load (propeller).

### 3.3.6 Actuators

The actuators employed within the simulation process were conceptually classified into the subsequent categories:

1. **Position Servomechanism**:
   - **Objective**: The primary goal of a position servomechanism is to pursue a desired position or angle of a system, such as an actuator.
   - **Operation**: The control system compares the current position with the desired one; such comparison generates a command signal aimed at minimizing any existing position error.
2. **Velocity Servomechanism**:
   - **Objective**: The primary objective of a velocity servomechanism is to regulate and maintain a specific velocity or speed of a system, such as a motor.
   - **Operation**: The control system measures the current velocity of the system and compares it to the desired velocity. It generates a control signal to adjust the system's velocity to match the desired value, minimizing any velocity error.

#### Position Servo

The first type of servomechanism controls the movement of the aerodynamic surfaces $\delta_e$ and $\delta_a$. While the detailed modeling of this mechanism isn't covered here, it's important to address any delays introduced by the system in the aircraft's overall dynamics. These delays can potentially destabilize the system, so it's crucial to simulate them accurately to avoid overestimating the control system's performance.

We can think of the actuator as a black-box that, given a commanded deflection as an input, will output an actual angular displacement of the aerodynamic surface with a certain delay.

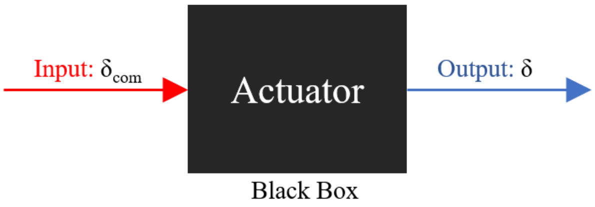


**Figure 3.10:** Position Servo

The transfer function in the Laplace domain used for delay simulation is of the following type [12]:

$$\frac{output}{input} = \frac{\delta}{\delta_{com}}(s) = \frac{K}{s+K} \tag{3.49}$$

In state space representation, the corresponding differential equation is:

$$\dot{\delta} = K(\delta_{com} - \delta) \tag{3.50}$$

where $K$ is a constant that can be tuned accordingly to the amount of delay we are willing to introduce.
In this work K was selected to be 20.
The corresponding step response and bode plots are shown in the next page.

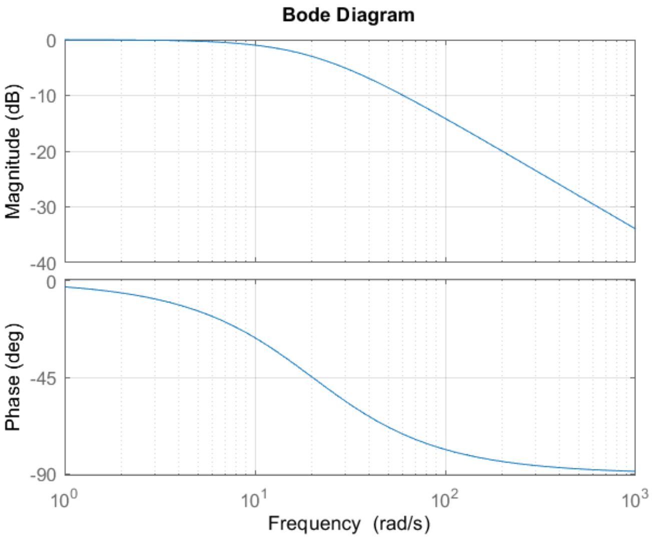


**Figure 3.11:** Actuator Bode plots

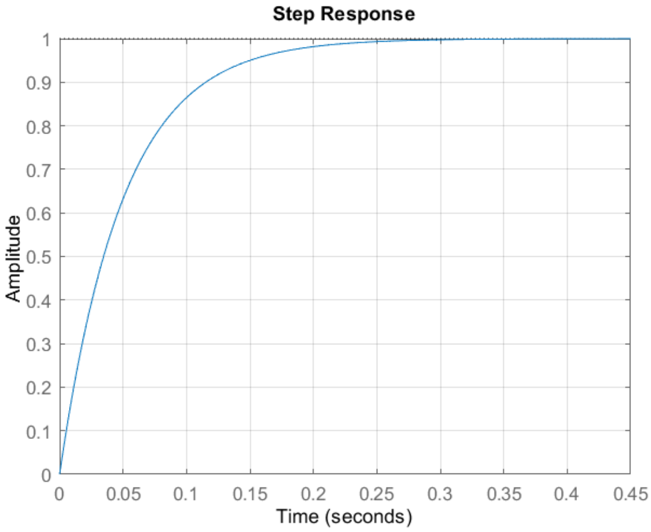


**Figure 3.12:** Actuator Step Response

As we can intuitively deduce from the step response, when a static step command is issued, the actuator doesn't respond immediately but exhibits some latency before reaching the commanded deflection. The Bode phase plot, on the other hand, illustrates the delay introduced by the transfer function based on the frequency content of any dynamical input.

Additionally, the output of the transfer function is saturated according to the aerodynamic surfaces deflection limits, which were selected as:

- $\delta_e \in [-24, +24]^o$
- $\delta_a \in [-24, +24]^o$

### Velocity Servo

The second servomechanism category is utilized to regulate the rotational speed of the propellers. In this scenario, a higher-fidelity electro-mechanical servomechanism model was constructed to integrate actual data from the DC motor datasheet, which was selected to drive the propellers.
The motor type was chosen in a prior design trade-off study, guided by an assessment of the power required to maintain the aircraft in flight across all phases of the mission.

A 1D mathematical model for the mechanical component of the actuator can be derived by applying Newton's second law for rotation as shown in the following picture.

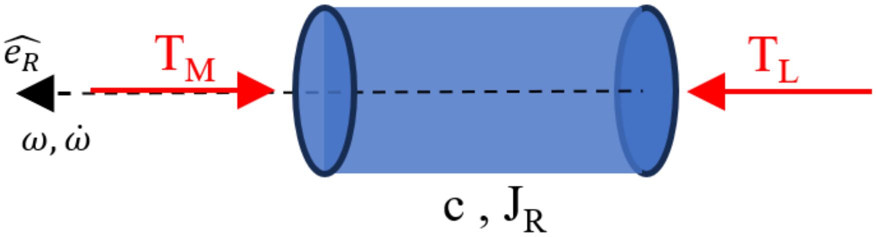


**Figure 3.13:** DC Motor mechanics

$T_M$ and $T_L$ are respectively the motor and load torques, whereas $c$ and $J_R$ are the viscous friction and inertia coefficients. The coefficients already include the propeller characteristics.
The motor torque comes from the electric model which can be approximated with a simple RL circuit, resulting in the following [8]:

$$\begin{cases} J_R\dot{\omega} + c\omega = T_M - T_L \\ L\frac{di}{dt} + Ri = V_A - K_b\omega \end{cases} \tag{3.51}$$

$V_A$ represents the input voltage, whereas $K_b\omega$ is referred to as back EMF, which factors in the conversion of electrical energy into mechanical energy. Apart from the influence of back EMF, the interdependency between the two equations arises from the motor's torque. Specifically, $T_M$ is typically contingent on the electrical current $i$ flowing through the circuit, a current that is regulated by the input voltage. A controller will provide an input voltage in response to the difference between the desired rotational speed and the present speed of the propeller. For achieving this, a sensor capable of measuring the rotation rate is essential.

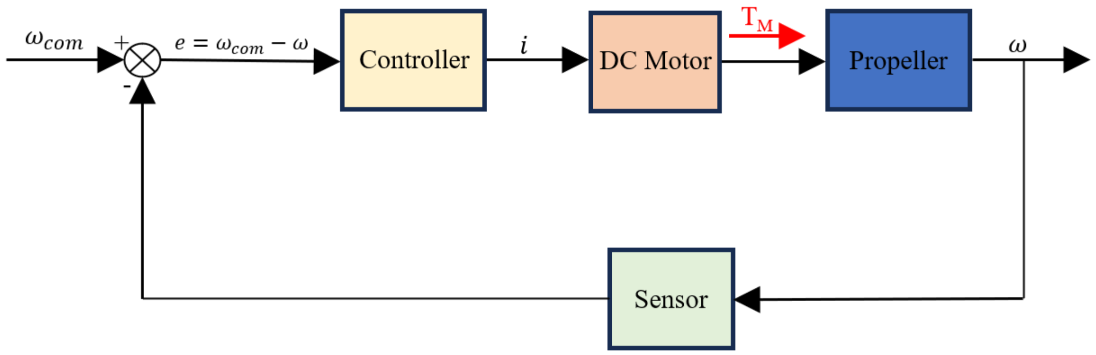


**Figure 3.14:** DC Motor Block Diagram

The simulation employs DC motor data extracted from the EMRAX 348 datasheet. The table below presents the key numerical attributes [13].

| Property | | Unit |
|---|---|---|
| $K_b$ | 0.15 | $V/RPM$ |
| $K_t$ | 1.92 | $Nm/A$ |
| $R$ | 13.15e-3 | $\Omega$ |
| $L$ | 185.3 | $\mu H$ |
| $T_{M_{max}}$ | 1000 | $Nm$ |
| $\omega_{max}$ | 4500 | $RPM$ |
| $c$ | 0.05 | $Kgm^2/s$ |
| $V_{A_{max}}$ | 1000 | $V$ |
| $J_R$ | 0.22042 | $Kg \cdot m^2$ |

**Table 3.5:** EMRAX 348

## 3.4 Plant and actuators in Simulink environment

In the figure below the two sub-models for actuators and plant (i.e. physical system to be controlled) are shown.

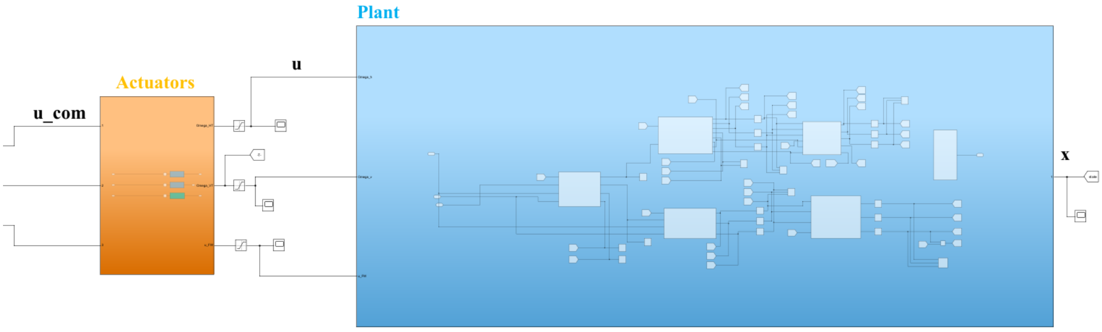


**Figure 3.15:** Plant + Actuators in Simulink environment

The first block models the actuators. The second one, instead, is the Simulink implementation of the non linear state space model of the aircraft. The control system provides a commanded input $\boldsymbol{u}_{com}$ to the actuators, which in turn will output an actual displacement/rotational speed $\boldsymbol{u}$ to the plant. The physical saturation limits restrict the actual aircraft's input appropriately.
If we expand the actuation block, we can see that it is formed by the same electro-mechanical model for both lift and thrust propellers and a first order differential equation for aerodynamic control surfaces.

The plant is made by 4 main blocks. The first one calculates external forces and torques exerted on the system which are then passed to the blocks implementing the dynamical equations for rotation and translation. The dynamical equations are integrated one time and the outcome of such integration is the passed onto the kinematics blocks. A second integration will provide the variables needed to complete the full state.

The code below shows how forces and moments are computed in body components.

```
function [forces_B, moments_B, F_c, M_c] = Actions_B(state, u_FW,
    Omega_h,Omega_v,rho, HT,S, b, c, dba, VT)

% state = [X Y Z u v w phi theta psi p q r]'
% u = [de da Tx1 Tx2 Tz taux tauy tauz]';
C_BN = Ned2Body(state(9), state(8), state(7));
```

```
x = [state(4:6) ; state(10:12) ; state(7:9)];
V = sqrt( sum(state(4:6).^2));

Alpha = atan2(state(6),state(4));
Beta = asin(state(5)/V);


%Aerodynamic Actions
if state(4) < 2
    A_a = zeros(6,1);
    C_WB = eye(3);
else
    C_WB = Body2Wind(Alpha, Beta);
    A_a = Aero_FandM(x,u_FW,rho,S,b,c,dba);
%A_a = [D,S,L,l,m,n]
end

%Horizontal Thrust Actions
A_h = HT.AM * Omega_h.^2;
%Vertical Thrust Actions
A_v = VT.AM * Omega_v.^2;
% .AM stands for allocation matrix

%Forces and Moments computation
forces_B = C_WB' * [-A_a(1); A_a(2); -A_a(3)] + [A_h(1); 0; -A_v(1)];
moments_B = A_a(4:6) + A_h(2:4) + A_v(2:4);

%Control forces and moments
M_c = 0.5*rho*V^2*S*[b 0 0; 0 c 0; 0 0 0]*[0 dba.Cl.da; dba.Cm.de 0;
    0 dba.Cn.da]*u_FW + [0;0;A_h(end)] + A_v(2:end);
F_c = [A_h(1); 0; -A_v(1)] ;
```

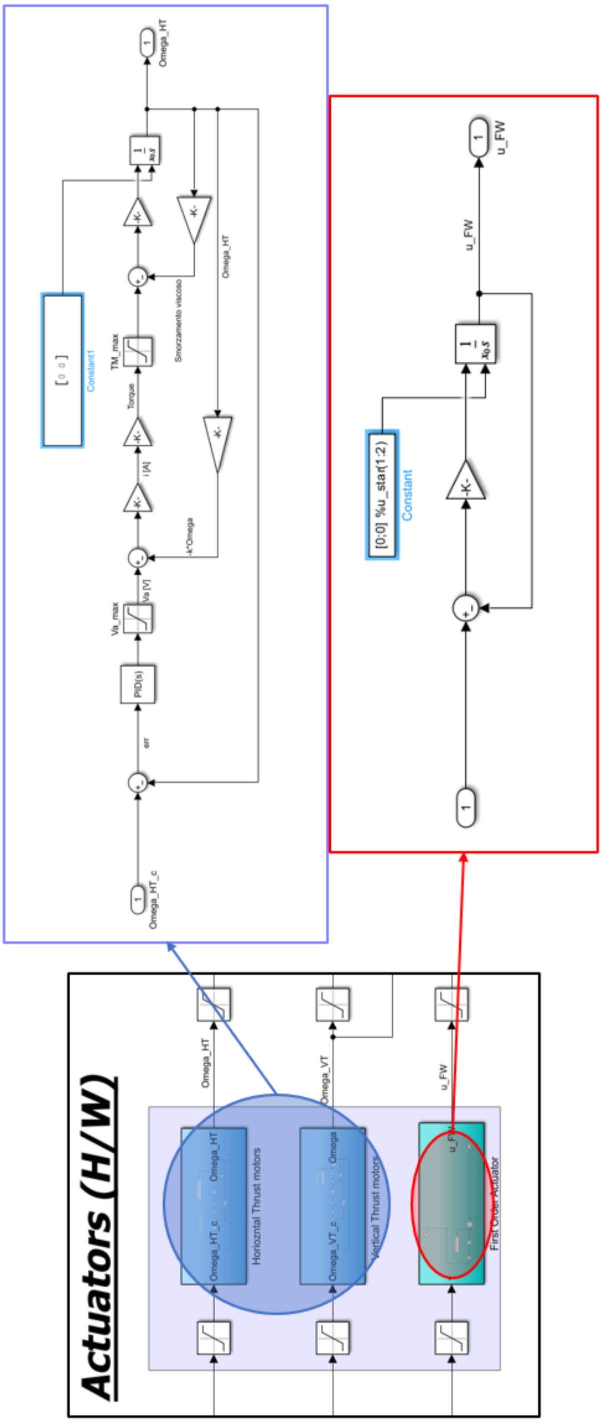


**Figure 3.16:** Actuator models

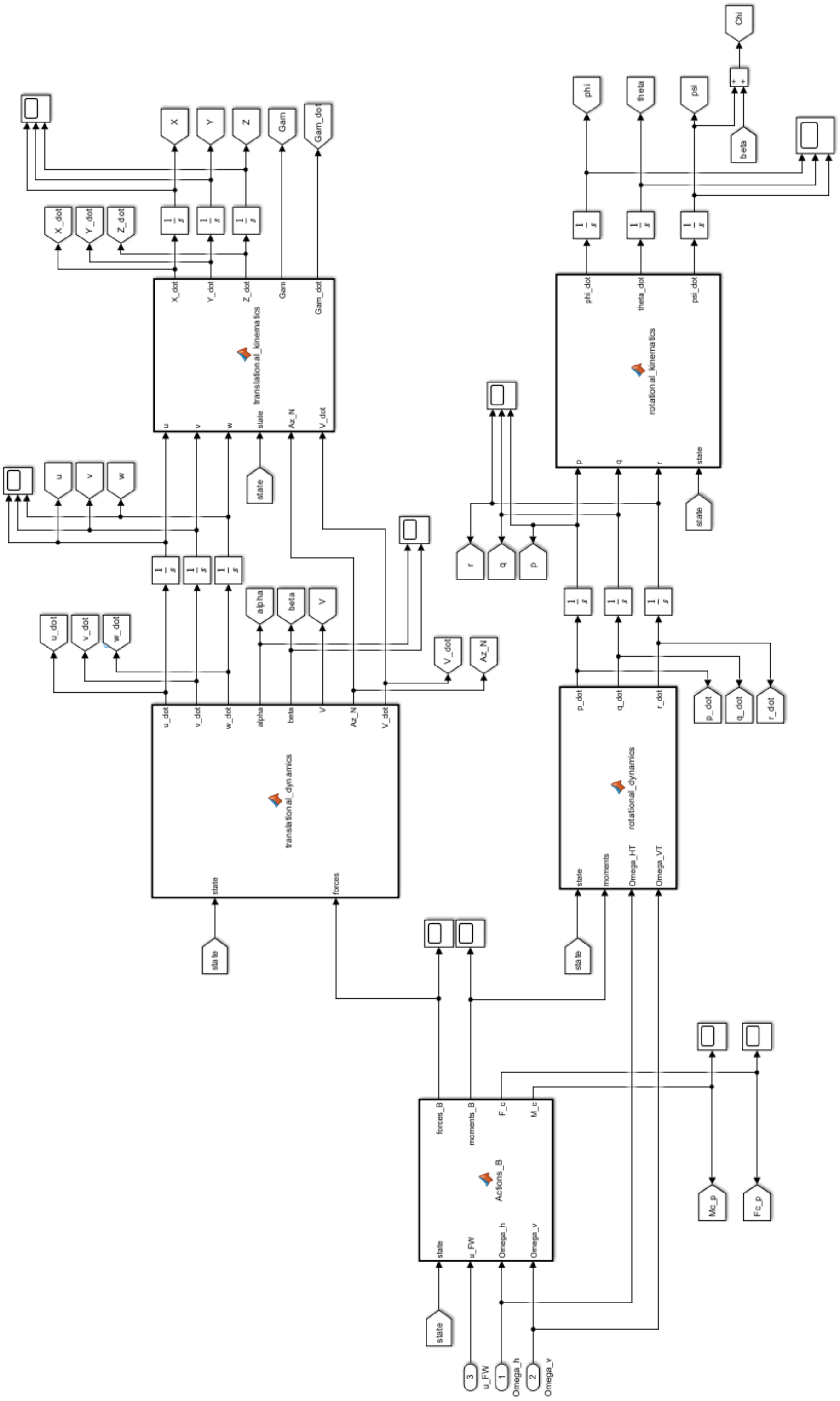


**Figure 3.17:** Plant Simulink model

# Chapter 4

# Trim Conditions and Linear Stability Analysis

This section is dedicated to assessing the linear stability of the system under different trim conditions during steady-level flight, with variations in speed. To linearize the aircraft dynamics, the initial step involves identifying equilibrium points. Subsequently, the eigenvalues of the state matrix are calculated.

## 4.1 Trim conditions

In this work, the equilibrium conditions are found under the assumption of steady-level, longitudinal flight. In mathematical terms this can be expressed as:

$$\text{Equilibrium point:} \iff \dot{\boldsymbol{x}}_{tr} = \frac{d}{dt}\begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \phi \\ \theta \\ \psi \end{bmatrix}_{tr} = \mathbf{0} \tag{4.1}$$

$$\text{Longitudinal flight:} \iff \boldsymbol{x}_{tr} = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \phi \\ \theta \\ \psi \end{bmatrix}_{tr} = \begin{bmatrix} u* \\ 0 \\ w* \\ 0 \\ 0 \\ 0 \\ 0 \\ \theta* \\ \psi* \end{bmatrix} = \boldsymbol{cost} \tag{4.2}$$

The trim problem can be resolved by formulating a constrained optimization problem [14], which can be solved using the MATLAB function *fmincon.*

Considering the previously discussed assumptions, further simplification of the problem includes setting $\dot{\mathbf{\Phi}} = \mathbf{0}$ and $\dot{v} = 0$. The first set of equations arises directly from these assumptions and doesn't introduce uncertainties into the problem. The second equation, however, may not be precisely accurate due to the presence of unbalanced side forces. Nonetheless, the error introduced by setting $\dot{v} = 0$ is almost negligible.

Consequently, the cost function is streamlined by excluding the derivatives of Euler angles and side-velocity from consideration upfront:

$$J = \dot{u}^2 + \dot{w}^2 + \dot{p}^2 + \dot{q}^2 + \dot{r}^2 \tag{4.3}$$

The objective is to minimize the cost function, which is equivalent to finding an equilibrium point that zeroes out the dynamical system. To achieve this, the minimum of J must be determined while exploring the search space variables $\boldsymbol{z}^T = [T_{x1}, T_{x2}, T_z, \delta_e, \delta_a, \alpha, \theta, \tau_x, \tau_y, \tau_z]$ under variation. It is important to notice that J needs to be written explicitly in terms of $\boldsymbol{z}$.

Below are the constraints for the search space variables ([15]), assuming a predefined velocity, for the most general case, which is the TR mode. It's important to note that the FW mode can be considered a sub-case of the TR mode in which the MR control variables are set to zero.

1. Linear Constraints:

$$\begin{cases} \gamma = \theta - \alpha = 0 \\ l_{\delta_a}\delta_a(1 - f(V)) - f(V)\tau_x = 0 \\ m_{\delta_e}\delta_e(1 - f(V)) - f(V)\tau_y = 0 \\ d_{Y_H}(T_{x1} - T_{x_2})(1 - f(V)) - f(V)\tau_z = 0 \end{cases} \tag{4.4}$$

Where $f(V)$ is the allocation function for redundant actuators in TR mode as presented in chapter 2.

2. Inequality Constraints:

$$\begin{cases} T_{x_1}, T_{x_2}, T_z > 0 \\ \alpha, \theta \in [-16{,}16]^o \\ \delta_e, \delta_a \in [-24{,}24]^o \\ \Omega^2_{V_i} > 0 \;\; \forall \;\; i = 1:6 \quad \rightarrow \quad [M]^\dagger [Tz, \tau_x, \tau_y, \tau_z]^T > 0 \end{cases} \tag{4.5}$$

Where $[M]$ is the allocation matrix as reported in equation (3.30).

3. Nonlinear Constraints:
the algorithm better minimizes the cost function if the dynamical equations, set to zero, are used as nonlinear constraints themselves:

$$\frac{d}{dt}\begin{bmatrix} u \\ w \\ p \\ q \\ r \end{bmatrix} = \mathbf{0} \tag{4.6}$$

## 4.2 FW linear stability

Linear stability for the linearized equations is assessed for four different trim velocities, namely $V_{tr} \in \{50{,}55{,}65{,}75\}$ $[\frac{m}{s}]$.

As an example, considering $V = 55\frac{m}{s}$, the first step is to verify whether the solution obtained from the minimization problem indeed represents an equilibrium point. This can be done by simulating the nonlinear augmented plant (plant + actuators) to ensure that for constant trim inputs, the system stays at rest. The results for this specific case are presented below.

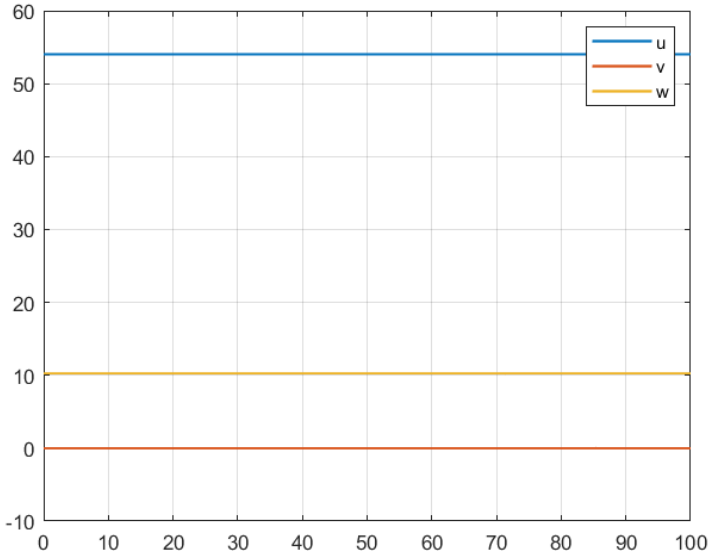


**Figure 4.1:** u,v,w vs t

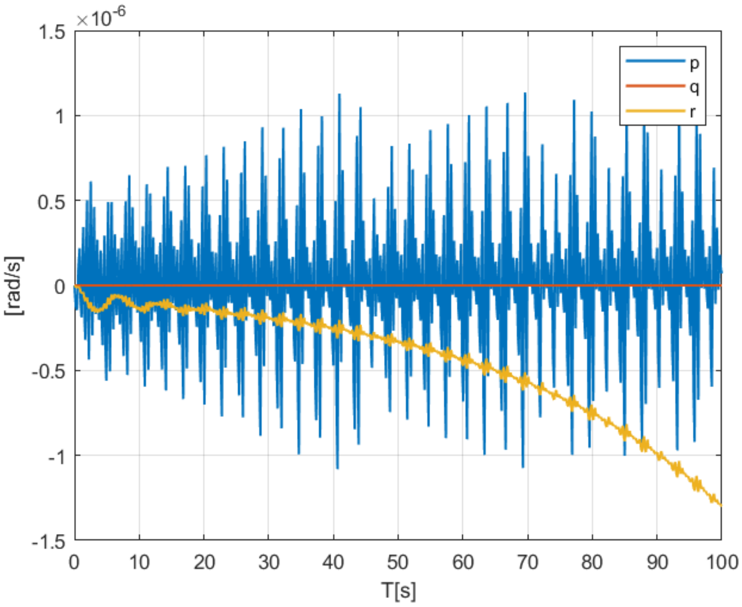


**Figure 4.2:** p,q,r vs t

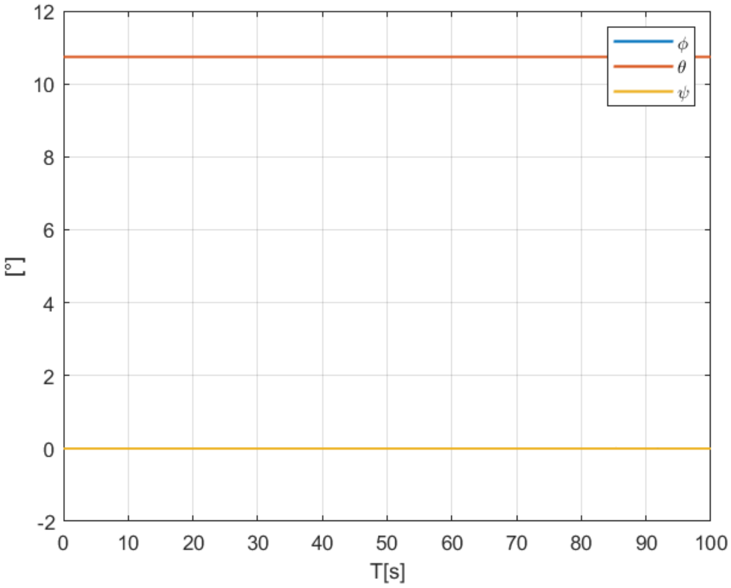


**Figure 4.3:** $\phi, \theta, \psi$ vs t

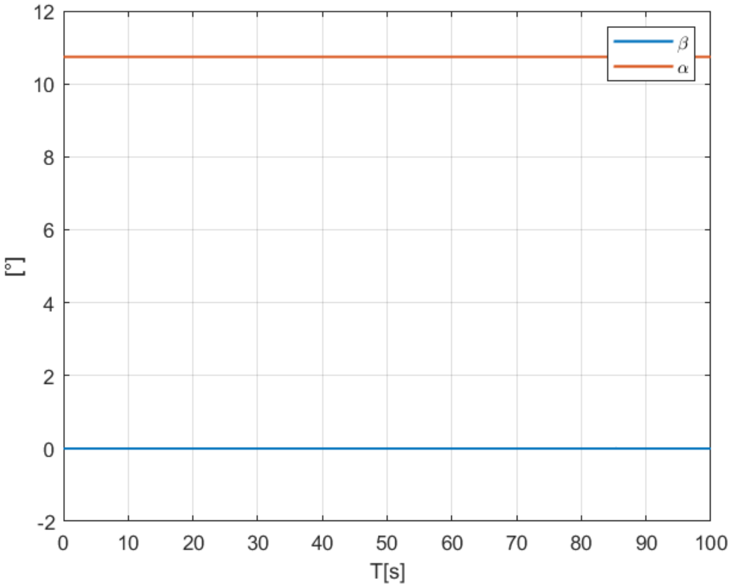


**Figure 4.4:** $\alpha, \beta$ vs t

Based on the plots, we can conclude that the calculated trim point is satisfactory.

Next, the trim values for state and control variables can be used to compute the state matrices $\mathbb{A}_{long}$ and $\mathbb{A}_{lat}$. Finally, their eigenvalues can be found. Each real eigenvalue or couple of complex conjugate pairs represents a natural mode of the system. Calculating the actual eigenvalues for this specific case yields:

$$\text{Longitudinal plane} \rightarrow \begin{cases} \lambda_{SP} = -0.8229 \pm 2.5906i & \rightarrow \text{ Short Period} \\ \lambda_{PH} = -0.0021 \pm 0.2486i & \rightarrow \text{ Phugoid} \end{cases} \tag{4.7}$$

$$\text{Latero-directional plane} \rightarrow \begin{cases} \lambda_{R} = -6.0253 & \rightarrow \text{ Roll} \\ \lambda_{S} = -0.0003 & \rightarrow \text{ Spiral} \\ \lambda_{DR} = -0.0249 \pm 0.9512i & \rightarrow \text{ Dutch Roll} \end{cases} \tag{4.8}$$

The generic solution to a linear ODE is of the form $Ce^{\lambda t} = Ce^{Re(\lambda)t}\sin(\omega t + \phi)$.

To ensure system stability, the solution must approach 0 as time $(t)$ increases. It's evident that the absolute value of the sine function cannot exceed 1. Therefore, for the solution to remain bounded and converge to zero, the real part $(Re(\lambda))$ of the eigenvalues must be negative.

In this specific case, all modes are stable. However, it's important to note that very lightly damped modes could exhibit unstable behavior in a nonlinear scenario. This analysis remains valid for small perturbations from the trim point.

A simulation to show the linear behaviour of the aircraft (i.e. its linear modes) can be performed by adding small perturbations to the initial state of the system at rest.

$$\boldsymbol{x}_0 = \boldsymbol{x}_{tr} + \delta\boldsymbol{x} \tag{4.9}$$

In the case of a longitudinal perturbation, two primary modes, Short Period and Phugoid, become active. Each longitudinal state variable responds to the perturbation as a linear combination of these two modes. Theoretical findings indicate that variables such as $u$ and $\theta$ are more influenced by the Phugoid mode, while variables like $w$ and $q$ exhibit more significant variations in the short term.

The subsequent plots illustrate the aircraft's longitudinal response to speed perturbations.

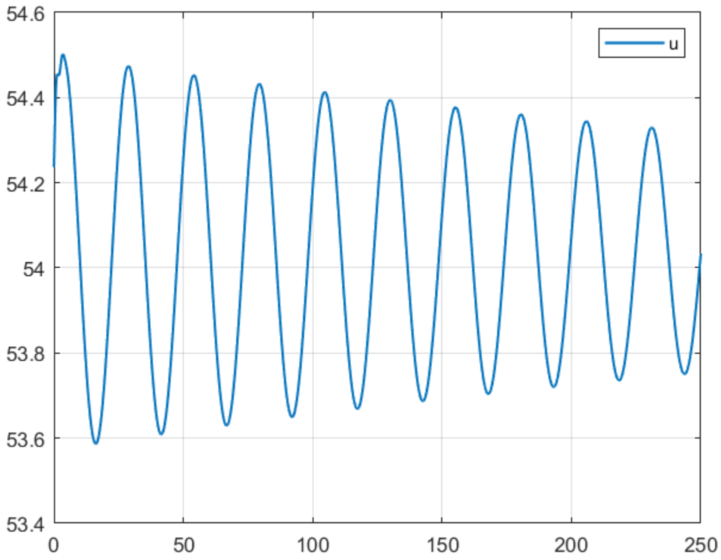


**Figure 4.5:** u vs t

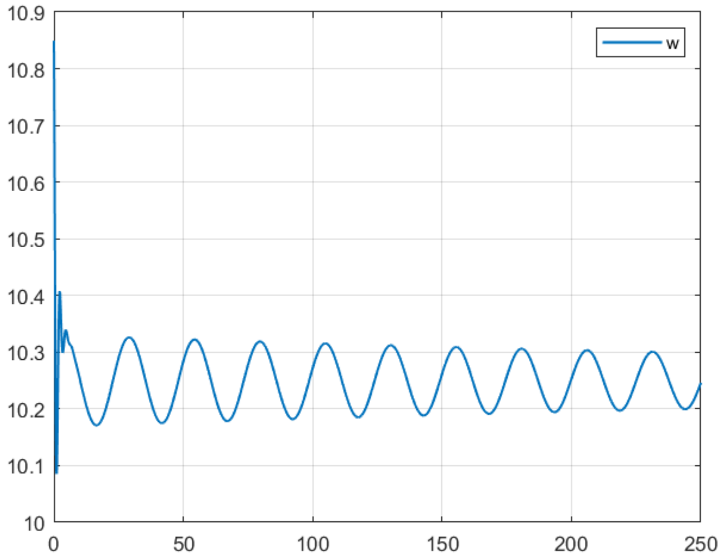


**Figure 4.6:** w vs t

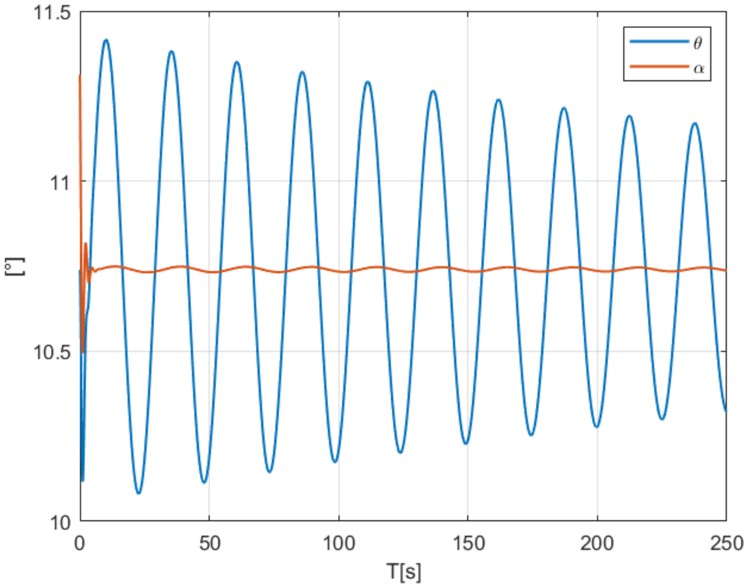


**Figure 4.7:** $\theta, \alpha$ vs t

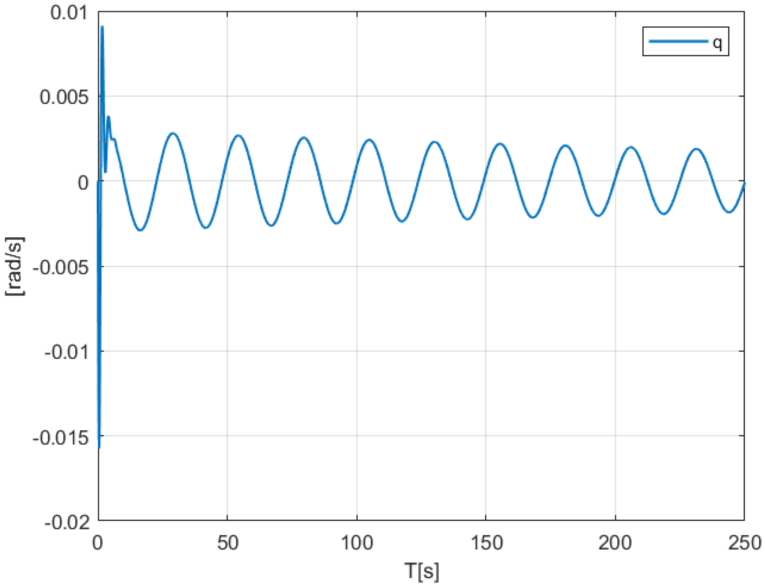


**Figure 4.8:** q vs t

The half-life time of each mode can be determined based on its eigenvalue using the formula $t_{1/2} = -\frac{log(2)}{Re(\lambda)}$. In the case of the Short Period mode, it takes less than one second for its oscillations to halve. On the other hand, for the Phugoid mode, it requires approximately 340 seconds for the exponential envelope of its oscillations to halve.

Concerning the lateral modes, Dutch Roll stands out as the only mode characterized by oscillatory behavior. In contrast, the other two modes exhibit purely exponential decay, which presents challenges in visualization due to their extremely short or lengthy half-life times. For instance, the Spiral mode has such light damping that it takes nearly 2000 seconds to diminish significantly. Conversely, the Roll mode experiences rapid decay within a fraction of a second. The following plot illustrates the behavior of the Roll mode. In the short term, it indeed demonstrates purely exponential decay before other modes become prominent.

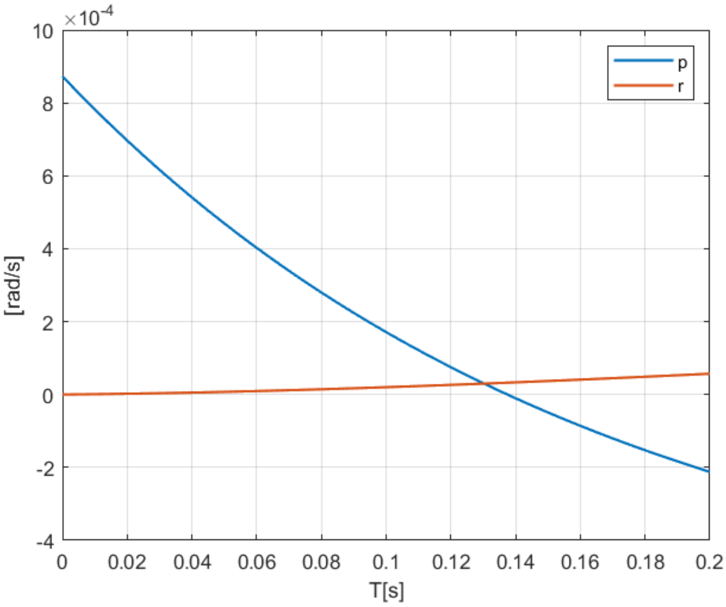


**Figure 4.9:** p vs t

For longer simulation time, the other modes become visible as well.

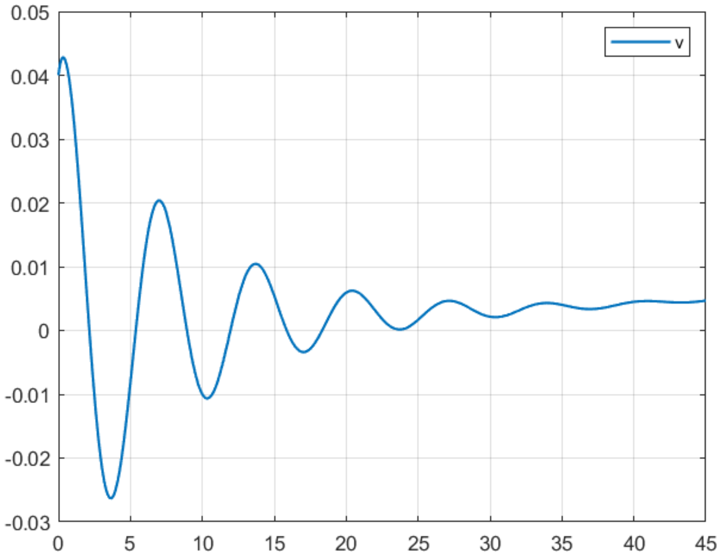


**Figure 4.10:** v vs t

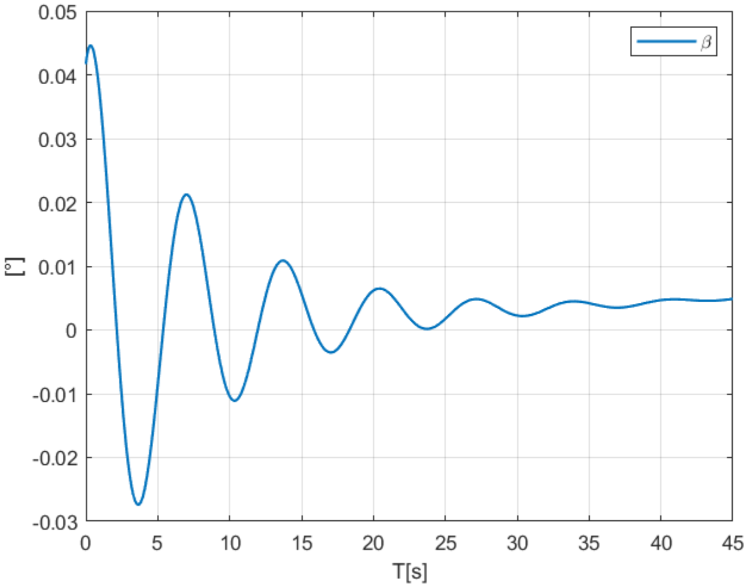


**Figure 4.11:** $\beta$ vs t

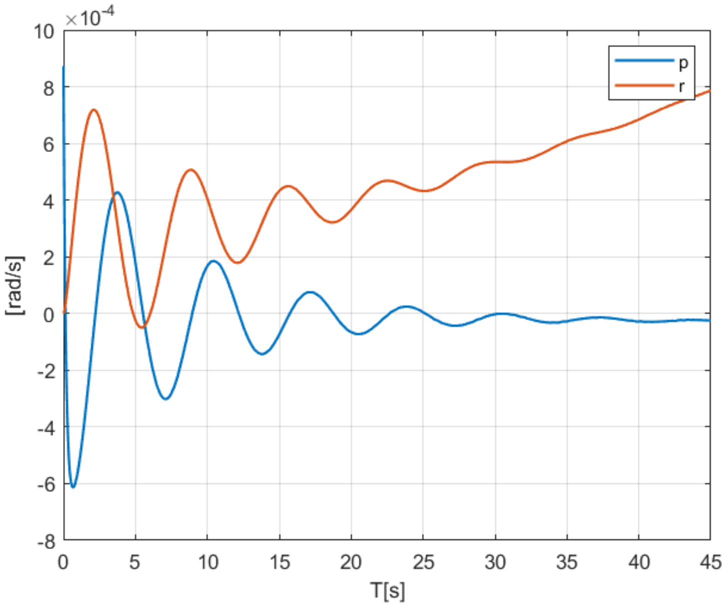


**Figure 4.12:** p,r vs t

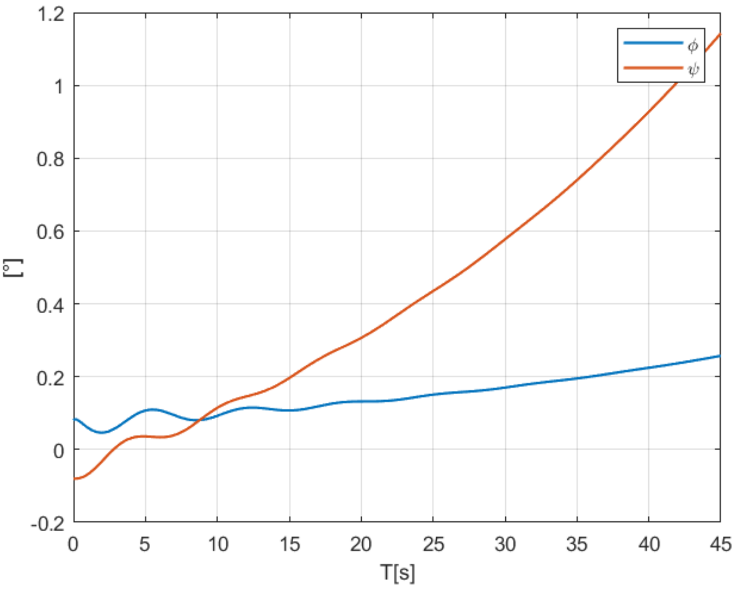


**Figure 4.13:** $\phi, \psi$ vs t

Even though the linear analysis indicates the stability of all lateral modes, the visual evidence in the plots clearly shows a departure from the trim conditions.

As mentioned earlier, it's a common occurrence for lightly stable linear modes to become unstable in a nonlinear simulation.

At the end of the chapter , a table is presented, showing the eigenvalues associated with the FW modes at different trim velocities.

## 4.3 TR linear stability

In Transition mode, the selected trim velocities for linear stability analysis are $V_{tr} \in \{16{,}25{,}34{,}43\}\ [\frac{m}{s}]$.

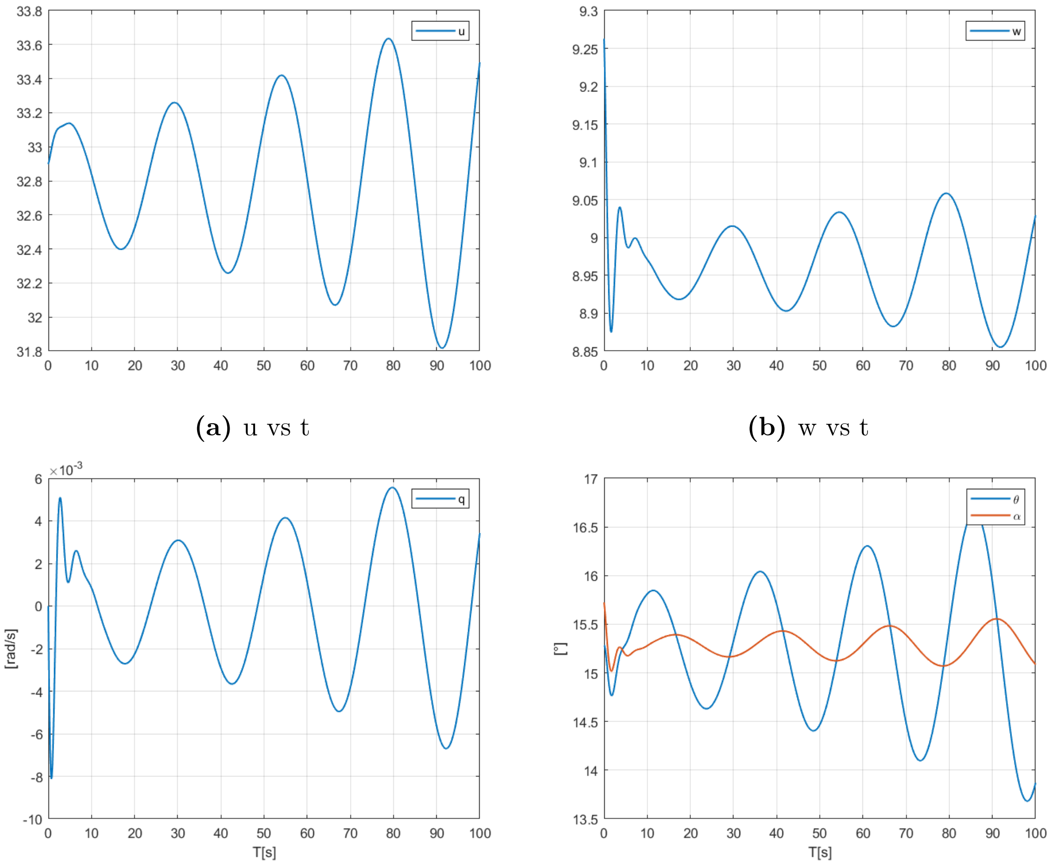
All the considerations made for FW flight remain applicable in this flight regime. However, it's important to note that the linearized equations differ from the previous ones due to an additional thrust term acting along $-\hat{k}_b$.

Below are the results for $V_{tr} = 34\frac{m}{s}$:

$$\text{Longitudinal plane} \rightarrow \begin{cases} \lambda_{SP} = -0.5233 \pm 1.6013i \quad \rightarrow \quad \text{Short Period} \\ \lambda_{PH} = 0.0012 \pm 0.2533i \quad \rightarrow \quad \text{Phugoid} \end{cases} \tag{4.10}$$

$$\text{Latero-directional plane} \rightarrow \begin{cases} \lambda_{R} = -3.9137 \quad \rightarrow \quad \text{Roll} \\ \lambda_{S} = -0.0004 \quad \rightarrow \quad \text{Spiral} \\ \lambda_{DR} = -0.0022 \pm 0.6489i \quad \rightarrow \quad \text{Dutch Roll} \end{cases} \tag{4.11}$$

Upon analyzing the numerical values of the eigenvalues, it becomes evident that almost all modes exhibit reduced stability compared to the previous case. Notably, the Phugoid mode has shifted towards instability.



**(a)** u vs t

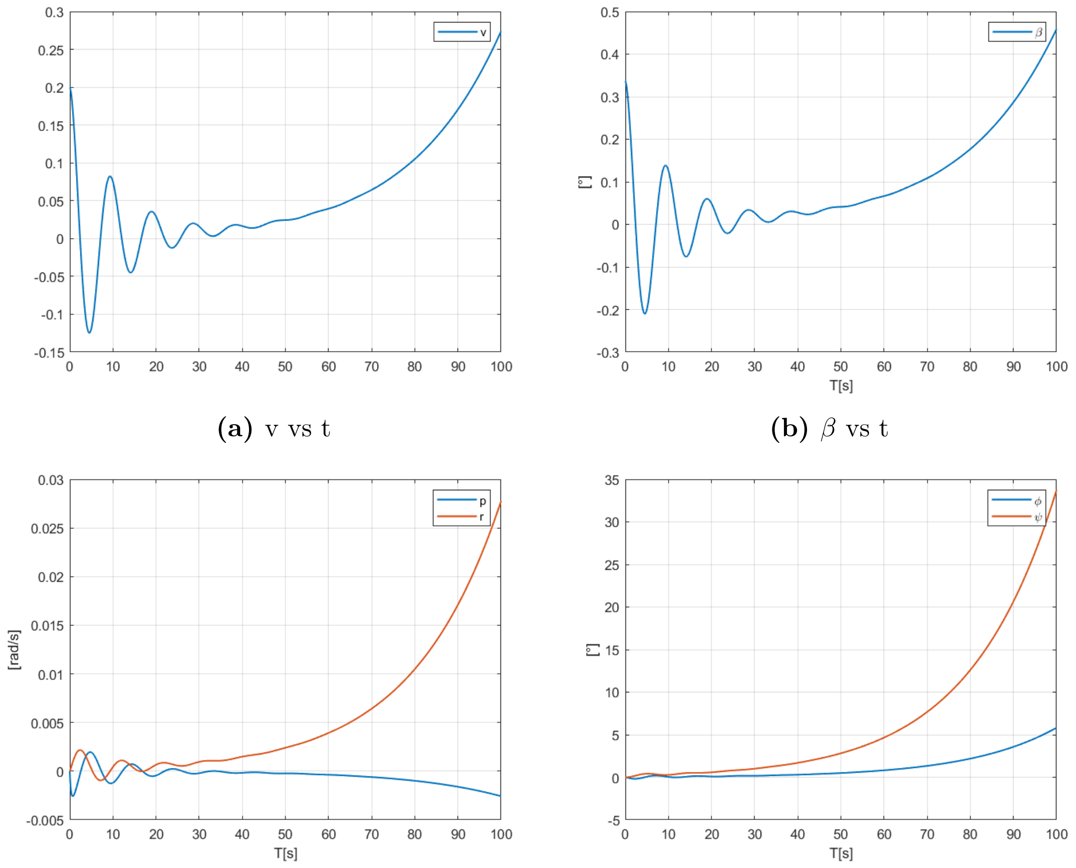
**(b)** w vs t

**(c)** q vs t

**(d)** $\theta$,$\alpha$ vs t

**Figure 4.14:** Longitudinal linear modes in Transition regime

The figures presented above, derived from an open-loop simulation of the augmented plant under small perturbations, provide concrete confirmation of our theoretical analysis. It is now evident that the Phugoid mode has transitioned into an unstable state, with oscillations that exhibit a divergent behavior, eventually tending towards infinity.



**(a)** v vs t

**(b)** $\beta$ vs t

**(c)** p,r vs t

**(d)** $\phi$,$\psi$ vs t

**Figure 4.15:** Latero-directional linear modes in Transition regime

In the lateral-directional plane, our analytical analysis continues to indicate stability for all modes. However, mirroring the behavior observed in the FW mode, the Spiral mode exhibits such light damping that even a small perturbation in a nonlinear simulation results in an unstable aircraft response.

When we consider the system as a continuum, moving away from the discrete state machine view, we can observe how the roots of the linear system change as the speed increases. This can be visualized by plotting the eigenvalues on the complex plane.

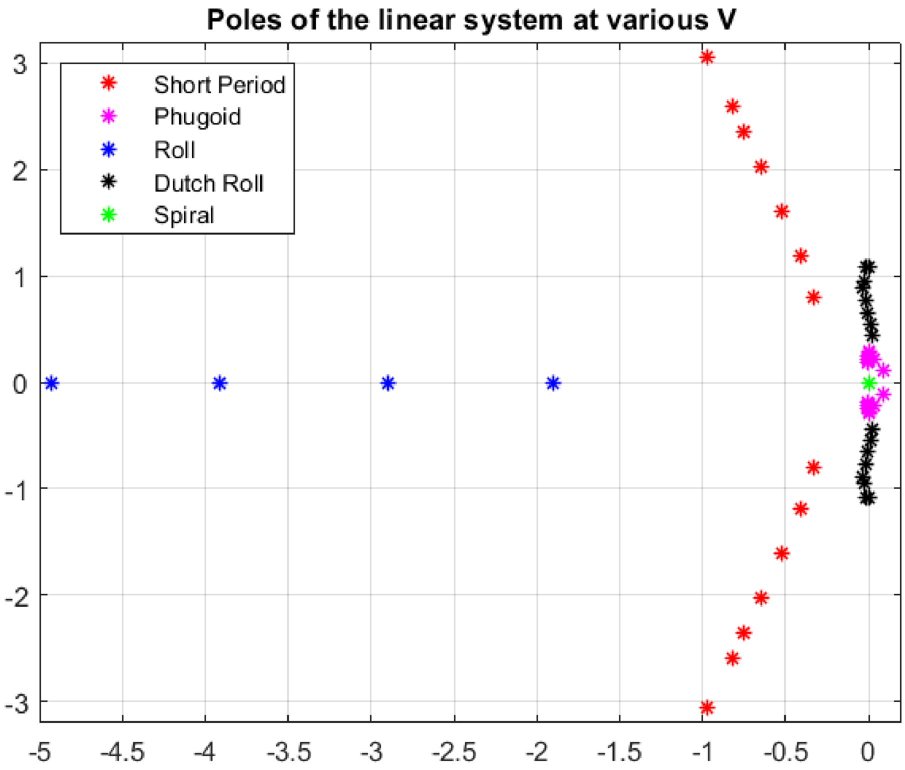


**Figure 4.16:** Root Locus w.r.t. velocity variations

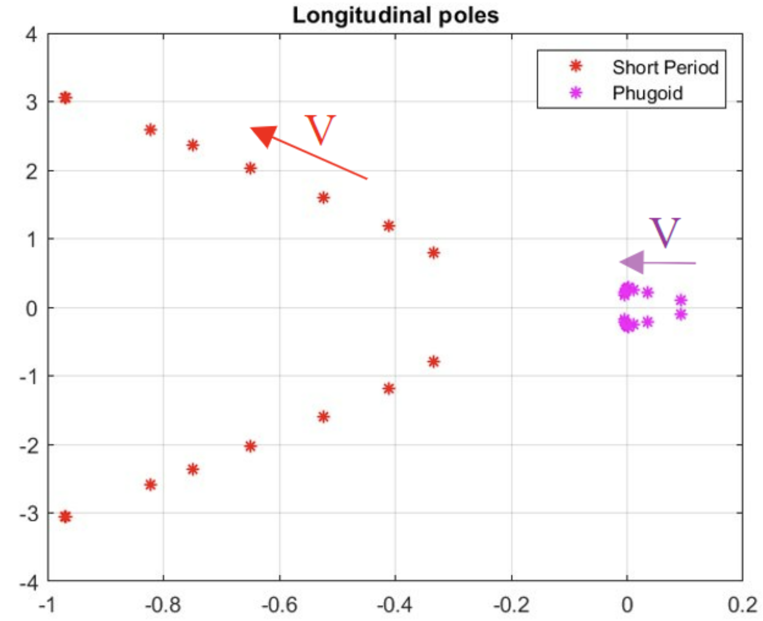


**Figure 4.17:** Longitudinal Poles

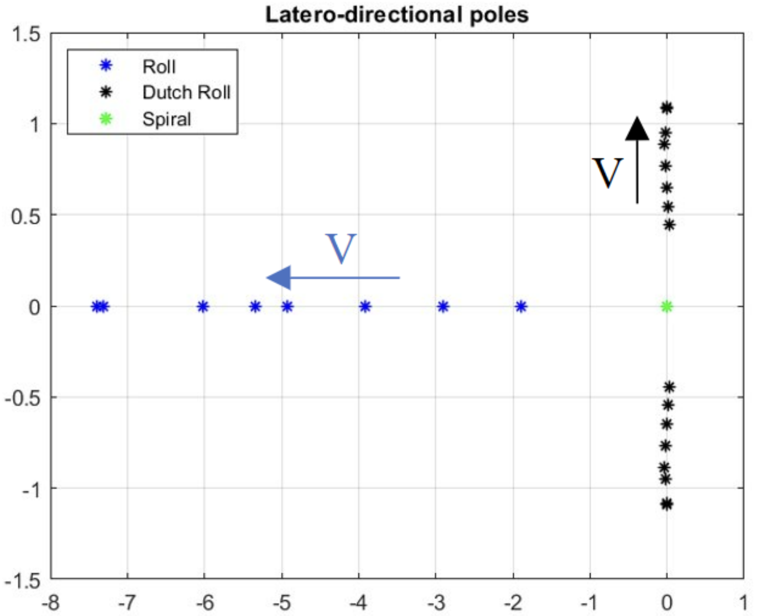


**Figure 4.18:** Lateral-Directional Poles

| | | $16\frac{m}{s}$ | $25\frac{m}{s}$ | $34\frac{m}{s}$ | $43\frac{m}{s}$ |
|---|---|---|---|---|---|
| **Long.** | SP | $-0.3335 \pm 0.7954i$ | $-0.4106 \pm 1.1823i$ | $-0.5233 \pm 1.6013i$ | $-0.6489 \pm 2.0259i$ |
| | PH | $0.0929 \pm 0.1062i$ | $0.0348 \pm 0.2098i$ | $0.012 \pm 0.2533i$ | $0.002 \pm 0.2863i$ |
| **Lat.** | DR | $0.0273 \pm 0.4467i$ | $0.0097 \pm 0.5411i$ | $-0.0022 \pm 0.6489i$ | $-0.0116 \pm 0.7667i$ |
| | RM | $-1.8982$ | $-2.9002$ | $-3.9137$ | $-4.9323$ |
| | SM | $-0.000417$ | $-0.000458$ | $-0.000435$ | $-0.000394$ |

**Table 4.1:** TR linear modes

| | | $50\frac{m}{s}$ | $55\frac{m}{s}$ | $65\frac{m}{s}$ | $75\frac{m}{s}$ |
|---|---|---|---|---|---|
| **Long.** | SP | $-0.7503 \pm 2.3556i$ | $-0.8229 \pm 2.5906i$ | $-0.9694 \pm 3.0612i$ | $-1.1167 \pm 3.5319i$ |
| | PH | $-0.0007 \pm 0.2727i$ | $-0.0021 \pm 0.2486i$ | $-0.0037 \pm 0.2113i$ | $-0.0047 \pm 0.1841i$ |
| **Lat.** | DR | $-0.0335 \pm 0.8874i$ | $-0.0249 \pm 0.9512i$ | $-0.011 \pm 1.0867i$ | $0.0001 \pm 1.2286i$ |
| | RM | $-5.3335$ | $-6.0253$ | $-7.3091$ | $-8.5194$ |
| | SM | $-0.0004$ | $-0.0003$ | $-0.0003$ | $-0.0003$ |

**Table 4.2:** FW linear modes

# Chapter 5

# Control techniques

A common approach in designing a complete control system for an air vehicle is to make use of the so called Time Scale Separation Principle relying on different transient decaying speeds for the various state variables. To tailor a control system effectively for a specific plant, it is essential to first assess the time scale of each variable. This assessment can be carried out through analytical methods like singular perturbations theory ([16],[17]) or by conducting simulations to evaluate the open-loop response of the aircraft to various inputs.

For conventional aircrafts, it is generally safe to assume that linear dynamics are slower than attitude dynamics. This greatly simplifies the control system design by dividing it into two distinct loops: a faster inner loop for attitude tracking and a slower outer loop for position tracking. The whole inner loop acts as an actuator for the outer loop and such partition is based on the following assumptions:

1. The inner loop is sufficiently fast to treat linear velocity as constant within a finite time horizon.
2. The outer loop is slow enough to consider attitude changes (governed by the inner loop) as instantaneous. However, an incorrect assessment of this second assumption can lead to instability. To mitigate this (i.e. to guarantee stability), gains for the internal loop must be set higher than those for the external loop [18],[19].

The primary advantage of this approach is that if the sum of the relative degrees of each subsystem equals the total order of the system as a whole, then no uncontrolled internal dynamics exist [17]. Consequently, this approach offers a significantly simplified mathematical framework for designing a control system.

The following page features an illustration depicting the various time scales present in aircraft dynamics. The evolution of any variable over time concerning a particular input is characterized by an order number, which is linked to the number of integrations in between.

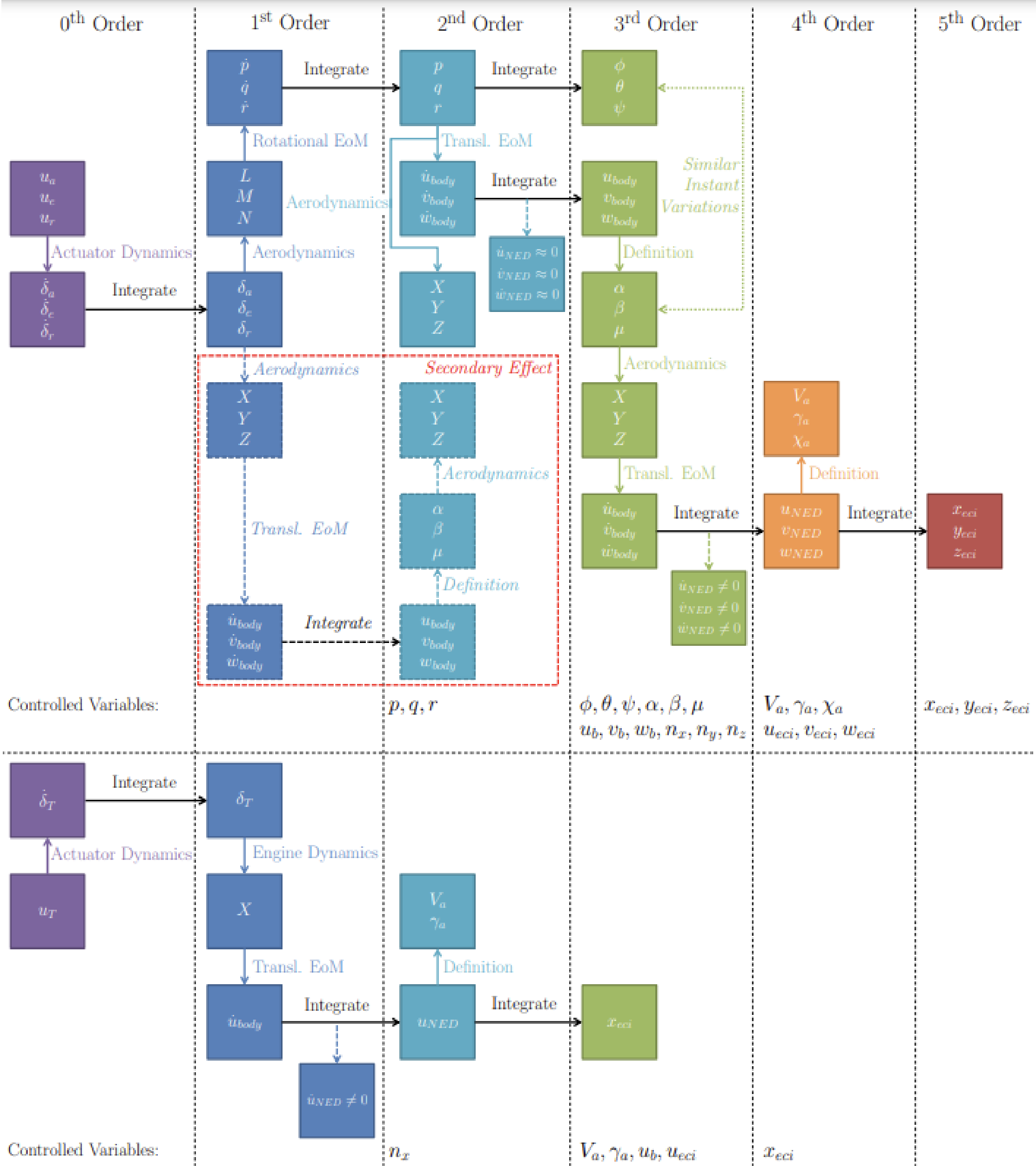


**Figure 5.1:** Time scales. Credit: R.C.van't Veld [17]

The following subsections serve as an introduction into the foundations of control theory needed for this thesis project.

## 5.1 PID

PID (Proportional-Integral-Derivative) is a feedback-based control technique used to minimize the error between a reference setpoint and the actual value of a variable within a system. It achieves this by calculating the difference between the reference and actual values and generating a control action as the sum of three contributions:

1. **P**: This component produces a control action that is proportional to the instantaneous error between the desired value and the actual value of the reference variable. In other words, it responds to the current error and attempts to bring it to zero.

2. **I**: The integral component produces a control action that is proportional to the integral (accumulated sum) of the error over time. It considers the history of the error and helps eliminate any long-term steady-state errors by continuously adjusting the control output.

3. **D**: The derivative component produces a control action that is proportional to the rate of change (derivative) of the error with respect to time. It provides insight into the future trend of the error and helps dampen any rapid changes in the error, thus contributing to stability.

By combining these three components, a PID controller can effectively regulate a system, bringing it to the desired setpoint and maintaining it there with minimal error. The specific gains or coefficients assigned to each component (Kp, Ki, and Kd) determine the controller's behavior and are typically tuned to achieve desired system performance.

In this work, the PID controller was implemented using the Simulink software, incorporating a modification in the form of a low-pass filter added to the derivative (D) contribution [20]. This modification is intended to attenuate any high-frequency disturbances. As a result, the controller's transfer function can be expressed as follows:

$$\frac{PID}{E}(s) = K_P + \frac{K_I}{s} + K_D \frac{Ns}{s+N} \tag{5.1}$$

as opposed to its basic formulation, which is:

$$\frac{PID}{E}(s) = K_P + \frac{K_I}{s} + K_D s \quad \rightarrow \quad PID(t) = K_P e(t) + K_D \dot{e}(t) + K_I \int_{t_0}^{t} e(\tau) d\tau \tag{5.2}$$

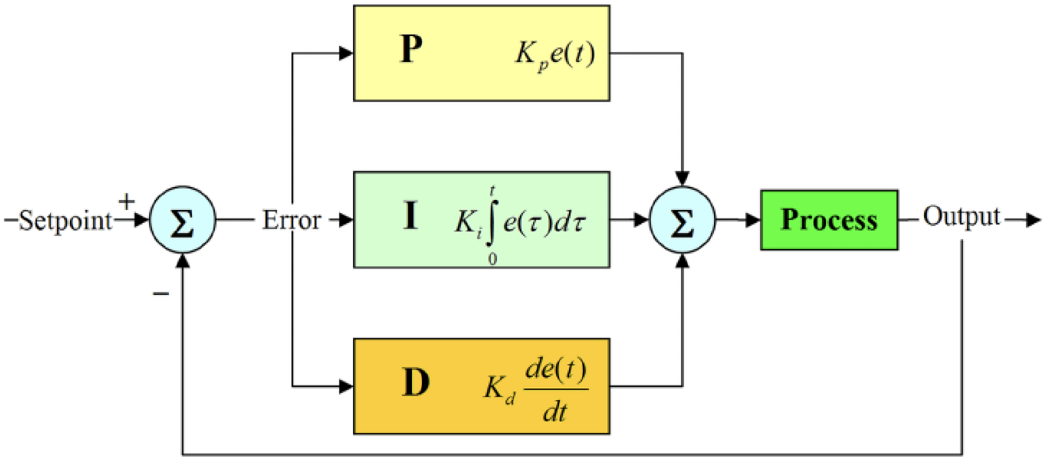


**Figure 5.2:** PID structure

## 5.2 Non-Linear Dynamic Inversion (NDI)

The dynamics of an aircraft are typically linear only under specific flight conditions within a limited range. The conventional approach involves assessing various trim points across the flight envelope and adjusting control gains for each specific condition, as these gains tend to work effectively only around that particular flight condition.

However, to bypass the challenge of gain-scheduling, there is a need to develop a nonlinear controller. Nonlinear Dynamic Inversion has demonstrated its effectiveness as a relatively straightforward method for controlling nonlinear systems while leveraging linear control techniques.

For NDI to be applicable, certain prerequisites must be met: the system should exhibit control affinity, and it must be restructured into a companion form. This transformation facilitates the application of NDI, allowing for more robust control across a broader range of flight conditions, including those where the system's dynamics are nonlinear.

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} & \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{u} \in \mathbb{R}^m \\ \boldsymbol{y} = \boldsymbol{H}(\boldsymbol{x}) & \boldsymbol{y} \in \mathbb{R}^p \end{cases} \tag{5.3}$$

Each output $y_i$ needs to be differentiated until any input belonging to $\boldsymbol{u}$ appears explicitly in the equation. Before deriving the controller, we need to define the Lie derivative $L_{(\cdot)}$.

Given any function $V(\boldsymbol{x}(t))$, its derivative w.r.t. time can be expressed in terms of Lie derivatives, passing through partial differentiation first:

$$\frac{dV(\boldsymbol{x}(t))}{dt} = \frac{\partial V}{\partial \boldsymbol{x}}\frac{d\boldsymbol{x}}{dt} = \frac{\partial V}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} = \frac{\partial V}{\partial \boldsymbol{x}}(\boldsymbol{F}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}) = L_f V(\boldsymbol{x}) + L_g V(\boldsymbol{x})\boldsymbol{u} \quad (5.4)$$

We can apply (5.4) to each of the $p$ output equations. Knowing that the $L_{(\cdot)}$ operator is linear and assuming that each $i-th$ component will be dependent on $\boldsymbol{u}$ on its $r_i - th$ derivative, the following can be obtained:

$$\dot{y}_i = y_i^{(1)} = \frac{\partial H_i}{\partial \boldsymbol{x}}\frac{d\boldsymbol{x}}{dt} = L_f H_i(\boldsymbol{x}) \quad (5.5)$$

$$\cdot$$

$$\cdot$$

$$y_i^{(r_i)} = L_f^{r_i} H_i(\boldsymbol{x}) + \sum_{j=1}^{m} L_{g_j} L_f^{r_i - 1} H_i(\boldsymbol{x}) u_j \quad (5.6)$$

The whole system can be rewritten in matrix form as:

$$\begin{bmatrix} y_1^{r_1} \\ \cdot \\ \cdot \\ y_p^{r_p} \end{bmatrix} = \begin{bmatrix} L_f^{r_1} H_1(\boldsymbol{x}) \\ \cdot \\ \cdot \\ L_f^{r_p} H_p(\boldsymbol{x}) \end{bmatrix} + \begin{bmatrix} L_{g_1} L_f^{r_1-1} H_1(\boldsymbol{x}) & \cdot & L_{g_m} L_f^{r_1-1} H_1(\boldsymbol{x}) \\ \cdot & & \cdot \\ \cdot & & \cdot \\ L_{g_1} L_f^{r_p-1} H_p(\boldsymbol{x}) & \cdot & L_{g_m} L_f^{r_p-1} H_p(\boldsymbol{x}) \end{bmatrix} \boldsymbol{u} = T_F(\boldsymbol{x}) + T_G(\boldsymbol{x})\boldsymbol{u} \quad (5.7)$$

Assuming $T_G$ is invertible and defining $y_i^{r_i} = v_i$ , $\forall i$ ($v_i$ = i-th virtual input) the actual control variables $u_i$ can be defined in terms of the virtual input $v_i$:

$$\boldsymbol{u} = T_G(\boldsymbol{x})^{-1}(\boldsymbol{v} - T_F(\boldsymbol{x})) \quad (5.8)$$

Since the virtual input $v_i$ will act only on the output $y_i$ the $T_G$ matrix is knows as *decoupling matrix.*

In this work, dynamic inversion will be used only for square systems where $r = r_1 + \cdots + r_p = n$. This case is simpler since there is no uncontrolled internal dynamics to be stabilized.

Dynamic inversion is basically a change of coordinates that transforms a nonlinear system into a new linear one in which all the classical linear control methodologies can be applied.

Defining $\boldsymbol{\mu}^T = [y_1, \cdot\cdot, y_1^{r_1-1}, \cdots, y_p, \cdot\cdot, y_p^{r_p-1}]$, the new linear system to be controlled is the following [21],[22]:

$$\begin{cases} \dot{\boldsymbol{\mu}} = \mathbb{A}\boldsymbol{\mu} + \mathbb{B}\boldsymbol{v} & \qquad \boldsymbol{\mu} \in \mathbb{R}^r, \boldsymbol{v} \in \mathbb{R}^p \\ \dot{\boldsymbol{y}}_{new} = \boldsymbol{\mu} \end{cases} \tag{5.9}$$

Where the state space matrices are:

$$\mathbb{A} = \begin{bmatrix} 0 & \mathbb{I}_{r_1-1} & 0 & \cdot & \cdot & 0 \\ 0 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & I_{r_2-1} & \cdot & \cdot & 0 \\ 0 & 0 & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & 0 & I_{r_p-1} \\ 0 & 0 & \cdot & \cdot & \cdot & 0 \end{bmatrix} \quad , \quad \mathbb{B} = \begin{bmatrix} \mathbf{0}_{(r_1-1)\times 1} \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{0}_{(r_p-1)\times 1} \\ 1 \end{bmatrix} \tag{5.10}$$

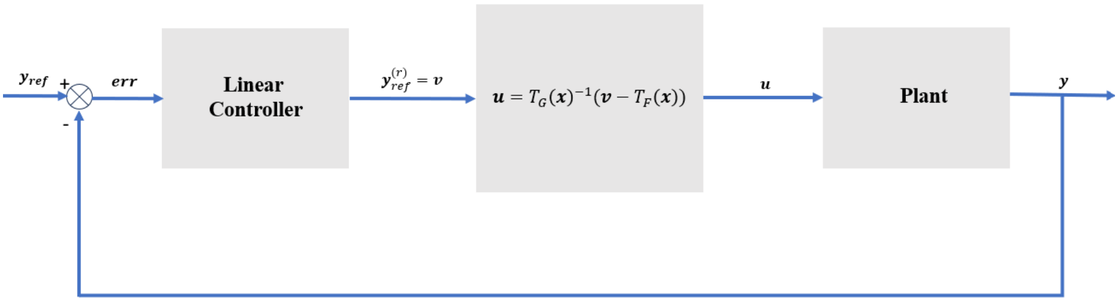


**Figure 5.3:** Nonlinear dynamic inversion

Nonlinear Dynamic Inversion (NDI) can offer effective control solutions for nonlinear dynamical systems. However, its performance can be limited by several significant drawbacks, as outlined below:

- NDI requires the system to be affine in terms of the control inputs represented by $\boldsymbol{u}$. This assumption is particularly stringent, especially when employing more intricate and accurate aerodynamic models in control system design.
- Another notable limitation lies in its strong reliance on the accuracy of the mathematical model. NDI is sensitive to uncertainties arising from model mismatches [23]. Unfortunately, uncertainties are inevitable in both the aerodynamic and structural models..

- NDI often places a greater computational burden compared to other control techniques.

## 5.3 Incremental Non-Linear Dynamic Inversion (INDI)

All of the aforementioned challenges associated with NDI can be effectively addressed through an incremental approach. The primary aim of this technique is to liberate the control system from strong dependencies on precise mathematical models and enhance its robustness to inaccuracies and disturbances. One direct result of the mathematical framework of this incremental approach is a control technique that is less computationally demanding. Additionally, it offers the capability to handle systems that do not exhibit control-affine properties.

To accomplish reduced realiance on the model, the complexity of real-world dynamical phenomena is shifted away from the equations and instead placed onto sensors. This approach is commonly referred to as the sensor-based approach, as it relies on sensor measurements in every iteration of the control system. Consequently, all unmodeled dynamics are introduced into the control law through actual state and state derivative measurements of the aircraft [24].

The fundamental concept of Incremental Nonlinear Dynamic Inversion (INDI) revolves around the idea of making adjustments to control variables based on past sensor measurements. This approach utilizes time-scale separation, building upon the notion of control effectiveness.
Control effectiveness, at its core, quantifies how a modification in a controlling variable influences a controlled variable. When extending this concept over time, we can differentiate between slow and fast dynamics. This distinction is made by considering the strength of the input's impact on the output within an arbitrarily small time interval [25].

In essence, the core idea behind INDI is to make gradual adjustments to control variables by leveraging historical sensor data. This approach employs the concept of control effectiveness to differentiate between different time scales within a system's dynamics. Specifically, we can view the slower dynamics as the outer loop and the faster dynamics as the inner loop. In this framework, from the perspective of the outer loop, it is assumed that any reference value for the inner dynamics is quasi-instantly achieved.

In the most general case, we can assume to be dealing with a nonlinear, non control-affine dynamical system:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \qquad \boldsymbol{x} \in \mathbb{R}^n \ , \ \boldsymbol{u} \in \mathbb{R}^m \tag{5.11}$$

Under the assumption of infinitesimal time increment, we can Taylor expand the equation about the current point in time $(\boldsymbol{x}(t_0), \boldsymbol{u}(t_0))$. For a sufficiently small $\Delta t$, a first order approximation can be used:

$$\dot{\boldsymbol{x}} \approx \dot{\boldsymbol{x}}_0 + \nabla_x \boldsymbol{f}_{(\boldsymbol{x}_0, \boldsymbol{u}_0)}(\boldsymbol{x} - \boldsymbol{x}_0) + \nabla_u \boldsymbol{f}_{(\boldsymbol{x}_0, \boldsymbol{u}_0)}(\boldsymbol{u} - \boldsymbol{u}_0) \tag{5.12}$$

Where the Jacobian matrices w.r.t. $\boldsymbol{x}$ and $\boldsymbol{u}$ are defined as:

$$\nabla_x \boldsymbol{f}_0 = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdot & \cdot & \frac{\partial f_1}{\partial x_n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \frac{\partial f_n}{\partial x_1} & \cdot & \cdot & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{(\boldsymbol{x}_0, \boldsymbol{u}_0)} \qquad \in \mathbb{R}^{n \times n} \tag{5.13}$$

$$\nabla_u \boldsymbol{f}_0 = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdot & \cdot & \frac{\partial f_1}{\partial u_m} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \frac{\partial f_n}{\partial u_1} & \cdot & \cdot & \frac{\partial f_n}{\partial u_m} \end{bmatrix}_{(\boldsymbol{x}_0, \boldsymbol{u}_0)} \qquad \in \mathbb{R}^{n \times m} \tag{5.14}$$

Given the assumption of quasi-instantaneous actuators, where control actions can be applied very quickly, there often exists a time scale separation that remains valid within a small time sample. This separation implies that the control effectiveness of the input vector $\boldsymbol{u}$ is significantly greater than that of the state vector $\boldsymbol{x}$.
Intuitively, this means that a change in the control input $\boldsymbol{u}$ has a much faster impact on the system compared to a change in the system state $\boldsymbol{x}$ itself. In other words, the cause of a change in the system (control input) happens much faster than the effect (system response).
Under these conditions, the following approximation can be considered valid:

$$\nabla_x \boldsymbol{f}_0 \Delta \boldsymbol{x} << \nabla_u \boldsymbol{f}_0 \Delta \boldsymbol{u} \tag{5.15}$$

This approximation likely represents a simplification or mathematical relationship that reflects the dominance of control input changes over state changes within a short time frame. The exact form of the approximation would depend on the specific context and equations being used.
As for traditional NDI, considering full-state feedback (i.e. $\boldsymbol{x} = \boldsymbol{y}$), we can equal $\dot{\boldsymbol{x}}$

to the virtual input $\boldsymbol{v}$ and obtain the following equation:

$$\boldsymbol{u} = \boldsymbol{u}_0 + (\nabla_u \boldsymbol{f}_0)^{\dagger}(\boldsymbol{v} - \dot{\boldsymbol{x}}_0) \tag{5.16}$$

From this expression, several key points can be deduced:

- The linearization about the current point in time makes this technique applicable even to non-affine systems..
- $(\nabla_u \boldsymbol{f}_0)^{\dagger}$ represents the Moore-Penrose (MP) inverse of the Control Effectiveness Matrix $\nabla_u \boldsymbol{f}_0$. This extends the technique's applicability to systems that are neither square nor invertible. Additionally, the MP inverse can be replaced with solving an optimization problem to account for control saturation constraints and power constraints, particularly when dealing with redundant actuators [26],[27],[28].
- Knowledge of the plant is limited to the *Control Effectiveness Matrix*, which can be computed analytically or can be derived from flight logs and parameter estimation techniques. Additionally, the matrix can be adapted online for a more robust and fault tolerant approach [24],[29]. The rest of the system is known through acceleration and actuator output measurements. Any model inaccuracy is, indeed, fed back to the system through sensors.
- All variables with a '0' subscript must be either measured or estimated. $\dot{\boldsymbol{x}}_0$ typically involves angular accelerations, which cannot be directly measured with a conventional IMU (Inertial Measurement Unit). Therefore, they need to be estimated from rate measurements. To mitigate noise amplification during differentiation, a pseudo-differentiation technique involving a low-pass Butterworth filter is commonly used:

$$H(s) = \frac{\sigma_n^2}{s^2 + 2\zeta\sigma_n s + \sigma_n^2} \tag{5.17}$$

- One of the primary challenges introduced by Incremental Nonlinear Dynamic Inversion (INDI) is delay, which can threaten the closed-loop stability of systems. This delay is mainly caused by:

  1. Actuator dynamics, as actuators are not instantaneous. The first works on INDI would neglect their presence. However, for Time Scale separation to be applied, it is fundamental to assess the effect of actuation lag on the overall system.

2. Filtering for pseudo-differentiation introduces a lag on the derived variables which can destabilize the system. The problem can be solved by time synchronizing the equation:

$$\boldsymbol{u} = \boldsymbol{u}_f + (\nabla_x \boldsymbol{f})_f^{\dagger}(\boldsymbol{v} - \dot{\boldsymbol{x}}_f) \tag{5.18}$$

   Where the expression is expanded about the point in time $t_f$ rather than $t_0$. This involves making sure that all measurements are delayed by the same amount by applying the same filter to all.

3. Sensors also introduce delays due to their inherent dynamics, which affect the timing of variables.

It's essential to emphasize that earlier research findings have demonstrated that INDI is more sensitive to lag introduced by filtering than it is to actuator delays [17].

### 5.3.1 Actuator Dynamics

Various methods have been developed to address actuator delay and constraints, including saturation and rate limits:

- A fundamental approach involves using a reference model with Pseudo Control Hedging (PCH). This approach accounts for the inherent dynamics of actuators and considers output physical limits [30].
- Recent advancements have introduced an extended Incremental Nonlinear Dynamic Inversion (E-INDI) technique that employs virtual inputs, denoted as $\boldsymbol{v}$, similarly to classical Nonlinear Dynamic Inversion (NDI). This extension enhances the INDI framework and provides a way to incorporate actuators directly into the control law [31].

Here's an insight into how actuators are included in the derivation of E-INDI:

$$v_i = y_i^{(r_i)} \;\; \forall i \quad \rightarrow \quad \boldsymbol{v} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}) \tag{5.19}$$

Extended INDI is based on a further differentiation of the precedent equation so as to obtain:

$$\dot{\boldsymbol{v}} = \frac{\partial \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{x}} \cdot \dot{\boldsymbol{x}} + \frac{\partial \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{u}} \cdot \dot{\boldsymbol{u}} = A(\boldsymbol{x}, \boldsymbol{u}) \cdot \dot{\boldsymbol{x}} + B(\boldsymbol{x}, \boldsymbol{u}) \cdot \dot{\boldsymbol{u}} \tag{5.20}$$

Assuming $B$ to slowly varying and $A \cdot \dot{\boldsymbol{x}} = \boldsymbol{d}_{\dot{x}}$ to be acting as a negligible disturbance (as for conventional INDI), keeping in mind section 3.3.6, we can introduce the actuator dynamics as follows:

$$\frac{\boldsymbol{u}}{\boldsymbol{u}_c} = \boldsymbol{K}_A \cdot \boldsymbol{F}_A(s) \quad \rightarrow \quad \Delta\boldsymbol{u}(s) = \boldsymbol{u}_c(s) - \boldsymbol{u}(s) = [(\boldsymbol{K}_A \cdot \boldsymbol{F}_A(s))^{-1} - \boldsymbol{I}] \cdot \boldsymbol{u}(s) \tag{5.21}$$

Introducing differentiation in the Laplace domain yields:

$$\Delta\boldsymbol{u}(s) = \boldsymbol{u}_c(s) - \boldsymbol{u}(s) = \frac{(\boldsymbol{K}_A \cdot \boldsymbol{F}_A(s))^{-1} - \boldsymbol{I}}{s} \cdot \dot{\boldsymbol{u}}(s) = \boldsymbol{G}_A(s) \cdot \dot{\boldsymbol{u}}(s) \tag{5.22}$$

Similarly we can specify a desired behaviour for the pseudocontrol as:

$$\dot{\boldsymbol{v}}_{des}(s) = \boldsymbol{G}_V(s) \cdot \Delta\boldsymbol{v}(s) \tag{5.23}$$

Going back to (5.20), neglecting state derivative terms, we obtain:

$$B \cdot (\boldsymbol{G}_A^{-1}(s) \cdot \Delta\boldsymbol{u}(s)) = \boldsymbol{G}_V(s) \cdot \Delta\boldsymbol{v}(s) \tag{5.24}$$

$$\vdots$$

$$\Delta\boldsymbol{u}(s) = \boldsymbol{G}_A(s) \cdot B^{-1} \cdot \boldsymbol{G}_V(s) \cdot \Delta\boldsymbol{v}(s) \tag{5.25}$$

The matrix to be inverted already contains information about control effectors. Furthermore, more advanced allocation algorithms can readily introduce rate limits and saturation constraints into the control system. The disturbance effect represented by $\boldsymbol{d}_{\dot{x}}$ can be estimated in real-time and integrated into the algorithm by comparing the actual behavior of the pseudo-control with the desired behavior specified by the transfer function $\boldsymbol{G}_V(s) = \boldsymbol{K}_V \cdot \boldsymbol{F}_V(s)$. The control law in its most general form known as ANDI (Actuator NDI) would be:

$$\Delta\boldsymbol{u}(s) = \boldsymbol{G}_A(s) \cdot B^{-1} \cdot (\boldsymbol{G}_V(s) \cdot \Delta\boldsymbol{v}(s) - \boldsymbol{d}_{\dot{x}}) \tag{5.26}$$

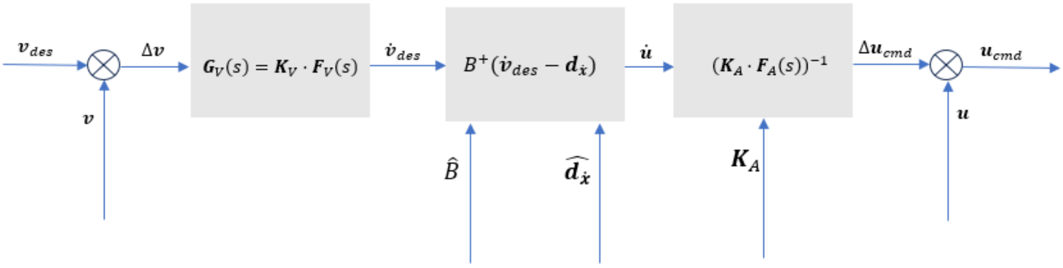


**Figure 5.4:** Extended INDI

The extended approach holds great promise, but its performance is strongly contingent on the precision of parameter estimation for the control effectors. Precise parameter estimation is a critical factor in the successful implementation and effectiveness of this approach.

In contrast, PCH in its basic form offers a less complex yet more straightforward method to address actuator limits. It requires minimal or no parameter estimation, making it a more accessible option for dealing with actuator constraints.

### 5.3.2 Delay prediction

Delay has the potential to jeopardize the stability of a closed-loop system when controlled using an incremental approach that relies on time scale separation.

One straightforward solution to mitigate the issues of time delay and the availability of angular accelerations is to employ a linear predictive filtering method (as proposed by Sieberling in [32]). It is essential for this filter to have minimal reliance on the mathematical model to align with what is the primary objective of INDI.

Mathematically the effect of non synchronized measurements can be described as follows:

$$\Delta \boldsymbol{u} = \boldsymbol{B}_0^{\dagger}(\boldsymbol{v} - \dot{\boldsymbol{x}}_{\tau}) \tag{5.27}$$

The control law will use measurements affected by lag. Consequently, the plant is not perfectly linearized:

$$\dot{\boldsymbol{x}} \approx \dot{\boldsymbol{x}}_0 + \boldsymbol{B}_0 \Delta \boldsymbol{u} = \dot{\boldsymbol{x}}_0 + \boldsymbol{B}_0 \boldsymbol{B}_0^{\dagger}(\boldsymbol{v} - \dot{\boldsymbol{x}}_{\tau}) \tag{5.28}$$

Yielding:

$$\dot{\boldsymbol{x}} = \dot{\boldsymbol{x}}_0 + \boldsymbol{v} - \dot{\boldsymbol{x}}_\tau \quad \rightarrow \quad \dot{\boldsymbol{x}}_{\Delta\tau} = \dot{\boldsymbol{x}}(t_0) - \dot{\boldsymbol{x}}(t_0 - \tau) \tag{5.29}$$

The predictive filter approach tries to tackle the issue of time synchronization by predicting the variable one time-step ahead rather then delaying all the variables through the use of a common lag filter.

The mathematical expression for a linear predictive filter is as follows:

$$\dot{x}_0 = \sum_{i=1}^{n} [\theta_x x_{t_0 - i\tau} + \theta_r r_{t_0 - i\tau}] + \epsilon \tag{5.30}$$

The value for $\dot{x}$ at the current point in time $t_0$ is a linear function of $n$ precedent measurements of $x$ and $n$ precedent reference values $r = x_{ref}$. The parameters $\theta$ can be found through Least Square Estimation using a large set of data.

A more recent approach, as presented by Steffensen et al. in [33], offers promise in addressing both delay and synchronization issues simultaneously. This approach aims to obtain the output derivative when it's not readily available through conventional IMU systems.
In this approach, appropriately filtered measurements are integrated with data from an online model of the plant, as illustrated below:

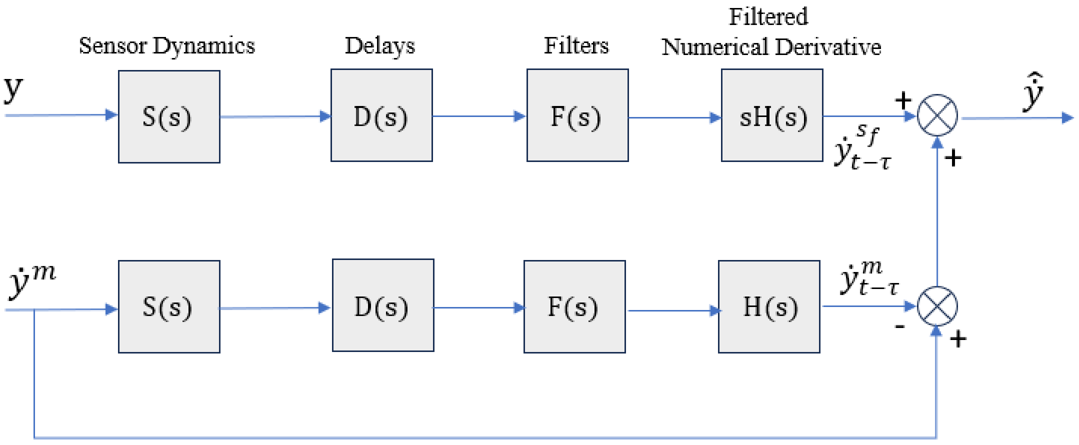


**Figure 5.5:** Complementary Filter

The estimate of the output derivative $\hat{\dot{y}}$ is a linear combination of the delayed sensor measurement (appropriately filtered) $\dot{y}^{sf}_{t-\tau}$, and the difference between a non delayed estimate of $\dot{y}$ from an on board mathematical model of the plant and the same quantity delayed by the same $\Delta t$ as the original output $y$, resulting in:

$$\hat{\dot{y}}_t = \dot{y}_t^m + (\dot{y}_{t-\tau}^{s_f} - \dot{y}_{t-\tau}^m) \tag{5.31}$$

All the transfer functions involved need appropriate inizialization.

The purpose of this section is to demonstrate the full potential of an incremental approach in developing a versatile, robust, and straightforward control law that can be deployed across various platforms thanks to the minimization of its reliance on a complex mathematical model. We've shown how recent developments in incremental control theory highlight its suitability for real-world situations where actuators exhibit delays and measurements may not always be readily accessible or available at all.

In this study, a conventional INDI approach is employed, assuming that all the necessary quantities for feedback are accessible. The development of suitable estimation algorithms is not within the scope of this work and will be a subject of future research. Nonetheless, actuators will be incorporated into the study.

# Chapter 6

# Control System Derivation

This chapter marks the shift from theoretical control foundations to their practical application in the creation of a comprehensive control system. The focus here is on the development of dedicated controllers for MR, FW, and TR modes. Each of these controllers undergoes a meticulous evaluation with respect to their tracking performance and their ability to withstand wind disturbances.

## 6.1 Trajectory Generator

The design of a comprehensive guidance algorithm is not within the scope of this work. Therefore, a simple polynomial trajectory generator is employed through the Simulink block *Polynomial Trajectory*[34].

To create a third-order polynomial trajectory between two consecutive waypoints ($P_i$,$P_f$) with their corresponding timepoints ($t_i$,$t_f$) and boundary conditions for velocity ($V_i$,$V_f$), the following problem needs to be solved:

$$P(t) = At^3 + Bt^2 + Ct + D \;\; s.t. \;\; \begin{cases} P(t_i) = P_i = At_i^3 + Bt_i^2 + Ct_i + D \\ P(t_f) = P_i = At_f^3 + Bt_f^2 + Ct_f + D \\ \dot{P}(t_i) = V_i = 3At_i^2 + 2Bt_i + C \\ \dot{P}(t_f) = V_f = 3At_f^2 + 2Bt_f + C \end{cases} \tag{6.1}$$

Through the solution of the linear system that articulates the trajectory constraints, one can derive the complete set of coefficients $\{A, B, C, D\}$ that serve as the comprehensive definition for the polynomial trajectory. Once these coefficients are obtained, it becomes feasible to generate inertial position and velocity commands utilizing the trajectory generator. These commands subsequently establish the desired inertial trajectory for the vehicle to follow.

## 6.2 Multi Rotor mode

In MR mode the only active control is through lift propellers. This mode employs a direct force control approach, where the control vector consists of forces and moments generated by the propellers. These control inputs are subsequently converted into rotational speeds. The control vector for MR mode can be defined as follows:

$$\boldsymbol{u}_{MR} = [T_z, \tau_x, \tau_y, \tau_z]^T \tag{6.2}$$

During this phase, it's important to note that aerodynamic surfaces remain in the null position, and aerodynamic forces are solely dependent on the state vector $\boldsymbol{x}$. Furthermore, it's important to note that the aerodynamic database is applicable within a limited range of Angle of Attack (AoA). Hence, for simulation purposes, aerodynamic effects are exclusively taken into account during forward flight, whereas we assume that the aircraft is not influenced by airflow during vertical flight phases. Given the relatively low velocities during take-off and landing, this approximation remains reasonably accurate.

Defining the aerodynamic action vector as $\boldsymbol{A}(\boldsymbol{x}, \boldsymbol{u}) = [D, S, L, l_A, m_a, n_a]^T$, the following holds:

$$\boldsymbol{A}_A^T(\boldsymbol{x}, \boldsymbol{u}) = \begin{cases} [\boldsymbol{F}_A(\boldsymbol{x})^T, \boldsymbol{M}_A(\boldsymbol{x})^T] & if \ \ u \geq 3 \\ \boldsymbol{0}_{1\times 6} & if \ \ u < 3 \end{cases} \tag{6.3}$$

Where $\boldsymbol{V}^B \cdot \hat{i}_B = u = 3$ is assumed to be the condition for the beginning of forward motion. This condition is compatible with the mission profile to be analyzed, during which the MR mode is required to complete vertical and planar maneuvers separately. In case of combined motion, $\alpha$ should be used, instead.

### 6.2.1 Inner Loop

The inner loop, as derived in the subsequent subsection, remains consistent across all three states of the aerial platform. Variations in gains will solely apply to the MR mode. This implies that minimal gain scheduling is required and can be straightforwardly integrated into the controller.

The primary objective of the inner loop is to achieve a desired attitude expressed in terms of Euler angles, denoted as $\boldsymbol{\Phi} = [\phi, \theta, \psi]^T$. It is of paramount importance that the inner loop operates at a higher frequency than the outer loop. Indeed, in cascaded control with time scale separation, the entire inner loop should effectively act as an actuator for the outer loop.

Attitude is governed by the following set of equations:

$$\begin{cases} \dot{\boldsymbol{\Phi}} = H(\boldsymbol{\Phi})\boldsymbol{\omega}^B_{B/N} \\ \dot{\boldsymbol{\omega}}^B_{B/N} = -\boldsymbol{\omega}^B_{B/N} \times \boldsymbol{J}^B\boldsymbol{\omega}^B_{B/N} + (\boldsymbol{J}^B)^{-1}(-\boldsymbol{G}^B(\boldsymbol{x},\boldsymbol{u}) + \boldsymbol{M}^B_x(\boldsymbol{x}) + \boldsymbol{M}^B_u(\boldsymbol{u})) \end{cases} \tag{6.4}$$

The inner loop structure designed for attitude tracking involves the following strategy:

1. Initially, a comparison is made between the desired attitude and the estimated actual attitude to generate an error. This error is then transformed into the desired rate for Euler angles through the use of a linear controller.

2. The first set of equations establishes a kinematic constraint relating Euler rates to body angular velocities. Importantly, this equation is devoid of mathematical model uncertainties, allowing traditional NDI to be employed for generating desired body velocities as outputs.

3. A second linear controller is utilized to convert the error between desired and actual body angular rates into desired angular accelerations.

4. The output of the second linear controller can be employed to invert the second set of equations, facilitating the extraction of desired moments. It's worth noting that the dynamical equations for rotation do contain model uncertainties, necessitating the adoption of INDI to enhance the robustness of the inversion process.

As previously mentioned, in direct force control, the control vector contains moments and forces. Consequently, a control allocation algorithm will be introduced later as a means of effectively translating these moments and forces into actual actuator commands.

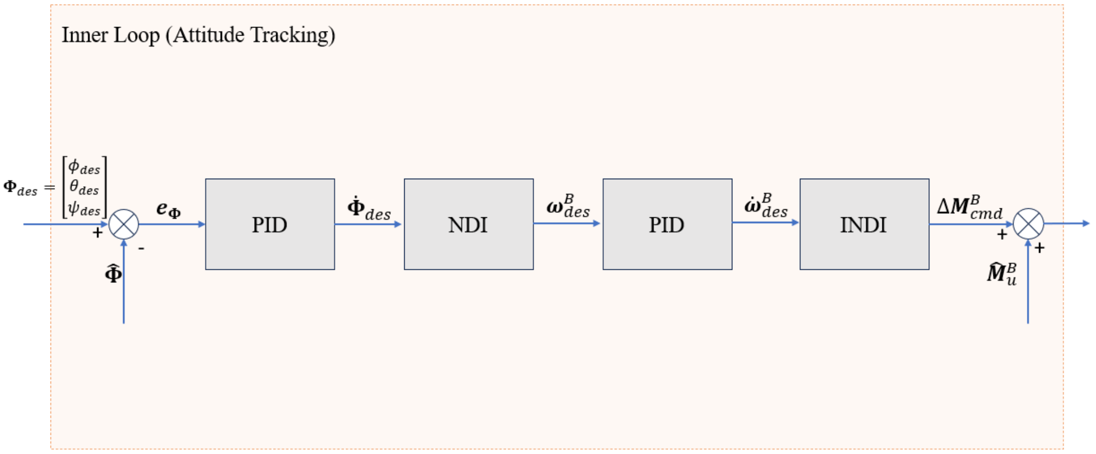

Inner Loop (Attitude Tracking)
$\boldsymbol{\Phi}_{des} = \begin{bmatrix}\phi_{des}\\ \theta_{des}\\ \psi_{des}\end{bmatrix}$
+
-
$\boldsymbol{e}_{\boldsymbol{\Phi}}$
$\hat{\boldsymbol{\Phi}}$
PID
$\dot{\boldsymbol{\Phi}}_{des}$
NDI
$\boldsymbol{\omega}^{B}_{des}$
PID
$\dot{\boldsymbol{\omega}}^{B}_{des}$
INDI
$\Delta\boldsymbol{M}^{B}_{cmd}$
+
+
$\hat{\boldsymbol{M}}^{B}_{u}$

**Figure 6.1:** Inner Loop

### INDI for rotational dynamics

Rewriting the rotational dynamics in terms of $\boldsymbol{x}$ and $\boldsymbol{u}$ as for eq. (3.40) and setting $\boldsymbol{u}_{FW}, \boldsymbol{u}_{HT} = \boldsymbol{0}$, the following holds:

$$\dot{\boldsymbol{x}}_3 = (\mathbf{J}^B)^{-1}(-\boldsymbol{x}_3 \times \mathbf{J}^B\boldsymbol{x}_3 + \boldsymbol{M}^B_A(\boldsymbol{x}_1, \boldsymbol{x}_3) + \boldsymbol{M}^B_T(\boldsymbol{u}_{VT}) - \boldsymbol{G}^B(\boldsymbol{x}_3, \boldsymbol{u}_{VT})) \tag{6.5}$$

Gyroscopic effects are small compared to control actions , so the equation can be furthered simplified into:

$$\mathbf{J}^B\dot{\boldsymbol{x}}_3 = \underbrace{-\boldsymbol{x}_3 \times \mathbf{J}^B\boldsymbol{x}_3 + \boldsymbol{M}^B_A(\boldsymbol{x}_1, \boldsymbol{x}_3)}_{\boldsymbol{f}(\boldsymbol{x})} + \underbrace{\boldsymbol{M}^B_T(\boldsymbol{u}_{VT})}_{\boldsymbol{u}=[u_2,u_3,u_4]^T} \tag{6.6}$$

Considering the moment produced by propellers as the control input itself (direct force control) and expanding (6.6) as a Taylor series yields:

$$\mathbf{J}^B\dot{\boldsymbol{x}}_3 = \mathbf{J}^B(\dot{\boldsymbol{x}}_3)_0 + \frac{\partial \boldsymbol{f}(\boldsymbol{x}_0)}{\partial \boldsymbol{x}}\Delta\boldsymbol{x} + \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{u}}\Delta\boldsymbol{u} + O(\Delta\boldsymbol{x}^2, \Delta\boldsymbol{u}^2) \approx \mathbf{J}^B(\dot{\boldsymbol{x}}_3)_0 + \Delta\boldsymbol{u} \tag{6.7}$$

Defining the virtual input $\boldsymbol{v} = \dot{\boldsymbol{x}}_3$ we can finally perform the inversion:

$$\Delta\boldsymbol{u} = \mathbf{J}^B(\boldsymbol{v} - \dot{\boldsymbol{x}}_{3_0}) \tag{6.8}$$

Assuming that (6.8) perfectly linearizes the plant, we can follow the same procedure as for (5.7) in order to obtain the new linear system to be controlled:

$$\boldsymbol{\mu} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad \rightarrow \quad \dot{\boldsymbol{\mu}} = \mathbb{A}\boldsymbol{\mu} + \mathbb{B}\boldsymbol{v} = \mathbb{I}_{3\times 3}\boldsymbol{v} \tag{6.9}$$

The linear controller chosen in this work is a classical PID in the Simulink block form, i.e. with the addition of a low pass filter for noise attenuation during error differentiation. Introducing the error dynamics for $\boldsymbol{e}_\omega = \boldsymbol{\omega}_{des} - \boldsymbol{\omega}$ into eq. (6.9), the system becomes:

$$\dot{\boldsymbol{\mu}} = \boldsymbol{v} = PID(\boldsymbol{e}_\omega) = \boldsymbol{K}_P \boldsymbol{e}_\omega(t) + \boldsymbol{K}_D \dot{\boldsymbol{e}}_\omega(t) + \boldsymbol{K}_I \int_{t_0}^{\tau} \boldsymbol{e}_\omega(\tau) d\tau \tag{6.10}$$

$\boldsymbol{K}_P$,$\boldsymbol{K}_I$,$\boldsymbol{K}_D$ are diagonal gain matrices in $\mathbb{R}^{3\times 3}$ whose elements $K_{ii}$ are selected through an appropriate tuning process. In this work the empirical Ziegler-Nichols method was used as a starting point, followed by trial and error fine tuning to obtain desired responses to step and sinusoidal inputs.

As mentioned in the past chapter, angular accelerations are not available through conventional IMUs. However, a complementary filter can be used for solving both the acceleration estimate and synchronization issues.

In this work, gyroscopic effects were considered to be negligible in the development of the controller. This assumption is especially valid due to the incremental nature of the control law. However, if gyro effects need to be included, the control law can be easily modified into [24]:

$$\boldsymbol{J}^B \Delta \dot{\boldsymbol{\omega}}^B_{B/N} = \frac{\partial(\boldsymbol{G}^B + \boldsymbol{M}^B_T)(\boldsymbol{\Omega}_{v0})}{\partial \boldsymbol{\Omega}_v} \Delta \boldsymbol{\Omega}_v \tag{6.11}$$

Recalling from the mathematical model that torques are linked to $\boldsymbol{\Omega}_v^2$ via a constant allocation matrix ($\boldsymbol{M}_{al}$), and taking into account the expression of gyroscopic effects, we can deduce:

$$\boldsymbol{J}^B \Delta \dot{\boldsymbol{\omega}}^B_{B/N} = \underbrace{(2\boldsymbol{M}_{al} diag[\boldsymbol{\Omega}_{v0}] - (\tilde{\boldsymbol{\omega}}^B_{B/N})_0 \boldsymbol{J}^B_R \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix})}_{\boldsymbol{B}=\text{Control Effectiveness Matrix}} \Delta \boldsymbol{\Omega}_v \tag{6.12}$$

In this case a pseudo-inverse is needed since the system is not square.

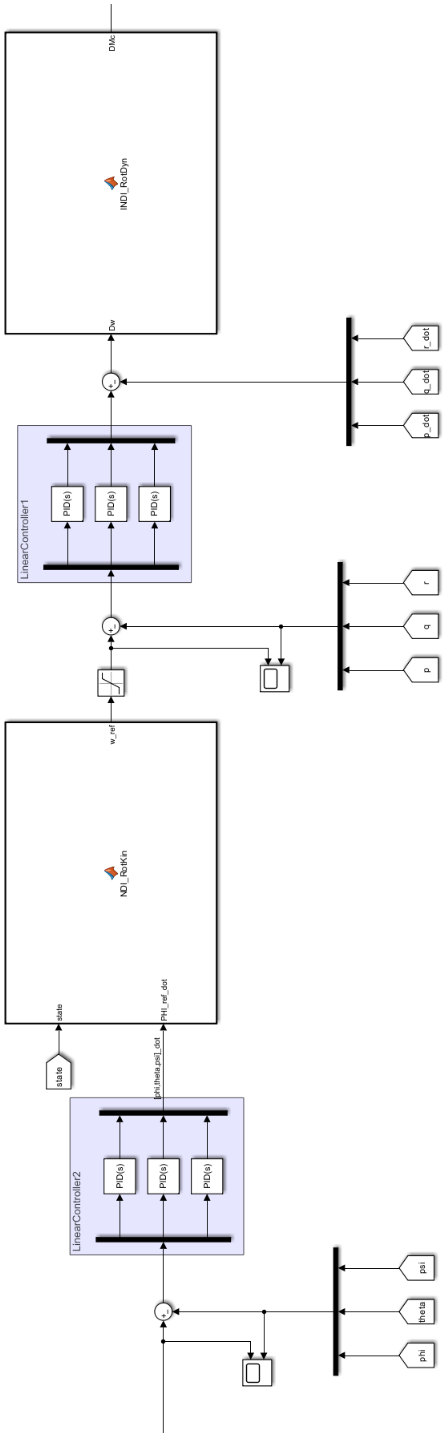


**Figure 6.2:** Inner Loop Simulink model

The gains selected for the linear controllers shown in table below.

| | $\boldsymbol{K_P}$ | $\boldsymbol{K_I}$ | $\boldsymbol{K_D}$ | $\boldsymbol{N}$ |
|---|---|---|---|---|
| $\boldsymbol{p}$ | 48 | 0 | 2.4 | 100 |
| $\boldsymbol{q}$ | 60 | 0 | 3 | 100 |
| $\boldsymbol{r}$ | 10 | 0 | 1.5 | 100 |

**Table 6.1:** Gains for body angular rate PID controllers

First the response of the inner loop to $\boldsymbol{\omega}^B_{(B/N)_{cmd}}$ inputs is presented.

1. The initial plot illustrates the response to two impulses: one at $+15\frac{deg}{s}$ and the other at $-10\frac{deg}{s}$. The desired angular rate is rapidly and precisely attained.

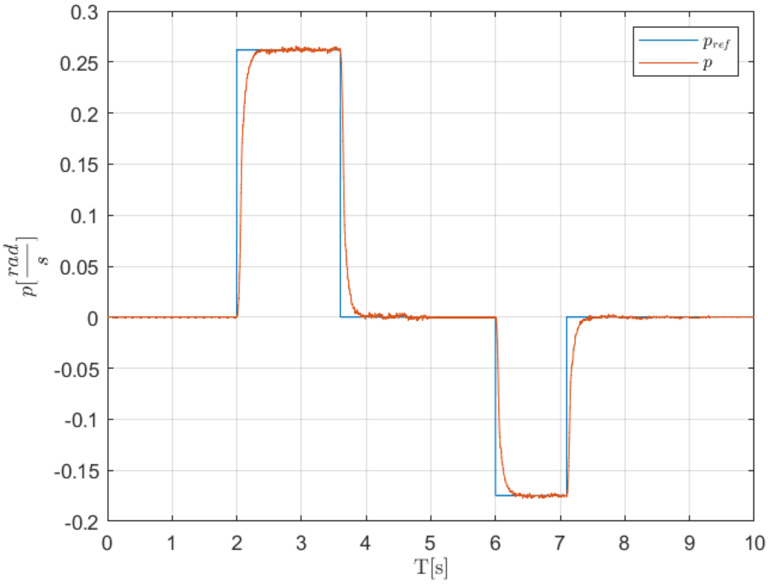


**Figure 6.3:** Response to double impulse $p_{des}$

2. Next the system response to a sinusoidal input is presented. The input is selected as: $p_{des}(t) = 17\frac{\pi}{180}sin(1.3t + 1)$. Once again, the controller is able to perfectly track the input.

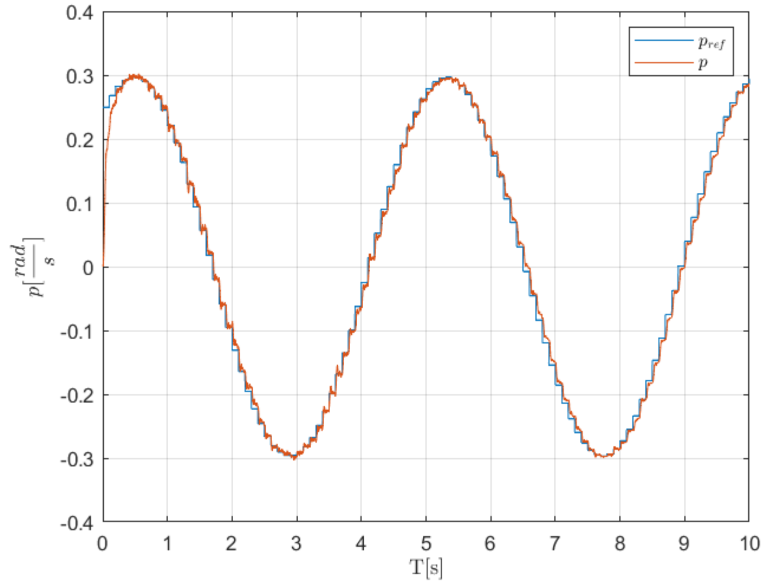


**Figure 6.4:** Response to sinusoidal $p_{des}$

Subsequently, responses to $q_{des}$ and $r_{des}$ are shown.

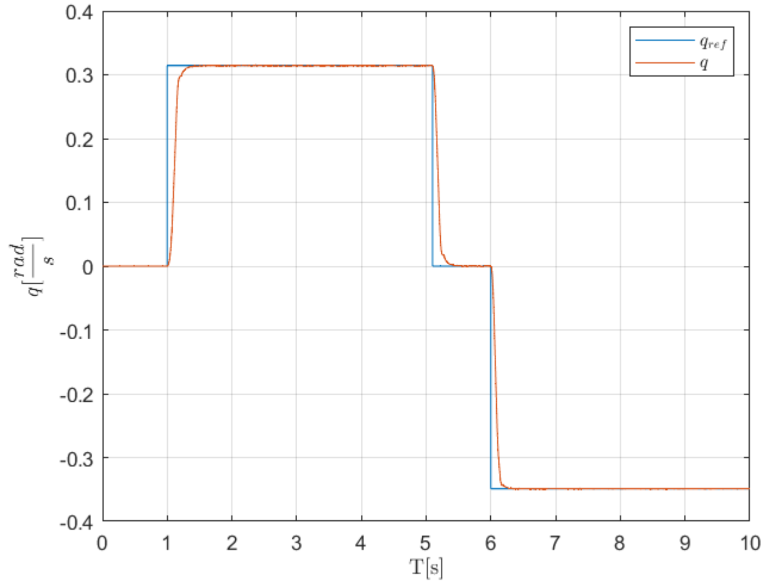


**Figure 6.5:** Response to double impulse $q_{des}$

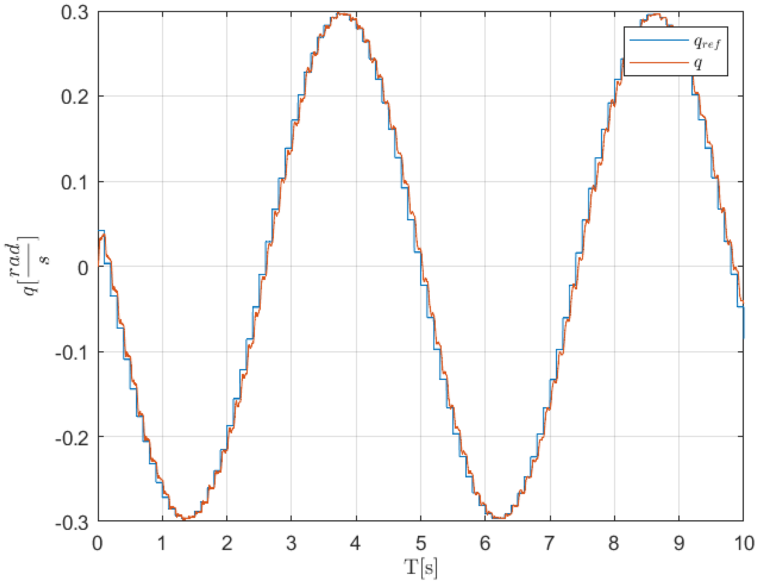


**Figure 6.6:** Response to sinusoidal $p_{des}$

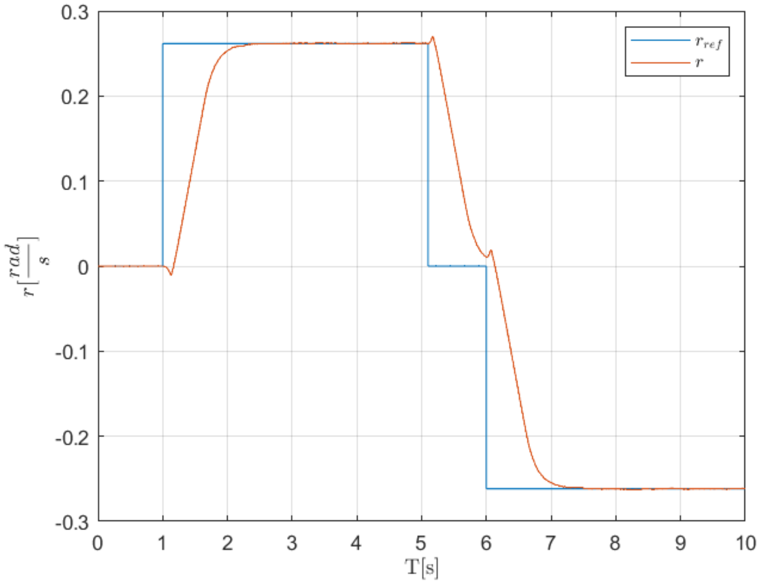


**Figure 6.7:** Response to double impulse $r_{des}$

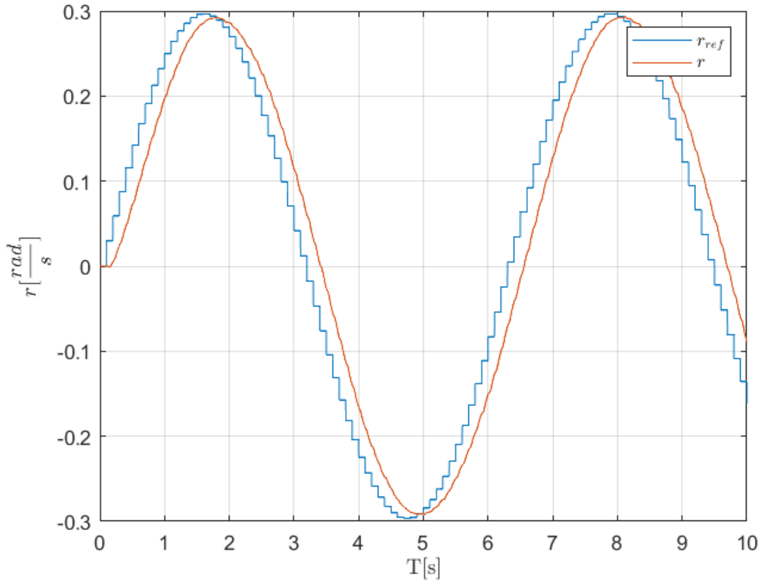


**Figure 6.8:** Response to sinusoidal $r_{des}$

### NDI for attitude kynematics

As stated earlier, for rotational kinematics a classical NDI approach can be used since no model uncertainties are present [26]. Rewriting the kinematic constraint in terms of state and control variables we get:

$$\dot{\boldsymbol{x}}_2 = H(\boldsymbol{x}_2)\boldsymbol{x}_3 \tag{6.13}$$

For this set of equations, $\boldsymbol{x}_3 = [p, q, r]^T$ acts as an input, while $\boldsymbol{x}_2 = \boldsymbol{\Phi} = [\phi, \theta, \psi]^T$ is the output. Applying NDI to (6.13), we obtain:

$$\boldsymbol{x}_3 = H(\boldsymbol{x}_2)^{-1}\boldsymbol{v} \tag{6.14}$$

$H(\boldsymbol{x}_2)$ is the *decoupling matrix* of this subsystem and allows us to obtain the new linear decoupled system to be controlled as:

$$\boldsymbol{\mu} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \quad \rightarrow \quad \dot{\boldsymbol{\mu}} = \mathbb{A}\boldsymbol{\mu} + \mathbb{B}\boldsymbol{v} = \mathbb{I}_{3\times3}\boldsymbol{v} \tag{6.15}$$

Like for INDI, a PID controller is used to drive the error $\boldsymbol{e}_\Phi = \boldsymbol{\Phi}_{des} - \boldsymbol{\Phi}$ to zero.

The same tuning method as for INDI is also used for this second set of PID controllers and yields the gains listed in the table below:

| | $\boldsymbol{K}_P$ | $\boldsymbol{K}_I$ | $\boldsymbol{K}_D$ | $\boldsymbol{N}$ |
|---|---|---|---|---|
| $\boldsymbol{\phi}$ | 2 | 0 | 0.1 | 100 |
| $\boldsymbol{\theta}$ | 3 | 0 | 0.12 | 50 |
| $\boldsymbol{\psi}$ | 1.6 | 0 | 0.15 | 100 |

**Table 6.2:** Gains for Euler Angles PID controllers

Next, tracking performance for step and step + sine wave commands for Euler angles are shown.

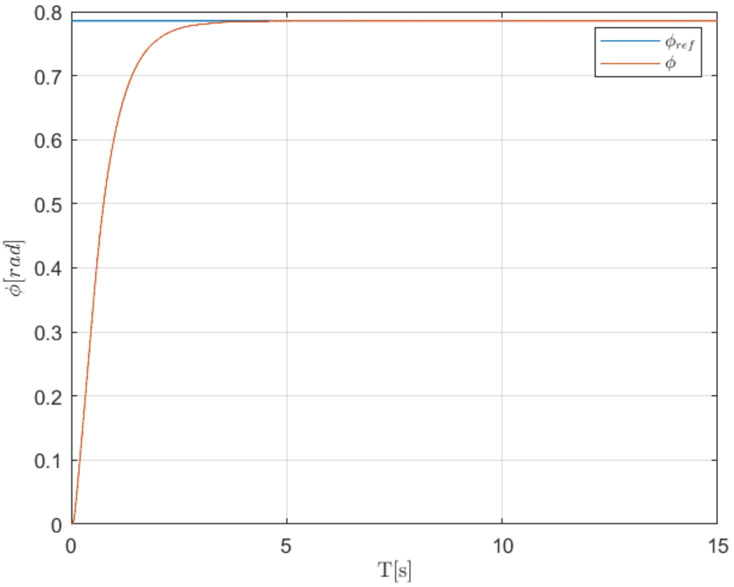


**Figure 6.9:** Tracking step $\phi_{ref}$

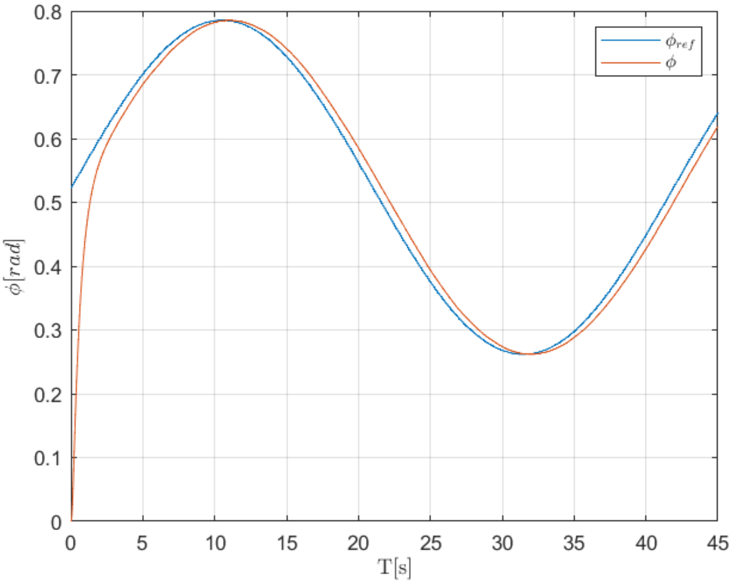


**Figure 6.10:** Tracking step + sine wave $\phi_{ref}$

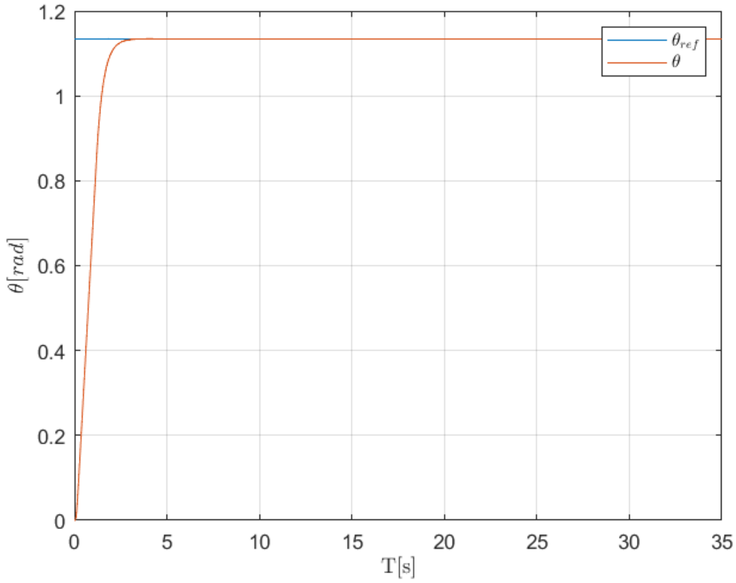


**Figure 6.11:** Tracking step $\theta_{ref}$

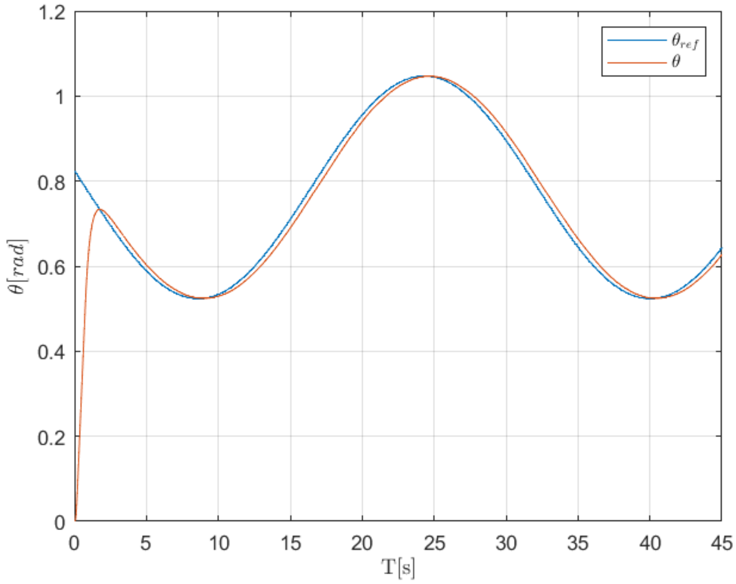


**Figure 6.12:** Tracking step + sine wave $\theta_{ref}$

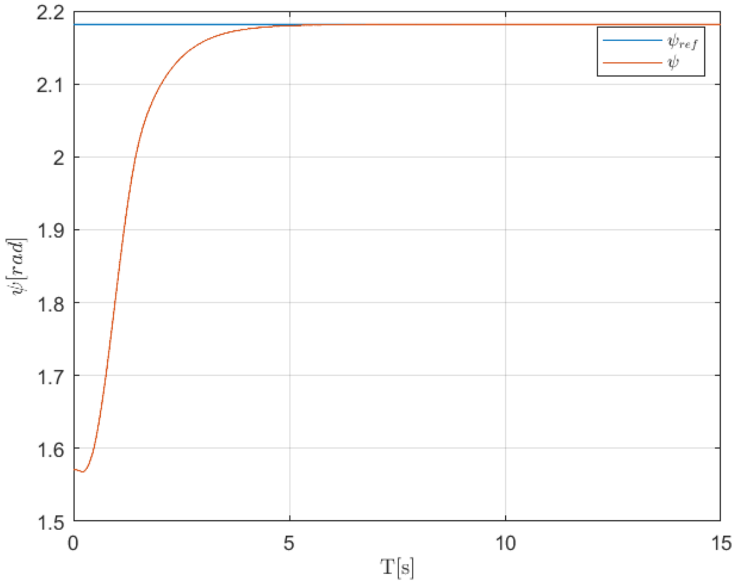


**Figure 6.13:** Tracking step $\psi_{ref}$

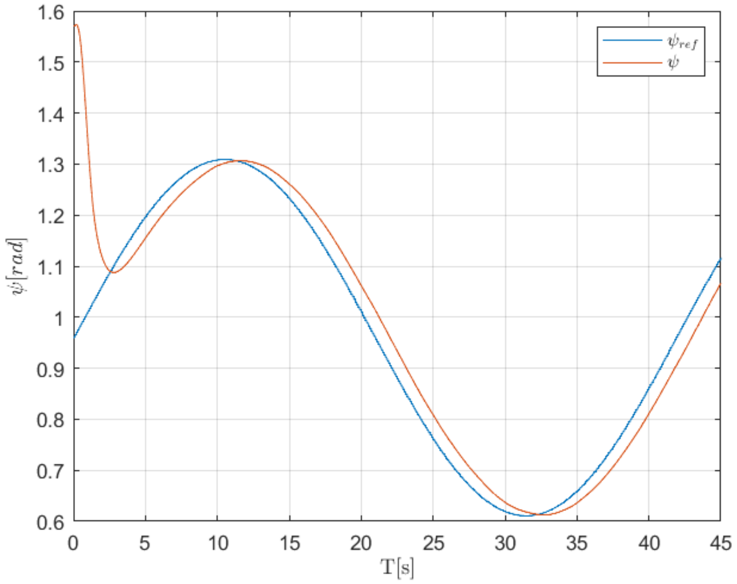


**Figure 6.14:** Tracking step + sine wave $\psi_{ref}$

### 6.2.2 Outer Loop

The outer loop control system is primarily responsible for following a desired trajectory in terms of inertial position. To achieve this, it receives velocity commands from a guidance algorithm and subsequently provides desired attitude to the inner loop and desired forces to an allocation algorithm.

From a mathematical perspective, the dynamical equations for linear motion are employed, after some manipulation. The vectorial equation for linear dynamics can be rewritten in terms of inertial accelerations as:

$$\boldsymbol{A}_i^N = \ddot{\boldsymbol{P}}^N = \begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} = \frac{1}{m}(C_{N/B}\boldsymbol{F}_T^B + m\boldsymbol{g}^N + C_{N/W}\boldsymbol{F}_W^W) \tag{6.16}$$

At low speeds, we can assume aerodynamic forces to be negligible and the expanded version of equation (6.16) becomes:

$$\ddot{\boldsymbol{P}}^N = \frac{1}{m}C_{N/B}(\boldsymbol{\Phi}) \begin{bmatrix} 0 \\ 0 \\ -T_z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{6.17}$$

The outer loop structure designed for trajectory tracking involves the following strategy:

1. A guidance algorithm generates references for inertial position and velocity. These references are used as inputs for a two-PID structure, which calculates desired inertial accelerations by considering position and velocity error functions.

2. The desired accelerations, obtained from the double PID controller, serve as a virtual input for the dynamical equations. As these equations are subject to model uncertainties, an incremental inversion method is employed. This results in the determination of desired attitude and desired forces.

3. The yaw angle $\psi$ does not appear in the rearranged version of equation 5.15 used for inversion, therefore its reference has to be computed differently. Instead of inputting $\psi$ into the inner loop, the azimuth angle $\chi_{ref}$ is injected. This choice is made because $\chi_{ref}$ can be readily computed from reference velocities, providing a more straightforward way to determine the desired yaw angle $\psi$.

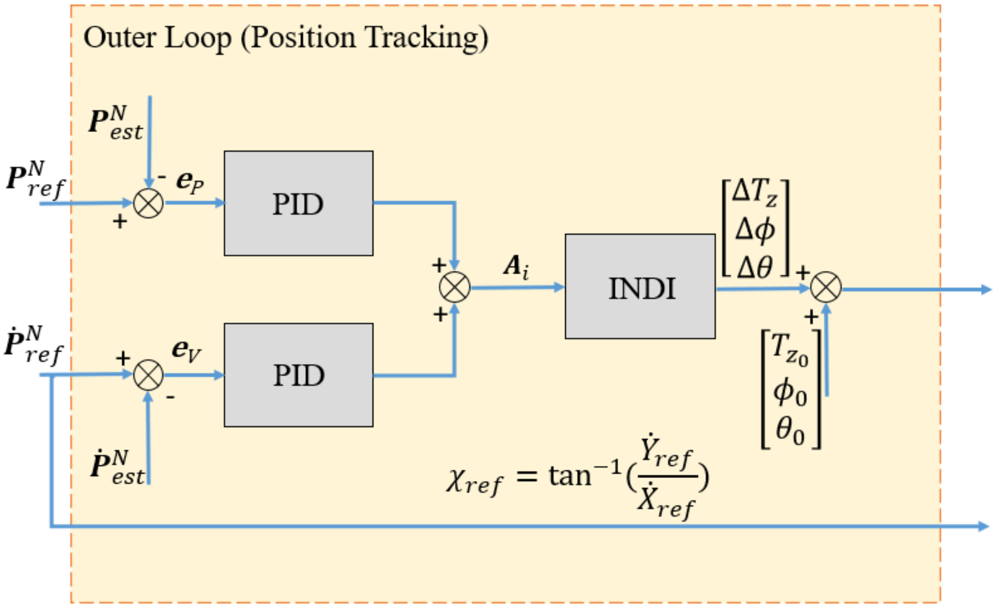

Outer Loop (Position Tracking)
$P^N_{est}$
$P^N_{ref}$
- $e_P$
+
PID
+
$A_i$
INDI
+
$\dot{P}^N_{ref}$
+ $e_V$
PID
-
$\dot{P}^N_{est}$
$\begin{bmatrix}\Delta T_z\\ \Delta\phi\\ \Delta\theta\end{bmatrix}$
+
+
$\begin{bmatrix}T_{z_0}\\ \phi_0\\ \theta_0\end{bmatrix}$
$\chi_{ref} = \tan^{-1}(\frac{\dot{Y}_{ref}}{\dot{X}_{ref}})$

**Figure 6.15:** Outer Loop

### INDI for translational dynamics

In order to derive the control system, the first two equations belonging to (**??**) need to be rotated in their body components through $C_{B/N}$ :

$$\boldsymbol{A}_i^B = C_{B/N}(\boldsymbol{\Phi})\begin{bmatrix}\ddot{X}\\ \ddot{Y}\\ \ddot{Z}\end{bmatrix} = C_{B/N}(\boldsymbol{\Phi})\begin{bmatrix}0\\0\\g\end{bmatrix} + \begin{bmatrix}0\\0\\-T_z\end{bmatrix} \tag{6.18}$$

Writing explicitly the first two equations yields:

$$\begin{cases}\ddot{X}^B = -g\sin\theta\\ \ddot{Y}^B = g\cos\theta\sin\phi\end{cases} \tag{6.19}$$

For this 2 × 2 system of non linear equations, $\Phi$ represents the output, while the inertial acceleration components, appropriately transformed in body coordinates, can be thought of as the input. The system is clearly non affine in $\Phi$, but INDI easily solves the issue as follows [35]:

$$\begin{cases}\ddot{X}^B = \ddot{X}_0^B - g\cos\theta_0\Delta\theta\\ \ddot{Y}^B = \ddot{Y}_0^B - g\sin\theta_0\sin\phi_0\Delta\theta + g\cos\theta_0\cos\phi_0\Delta\phi\end{cases} \tag{6.20}$$

To complete the outer loop we need to include a third equation for vertical acceleration. This time, the equation is kept in NED coordinates as follows[35]:

$$\ddot{Z} = g + C_{N/B}(\boldsymbol{\Phi})(3{,}3)(-T_z) = \underbrace{g - \frac{1}{m}\cos\phi\cos\theta\, T_z}_{f(\boldsymbol{\Phi},u_1)} \tag{6.21}$$

According to the time scale separation principle, in a short period of time, the effect of the actual output $T_z$ will be much more impactful than the influence of the Euler angles, therefore we can reduce eq. (6.21) to:

$$\ddot{Z} \approx \ddot{Z}_0 + \frac{\partial f(\boldsymbol{\Phi}_0, u_{1_0})}{\partial \boldsymbol{\Phi}}\Delta\boldsymbol{\Phi} + \frac{\partial f(\boldsymbol{\Phi}_0, u_{1_0})}{\partial u_1}\Delta u_1 \approx \ddot{Z}_0 - \frac{1}{m}\cos\phi_0\cos\theta_0\,\Delta u_1 \tag{6.22}$$

The incremental equations obtained through a Taylor first order expansion can be grouped into one matrix equation:

$$\underbrace{\Delta\begin{bmatrix}\ddot{X}^B\\ \ddot{Y}^B\\ \ddot{Z}^N\end{bmatrix}}_{(\boldsymbol{v}-\dot{\boldsymbol{x}}_0)} = \underbrace{\begin{bmatrix}-g\cos\theta_0 & 0 & 0\\ -g\sin\theta_0\sin\phi_0 & g\cos\theta_0\cos\phi_0 & 0\\ 0 & 0 & \frac{1}{m}\cos\phi_0\cos\theta_0\end{bmatrix}}_{\boldsymbol{B}\,=\,\text{Control Effectiveness Matrix}}\underbrace{\begin{bmatrix}\Delta\theta\\ \Delta\phi\\ \Delta T_z\end{bmatrix}}_{\Delta\boldsymbol{u}} \tag{6.23}$$

It must be noted that desired accelerations are provided by the PID controller in NED axes, which means they need to be rotated in their respective body components. Assuming that the inner loop is able to track attitude faster than position variations occur, such rotation can be applied using current estimated Euler angles. Should that not be the case, INDI can still be applied to the dynamical equations in NED components as a whole, resulting just in a slightly more complicated *Control Effectiveness Matrix.*

The desired accelerations are determined through the combination of two PID contributions. The first contribution focuses on correcting velocity errors ($\boldsymbol{e}_P = \boldsymbol{P}^N_{ref} - \boldsymbol{P}^N$), while the second contribution is tasked with minimizing position errors ($\boldsymbol{e}_V = \dot{\boldsymbol{P}}^N_{ref} - \dot{\boldsymbol{P}}^N$):

$$\ddot{\boldsymbol{P}}^N_{ref} = \begin{bmatrix}\boldsymbol{K}_{P_P} & \boldsymbol{K}_{P_V}\end{bmatrix}\begin{bmatrix}\boldsymbol{e}_P\\ \boldsymbol{e}_V\end{bmatrix} + \begin{bmatrix}\boldsymbol{K}_{D_P} & \boldsymbol{K}_{D_V}\end{bmatrix}\begin{bmatrix}\dot{\boldsymbol{e}}_P\\ \dot{\boldsymbol{e}}_V\end{bmatrix} + \begin{bmatrix}\boldsymbol{K}_{I_P} & \boldsymbol{K}_{I_V}\end{bmatrix}\begin{bmatrix}\int \boldsymbol{e}_P\\ \int \boldsymbol{e}_V\end{bmatrix} \tag{6.24}$$

Position error control allows the UAV (drone) to make corrections not only when there is a deviation in speed but also when there is a positional offset from the desired location. This means that the drone will adjust its acceleration and trajectory to quickly bring itself back to the desired position, even if its speed already matches the desired velocity.

As previously mentioned, the Euler angle $\psi$ is not directly considered during inversion, which means its reference value needs to be determined differently. In this context, yaw primarily serves to correct off-track errors. Therefore, the azimuth angle can be leveraged as a reference value for calculating the appropriate angular rate of yaw, denoted as $\dot{\psi}$, in the inner loop:

$$\chi_{ref} = \tan^{-1}(\frac{\dot{Y}_{ref}}{\dot{X}_{red}}) \tag{6.25}$$

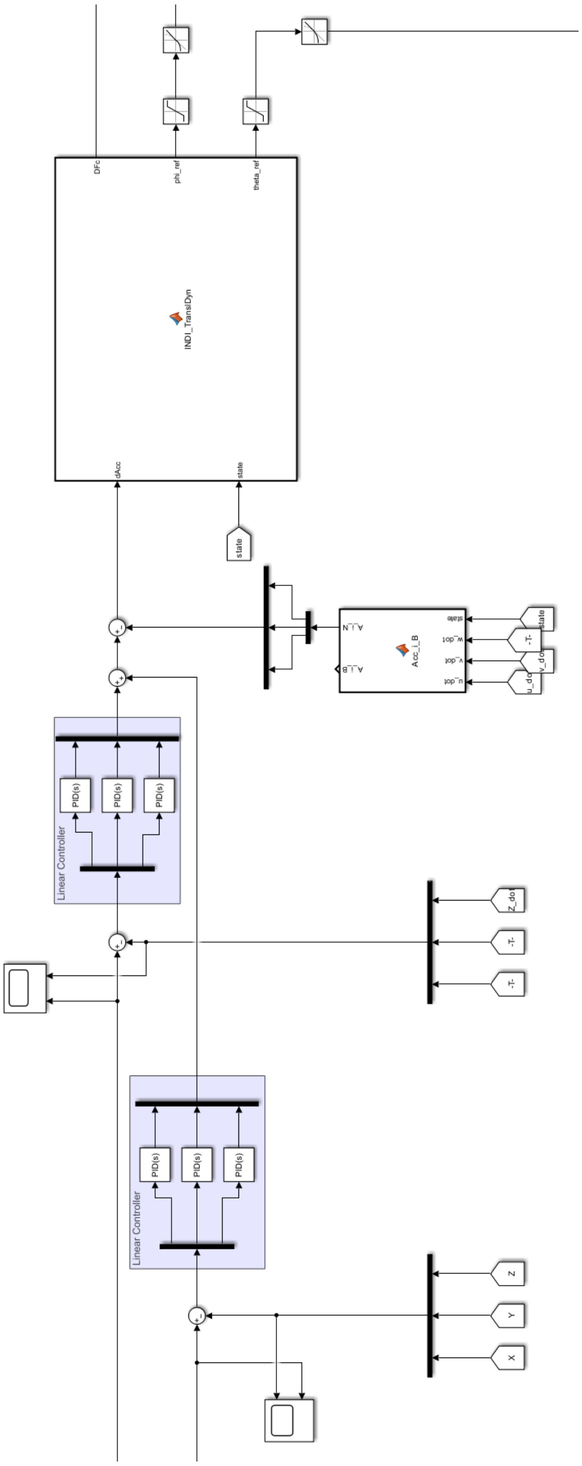


**Figure 6.16:** Outer Loop Simulink model

Through the same tuning approach as earlier, the following gains are obtained:

| | $\boldsymbol{K_P}$ | $\boldsymbol{K_I}$ | $\boldsymbol{K_D}$ | $\boldsymbol{N}$ |
|---|---|---|---|---|
| $\boldsymbol{X}$ | 0.6 | 0 | 1 | 100 |
| $\boldsymbol{Y}$ | 0.6 | 0 | 0.01 | 100 |
| $\boldsymbol{Z}$ | 15 | 0 | 0 | 0 |
| $\boldsymbol{\dot{X}}$ | 1.3 | 0 | 0.5 | 100 |
| $\boldsymbol{\dot{Y}}$ | 1.2 | 0 | 0.7 | 100 |
| $\boldsymbol{\dot{Z}}$ | 10 | 0.2 | 0.005 | 100 |

**Table 6.3:** Gains for inertial velocity and position PIDs

Next, the performance of velocity tracking is shown for a planar maneuver and a vertical one.

1. Planar maneuver composed of a step in $\dot{X}$ and a step + sine wave in $\dot{Y}$:

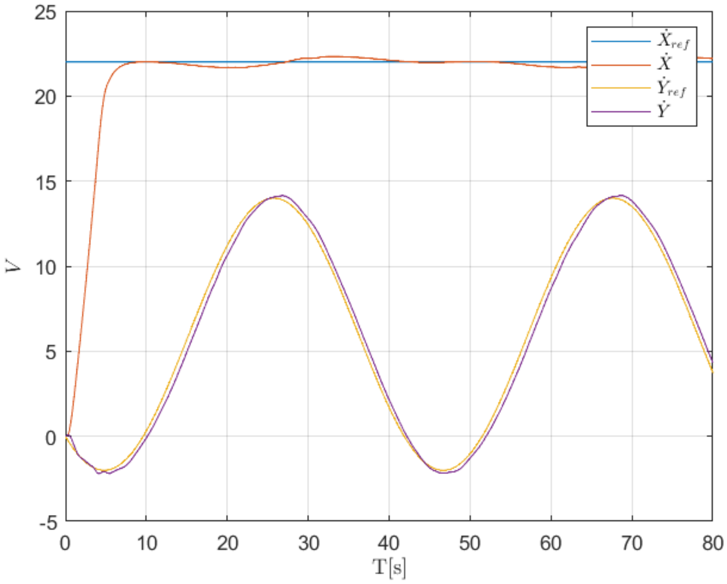


**Figure 6.17:** Tracking $\dot{X}_{ref}$ and $\dot{Y}_{ref}$

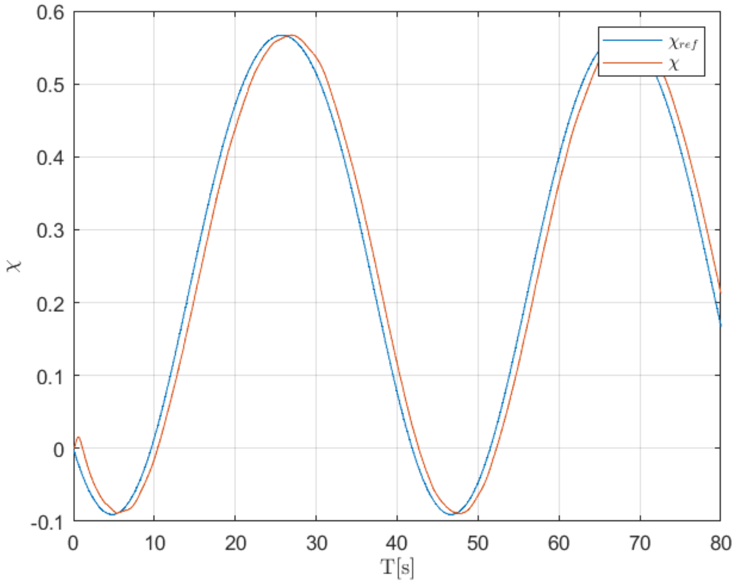


**Figure 6.18:** Tracking $\chi_{ref}$

2. Vertical maneuver composed of a step + sine wave in $\dot{Z}$:

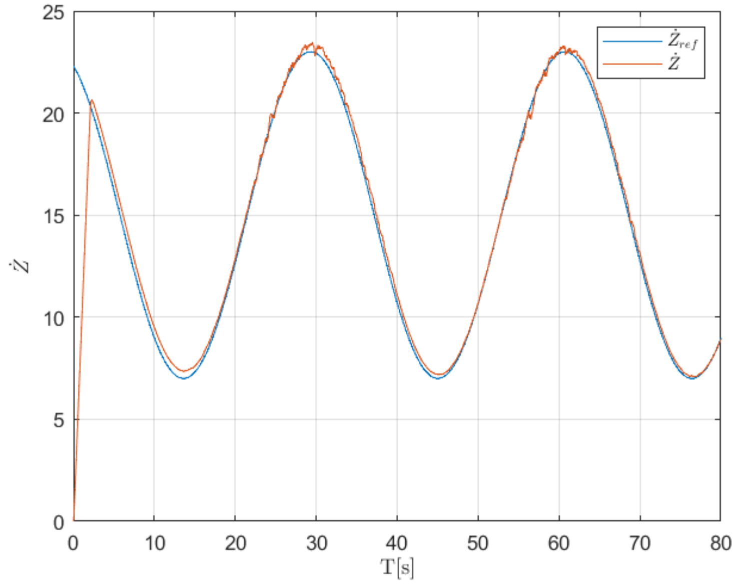


**Figure 6.19:** Tracking $\dot{Z}_{ref}$

To assess the robustness of the controller to wind inputs, the equations of motion need to include wind speed. In that case we can use the chain rule for linear velocities, i.e. we can consider the inertial speed of the aircraft as the sum of its relative speed w.r.t. the wind vector ($\boldsymbol{V}_{cm/W}$),also used for aerodynamic actions, and the velocity of wind w.r.t. to the inertial frame ($\boldsymbol{V}_{W/E}$) [12]:

$$\boldsymbol{V}_{cm/G} = \boldsymbol{V}_{cm/W} + \boldsymbol{V}_{W/E} \tag{6.26}$$

If only steady wind is considered, the dynamics of the aircraft becomes:

$$\dot{\boldsymbol{V}}^B_{cm/W} = \frac{1}{m}\boldsymbol{F}^B + C_{B/N}(\boldsymbol{\Phi})\boldsymbol{g}^N - \boldsymbol{\omega}^B_{B/N} \times \boldsymbol{V}^B_{cm/W} \tag{6.27}$$

Results for the following wind input in NED coordinates in $\frac{m}{s}$ are shown:

$$\begin{cases} V_{W_X} = 7.5 \\ V_{W_Y} = pulse(3,3,18) + pulse(-2,50,20) \\ V_{W_Z} = 1 \end{cases} \tag{6.28}$$

The arguments of the *pulse* function are respectively amplitude, starting time and duration.

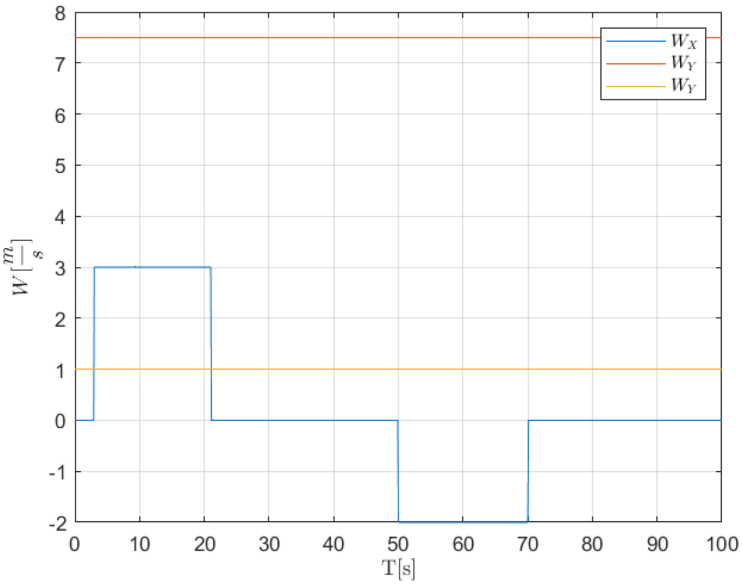


**Figure 6.20:** Wind velocities in NED axes

The aircraft response in terms of inertial position and Euler angles is:

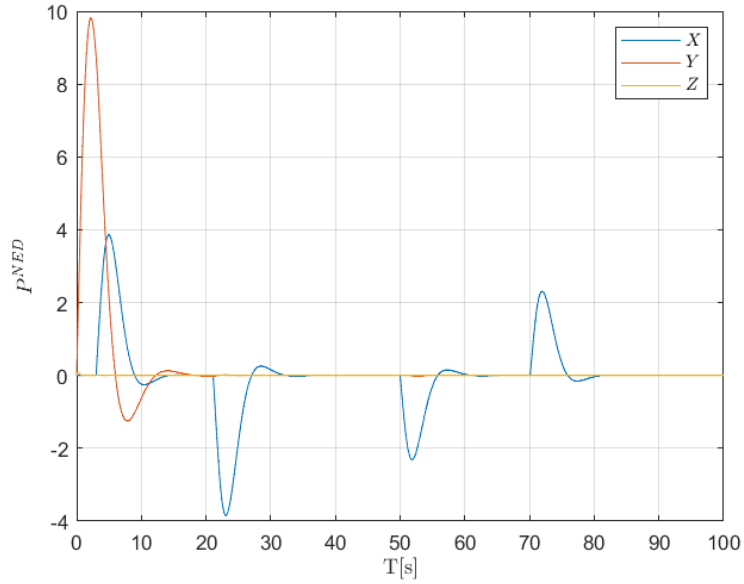


**Figure 6.21:** A/C inertial displacement

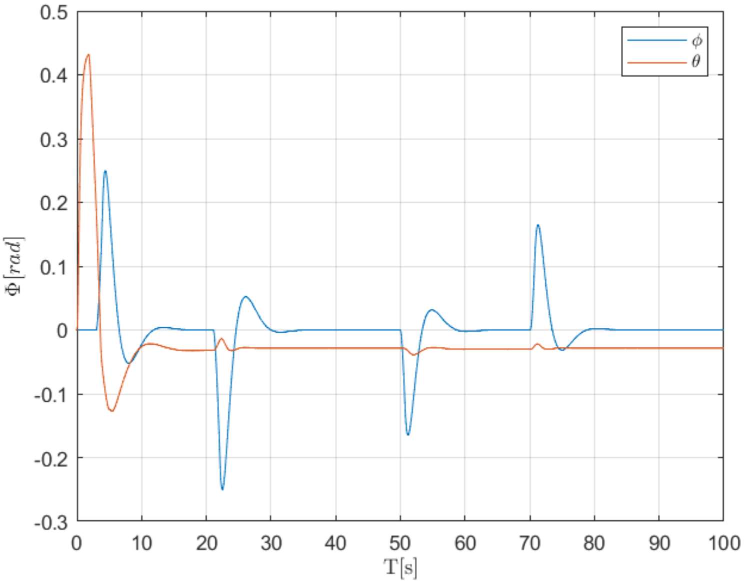


**Figure 6.22:** Euler angles

Despite the influence of wind, the UAV demonstrates a rapid recovery to its initial (zero) position. Furthermore, Figure 6.22 illustrates the stabilization of the pitch angle ($\theta$) to a non-zero value. This adjustment is necessary as the UAV needs to pitch forward to generate horizontal thrust, counteracting the aerodynamic forces introduced by the external wind.

Finally, a simulation for a simple waypoint-tracking mission is shown, with some wind disturbances acting on the system.

The matrix of waypoints to be followed is given in NEU (North-East-Up) coordinates:

$$\boldsymbol{WPTs}^N = \begin{bmatrix} \boldsymbol{N}^T \\ \boldsymbol{E}^T \\ \boldsymbol{H}^T \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 305 & 610 & 610 & 610 \\ 0 & 500 & 1000 & 1000 & 1000 & 500 & 0 \end{bmatrix} \tag{6.29}$$

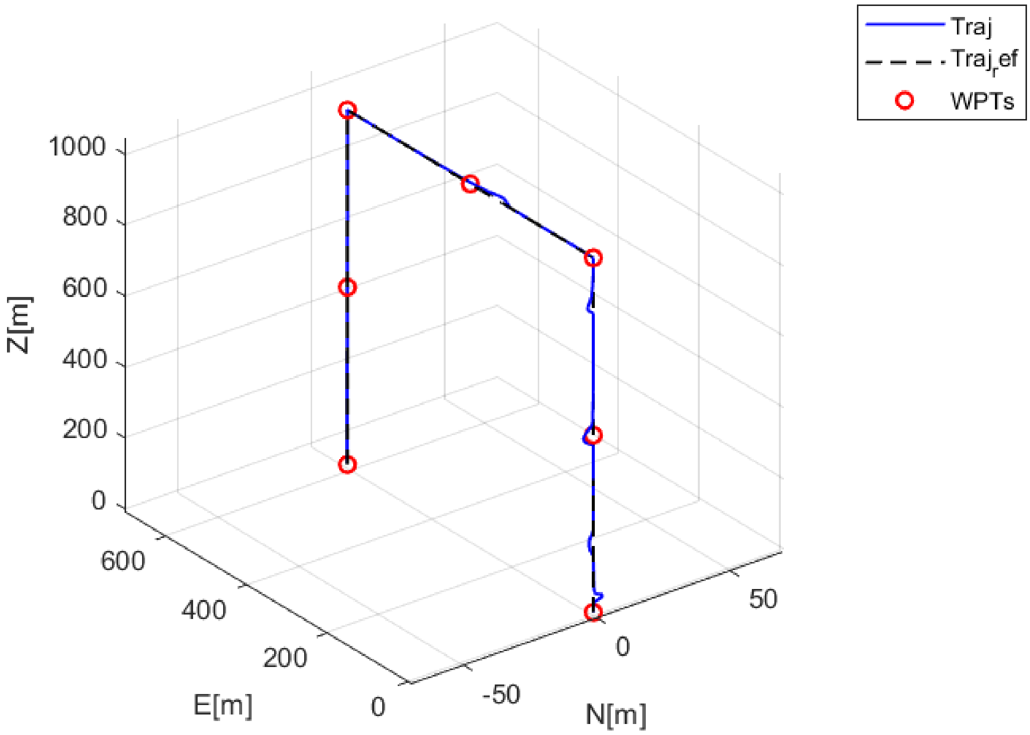


**Figure 6.23:** 3D trajectory

The effect of external disturbances on the inertial position of the drone is visualized below:

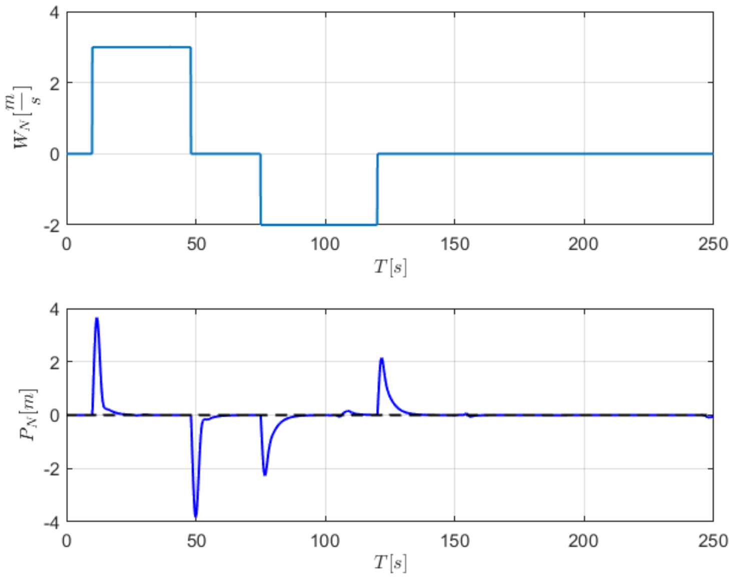


**Figure 6.24:** $P_N$ vs $W_N$

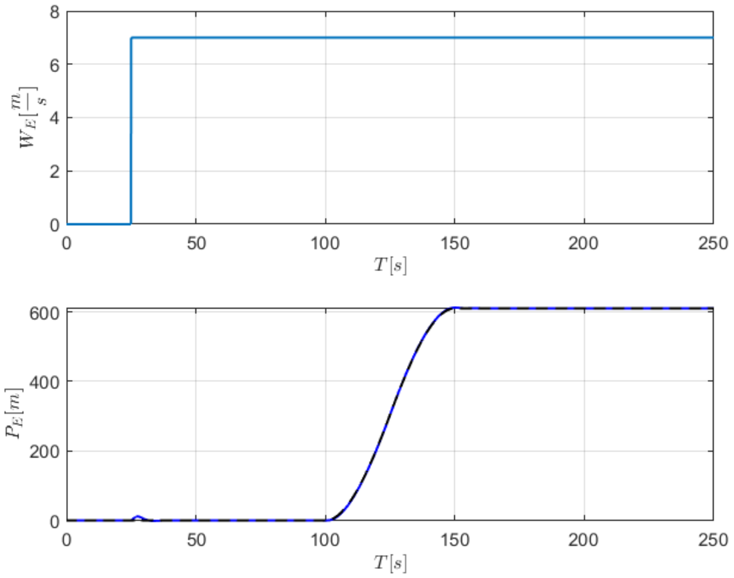


**Figure 6.25:** $P_E$ vs $W_E$

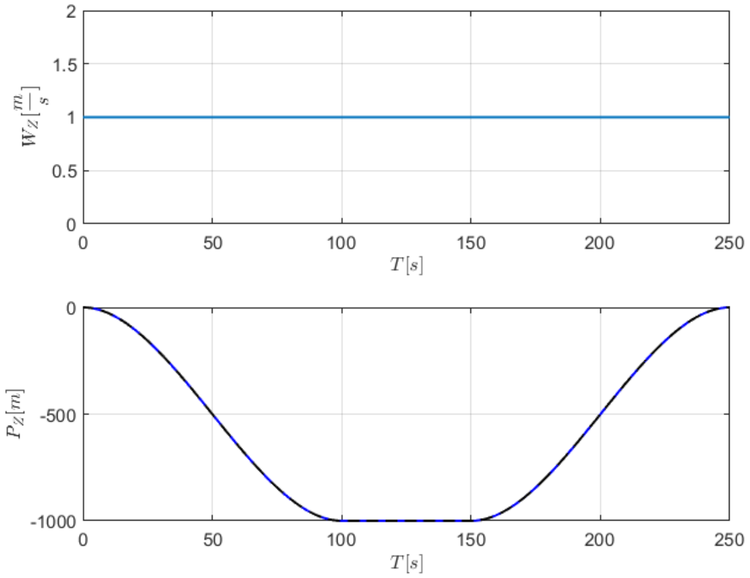


**Figure 6.26:** $P_H$ vs $W_H$

Following the time-history of the control variable $\mathbf{\Omega}_V$ is shown:

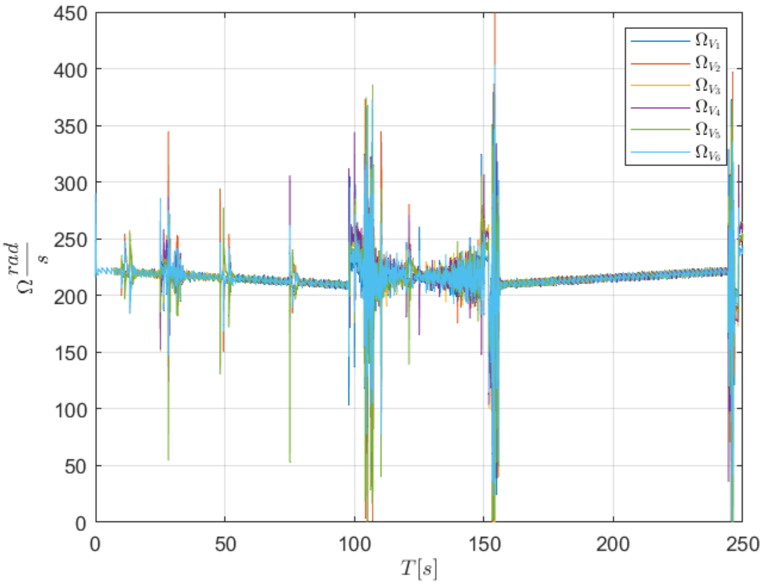


**Figure 6.27:** $\mathbf{\Omega}_V$

## 6.3 Fixed Wing mode

In FW mode, the control of the drone involves two distinct types of mechanisms. The first is the utilization of thrust motors, responsible for managing both horizontal thrust and yaw moments. The latter control is achieved by adjusting the speeds of these motors differentially. Additionally, aerodynamic surfaces come into play for generating moments along the aircraft's body axes, specifically the X-axis and Y-axis. When we consider direct force control, these control variables are translated into the following control vector:

$$\boldsymbol{u}_{FW} = [Tx(\boldsymbol{\Omega}_H), l(\delta_a), m(\delta_a), n(\boldsymbol{\Omega}_H)]^T \tag{6.30}$$

During this phase of the mission, the lift motors are turned off and retracted within the fuselage to enhance aerodynamic efficiency.

### 6.3.1 Inner Loop

Regarding the inner loop, direct force control enables the application of the same strategy that was previously developed for the MR mode. The only distinction lies in the gains of the linear controllers, which, however, will remain consistent during the transition mode.

It's important to highlight that even when employing a more intricate aerodynamic model, the incremental approach still permits the utilization of linear simplified aerodynamics due to its limited reliance on the mathematical model of the plant.

In the case of utilizing a nonlinear aerodynamic model, the general equation for rotational dynamics can be expressed as follows:

$$\boldsymbol{J}^B \dot{\boldsymbol{\omega}}^B_{B/N} = \underbrace{-\boldsymbol{\omega}^B_{B/N} \times \boldsymbol{J}^B \boldsymbol{\omega}^B_{B/N}}_{f(\boldsymbol{x})} \underbrace{-\boldsymbol{G}^B(\boldsymbol{\omega}^B_{B/N}, \boldsymbol{\Omega}_H) + \boldsymbol{M}^B_A(\boldsymbol{x}, \boldsymbol{u}_{FW})}_{g(\boldsymbol{x},\boldsymbol{u})} \underbrace{+\boldsymbol{M}^B_T(\boldsymbol{\Omega}_H)}_{h(\boldsymbol{u})} \tag{6.31}$$

Using first order Taylor expansion and neglecting state dependent terms (Time Scale Separation) yields:

$$\boldsymbol{J}^B \Delta\dot{\boldsymbol{\omega}}^B_{B/N} \approx (\frac{\partial(\boldsymbol{G}^B + \boldsymbol{M}^B_T)}{\partial \boldsymbol{\Omega}_H})_0 \Delta\boldsymbol{\Omega}_H + (\frac{\partial \boldsymbol{M}^B_A}{\partial \boldsymbol{u}_{FW}})_0 \Delta\boldsymbol{u}_{FW} \tag{6.32}$$

As expected, INDI allows us to rewrite the equation in control-affine form.

Next, the performance of the inner loop is shown for step and sinusoidal inputs, starting from the following cruise trim conditions:

$$\boldsymbol{x}_{tr} = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \phi \\ \theta \\ \psi \end{bmatrix}_{tr} = \begin{bmatrix} 59.29\ [\frac{m}{s}] \\ 0\ [\frac{m}{s}] \\ 9.21\ [\frac{m}{s}] \\ 0\ [\frac{rad}{s}] \\ 0\ [\frac{rad}{s}] \\ 0\ [\frac{rad}{s}] \\ 0\ [rad] \\ 0.154\ [rad] \\ 0\ [rad] \end{bmatrix} \tag{6.33}$$

$$\boldsymbol{u}_{tr} = \begin{bmatrix} \delta_e \\ \delta_a \\ \Omega_{H_1} \\ \Omega_{H_2} \end{bmatrix}_{tr} = \begin{bmatrix} -15.78\ [deg] \\ 2e-4\ [deg] \\ 1447.6\ [Rpm] \\ 1447.6\ [Rpm] \end{bmatrix} \tag{6.34}$$

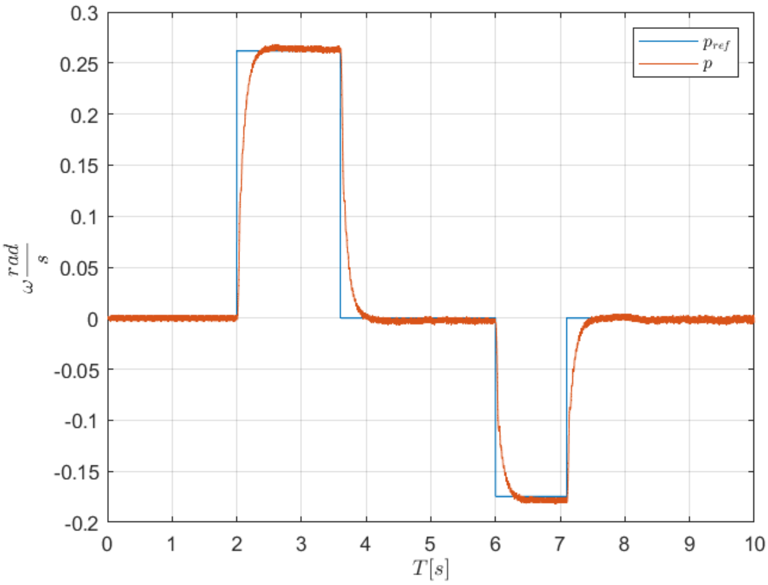


**Figure 6.28:** Response to double impulse $p_{des}$

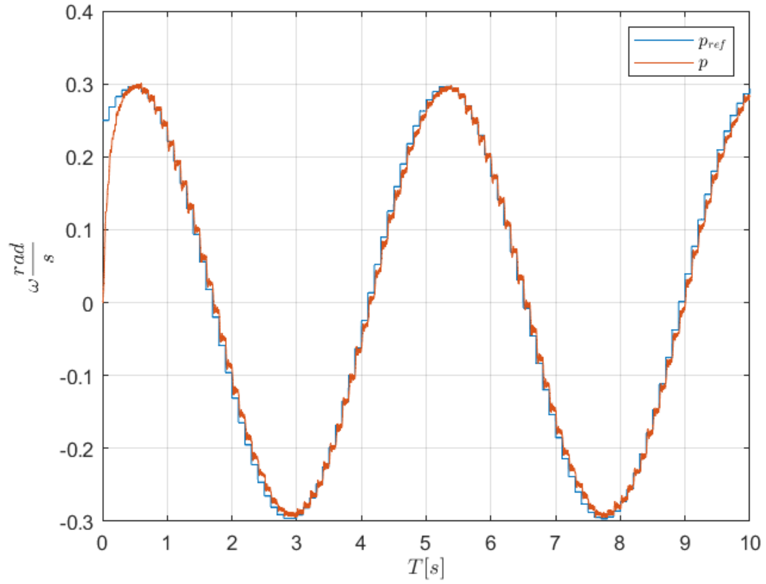


**Figure 6.29:** Response to sinusoidal $p_{des}$

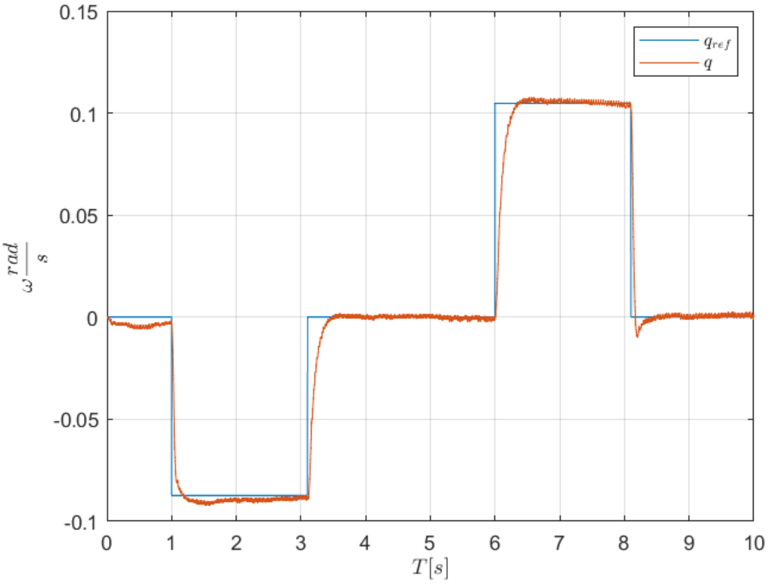


**Figure 6.30:** Response to double impulse $q_{des}$

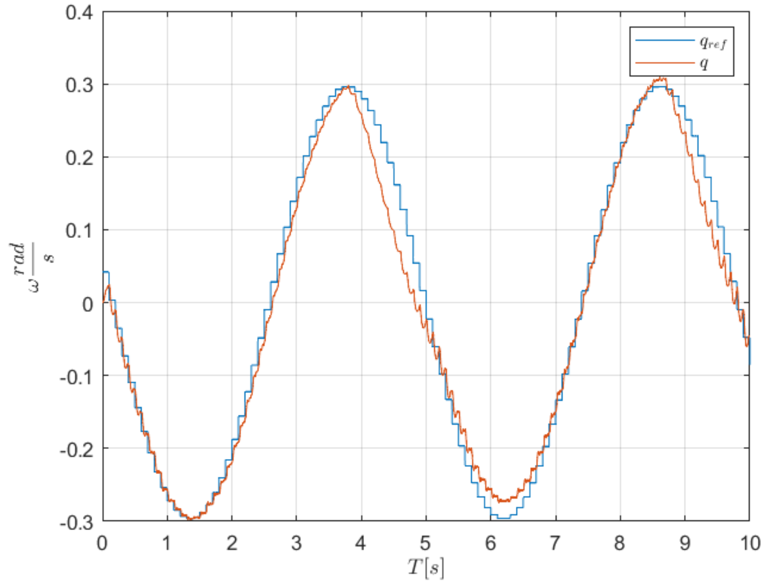


**Figure 6.31:** Response to sinusoidal $q_{des}$

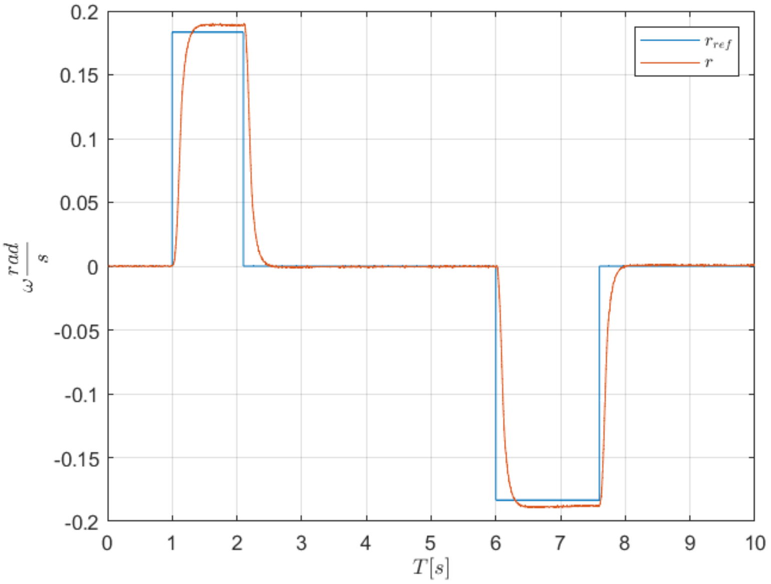


**Figure 6.32:** Response to double impulse $r_{des}$

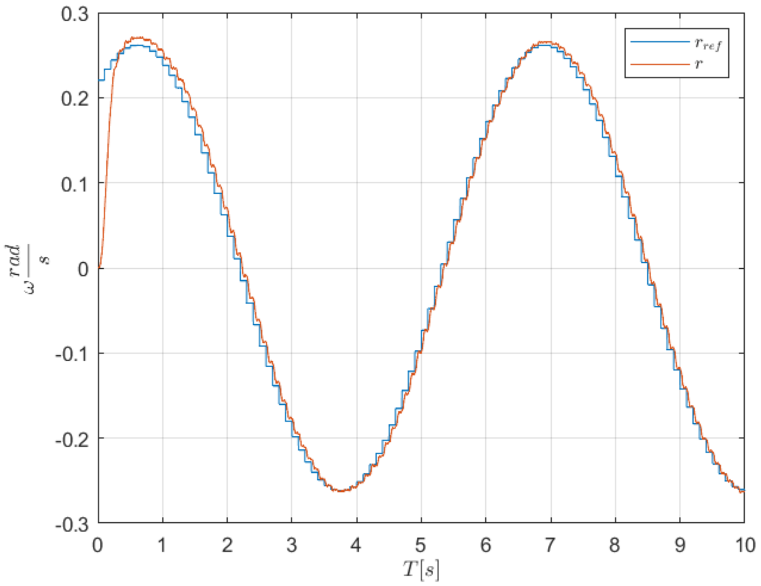


**Figure 6.33:** Response to sinusoidal $r_{des}$

The inner loop for FW mode differs from the MR phase in its objective to maintain a zero Angle of Sideslip ($\beta$) to enhance flight efficiency. To achieve this, the desired yaw rate $\dot{\psi}_{des}$ is generated through a sideslip controller, wherein the error $e_\beta = 0 - \beta$ is converted into $\dot{\psi}_{des}$ using a linear controller[35].

Additionally, in fixed-wing flight, turning is primarily achieved through the roll angle ($\phi$). Thus, the azimuth angle $\chi_{ref}$ obtained from the outer loop is converted into a reference roll angle ($\phi_{ref}$) using a linear controller.

Subsequently, the performance of Euler angles tracking is presented. Each plot includes $\beta$ over time to assess how effectively the sideslip controller maintains it at zero. Moreover, in fixed-wing flight, yaw ($\psi$) is not of primary interest; instead, the azimuth angle is depicted. As mentioned earlier, azimuth tracking is indeed achieved by rolling the aircraft via $\phi$.

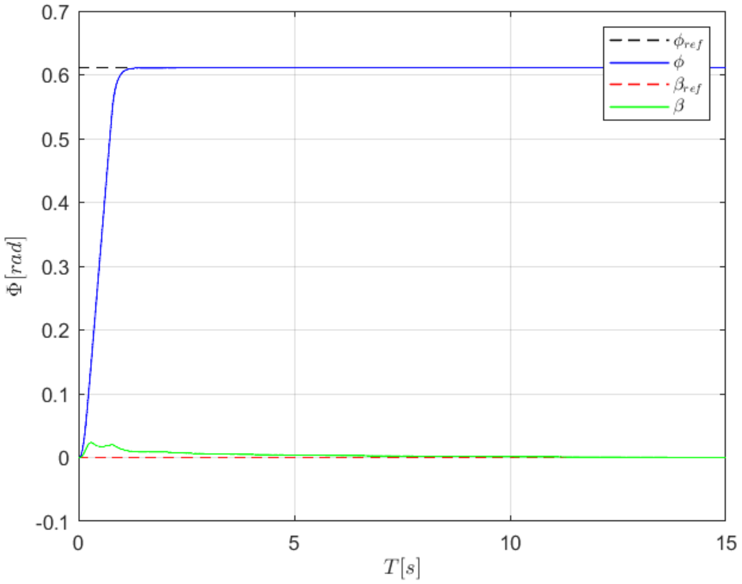


**Figure 6.34:** Response to step $\phi_{des}$

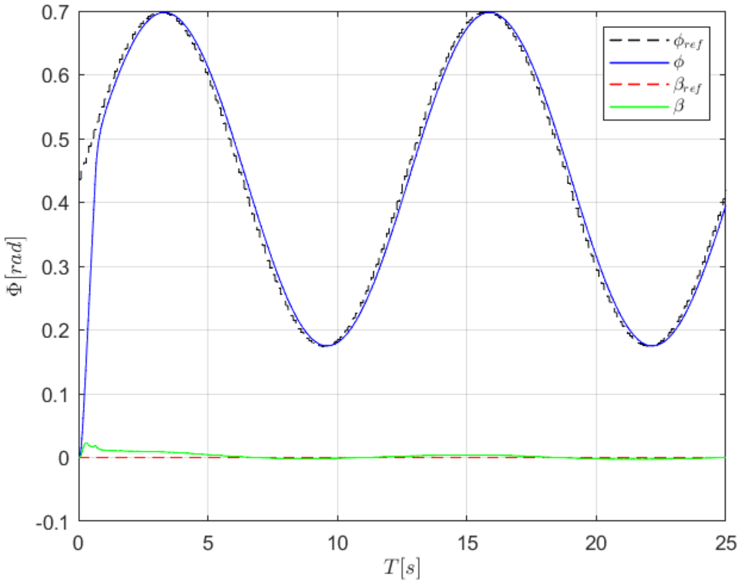


**Figure 6.35:** Response to step + sine wave $\phi_{des}$

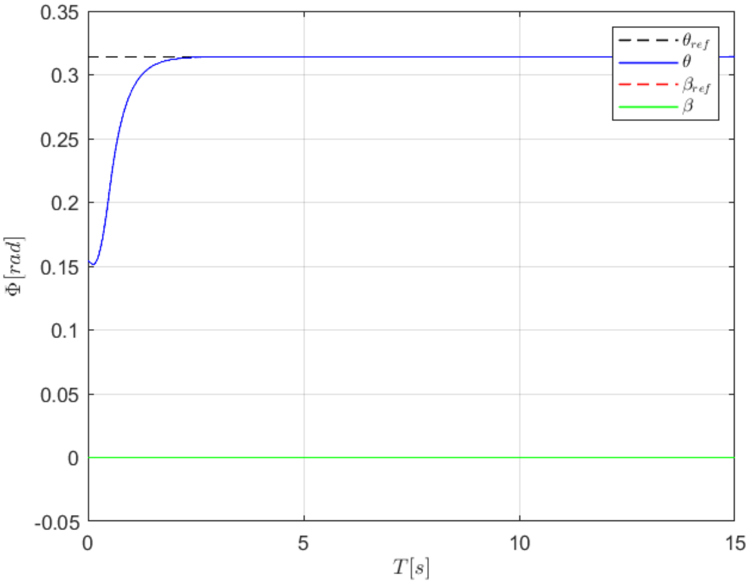


**Figure 6.36:** Response to step $\theta_{des}$

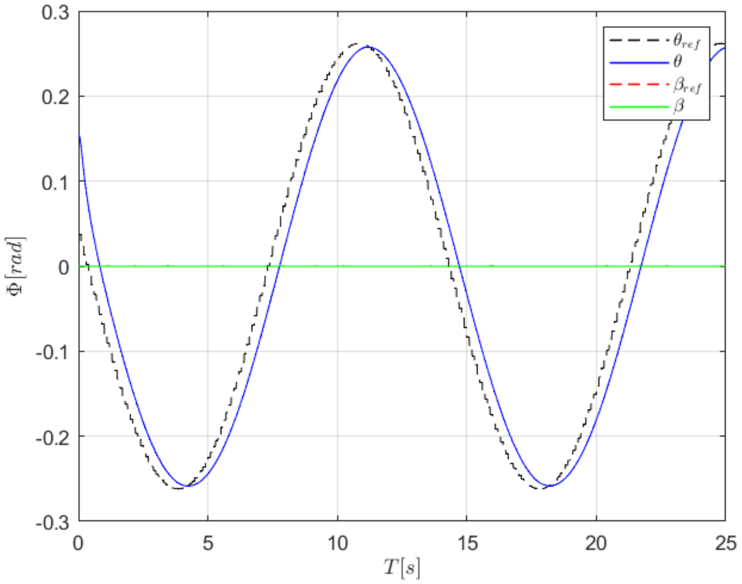


**Figure 6.37:** Response to step + sine wave $\theta_{des}$

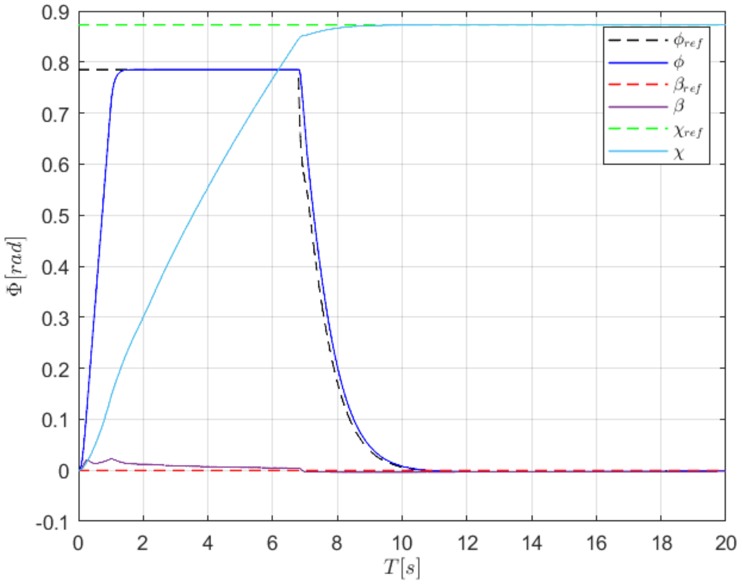


**Figure 6.38:** Response to step $\chi_{des}$

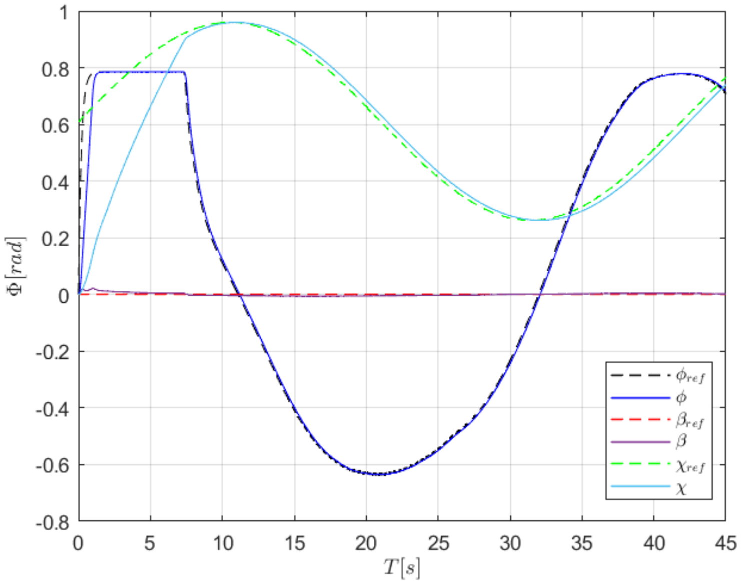


**Figure 6.39:** Response to step + sine wave $\chi_{des}$

It must be noted that some low pass filtering is performed on internal reference signals to attenuate high frequency oscillations.

### 6.3.2 Outer Loop

The outer loop, again, serves the purpose of tracking the inertial trajectory as well as providing references for the inner loop in terms of attitude. The external part of the controller will be different from the MR mode due to the different nature of sustained flight in which thrust is provided horizontally and flight level is achieved by exploiting aerodynamic lift.

For the FW controller, linear dynamics is conveniently written in fligh-path axes. By applying the transformation matrix $C_{Fp/N}(\gamma, \chi)$ to the linear dynamics, the following can be obtained:

$$\begin{bmatrix} m\dot{V} \\ mVc(\gamma)\dot{\chi} \\ -mV\dot{\gamma} \end{bmatrix} = C_{Fp/N} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + C_{Fp/N}C_{N/B} \begin{bmatrix} T_X \\ 0 \\ 0 \end{bmatrix} + C_{Fp/N}C_{N/W} \begin{bmatrix} -D \\ S \\ -L \end{bmatrix} \tag{6.35}$$

Under the assumption of attitude tracking being faster than trajectory tracking, we can further simplify the equations by neglecting the latero-directional motion. Eq. (6.35) becomes:

$$\begin{cases} \dot{V} = \frac{1}{m}(T_X \cos(\alpha) - D - mg\sin(\gamma)) \\ \dot{\gamma} = \frac{1}{mV}(T_X \sin(\alpha) + L - mg\cos(\gamma)) \end{cases} \tag{6.36}$$

The strategy for the outer loop controller involves the following steps:

1. A guidance algorithm provides inertial position and velocity. The desired position is compared with the actual one and the produced error is turned into an additional reference velocity by means of a PID controller. The sum of the two velocity components is the final reference to be followed.

2. Classical NDI is applied to the kinematic constraint between flight-path and inertial variables. Therefore, reference inertial velocity components are transformed into desired azimuth angle, flight-path angle and magnitude of the inertial velocity vector. The original approach would involve transforming these variables into their desired derivatives through linear controllers. However, the controller turned out to perform better if $\gamma$ and $\chi$ are directly passed to a PID controller and turned into desired pitch and roll angles.

3. $V$ and $\gamma$ still need to be transformed into $\dot{V}$ and $\dot{\gamma}$ by a set of PID controllers in order to correct the offset between their desired and actual values.

4. Finally, $\dot{V}$ and $\dot{\gamma}$ are used to incrementally invert the simplified dynamical equations (6.35) to provide the commanded thrust value $T_X$.

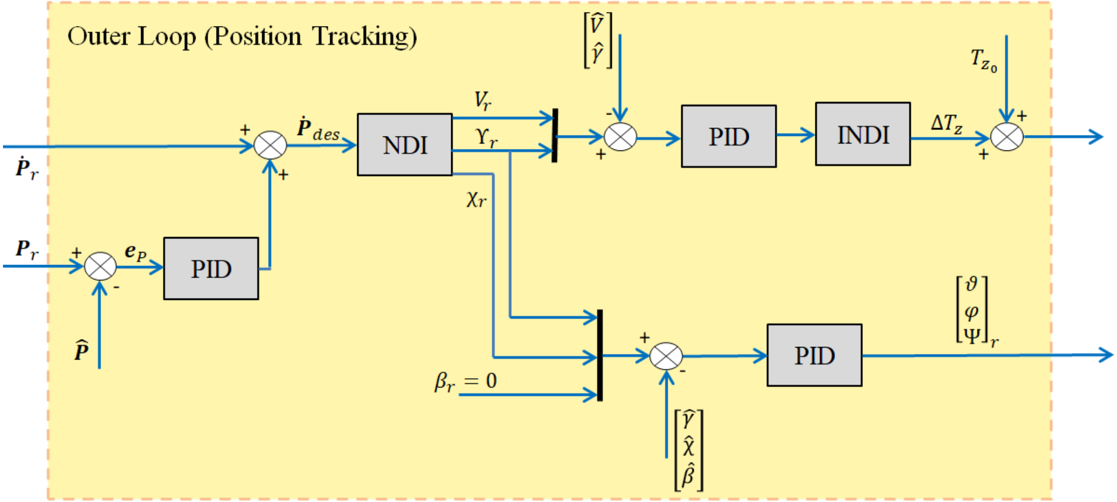

Outer Loop (Position Tracking)
$\dot{P}_r$
$P_r$
+
$e_P$
PID
-
$\hat{P}$
+
+
$\dot{P}_{des}$
NDI
$V_r$
$\Upsilon_r$
$\chi_r$
$\beta_r = 0$
$\begin{bmatrix}\hat{V}\\ \hat{\gamma}\end{bmatrix}$
-
+
PID
INDI
$\Delta T_z$
$T_{z_0}$
+
+
$\begin{bmatrix}\vartheta\\ \varphi\\ \Psi\end{bmatrix}_r$
+
-
PID
$\begin{bmatrix}\hat{\gamma}\\ \hat{\chi}\\ \hat{\beta}\end{bmatrix}$

**Figure 6.40:** Outer Loop

### INDI for translational dynamics

As mentioned earlier, the pitch angle $\theta$ is commanded to track the desired flight-path angle. As a result, $\gamma$ is tracked independently of the thrust provided by the engines and we can assume that the effect of $\gamma$ is sufficiently taken into account by sensor measurements. Therefore, we can reduce speed control to a single equation for INDI:

$$m\dot{V} = \underbrace{T_X\cos(\alpha)}_{u_1\cdot g(\boldsymbol{x})}\underbrace{-D - mg\sin(\gamma)}_{f(\boldsymbol{x})} \tag{6.37}$$

Even though the equation is control-affine from the perspective of direct force control, INDI is still preferred over classical inversion because any uncertainty in the model is fed back to the control system in the form of sensor measurements. Applying INDI as shown in the previous sections yields:

$$\Delta T_X = m\frac{\dot{V} - \dot{V}_0}{cos(\alpha_0)} \tag{6.38}$$

As usual, all the variables with a *’0’* subscript need to be estimated. In this case $\dot{V}_0$ is the time derivative of the velocity vector at current point in time, and it can be obtained through appropriate manipulation of accelerometer measurements.

The desired acceleration is obtained through a PID controller, which transforms the velocity error $e_V = V_{des} - V$ into a commanded acceleration:

$$\dot{V} = K_{P_V} e_V + K_{D_V} \dot{e}_V + K_{I_V} \int e_V d\tau \tag{6.39}$$

### NDI for translational kinematics

The desired velocity used for computing $e_V$ comes from a kinematical relationship between inertial and flight-path variables.
In mathematical terms, such kinematical constraint can be expressed as follows:

$$\dot{\boldsymbol{P}}^N = C_{N/Fp}(\chi, \gamma) \begin{bmatrix} V \\ 0 \\ 0 \end{bmatrix} \rightarrow \begin{cases} \dot{X} = V\cos(\gamma)\cos(\chi) \\ \dot{Y} = V\cos(\gamma)\sin(\chi) \\ \dot{Z} = V\sin(\gamma) \end{cases} \tag{6.40}$$

This set of equations does not contain any model uncertainties and lends itself well to classical dynamic inversion[26]. A proper rearrangement of the inverse relationships results in:

$$\begin{cases} V_{des} = \sqrt{\dot{X}^2 + \dot{Y}^2 + \dot{Z}^2} \\ \chi_{des} = \tan^{-1}(\frac{\dot{Y}}{\dot{X}}) \\ \gamma_{des} = \sin^{-1}(\frac{-\dot{Z}}{V}) \end{cases} \tag{6.41}$$

This is one of the rare cases in which NDI can be applied to non control-affine systems. We can think of $[V, \gamma, \chi]^T$ as the input $\boldsymbol{u}$ and $\dot{\boldsymbol{P}}^N$ as the output, which can be conveniently taken as the virtual input $\boldsymbol{v}$. The result is the following nonlinear relationship:

$$\boldsymbol{u} = \begin{bmatrix} V_{des} \\ \chi_{des} \\ \gamma_{des} \end{bmatrix} = \boldsymbol{f}( \begin{bmatrix} \dot{X}_{des} \\ \dot{Y}_{des} \\ \dot{Z}_{des} \end{bmatrix} ) = \boldsymbol{f}(\boldsymbol{v}) \tag{6.42}$$

To understand where the virtual input comes from, we need to consider the guidance algorithm. In this context, such algorithm provides a reference inertial position as well as a reference inertial velocity. In order for the control system to be able to correct position errors, the commanded velocity has to be equipped with an extra term which provides a $\Delta\dot{\boldsymbol{P}}^N$ when a position offset occurs. An easy solution is to pass the position error ($\boldsymbol{e}_P = \boldsymbol{P}_g^N - \boldsymbol{P}^N$) to a PID controller that gives an extra velocity term only when position corrections are needed.

The resulting expression for $\boldsymbol{v}$ is:

$$\dot{\boldsymbol{P}}^N_{des} = \dot{\boldsymbol{P}}^N_g + \begin{bmatrix} \boldsymbol{K}_P & \boldsymbol{K}_D & \boldsymbol{K}_I \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_P \\ \dot{\boldsymbol{e}}_P \\ \int \boldsymbol{e}_P d\tau \end{bmatrix} \tag{6.43}$$

The subscript *'g'* denotes a variable sourced from the guidance algorithm.

Finally, the reference attitude for the inner loop is yielded by:

- Two PID controllers that convert $\gamma_{des}$ and $\chi_{des}$ respectively into reference pitch and yaw angles ($[\theta, \phi]_{des}$).
- One Sideslip controller consisting of a PID that transforms the reference AoS ($\beta_{ref} = 0$) into a desired yaw rate $\dot{\psi}$.

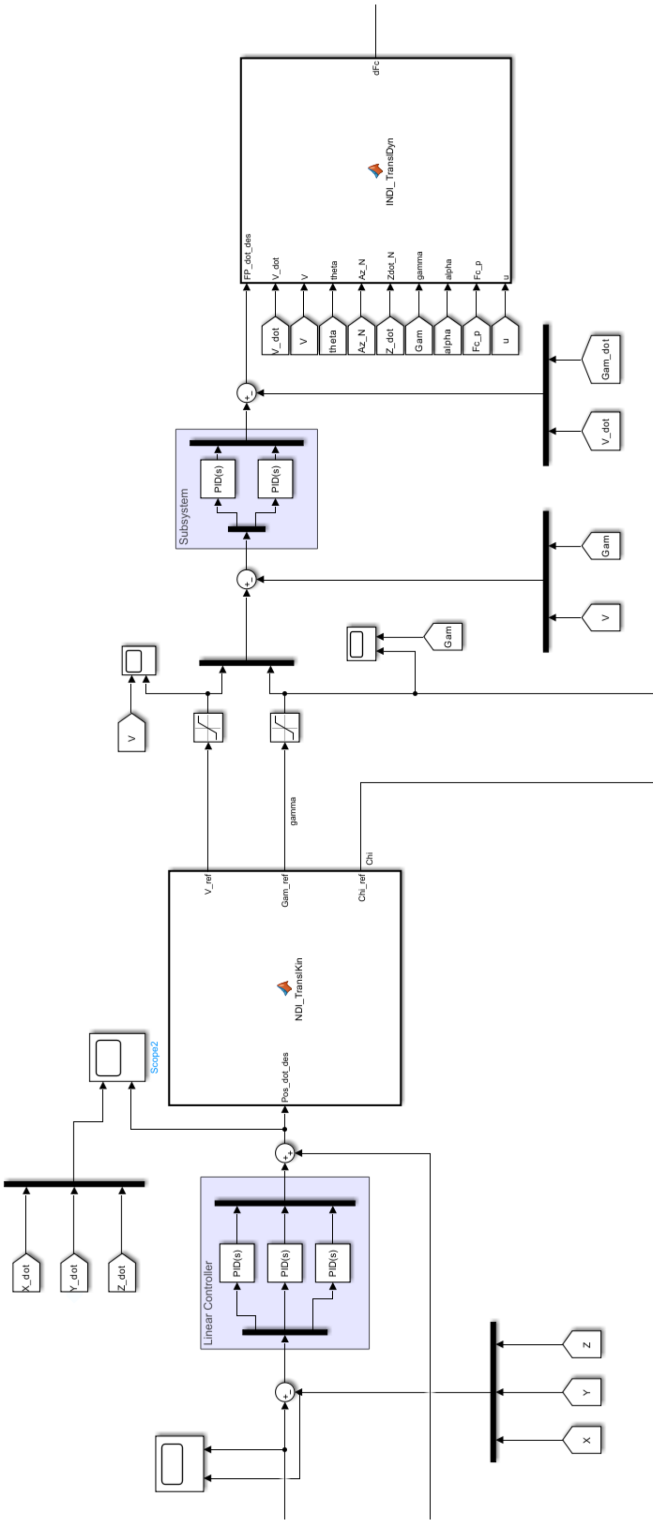


**Figure 6.41:** FW Outer Loop Simulink model

The outer loop's performance is demonstrated during a combined maneuver, involving a simultaneous step and sine wave input for all flight-path variables, including $V$, $\gamma$, and $\chi$. The dynamic input applied to the azimuth angle serves as a validation of the controller's effectiveness, in spite of the approximations in the linear dynamics equations used in the outer loop.

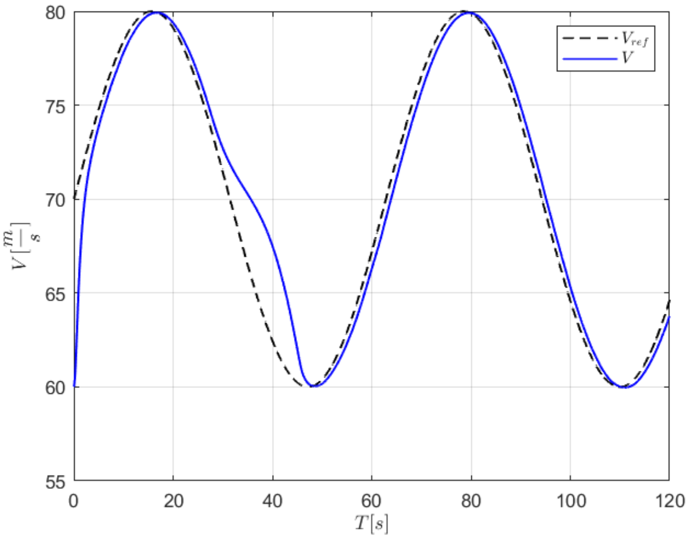


**Figure 6.42:** $V$ tracking for step + sine wave $V_{ref}$

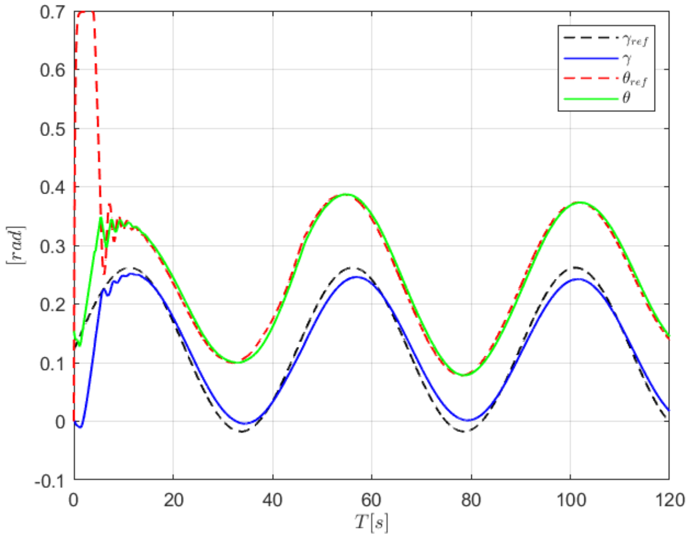


**Figure 6.43:** $\gamma$ tracking for step + sine wave $\gamma_{ref}$

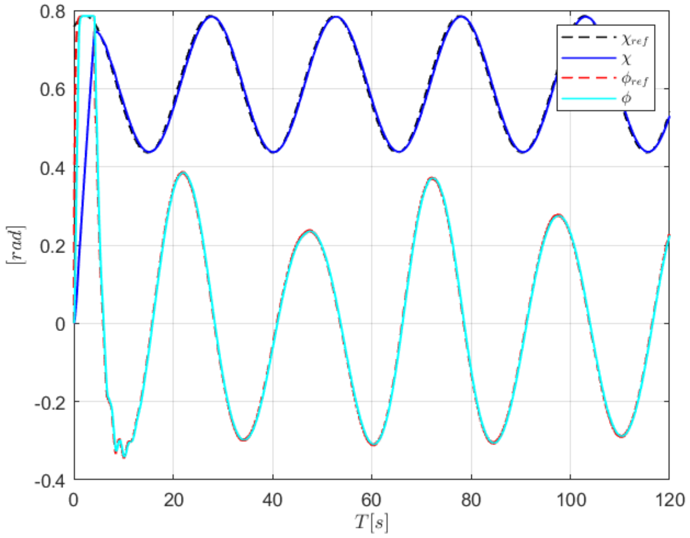


**Figure 6.44:** $\chi$ tracking for step + sine wave $\chi_{ref}$

Given the kinematic relationship between velocities and flight-path variables, assessing the system response to $[V, \chi, \gamma]^T$ inputs is the same as evaluating the tracking performance to velocity commands.

The robustness of the controller to external atmospheric disturbances is tested in a simulated mission through the following NEU waypoints:

$$\boldsymbol{WPTs}^N = \begin{bmatrix} \boldsymbol{N}^T \\ \boldsymbol{E}^T \\ \boldsymbol{H}^T \end{bmatrix} = \begin{bmatrix} 0 & 1e3 & 1e4 & 1.5e4 & 2.5e4 \\ 0 & 0 & 0 & 5e3 & 1e4 \\ 1e3 & 1e3 & 2e3 & 2e3 & 2e3 \end{bmatrix} [m] \tag{6.44}$$

The 3D real trajectory is compared against the reference trajectory issued by the guidance algorithm in the following plot:

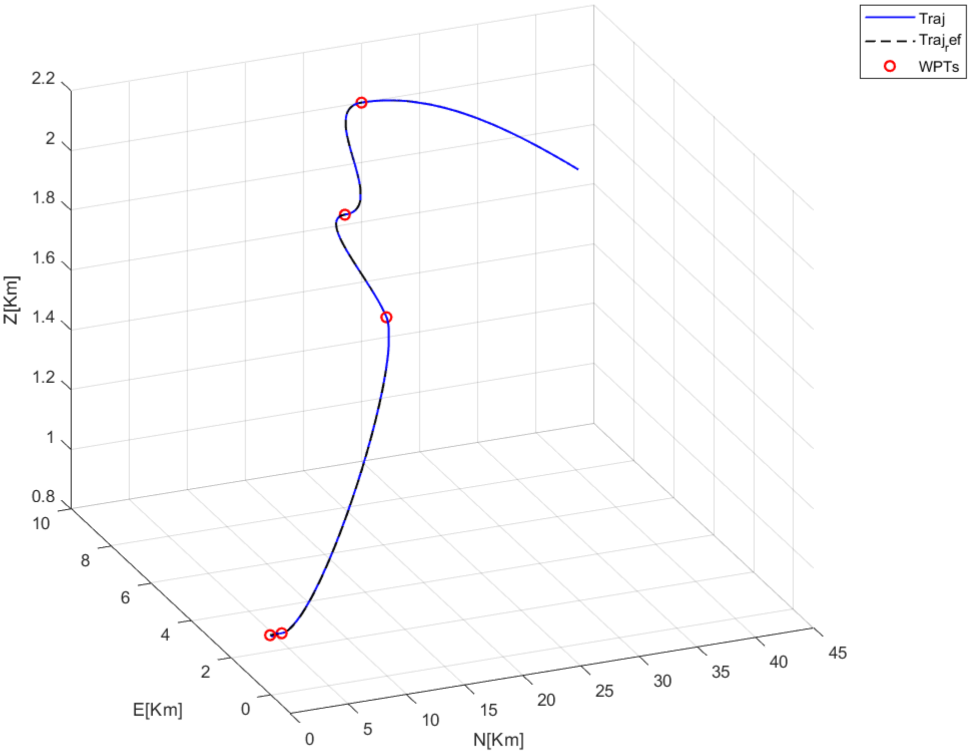


**Figure 6.45:** 3D trajectory

At such high speeds, the aircraft's center of mass covers a considerable distance, making it challenging to discern the actual deviation of the flight path from the ideal trajectory. Therefore, to assess the impact of external wind, the figures below illustrate the evolution of the inertial position error over time:

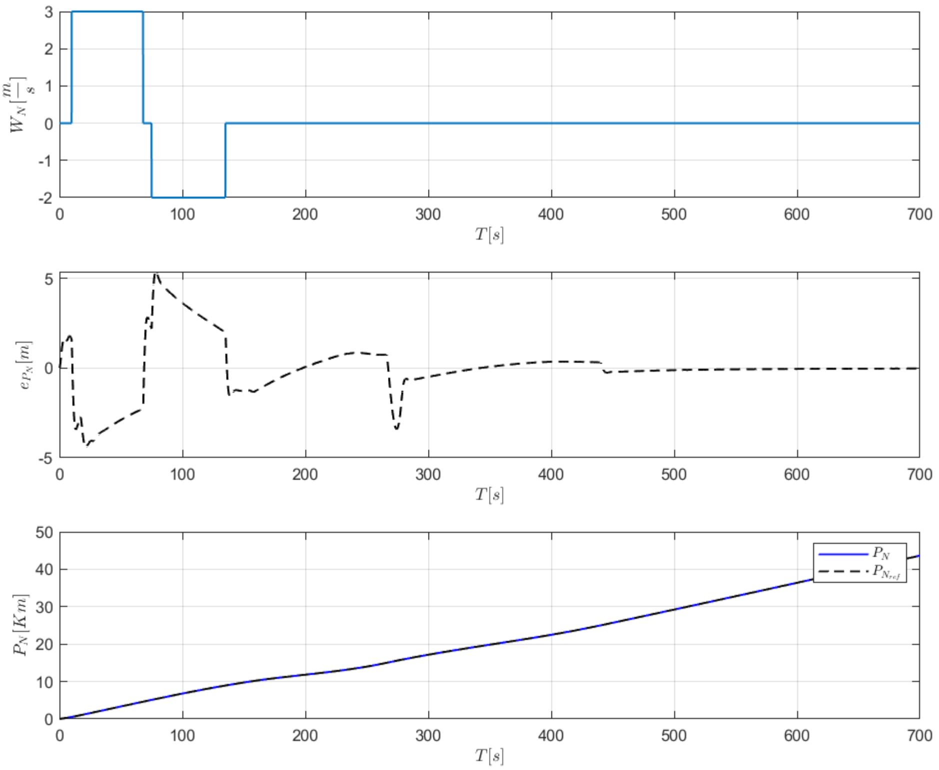


**Figure 6.46:** Effect of North wind on North inertial displacement

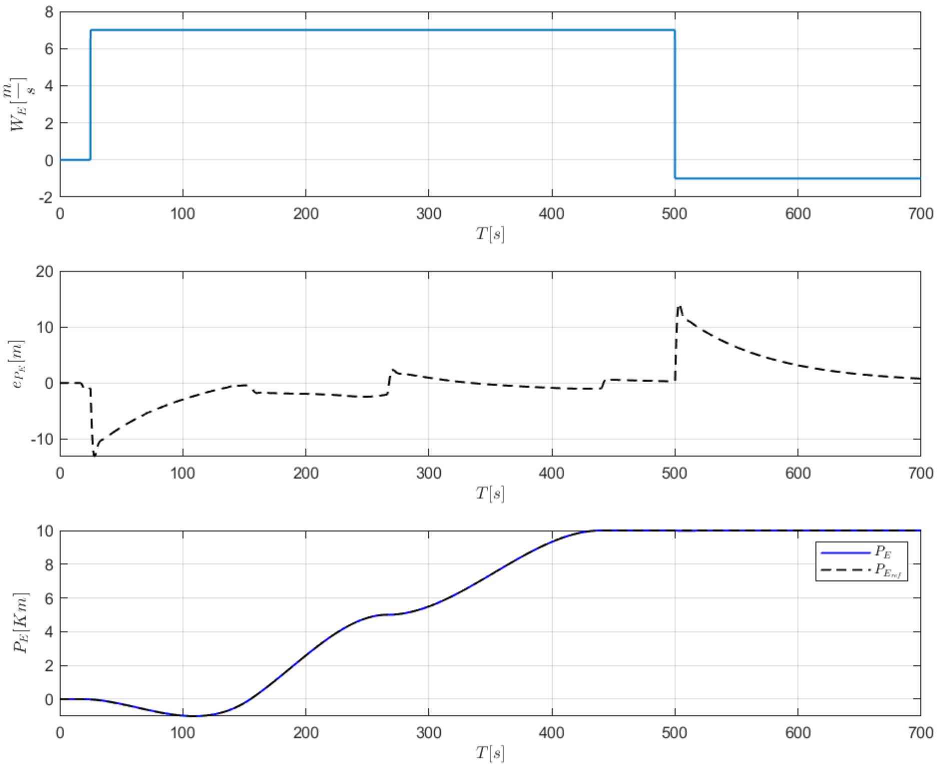


**Figure 6.47:** Effect of East wind on East inertial displacement

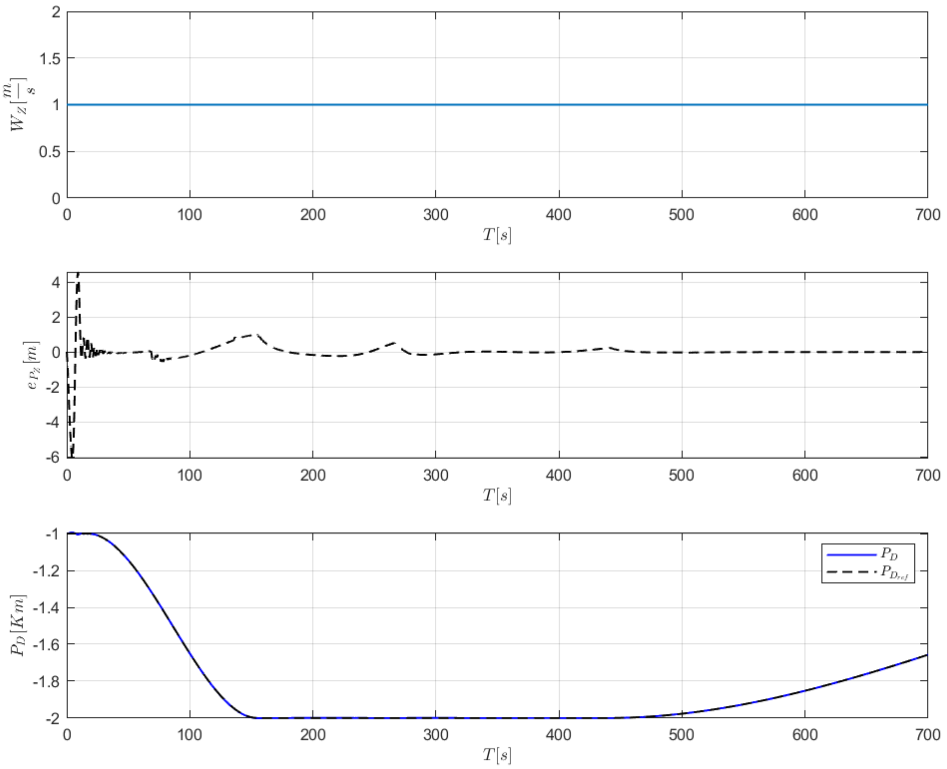


**Figure 6.48:** Effect of Down wind on Down inertial displacement

The control system exhibits effective responsiveness to external disturbances. The largest error induced by wind amounts to approximately 10 meters, in contrast to the distance traveled in the same direction, which is on the order of 10 kilometers.

The time history the flight-path variables is shown below.

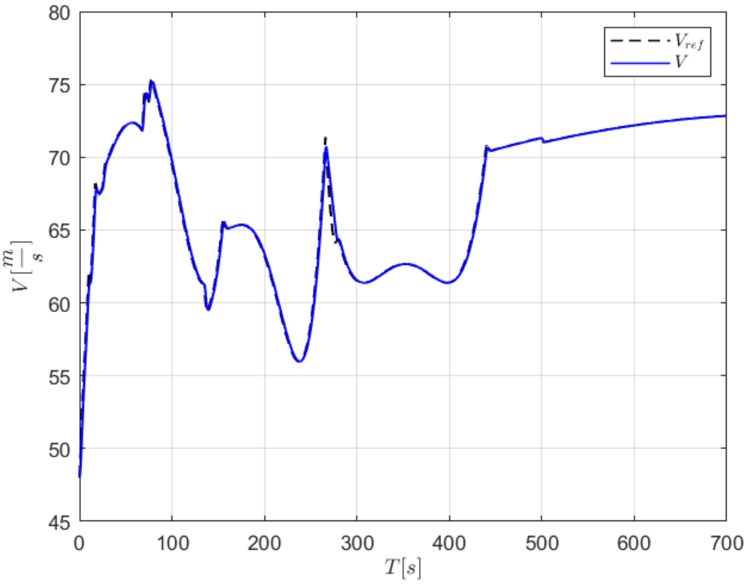


**Figure 6.49:** V vs t

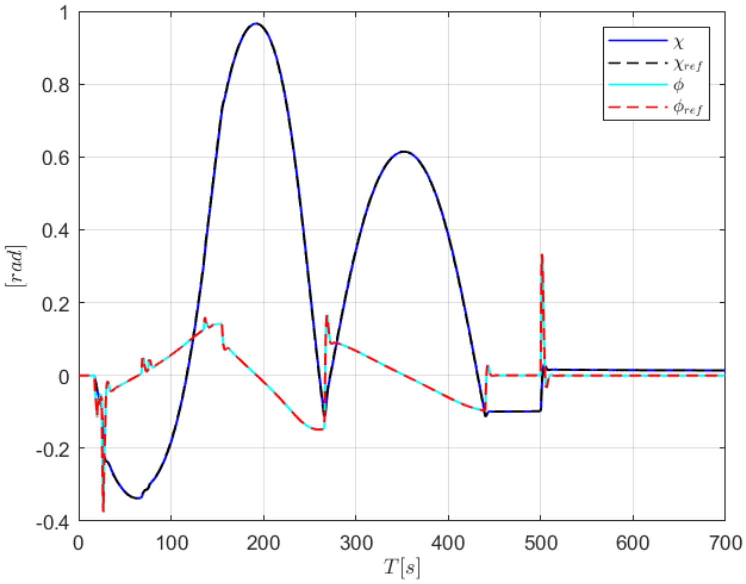


**Figure 6.50:** $\chi$ vs t

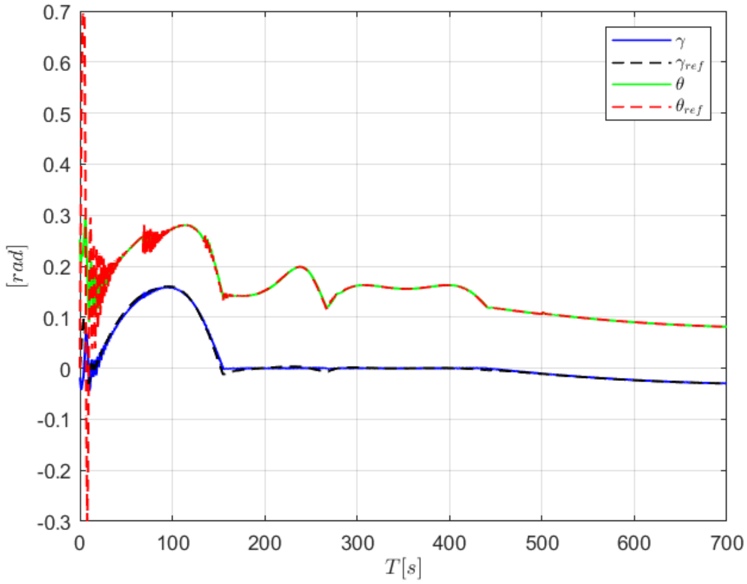


**Figure 6.51:** $\gamma$ vs t

The aircraft possesses the capability to maneuver and counteract external disturbances, all while maintaining a close-to-zero value for $\beta$ throughout the process.

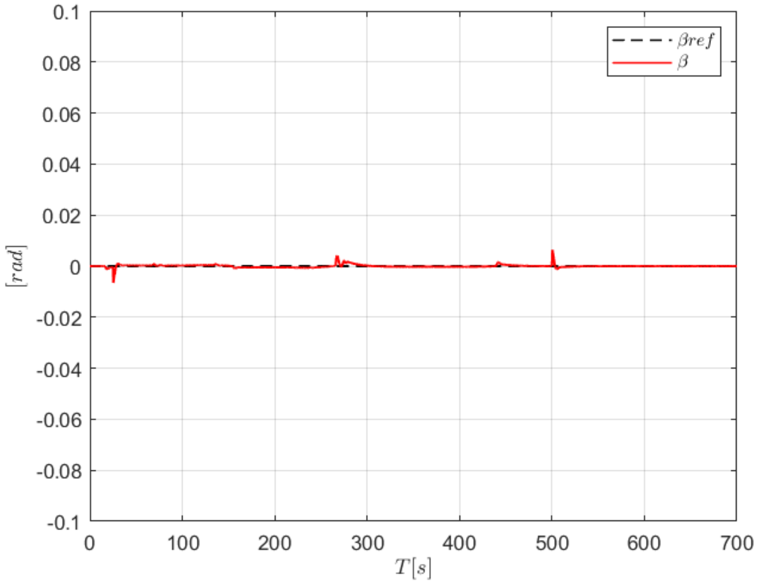


**Figure 6.52:** $\beta$ vs t

## 6.4 Transition/Hybrid mode

In a typical mission profile, the aircraft would need to smoothly transition between different flight modes, specifically from a Vertical Takeoff and Landing (VTOL) or Short Takeoff and Landing (STOL) mode, referred to as MR, to FW mode for efficient cruise. This transition should ideally involve a constant-altitude, linearly accelerated motion, where the aircraft gradually increases its forward speed until aerodynamic lift becomes sufficient to sustain it in midair. However, safety considerations dictate that the aircraft must also have the capability to maneuver in a Transitional mode. This TR mode is crucial to ensure safe and controlled operations during the transition phases.

The inherent complexity in controlling a hybrid mode aircraft stems from the presence of overactuation, where all control mechanisms are simultaneously active. However, one approach to simplify this complex control problem is to employ a direct force control method. With this method, the control vector $\boldsymbol{u}_{TR}$ is transformed as follows:

$$\begin{bmatrix} \delta_e \\ \delta_a \\ \Omega_{H1} \\ \Omega_{H2} \\ \Omega_{V1} \\ \cdot \\ \cdot \\ \Omega_{V6} \end{bmatrix} \rightarrow \begin{bmatrix} T_X \\ T_Z \\ M_X \\ M_Y \\ M_Z \end{bmatrix} \tag{6.45}$$

The control system operates by calculating the necessary control forces and moments needed for the aircraft's desired maneuvers. These computed control inputs are then translated into actual actuator commands through the use of a control allocation algorithm. This algorithm is responsible for distributing the required control inputs to the various actuators on the aircraft, ensuring that the desired control actions are effectively carried out.

### 6.4.1 Inner Loop

As previously discussed, the inner loop remains consistent with the Fixed-Wing (FW) mode, both in terms of control strategy and the gains used for the linear controllers. This represents a significant advantage offered by nonlinear control techniques, as they eliminate the need for gain scheduling.

The assessment of tracking performance for the inner loop begins with the following trim conditions, assuming steady level longitudinal flight.

$$\boldsymbol{x}_{tr} = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \phi \\ \theta \\ \psi \end{bmatrix}_{tr} = \begin{bmatrix} 28.364\ [\frac{m}{s}] \\ 0\ [\frac{m}{s}] \\ 6.039\ [\frac{m}{s}] \\ 0\ [\frac{rad}{s}] \\ 0\ [\frac{rad}{s}] \\ 0\ [\frac{rad}{s}] \\ 0\ [rad] \\ 0.2098\ [rad] \\ 0\ [rad] \end{bmatrix}, \quad \boldsymbol{u}_{tr} = \begin{bmatrix} \delta_e \\ \delta_a \\ \Omega_{H_1} \\ \Omega_{H_2} \\ \Omega_{V_1} \\ \Omega_{V_2} \\ \Omega_{V_3} \\ \Omega_{V_4} \\ \Omega_{V_5} \\ \Omega_{V_6} \end{bmatrix}_{tr} = \begin{bmatrix} -14.254\ [deg] \\ 0\ [deg] \\ 1992.5\ [Rpm] \\ 1992.5\ [Rpm] \\ 1806.7\ [Rpm] \\ 1667.2\ [Rpm] \\ 1514.9\ [Rpm] \\ 1514.9\ [Rpm] \\ 1667.2\ [Rpm] \\ 1806.7\ [Rpm] \end{bmatrix} \quad (6.46)$$

To begin with, the tracking of body angular velocity inputs is demonstrated. A combined input is applied to simultaneously stimulate all three body axes. Each component of this input is instructed to follow a combination of a sine wave and a double impulse input.

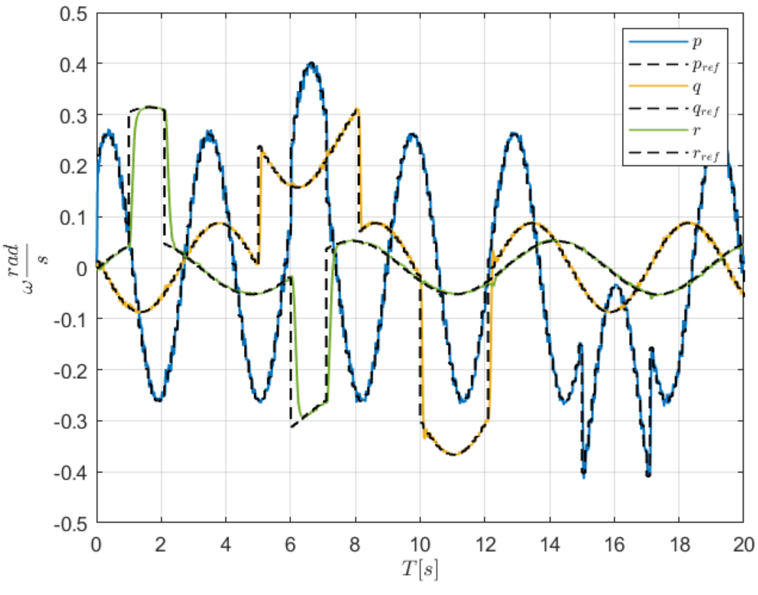


**Figure 6.53:** $\boldsymbol{\omega}^{B}_{B/N}$ tracking for sine + double impulse input

Moving on, the assessment of attitude tracking involves inputting a combination of a sine wave, a step function, and a double impulse simultaneously for both roll ($\phi$) and pitch ($\theta$) angles.

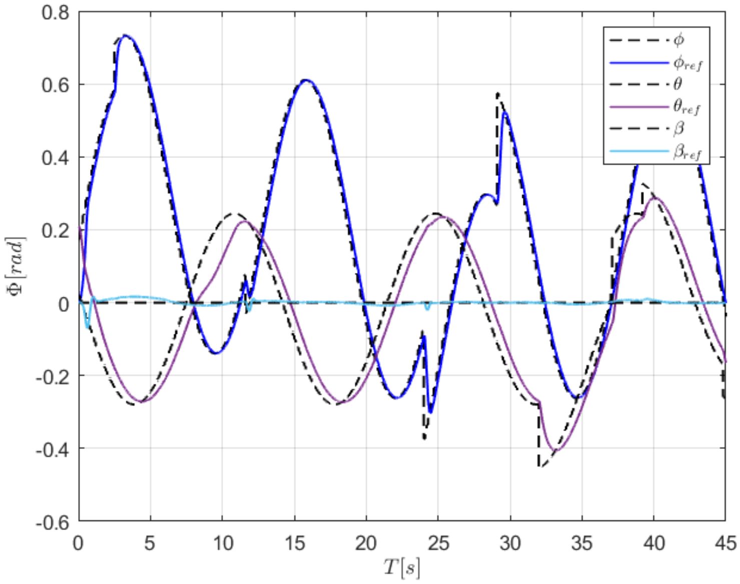


**Figure 6.54:** $[\phi, \theta]$ tracking for sine + step + double impulse input

As for FW mode, $\chi$ is preferred over yaw as it provides information about the inertial trajectory projected onto the tangential plane to the Earth's surface. Also in TR mode the control system tracks the azimuth angle via roll $\phi$.

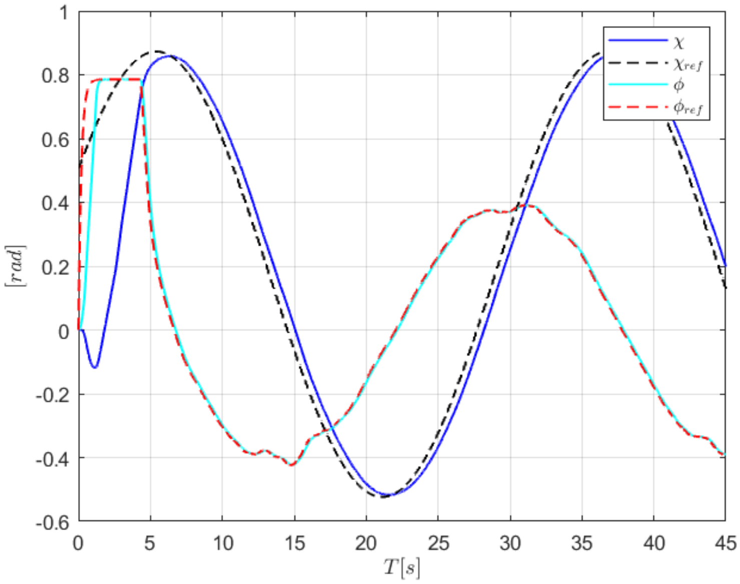


**Figure 6.55:** $\chi$ tracking for sine + step input

### 6.4.2 Outer Loop

The Fixed-Wing and Transitional modes share the same outer loop controller. They share the same gains for the PID blocks and they are nearly identical in terms of their control strategies. The only distinction between them is that during the transition phase, the control force vector has an additional component acting along the $-\hat{\boldsymbol{k}}_b$ axis.

As a result, the equations governing the linear dynamics, while maintaining the assumption of longitudinal motion, cannot be reduced to a single expression. Therefore, an INDI approach needs to be applied to the following 2 × 2 system:

$$\begin{cases} \dot{V} = \frac{1}{m}(T_X \cos(\alpha) - D - mg\sin(\gamma) - T_z \sin(\alpha)) \\ \dot{\gamma} = \frac{1}{mV}(T_X \sin(\alpha) + L - mg\cos(\gamma) + T_z \cos(\alpha)) \end{cases} \tag{6.47}$$

In this scenario, considering the low speeds and the relatively small influence of aerodynamic surface deflection on lift ($L_{\delta_e}\delta_e$), it's reasonable to assume that the contribution of $L_{\delta_e}\delta_e$ is minor compared to other control effects. Consequently, we can assume the two thrust terms are the only control variables involved.

When linearizing the system and neglecting the state-dependent terms, the simplified equations become:

$$\Delta \begin{bmatrix} \dot{V} \\ \dot{\gamma} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{mV_0} \end{bmatrix} \begin{bmatrix} \cos(\alpha_0) & -\sin(\alpha_0) \\ \sin(\alpha_0) & \cos(\alpha_0) \end{bmatrix}}_{\boldsymbol{B} = \text{Control Effectiveness Matrix}} \begin{bmatrix} \Delta T_X \\ \Delta T_Z \end{bmatrix} \tag{6.48}$$

In the TR mode, similar to what was done for FW flight, the reference flight path angle ($\gamma_{ref}$) from the NDI for linear kinematics could be passed through a PID controller to output $\theta_{ref}$ . However, in TR mode, the flight path angle ($\gamma$) is additionally influenced by the vertical thrust component ($T_Z$). Ideally, the controller should take into account the strong coupling between $\gamma$ and the velocity $V$ through $T_Z$ when determining the reference pitch angle.

In the literature, for the simpler case of zero vertical thrust ([26]), this coupling is addressed by extending the output vector for the linear dynamics to a 2 × 1 matrix consisting of $[T_X, \theta]^T$. Similarly, in this work, we can expand the output to a 3 × 1 matrix consisting of $[T_X, T_Z, \theta]^T$. The expression to be inverted (using a pseudo-inverse) becomes:

$$\Delta \begin{bmatrix} \dot{V} \\ \dot{\gamma} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{mV_0} \end{bmatrix} \begin{bmatrix} \cos(\alpha_0) & -\sin(\alpha_0) & (-T_{X_0}\sin(\alpha_0) - T_{Z_0}\cos(\alpha_0) - qSC_{D_{\alpha_0}}) \\ \sin(\alpha_0) & \cos(\alpha_0) & (T_{X_0}\cos(\alpha_0) - T_{Z_0}\sin(\alpha_0) + qSC_{L_{\alpha_0}}) \end{bmatrix}}_{\boldsymbol{B} = \text{Control Effectiveness Matrix}} \begin{bmatrix} \Delta T_X \\ \Delta T_Z \\ \Delta\theta \end{bmatrix} \tag{6.49}$$

It must be noted that $\Delta\alpha = \Delta\theta - \Delta\gamma \approx \Delta\theta$ because of time scale separation.

A comparison for the two different strategies is reported below to select the most suitable method from different inputs in terms of flight path variables $[V, \chi, \gamma]$.

1. The nominal scenario involves a straight, constant altitude transition. Therefore their performance to a step in V while keeping $\chi$ and $\gamma$ to zero is evaluated:

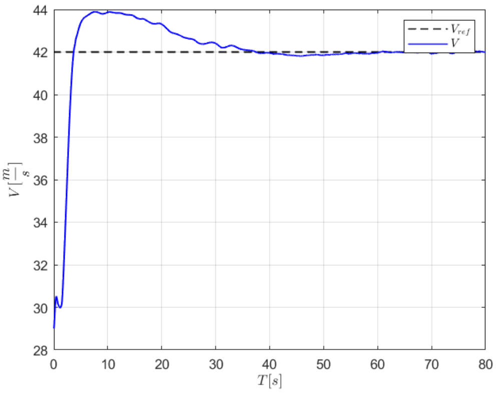


**Figure 6.56:** V tracking - strategy 1

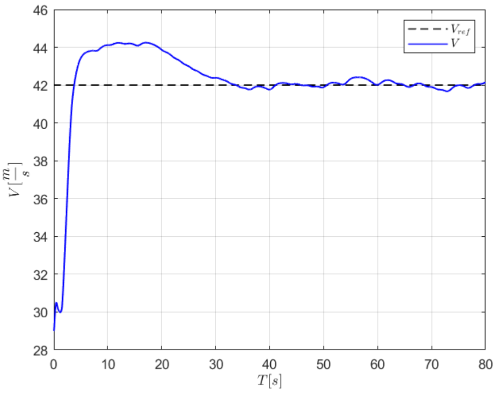


**Figure 6.57:** V tracking - strategy 2

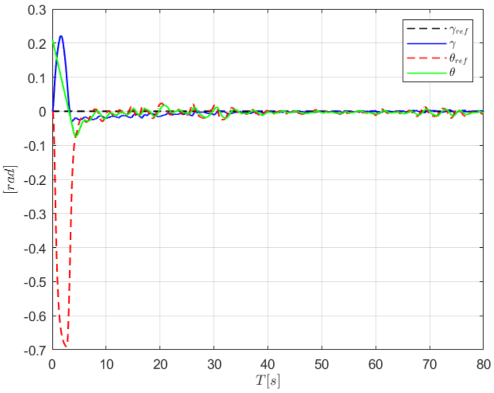


**Figure 6.58:** $\gamma$ regulation - strategy 1

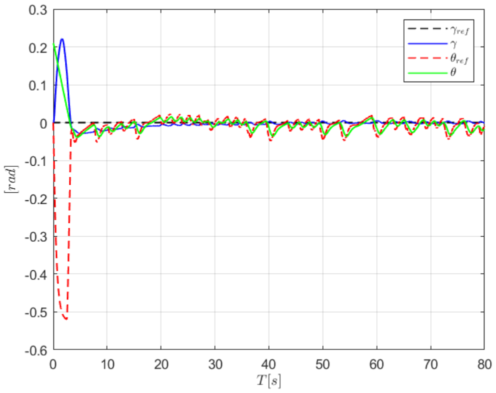


**Figure 6.59:** $\gamma$ regulation - strategy 2

In the nominal scenario, the basic strategy (strategy 1, same as for FW) consisting of using $\gamma_{ref}$ from NDI directly to produce $\theta_{ref}$ seems to perform better in terms of smaller overshoot for $V$ and smoother $\theta_{ref}$. However, future studies will involve playing around with appropriate filtering and gains.

2. Next a different scenario involving dynamic variations in the longitudinal plane is considered. $\chi$ is still kept at zero, however both $V$ and $\gamma$ are commanded to vary through a step + sine wave.

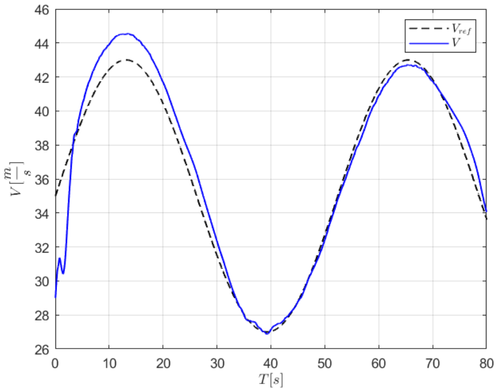


**Figure 6.60:** V tracking - strategy 1

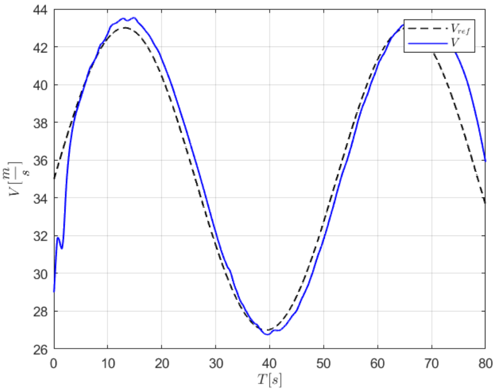


**Figure 6.61:** V tracking - strategy 2

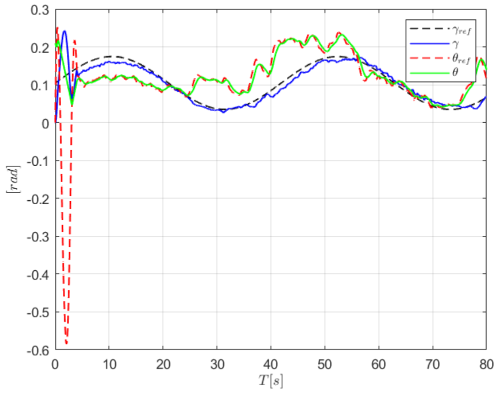


**Figure 6.62:** $\gamma$ tracking - strategy 1

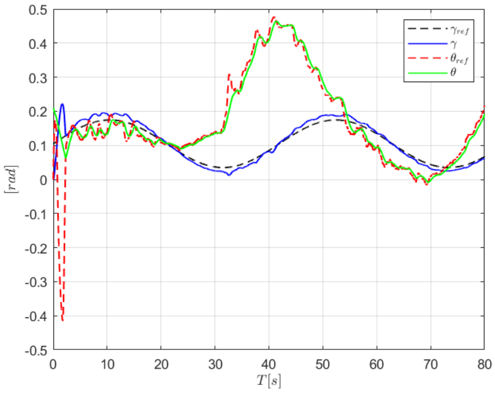


**Figure 6.63:** $\gamma$ tracking - strategy 2

Again, the two strategies show very similar performances. However, strategy 1 provides a smoother $\theta_{ref}$.

3. Next, a complete maneuver involving variations in azimuth is analysed. All three flight path variables are required to track a step + sine wave command. This is far from the nominal scenario, however the controller should be able to respond even in case of emergency.

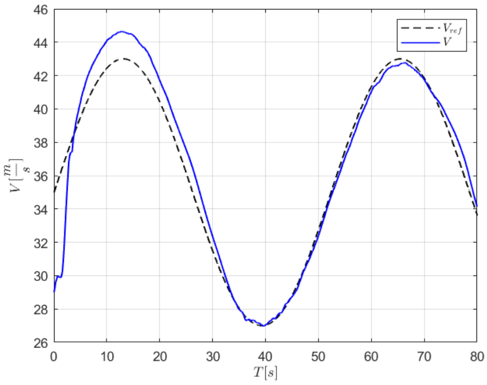


**Figure 6.64:** V tracking - strategy 1

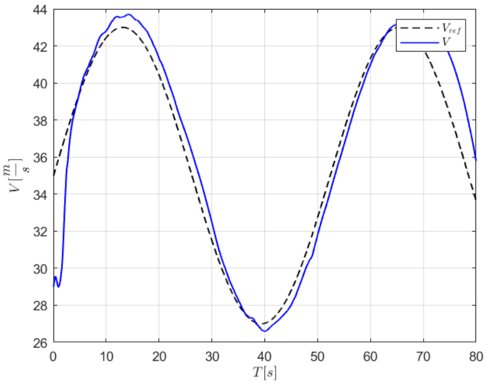


**Figure 6.65:** V tracking - strategy 2

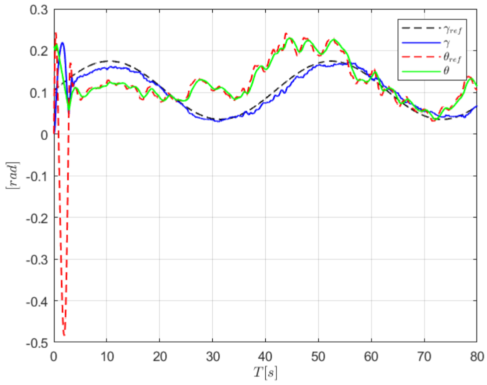


**Figure 6.66:** $\gamma$ tracking - strategy 1

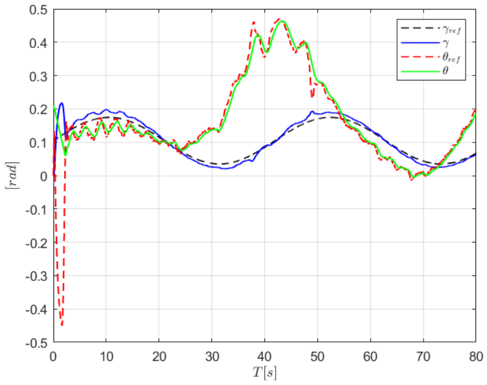


**Figure 6.67:** $\gamma$ tracking - strategy 2

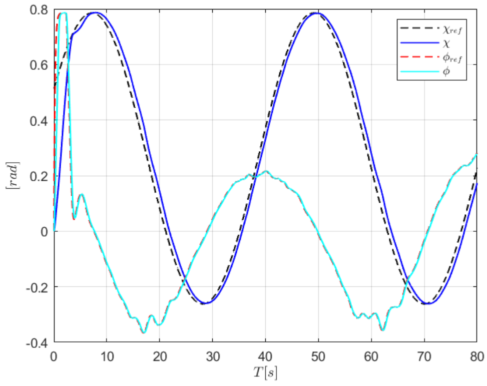


**Figure 6.68:** $\chi$ tracking - strategy 1

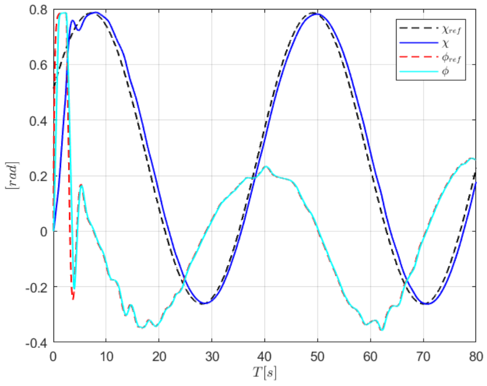


**Figure 6.69:** $\chi$ tracking - strategy 2

Similar to the findings in previous case studies, it seems that strategy 1 remains the more effective option in terms of achieving a smoother reference pitch.

The comparison between the two strategies can be quantified using two main criteria:

1. **Root Mean Square Error (RMSE)**: The pitch angle is used to track the flight path angle; therefore, we are interested in minimizing the flight path error rather than pitch. Therefore, it makes sense to use the RMSE for $\gamma$:

$$RMSE_{\gamma} = \sqrt{\sum_{i=1}^{N} \frac{(\gamma_{ref_i} - \gamma_i)^2}{N}} \tag{6.50}$$

   A reduced RMSE reflects enhanced tracking performance for the flight path angle.

2. **Standard Deviation ($\sigma$)**: To ensure passenger comfort, we aim for the controller to execute smooth attitude changes. This can be quantified as the standard deviation of the pitch angle, reflecting how much it oscillates around its mean value. The standard deviation of $\theta$ is calculated as:

$$\overline{\theta} = \sum_{i=1}^{N} \frac{\theta_i}{N} \quad \rightarrow \quad \sigma_{\theta} = \sqrt{\sum_{i=1}^{N} \frac{(\overline{\theta} - \theta_i)^2}{N}} \tag{6.51}$$

   A smaller Standard Deviation suggests fewer variations in attitude concerning the mean value, which indicates a more comfortable experience for the passenger.

To assess the full outer loop performance, the RMSE for velocity is employed as well. The following table presents such values for both strategies for all three cases analyzed:

| | | **Controller 1** | **Controller 2** |
|---|---|---|---|
| **Case 1** | $\sigma_\theta$ | 0.0217 | 0.0234 |
| | $RMSE_\gamma$ | 0.0262 | 0.0261 |
| | $RMSE_V$ | 1.959 | 2.003 |
| **Case 2** | $\sigma_\theta$ | 0.0517 | 0.1246 |
| | $RMSE_\gamma$ | 0.0171 | 0.0151 |
| | $RMSE_V$ | 1.133 | 1.195 |
| **Case 3** | $\sigma_\theta$ | 0.054 | 0.1201 |
| | $RMSE_\gamma$ | 0.0181 | 0.0143 |
| | $RMSE_V$ | 1.255 | 1.367 |

**Table 6.4:** Comparison Table

The table data indicates that Strategy 1 exhibits slightly less accurate tracking for $\gamma$ and slightly better performance for $V$ compared to Strategy 2. However, Strategy 1 significantly outperforms the second strategy in terms of minimizing pitch reference oscillations.

Finally, a complete maneuver plus wind disturbances is considered to investigate the robustness of the two different approaches. The NED components of simulated wind are as follows:

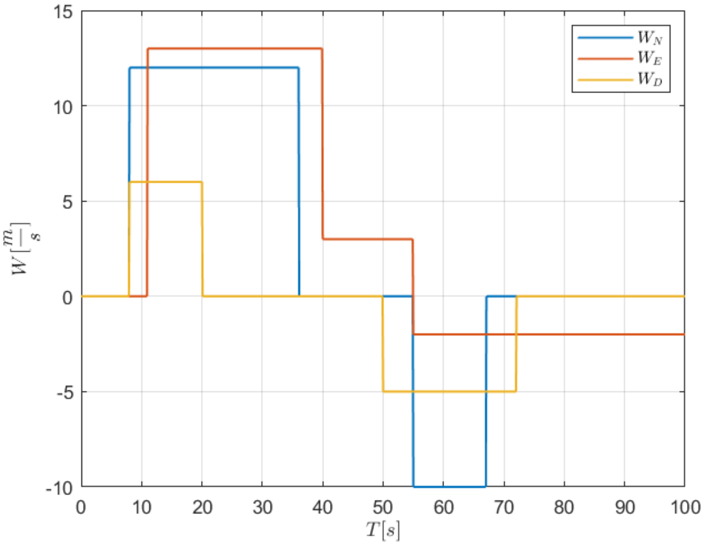


**Figure 6.70:** Wind velocity

Taking into account the presence of the previously mentioned wind disturbance, it was observed that only the basic controller could adequately track the input. This, in addition to the findings displayed in the previous table left no doubts in choosing strategy 1 for all upcoming comprehensive mission simulations. The performance of this selected strategy is illustrated below.

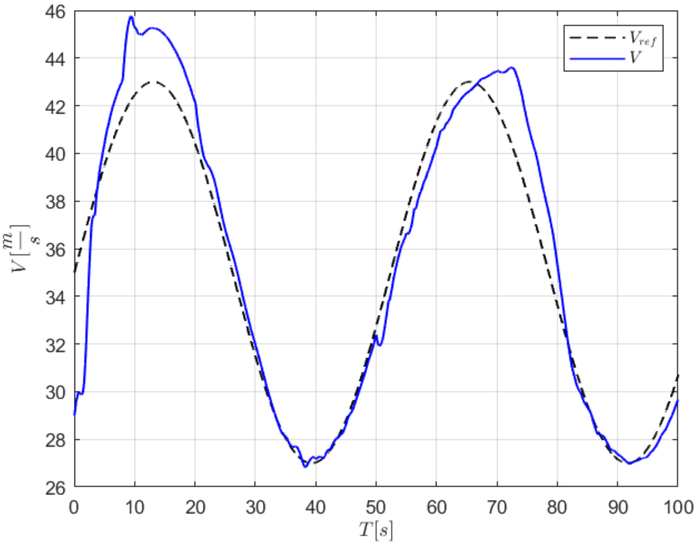


**Figure 6.71:** V tracking

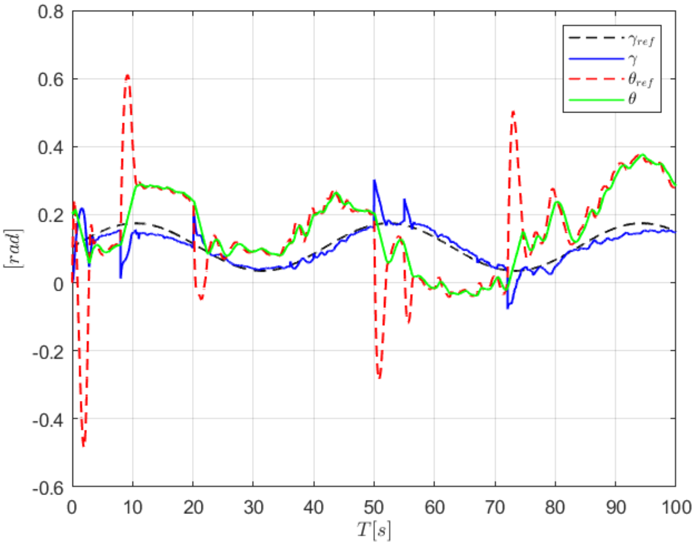


**Figure 6.72:** $\gamma$ tracking

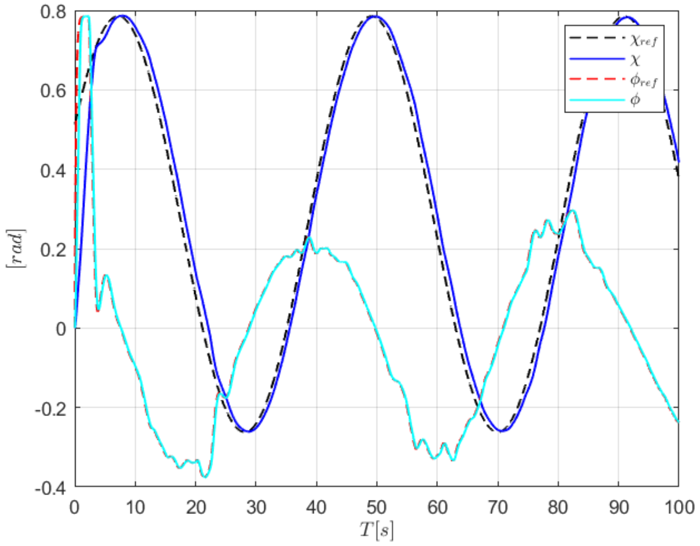


**Figure 6.73:** $\chi$ tracking

The results indicate that external inputs have minimal impact on the tracking of the heading angle ($\chi$). In contrast, variables in the longitudinal plane, such as airspeed ($V$) and flight path angle ($\gamma$), are influenced by the presence of wind. Nevertheless, the controller effectively manages and limits the effects of these disturbances, ensuring robust tracking performance.

## 6.5 Control Allocation Algorithm

For each state of the system, direct force control requires a control allocation algorithm in order to convert the desired forces and moments into tangible actuator commands. Knowing the mathematical relationship between control actions and actuation variables, this can be easily done by inverting such equation.

As stated in the previous chapters, INDI provides an increment in control actions and estimates of their current values are needed. Such estimates can be performed by using the mathematical expression between actions and actuation variables in the forward direction, where the current states of actuators are provided through sensor measurements.

In the most general case with 6DOF control, according to INDI, the control action to be converted into actuator inputs is given by the sum of the action increment and the current action estimate, as follows:

$$\boldsymbol{A}_C^B(t+dt) = \begin{bmatrix} \boldsymbol{F}_C^B \\ \boldsymbol{M}_C^B \end{bmatrix}(t+dt) = \hat{\boldsymbol{A}}_C^B(t) + \Delta \boldsymbol{A}_C^B \tag{6.52}$$

Once the required control action coming from the control system is computed, it can be converted into actuator input ($\boldsymbol{u}_A$):

$$\boldsymbol{A}_C^B = \boldsymbol{f}(\boldsymbol{u}_A) \;\; \rightarrow \;\; \boldsymbol{u}_A = \boldsymbol{f}^{-1}(\boldsymbol{A}_C^B) \tag{6.53}$$

### 6.5.1 MR control allocation

Control allocation for MR mode is relatively straightforward and entails a matrix inversion process. Drawing parallels with equation (3.30), the control action vector is connected to actuator variables through a fixed allocation matrix denoted as $M$:

$$\boldsymbol{A}_{MR}^B = \begin{bmatrix} T_Z \\ \tau_X \\ \tau_Y \\ \tau_Z \end{bmatrix} = [M]\boldsymbol{\Omega}_V^2 \;\; \rightarrow \;\; \boldsymbol{\Omega}_V = \sqrt{[M]^\dagger \boldsymbol{A}_{MR}^B} \tag{6.54}$$

It's important to emphasize that a Moore-Penrose pseudoinverse is employed due to the dimensions of the allocation matrix ($[M] \in \mathbb{R}^{4\times 6}$).

### 6.5.2 FW control allocation

In FW mode, the allocation matrix has to be appropriately constructed. First of all, gyroscopic effects due to thrust propellers and aerodynamic surface deflections effects on lift can be safely neglected. Under these assumptions, we can write the following:

$$\begin{cases} T_X = K_{T_H}(\Omega_{H1}^2 + \Omega_{H2}^2) \\ m_c = l_{\delta_a}\delta_a + K_{Q_H}(\Omega_{H1}^2 - \Omega_{H2}^2) \\ l_c = m_{\delta_e}\delta_e + K_{T_H} d_{Z_H}(-\Omega_{H1}^2 - \Omega_{H2}^2) \\ n_c = n_{\delta_a}\delta_a + K_{T_H} d_{Y_H}(\Omega_{H1}^2 - \Omega_{H2}^2) \end{cases} \tag{6.55}$$

The control allocation matrix is a direct consequence of equation (6.55), and the inversion process takes the following form:

$$\begin{bmatrix} \delta_e \\ \delta_a \\ \Omega_{H1}^2 \\ \Omega_{H2}^2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & K_{T_H} & K_{T_H} \\ 0 & l_{\delta_a} & K_{Q_H} & -K_{Q_H} \\ m_{\delta_e} & 0 & -K_{T_H} d_{Z_H} & -K_{T_H} d_{Z_H} \\ 0 & n_{\delta_a} & K_{T_H} d_{Y_H} & -K_{T_H} d_{Y_H} \end{bmatrix}^{-1} \boldsymbol{A}_{FW}^B \tag{6.56}$$

The final step involves taking the square root of the rotational speeds.

### 6.5.3 TR control allocation

The allocation process for TR mode involves a great number of actuator variables, therefore a direct inversion approach is not feasible.

First, the problem of functional redundancy has to be addressed. In this context, redundancy refers to the fact that some of the control actions can be produced by multiple actuators:

- $T_X \rightarrow \boldsymbol{\Omega}_H$
- $T_Z \rightarrow \boldsymbol{\Omega}_V$
- $m_c \rightarrow \boldsymbol{\Omega}_V + \delta_a$
- $l_c \rightarrow \boldsymbol{\Omega}_V + \delta_e$
- $n_c \rightarrow \boldsymbol{\Omega}_V + \boldsymbol{\Omega}_H$

This challenge is tackled by distributing the control actions among the functionally redundant actuators using a velocity function. Given the assumption that aerodynamic actions become increasingly effective as velocity rises, the allocation of control load can be partitioned based on the aircraft's flight speed in the following manner([7]):

$$f(V) = \frac{V - V_{TR_i}}{V_{TR_f} - V_{TR_i}} \quad \rightarrow \quad \begin{bmatrix} m_c \\ l_c \\ n_c \end{bmatrix} = \underbrace{\begin{bmatrix} m_c \\ l_c \\ n_c \end{bmatrix} f(V)}_{\text{FW control}} + \underbrace{\begin{bmatrix} m_c \\ l_c \\ n_c \end{bmatrix} (1 - f(V))}_{\text{MR control}} \tag{6.57}$$

Here, $V_{TR_i}$ and $V_{TR_f}$ represent the initial and final velocities during the transition phase, respectively. It's worth noting that more advanced and refined partitioning methods will be explored in future investigations. However, this approach allows for a gradual and smooth shift from the first state of the plant to its cruising FW condition.

The significant advantage of direct force control lies in its ability to compute control actions in a straightforward manner, and the partitioning of these actions can be achieved sequentially and with ease, regardless of the location of the functional redundancy within the system.

The second challenge that needs to be addressed stems from the fact that each actuator generates both desired control effects and undesired secondary ones. These undesired effects cannot be disregarded during INDI since they operate on the same time scale as control actions. Each actuation will generate the following:

- $\delta_e \quad \rightarrow \quad \begin{cases} \text{Control effects: } m_{\delta_e}\delta_e \\ \text{Undesired effects: } L_{\delta_e}\delta_e \end{cases}$
- $\delta_a \quad \rightarrow \quad \begin{cases} \text{Control effects: } l_{\delta_a}\delta_a \\ \text{Undesired effects: } n_{\delta_a}\delta_a \end{cases}$
- $\boldsymbol{\Omega}_H \quad \rightarrow \quad \begin{cases} \text{Control effects: } T_x \ , \ n_p \\ \text{Undesired effects: } m_p \ , \ l_p \ , \ gyro \end{cases}$
- $\boldsymbol{\Omega}_V \quad \rightarrow \quad \begin{cases} \text{Control effects: } T_z \ , \ \tau_x \ , \ \tau_y \ , \ \tau_z \\ \text{Undesired effects: } gyro \end{cases}$

It can be safely assumed that gyroscopic effects are negligible compared to other control components. However, other undesired effects still need to be taken into account. The problem can be solved using a cascaded inversion for predicting and compensating undesired actions.

1. First, the aerodynamic control actions are converted into commanded deflections:

$$\begin{bmatrix} \delta_e \\ \delta_a \end{bmatrix} = \begin{bmatrix} 0 & l_{\delta_a} \\ m_{\delta_e} & 0 \end{bmatrix}^{-1} \begin{bmatrix} l_c \\ m_c \end{bmatrix} f(V) \tag{6.58}$$

   The susbsequent undesired effects can be expressed as $L_{\delta_e}\delta_e$ and $n_{\delta_a}\delta_a$.

2. Following, the desired control actions from thrust propellers are considered. Their required values need to be corrected to compensate for the undesired effects of aerodynamic surfaces.

$$\begin{cases} T_x' = T_x - L_{\delta_e}\delta_e \sin(\alpha) \\ n_p' = n_c f(V)' = n_c f(V) - n_{\delta_a}\delta_a \end{cases} \tag{6.59}$$

   The actuators commands become:

$$\boldsymbol{\Omega}_H = \frac{1}{K_{T_H}} \begin{bmatrix} 1 & 1 \\ d_{Y_H} & -d_{Y_H} \end{bmatrix}^{-1} \begin{bmatrix} T_x' \\ n_p' \end{bmatrix} \tag{6.60}$$

This inversion will produce $K_{Q_H}(\Omega_{H1}^2 - \Omega_{H2}^2)$ and $-K_{T_H} d_{Z_H}(\Omega_{H1}^2 + \Omega_{H2}^2)$ as undesired moments respectively along $X_B$ and $Y_B$.

3. Finally, lift propellers are considered. Compensation of previous undesired effects leads to the following new requested actions:

$$\begin{cases} T_z' = T_z - L_{\delta_e}\delta_e \cos(\alpha) \\ \tau_x' = l_c(1 - f(V))' = l_c(1 - f(V)) - K_{Q_H}(\Omega_{H1}^2 - \Omega_{H2}^2) \\ \tau_y' = m_c(1 - f(V))' = m_c(1 - f(V)) + K_{T_H} d_{Z_H}(\Omega_{H1}^2 + \Omega_{H2}^2) \end{cases} \tag{6.61}$$

The MR allocation algorithm can, then, be applied to the new requested MR action vector:

$$\boldsymbol{\Omega}_V = \sqrt{[M]^\dagger \begin{bmatrix} T_z' \\ \tau_x' \\ \tau_y' \\ \tau_z \end{bmatrix}} \tag{6.62}$$

The allocation method described in this subsection allows to effectively predict and compensate undesired effects that operate on the same time scale as control actions and therefore cannot be neglected during the first order Taylor expansion of the dynamical equations.

# Chapter 7

# Full Mission simulation

The objective of this chapter is to present the outcomes of a comprehensive mission that involves transitions between different states. Each controller has been individually validated and demonstrated to function effectively. However, it is imperative to establish the controller's ability to smoothly transition between control variables in practice. This may necessitate the adjustment of gains w.r.t. previous simulations to ensure satisfactory performance during each phase of the mission as well as during the transitions between phases. It's worth noting that aerobrakes or alternative deceleration methods are not yet incorporated at this stage. Consequently, the mission is designed to minimize the need for the high-efficiency vehicle to execute abrupt decelerations or descents. The primary objective is indeed to determine whether the combination of controllers can effectively handle transitions between states.

The vehicle is requested to undergo all flight phases, starting from ground. As stated earlier, the results shown will focus more on the transition between states rather than each single phase, since the effectiveness of the single controllers was already tested.

The complete mission profile is shown below.

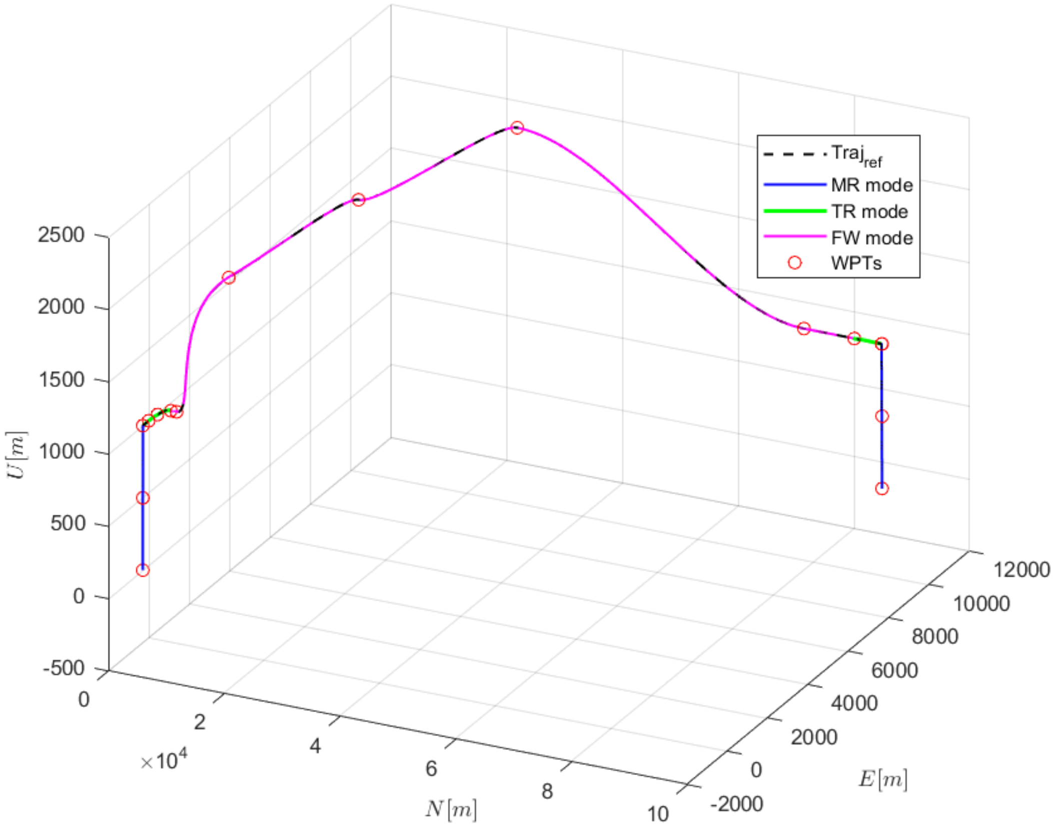


**Figure 7.1:** 3D trajectory for full mission

## 7.1 Forward Transition

In this context, ’forward transition’ pertains to the acceleration phase and encompasses two distinct state switches.

1. MR → TR

2. TR → FW

### 7.1.1 MR → TR

The first switch involves the activation of aerodynamic surfaces and thrust motors. We expect the vehicle to tilt backwards, given that forward propulsion now relies on thrust motors and vertical traction is entirely dedicated to counteracting gravity.

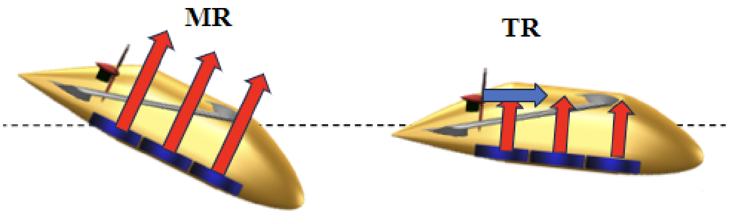


**Figure 7.2:** MR to TR

The state machine approach determines an on-off behaviour for FW control variables. This will certainly cause a fast change in pitch reference. In order for it to be attenuated, a limiter on the reference pitch rate is applied. It is necessary to limit accelerations for the sake of passengers’ comfort and structural integrity.

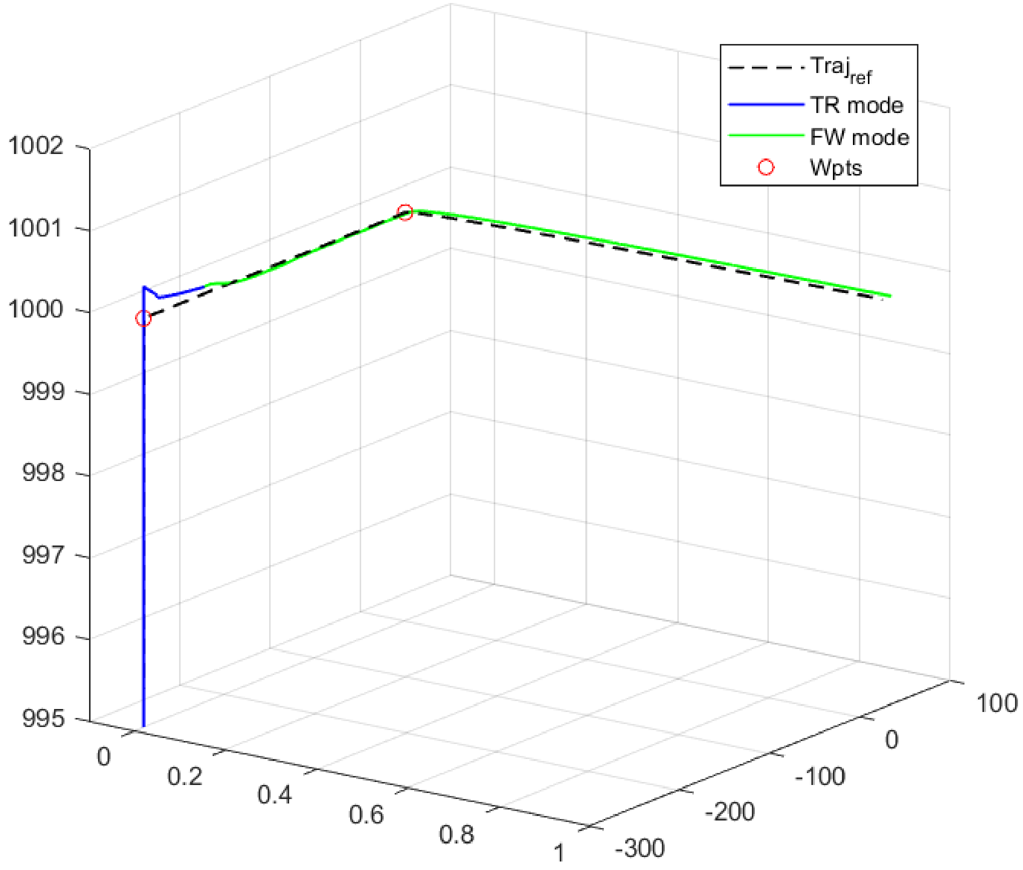


**Figure 7.3:** 3D plot for switch n.1

From the provided figure, it's evident that the transition results in a minor altitude increase. This can be attributed to the upward pitching motion, aligning the vertical thrust with the local vertical. This transient surge in lift contributes to the elevation gain. Notably, the precise switching point occurs while in linear motion, primarily affecting linear state variables.

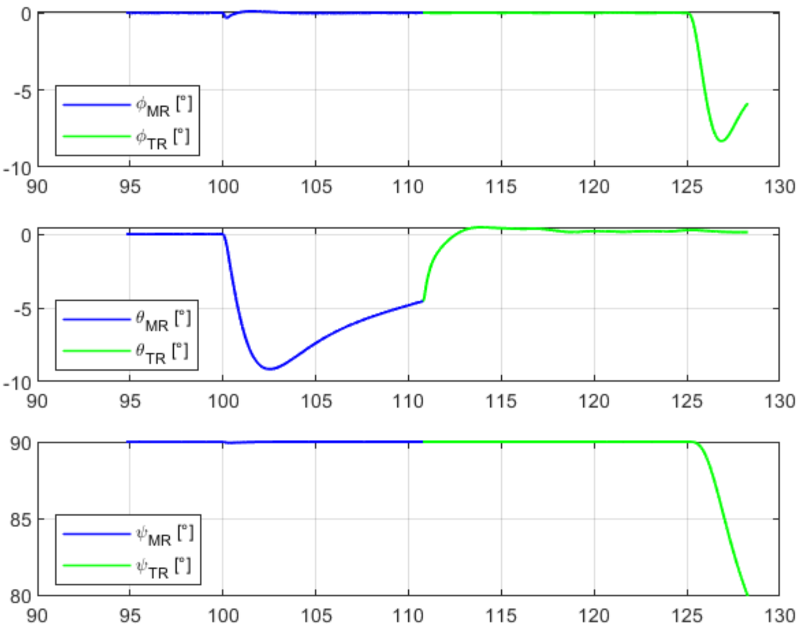


**Figure 7.4:** **Φ** for switch n.1

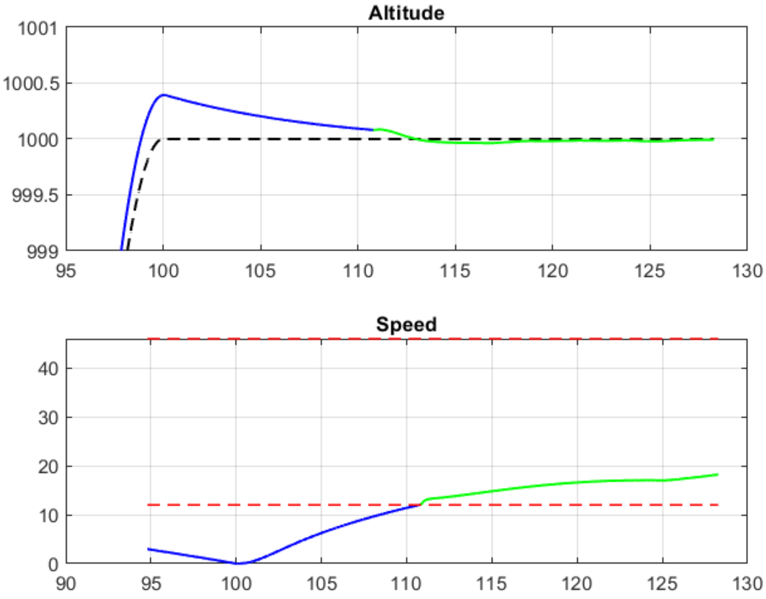


**Figure 7.5:** h and V vs t for switch n.1

As mentioned earlier, this switch causes the aircraft to pitch up. The change in attitude (Fig. 7.4) is not abrupt and residual oscillations decay in a few seconds. Figure 7.5 shows that the controller is able to respond to a height change very quickly resulting in an imperceptible altitude error.

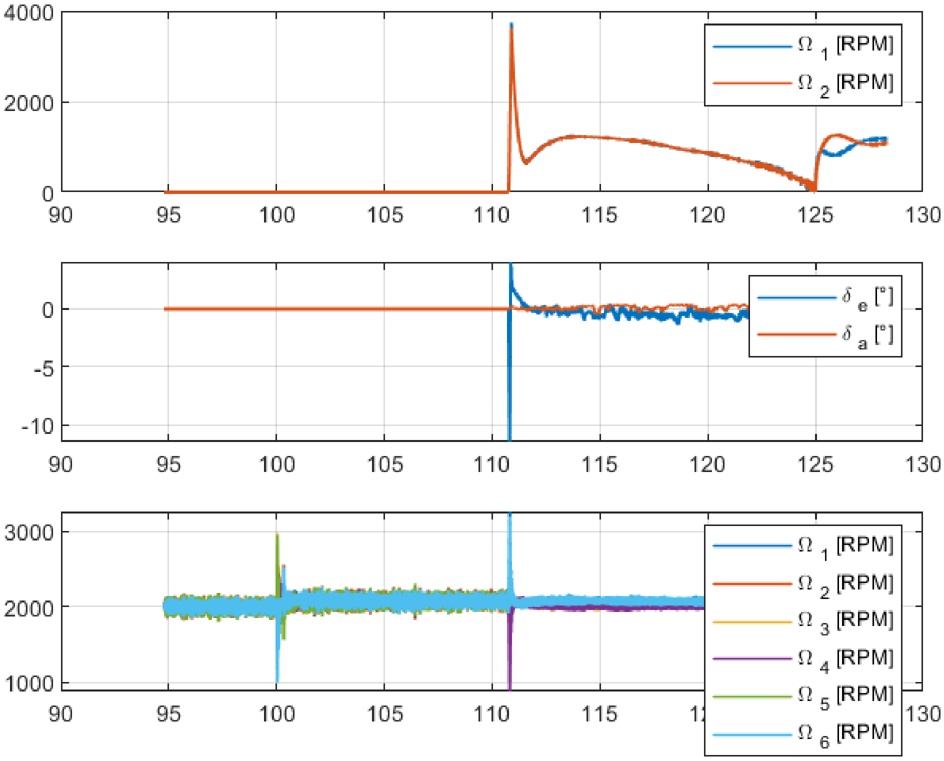


**Figure 7.6:** Control variables for switch n.1

Before the switching time point, all FW control variables are set to zero. After the transition takes place, aerodynamic surface deflections continues being small owing to the velocity based allocation algorithm.

### 7.1.2 TR → FW

The second transition involves deactivating the lift motors. Rather than adopting an on-off behavior, we've chosen to implement an exponential function for a gradual reduction in motor power. This approach aims to ensure a smoother and more controlled transition.

The deactivation of the lift motors is expected to result in a reduction of the upward force, which, in turn, may lead to a decrease in altitude. To counteract this effect, we anticipate the vehicle adjusting its Angle of Attack (AoA) to generate additional aerodynamic lift, compensating for the absence of the lift motors.

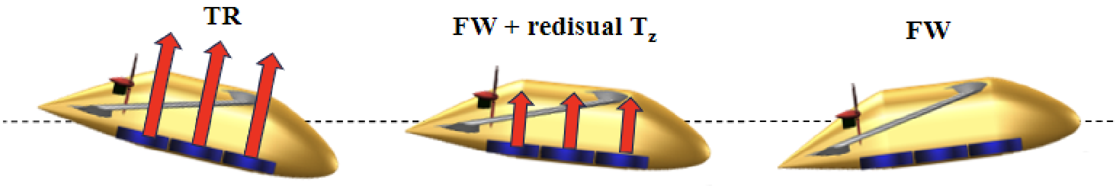


**Figure 7.7:** TR to FW

Again, a rate limiter for pitch is essential for avoiding unacceptable angular accelerations.

The 3D plot depicted below illustrates that, on this occasion, the switch takes place during a turning maneuver. As a result, any alterations in lateral-directional state variables should be attributed to the maneuver itself rather than the transition process.

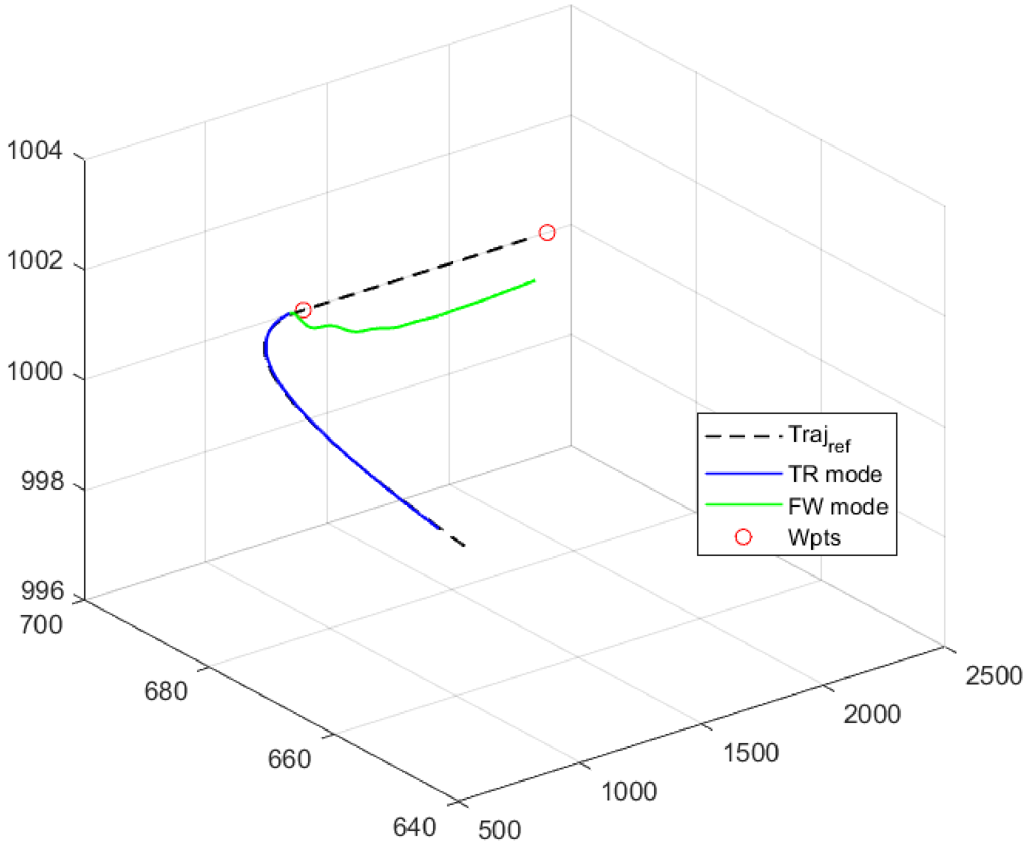


**Figure 7.8:** 3D plot for switch n.2

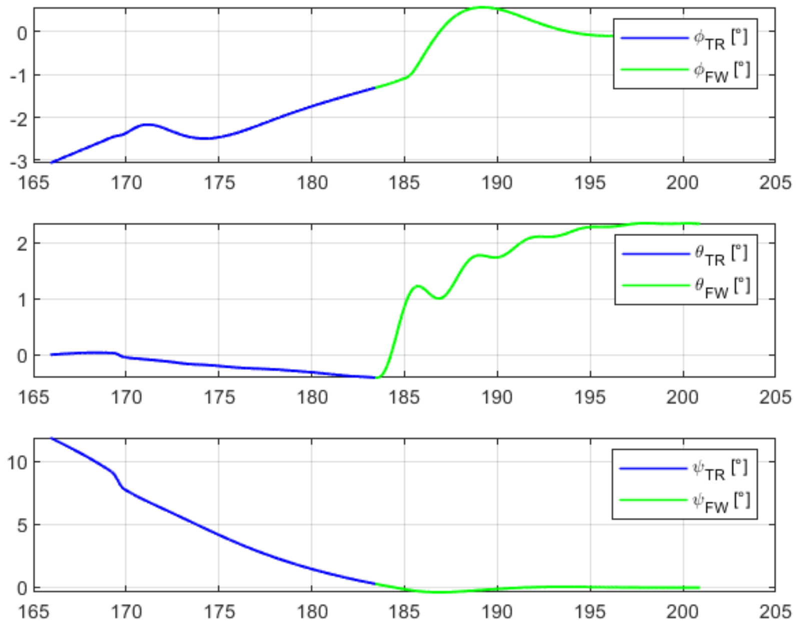


**Figure 7.9:** **Φ** for switch n.2

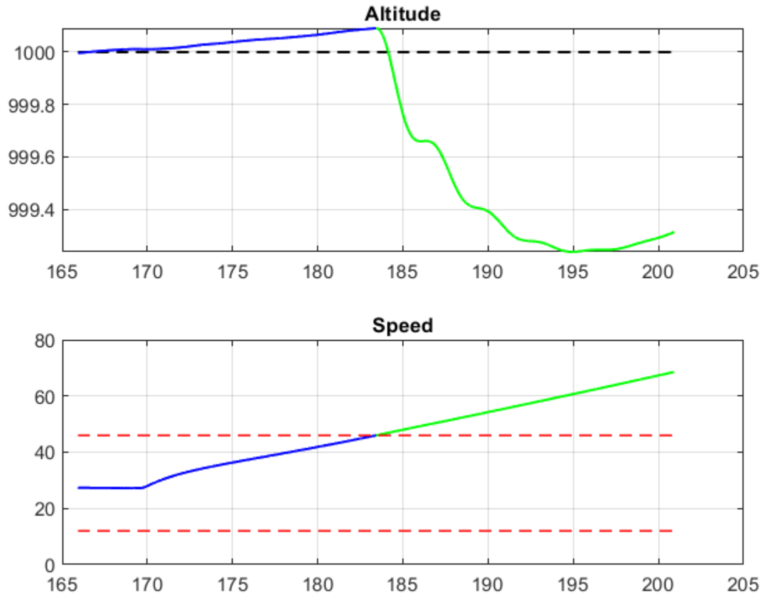


**Figure 7.10:** h and V for switch n.2

As mentioned earlier, the mode switch causes the aircraft to pitch up. The change in attitude is satisfactorily smooth and its small oscillations decay in very few seconds. As expected, there is a small altitude loss, however the controller is able to quickly respond and recover the requested flight level.

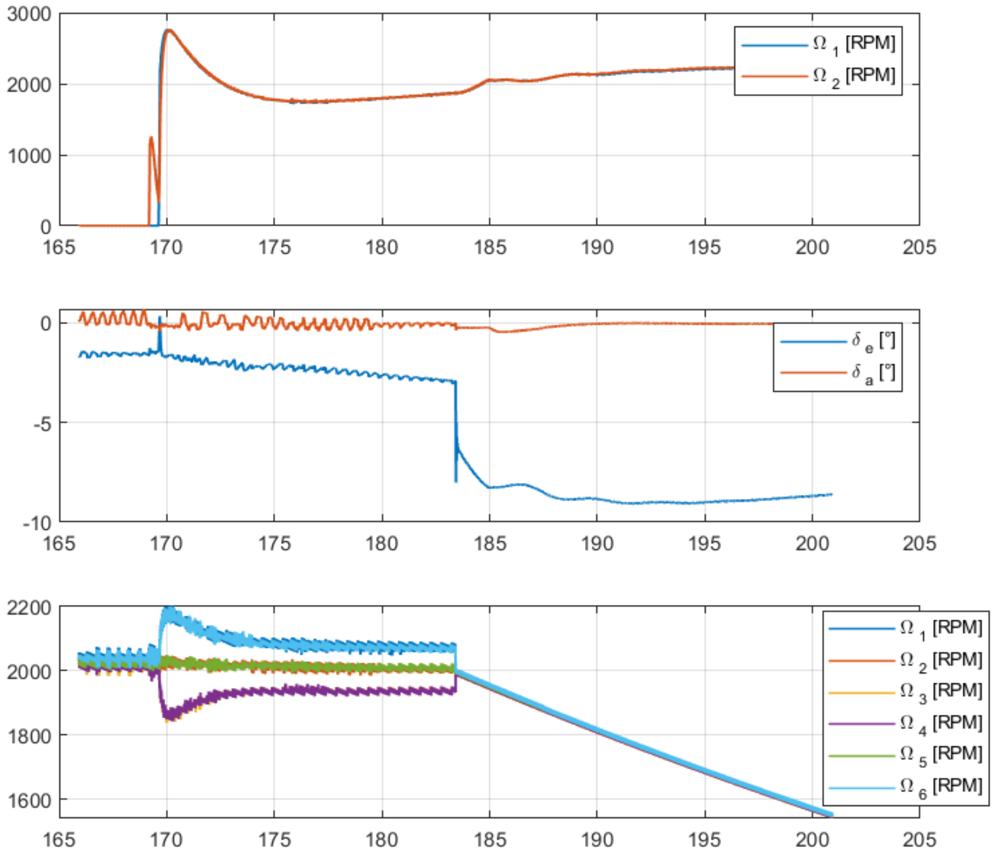


**Figure 7.11:** Control variables for switch n.2

The plot for control variables depicts the imposed exponential decrease of vertical thrust.

## 7.2 Backward transition

Backward transition is the reverse process of forward transition. During deceleration the aircraft will undergo the following:

1. FW → TR
2. TR → MR

### 7.2.1 FW → TR

First, the performance of the controller in switching from FW to hybrid is assessed. The main abrupt change that may cause problem involves the sudden starting of lift motors, which can provoke an overshoot in terms of altitude. This should be contrasted by the aircraft pitching nose down to decrease lift.

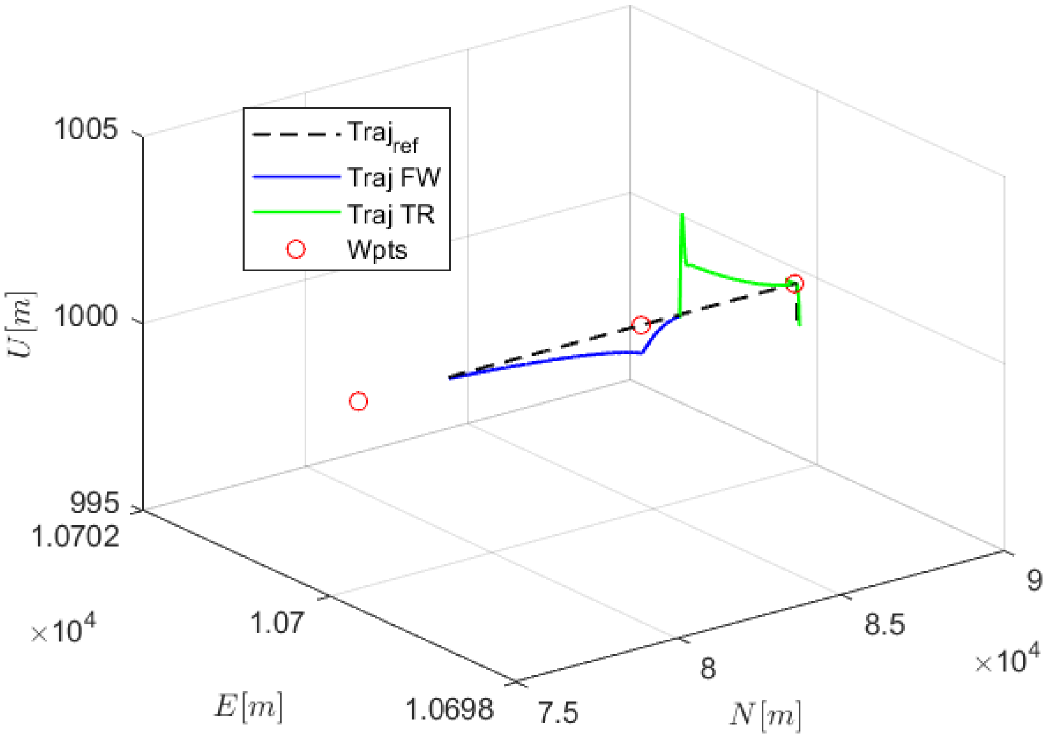


**Figure 7.12:** 3D plot for switch n.3

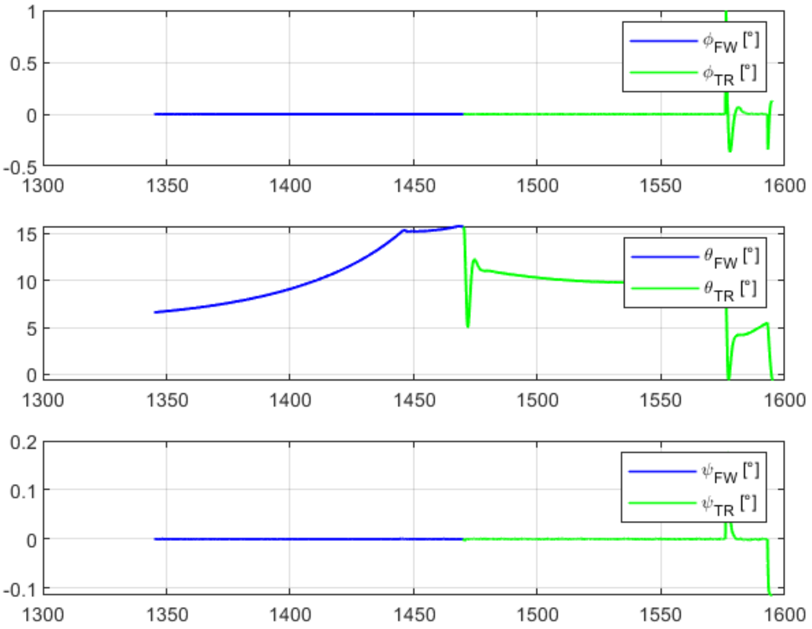


**Figure 7.13:** $\mathbf{\Phi}$ for switch n.3

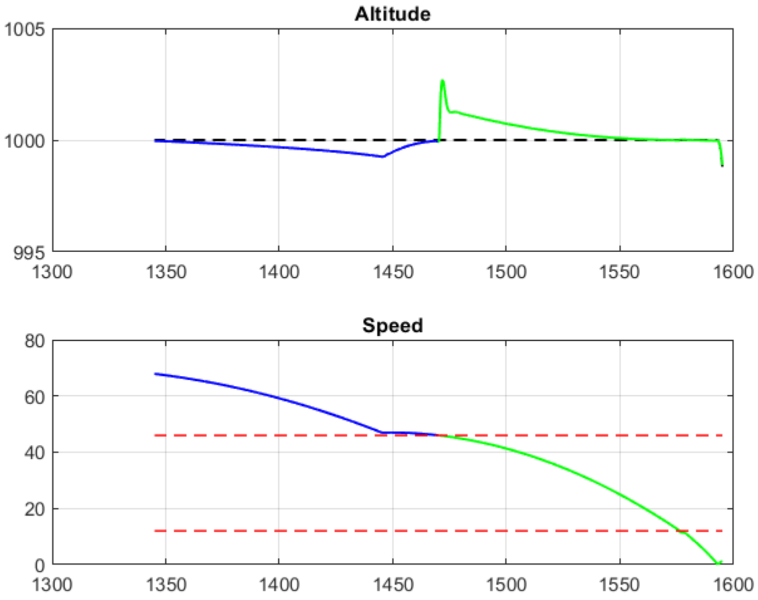


**Figure 7.14:** h and V for switch n.3

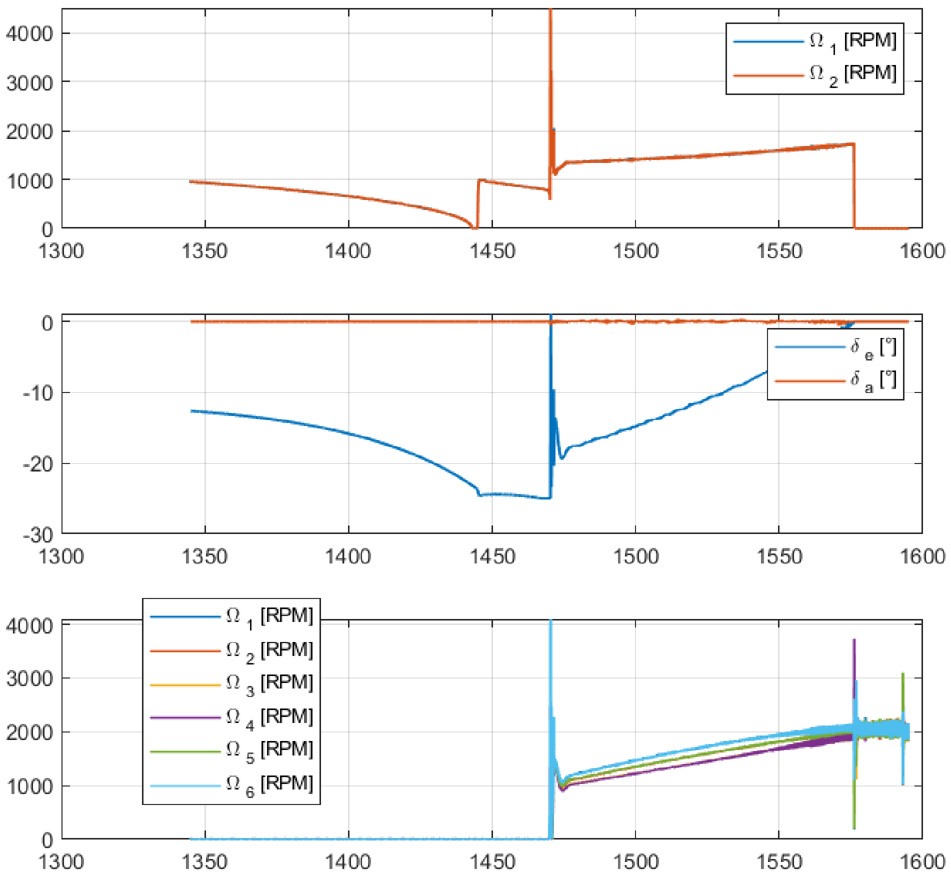


**Figure 7.15:** Control variables for switch n.3

### 7.2.2 TR → MR

The concluding transition involves a shift from hybrid mode to MR mode. The primary adaptation in control variables revolves around discontinuing horizontal thrust. To counterbalance this change, the aircraft must initiate a nose-down pitch, resulting in a slight increase in altitude due to the alignment of vertical thrust with the local vertical. In this configuration, the lift motors assume the responsibility of providing the new forward force necessary to sustain linear motion.

It's important to highlight that there is no need for a gradual reduction in power for the thrust motors. This is particularly true at low speeds, where aerodynamic drag is minimal, resulting in low levels of horizontal thrust.

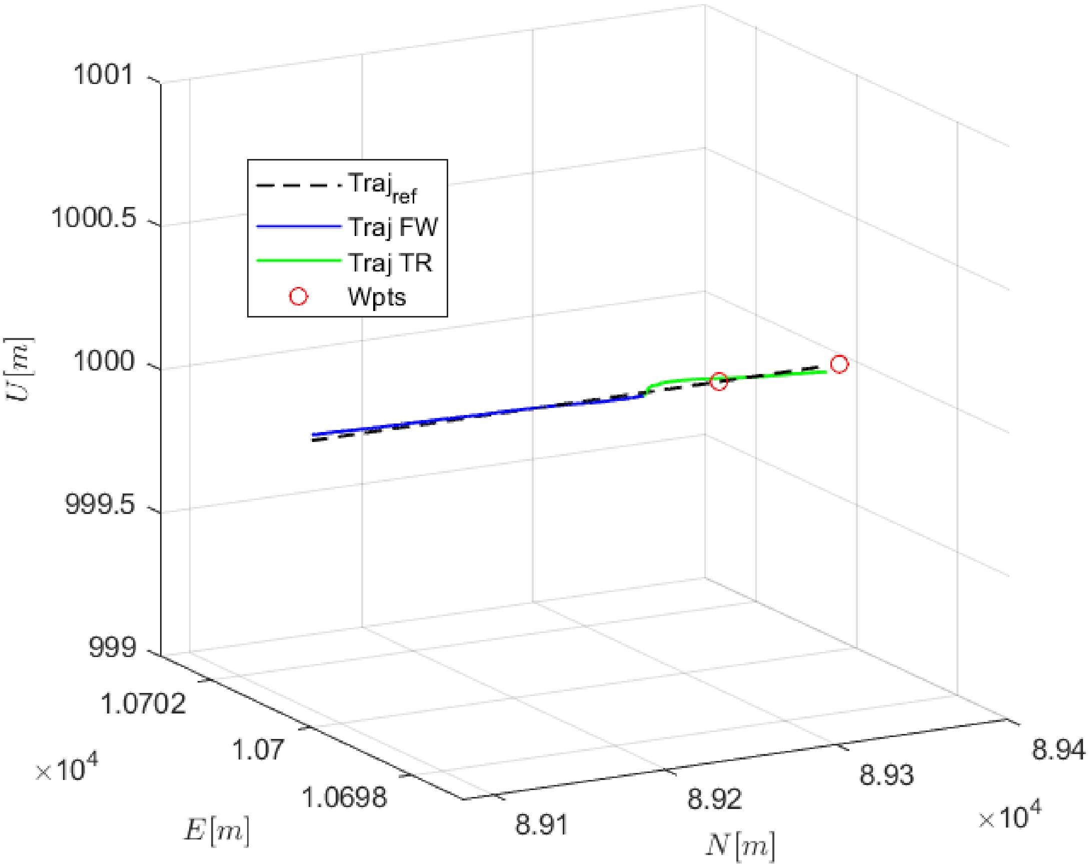


**Figure 7.16:** 3D plot for switch n.4

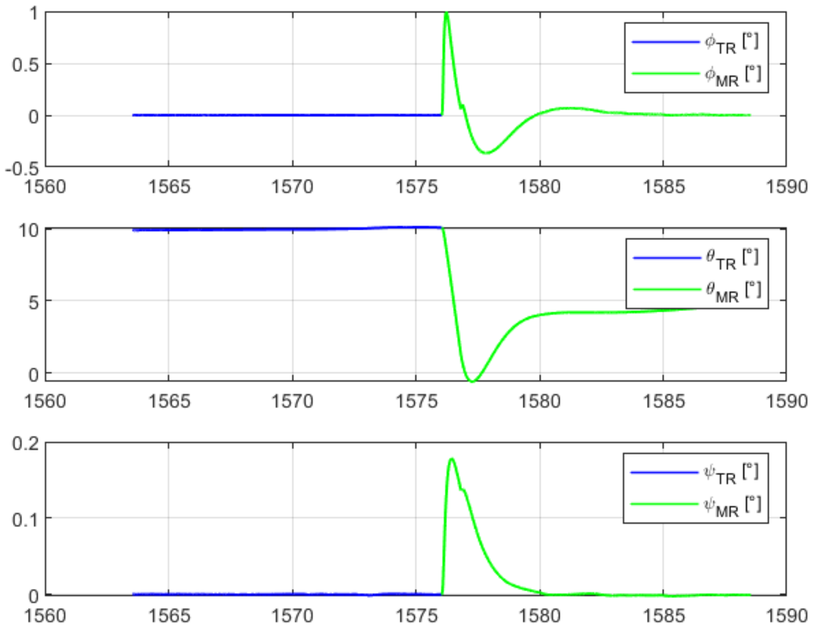


**Figure 7.17:** **Φ** for switch n.4

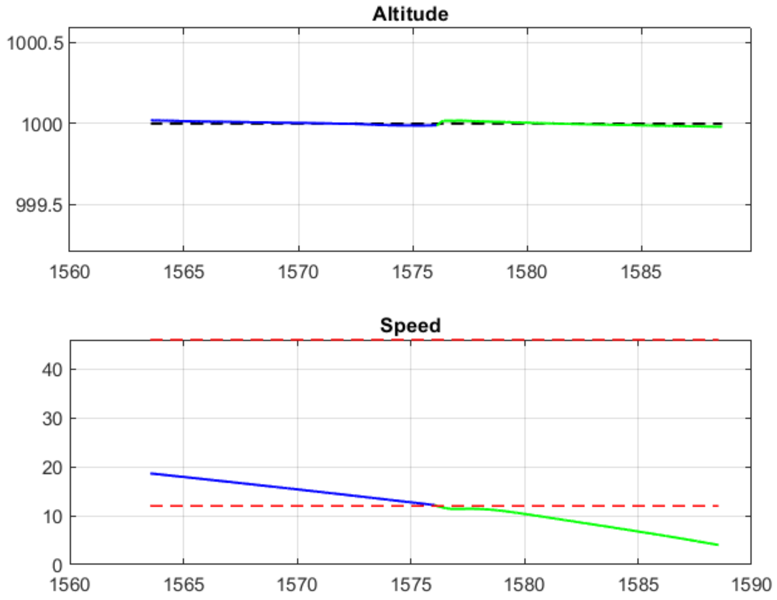


**Figure 7.18:** h and V for switch n.4

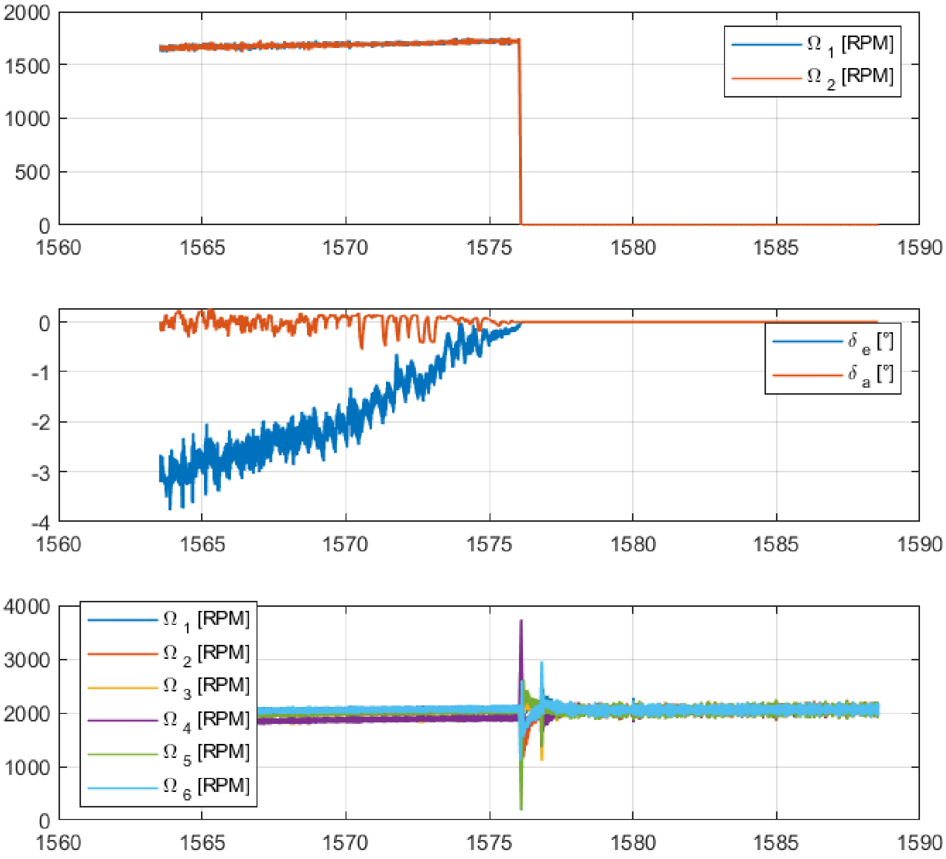


**Figure 7.19:** Control variables for switch n.4

## 7.3 Energy consumption

It is essential to quantify the energy required by the control system for the aircraft to terminate the mission. Clearly, the most energy intensive phases are MR and TR due to the presence of vertical thrust to balance gravity. In order to minimize battery weight, these two modes have to be terminated as quickly as possible.

For a primitive analysis, we can assume that the energy consumed by aerodynamic surfaces is much smaller than that requested by propellers:

$$P(t) = \sum_{i=1}^{6} P_{VT_i} + \sum_{j=1}^{2} P_{HT_j} + P_{\delta_e} + P_{\delta_a} \approx \sum_{i=1}^{6} P_{VT_i} + \sum_{j=1}^{2} P_{HT_j} \tag{7.1}$$

Knowing that the power required by each rotor can be expressed as $Q(\Omega){\cdot}\Omega = K_Q\Omega^3$, we can rearrange equation (7.1) as:

$$P(t) = K_{Q_V} \cdot \Omega^3_{V_1} + \cdots + K_{Q_V} \cdot \Omega^3_{V_6} + K_{Q_H} \cdot \Omega^3_{H_1} + K_{Q_H} \cdot \Omega^3_{H_2} \tag{7.2}$$

The time-history of requested power for the full mission is shown below.

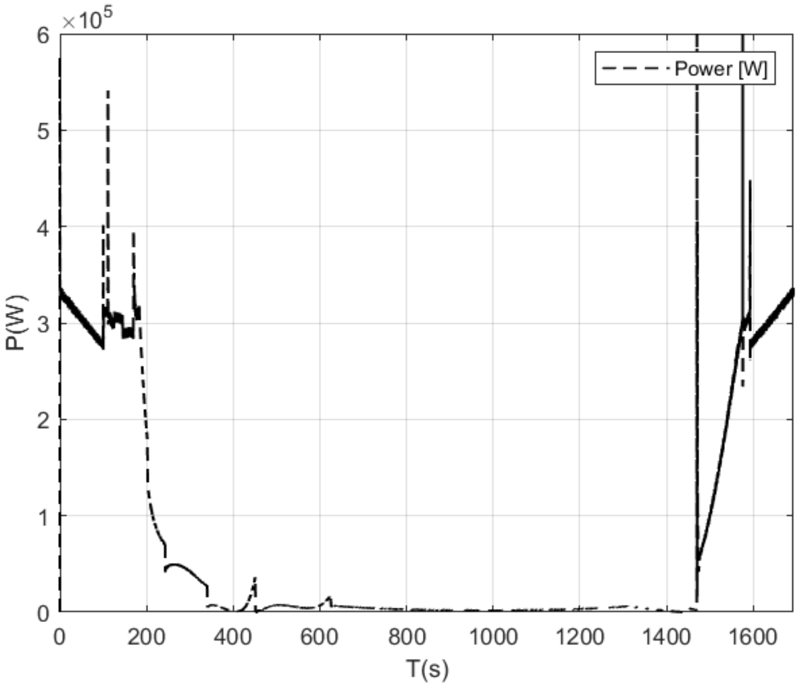


**Figure 7.20:** Power(Watts) over time

As anticipated, the instantaneous power demand when the lift motors are active is significantly higher, ranging from one to two orders of magnitude greater than during FW flight.

Finally, we can compute the energy consumed throughout the mission by integrating the consumed power over time:

$$E = \int_0^{t_f} (K_{Q_V} \cdot \Omega_{V_1}^3 + \cdots + K_{Q_V} \cdot \Omega_{V_6}^3 + K_{Q_H} \cdot \Omega_{H_1}^3 + K_{Q_H} \cdot \Omega_{H_2}^3) d\tau \tag{7.3}$$

In this context, fixed-step explicit Euler was used for numerical integration, therefore we obtain:

$$E \approx \sum_{k=1}^{N-1} [K_{Q_V} \sum_{i=1}^{6} (\Omega_{V_i}^{(k+1)^3} - \Omega_{V_i}^{(k)^3}) + K_{Q_H} \sum_{i=1}^{2} (\Omega_{H_i}^{(k+1)^3} - \Omega_{H_i}^{(k)^3})] \Delta t \tag{7.4}$$

This value was calculated to be 1.2514$e$8J.

Considering a classical Li-ion battery with[36]:

- Specific Energy Density(SED): $0.15 \frac{kW \cdot h}{Kg}$;
- Depth of Discharge(DoD): 65%;
- Efficiency($\eta$): 93%.

We can now estimate the weight of the battery needed to perform a similar mission as:

$$m = \frac{E}{DoD \cdot \eta \cdot SED} = 386.2Kg \tag{7.5}$$

Which is 18.4% of the total mass of the aircraft, conforming with similar vehicles.

## 7.4 Full Mission with Wind disturbances

The identical mission was simulated, incorporating wind disturbances introduced at various time points across all flight phases. Additionally, wind disturbances were strategically introduced at moments corresponding to the mode switches to do a preliminary evaluation of the robustness and effectiveness of the switching logic under non-nominal conditions.

Below is the plot of wind inputs in NED components for the whole mission.

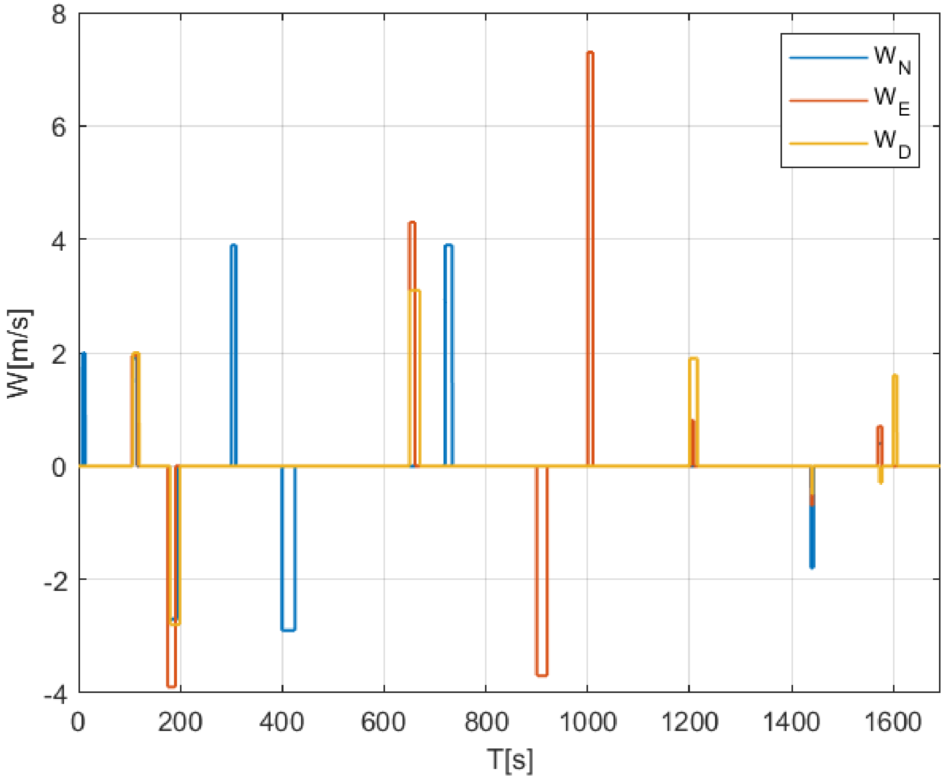


**Figure 7.21:** Wind disturbances for full mission

Following that, the controller's ability to mitigate external disturbances during mode switches is demonstrated through plots depicting the actual altitude and ground track compared to their respective reference values.

1. Switch n.1: MR → TR

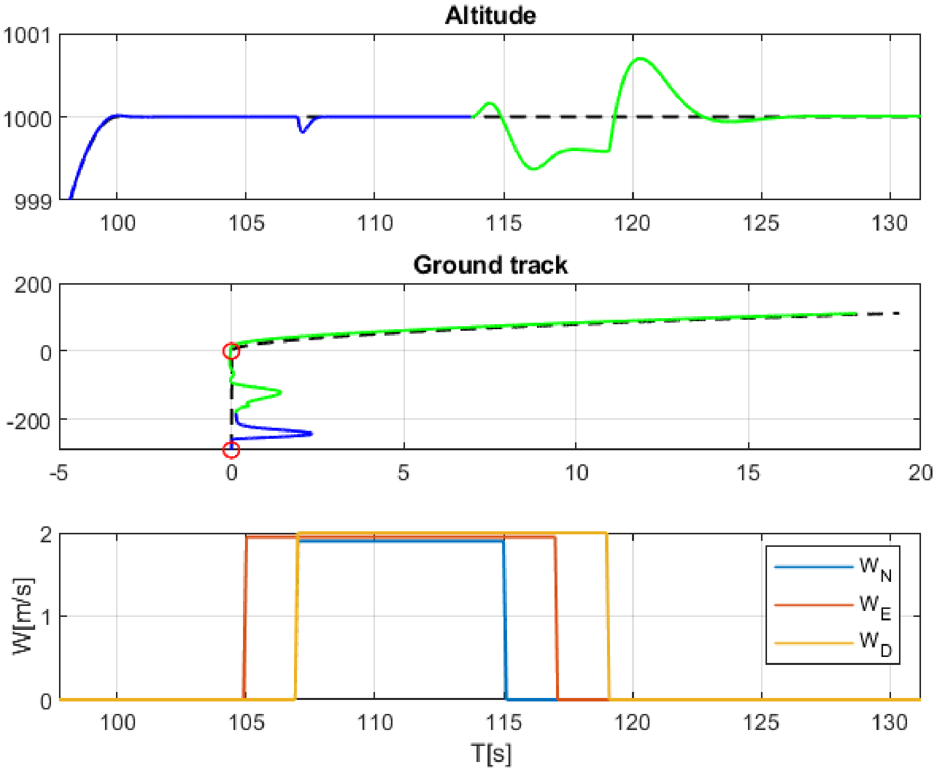


**Figure 7.22:** Switch n.1 under wind disturbances

2. Switch n.2: TR → FW

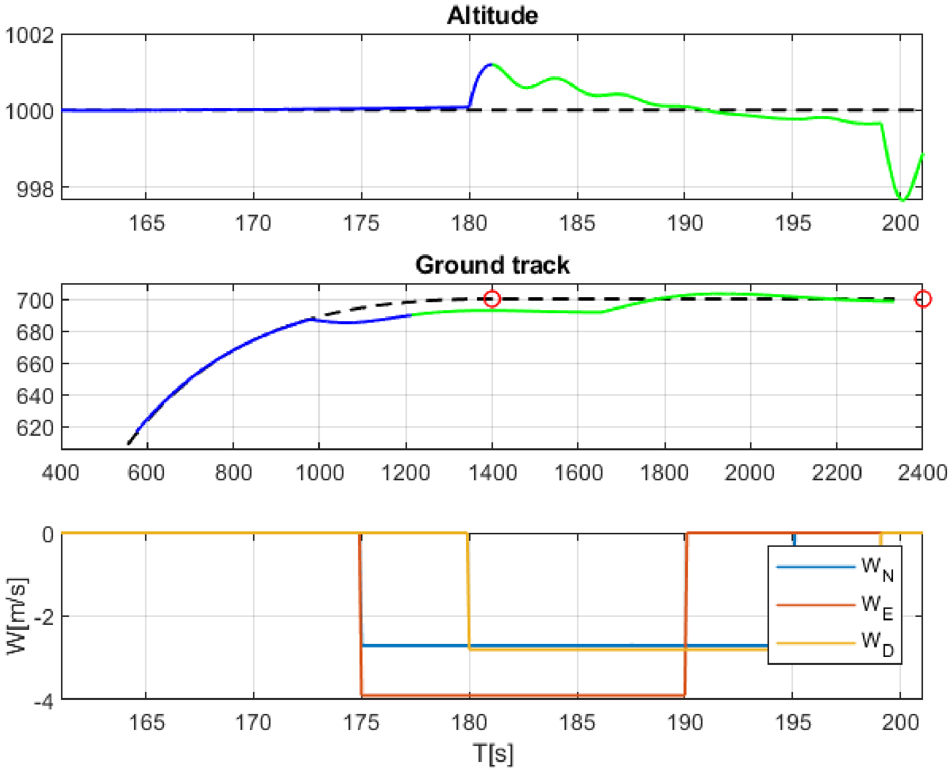


**Figure 7.23:** Switch n.2 under wind disturbances

3. Switch n.3: FW → TR

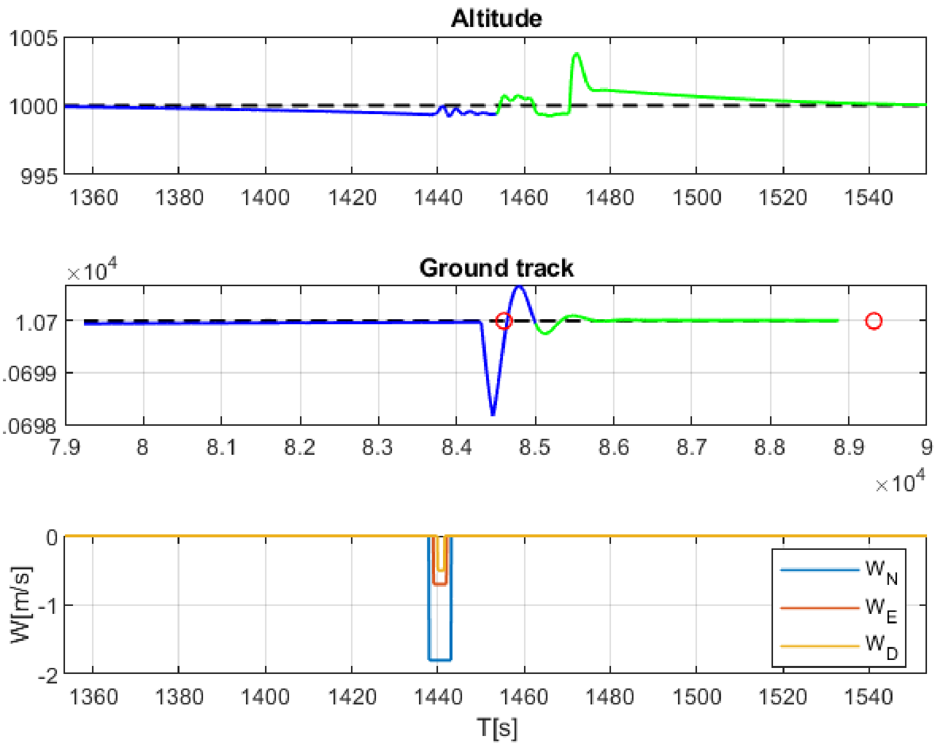


**Figure 7.24:** Switch n.3 under wind disturbances

4. Switch n.4: TR → MR

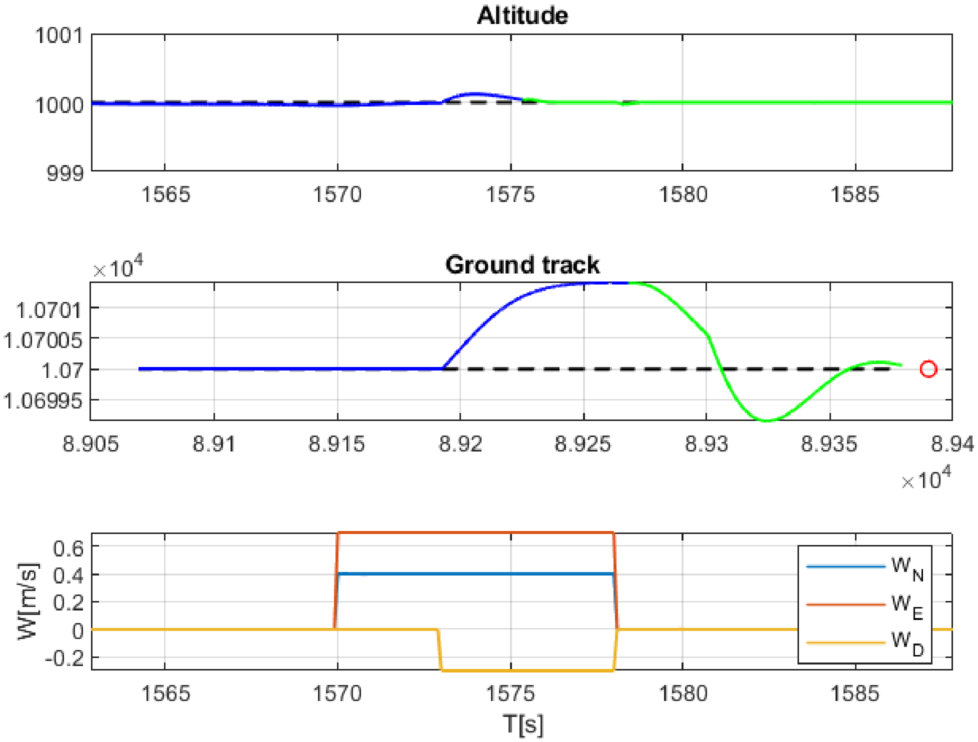


**Figure 7.25:** Switch n.4 under wind disturbances

# Chapter 8

# Realistic Mission simulation

In this final chapter, a practical application of TP technology is explored, highlighting its potential to address challenges faced by mountain communities. Politecnico, in collaboration with UNCEM (the National Union of Mountain Communities), envisions the use of this technology as a solution to issues related to these remote areas. Rather than relying on expensive cable car systems, the proposal suggests the deployment of STOL (Short Takeoff and Landing) vehicles.

Traditional cable car installations can be cost-prohibitive, and their reliability is often compromised by adverse weather conditions prevalent in mountainous regions. In contrast, STOL vehicles offer a promising alternative. They require less infrastructure, making use of available spaces as makeshift runways. This approach not only potentially reduces costs but also provides a more adaptable transportation option.

Mountainous areas typically offer ample space that can be repurposed as suitable airstrips or landing zones for STOL vehicles. Moreover, these vehicles may offer greater flexibility in operation during adverse weather, enhancing transportation reliability in challenging terrains.

Overall, the concept of employing STOL vehicles in mountain communities signifies a cost-effective and weather-resilient solution, utilizing existing open spaces to improve accessibility and transportation options for goods and people in these remote regions.

The selected route for simulation connects two mountain towns, Balme and Ceres, which are situated at different altitudes: Balme at 1400 meters and Ceres at 450 meters. Despite the relatively short ground distance of only 15 kilometers between them, the mission profile involves several phases:

1. **T/O**: The vehicle will initiate a vertical takeoff from the ground at Ceres (450 meters) and ascend to an altitude of 900 meters.
2. **Acceleration**: After takeoff, the aircraft will accelerate until it reaches its cruise speed of 61 meters per second.
3. **Cruise**: The aircraft will follow the river bed for approximately 15 kilometers. Interestingly, in this case, the cruise phase coincides with the ascending phase due to the difference in altitudes between Balme and Ceres.
4. **Deceleration**: Upon reaching the destination area, the aircraft will begin to decelerate until it comes to a complete stop.
5. **LNDG**: The aircraft will perform a vertical landing in Balme, which is at an altitude of 1400 meters.

During phases 2 to 4, the aircraft will maintain a constant climb angle of approximately 3.05 degrees. This mission profile outlines the key steps and altitudes involved in the simulated flight between Balme and Ceres.

The route was planned by marking specific waypoints along the river's path. These waypoints were then translated into NED inertial positions for navigation purposes. To provide a visual representation of the route, it can be plotted on Google Maps as shown in the next page.

Wind disturbances in the simulation were generated based on the prevailing wind conditions in the two cities. Subsequently, the wind rose diagrams for both locations are presented [37],[38]. These wind roses provide a visual representation of the typical wind patterns and frequencies of wind directions over a specific period, helping to simulate realistic wind scenarios for the given areas.

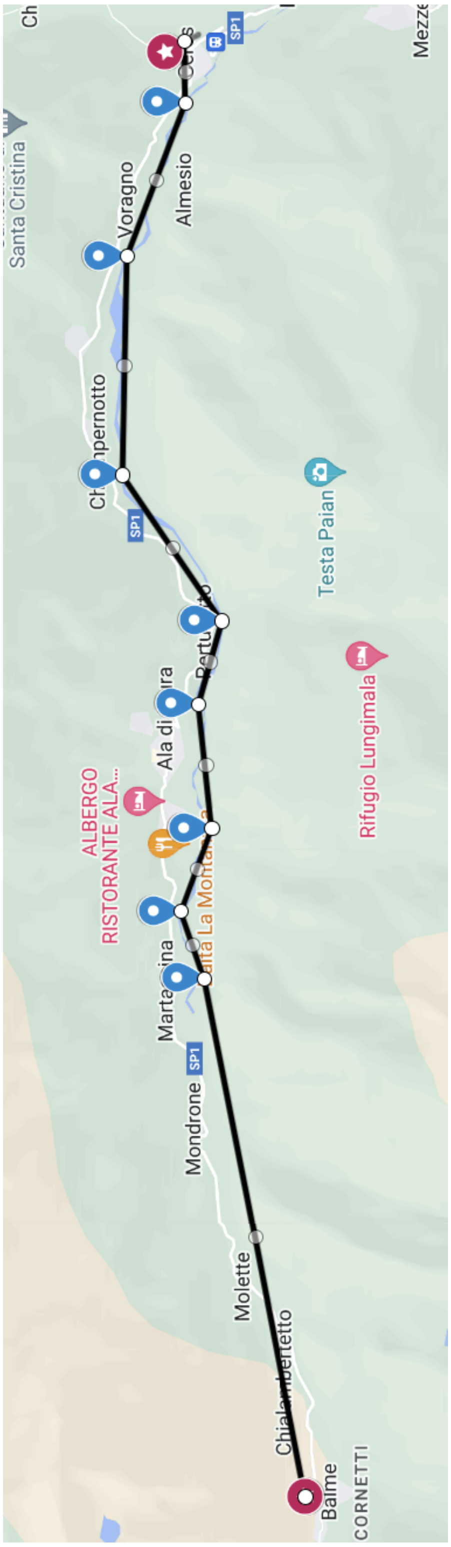


**Figure 8.1:** Ceres-Balme route

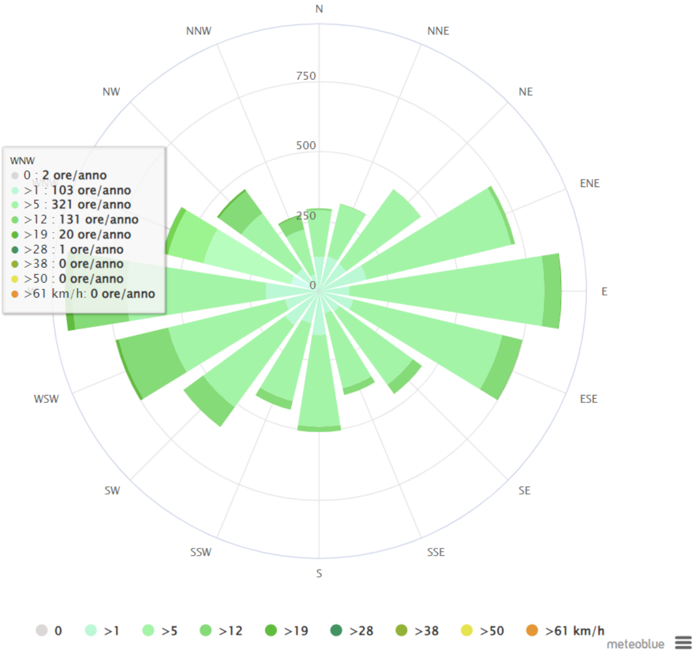


**Figure 8.2:** Wind rose for Ceres

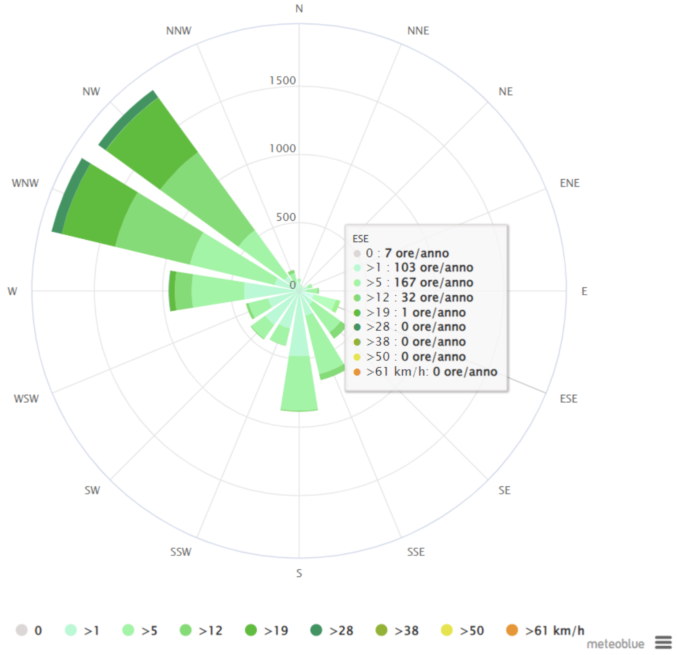


**Figure 8.3:** Wind rose for Balme

The polar wind charts indicate that throughout the entire year, the expected wind speeds fall within the range of 0.6 to 5 meters per second. Consequently, when simulating wind inputs, it is appropriate to consider values within this specified range to accurately represent the typical wind conditions experienced in the area. Wind disturbances expressed in body coordinates are shown below.

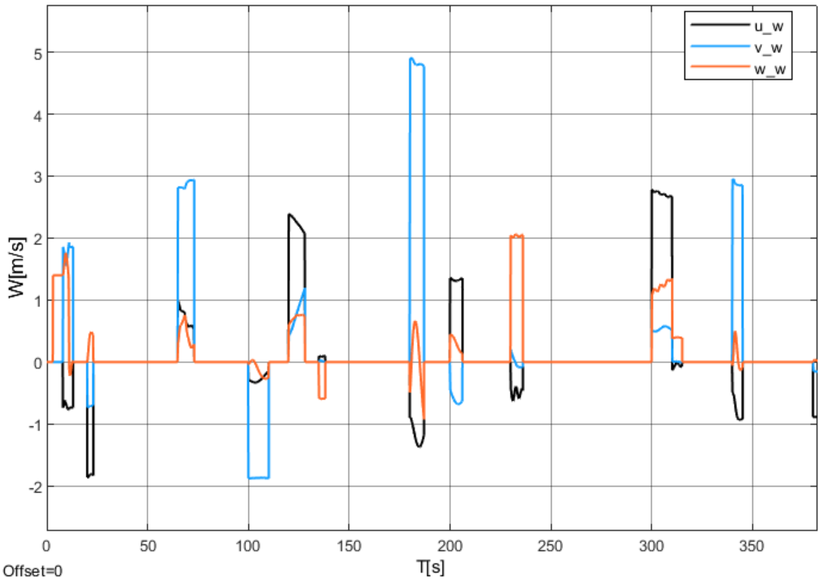


**Figure 8.4:** Wind disturbances

In the following page, the 3D and ground track trajectories of the aircraft are compared with the reference ones. However, to actually see the effect of wind disturbances on such a long period of time, it is necessary to plot also the inertial position error for each axis of the linear motion.

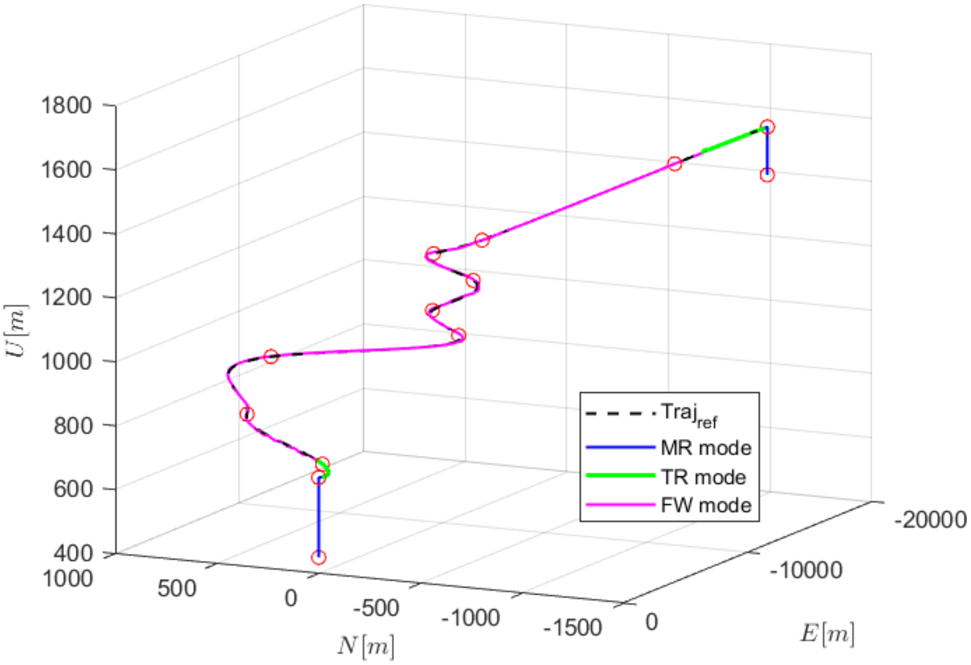


**Figure 8.5:** 3D trajectory for Ceres-Balme route

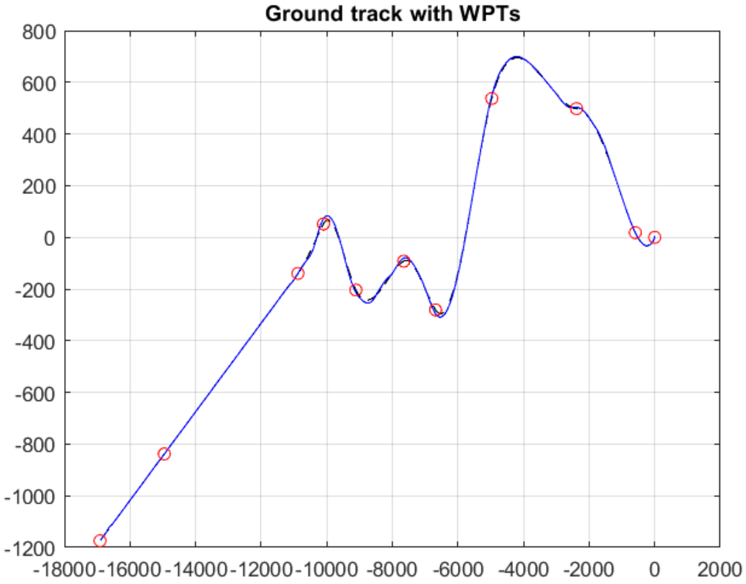


**Figure 8.6:** 2D planar trajectory for Ceres-Balme route

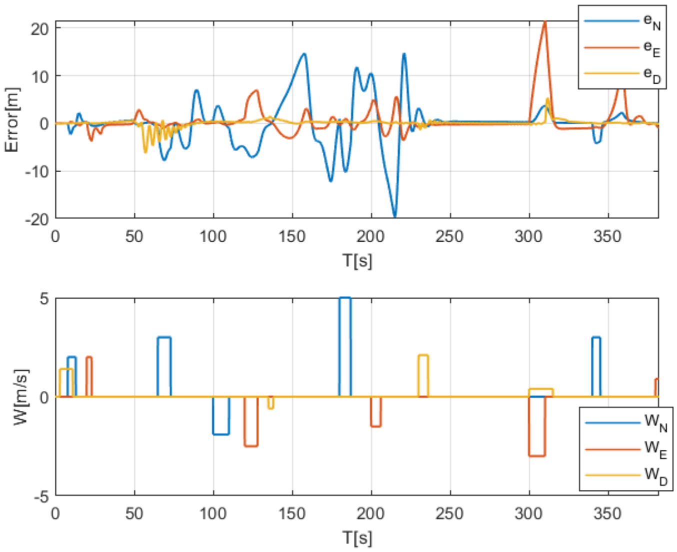


**Figure 8.7:** Wind effect on inertial position error

In practical scenarios, it is entirely normal for wind inputs to introduce positional errors, particularly when disturbances occur during maneuvering. The significant variability in roll angle is an indication of the significant maneuvering load on the vehicle as it attempts to track each waypoint, as shown below.

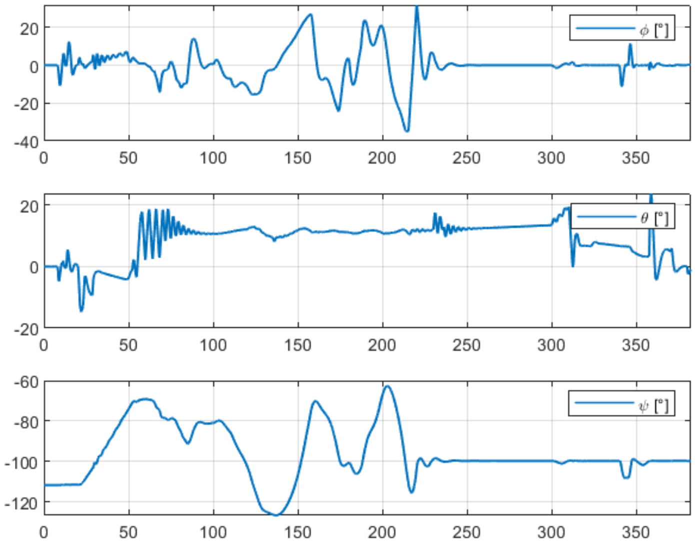


**Figure 8.8:** Euler angles for Ceres-Balme route

Changes in roll and yaw angles primarily occur during turning maneuvers, while rapid pitch adjustments typically take place during transitions between different states. As discussed in the previous chapter, the activation and deactivation of motors during these transitions can introduce discontinuities that the aircraft needs to adapt to. Nevertheless, with careful tuning of control gains and the implementation of rate limiters, it should be possible to ensure passenger comfort during these phases.

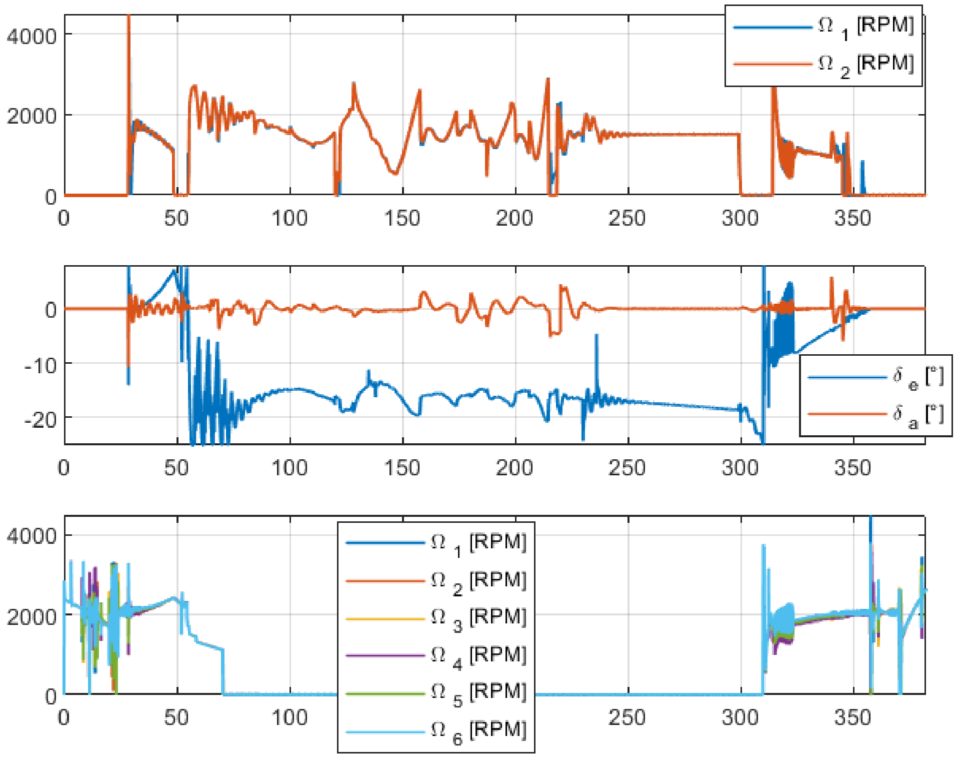


**Figure 8.9:** Control variables

Finally, control variables are shown. As is customary, the lift motors are intentionally deactivated in a gradual manner, employing an exponential decay approach, until they reach a specified threshold. This method is employed to prevent abrupt changes that could lead to sudden unacceptable discontinuities.

In summary, the aircraft demonstrates its capability to successfully accomplish the mission despite encountering step wind disturbances and navigating through demanding turns, which are essential for adhering to the course along the river bed. It's important to note that the primary objective of this work was not to determine the optimal trajectory, but rather to establish that the vehicle can effectively handle challenging routes. Consequently, the vehicle should be well-equipped to adhere to smoother and more efficient trajectories in future missions.

To perform a reverse simulation of the mission, involving a descent, it was hypothesized that the horizontal thrust propellers could generate reverse thrust. This adaptation was implemented to mitigate uncontrolled acceleration resulting from gravity during steep descents. To achieve this, adjustments to the control allocation algorithm were necessary. Once the required horizontal thrust for each propeller was determined, the desired rotational speed was computed as follows:

$$\begin{bmatrix} T_{X_1} \\ T_{X_2} \end{bmatrix} = K_{T_H} \begin{bmatrix} |\Omega_{H_1}|\Omega_{H_1} \\ |\Omega_{H_2}|\Omega_{H_2} \end{bmatrix} \rightarrow \begin{bmatrix} \Omega_{H_1} \\ \Omega_{H_2} \end{bmatrix} = \begin{bmatrix} sign(\Omega_{H_1}) & 0 \\ 0 & sign(\Omega_{H_2}) \end{bmatrix} \sqrt{\frac{1}{K_{T_H}} \begin{bmatrix} T_{X_1} \\ T_{X_2} \end{bmatrix}} \tag{8.1}$$

Results for the return mission are presented below in the following figures:

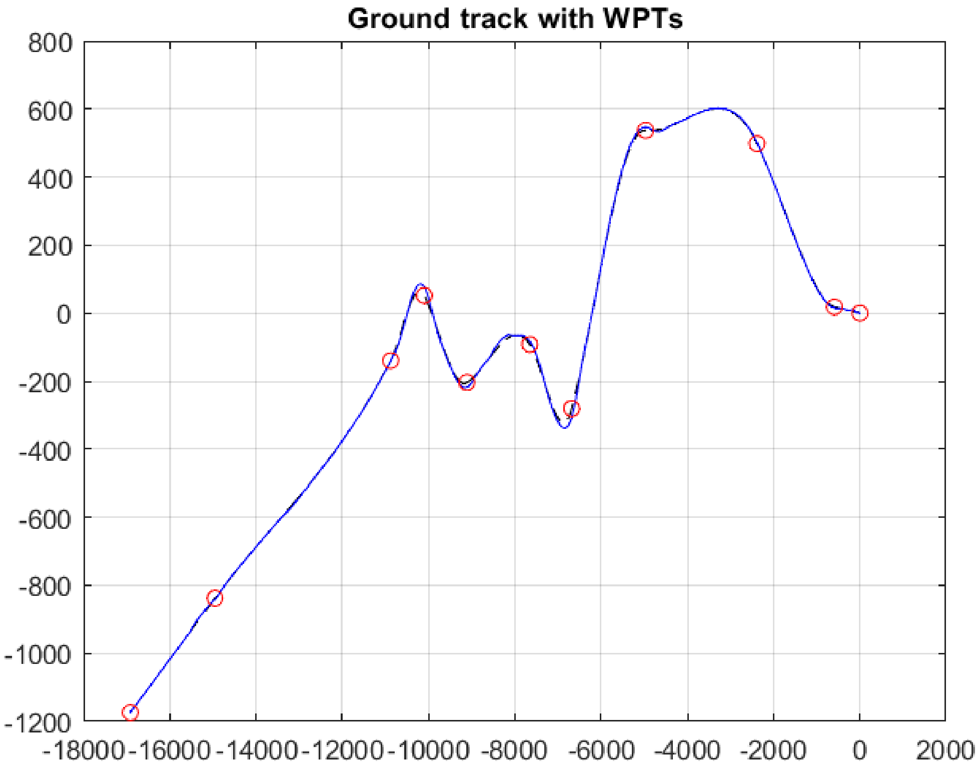


**Figure 8.10:** Ground Track for return mission

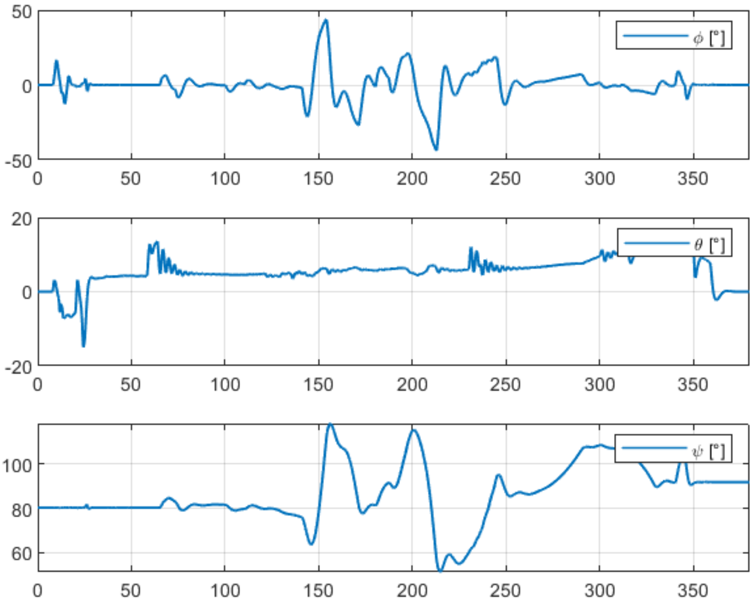


**Figure 8.11:** Euler angles for return mission

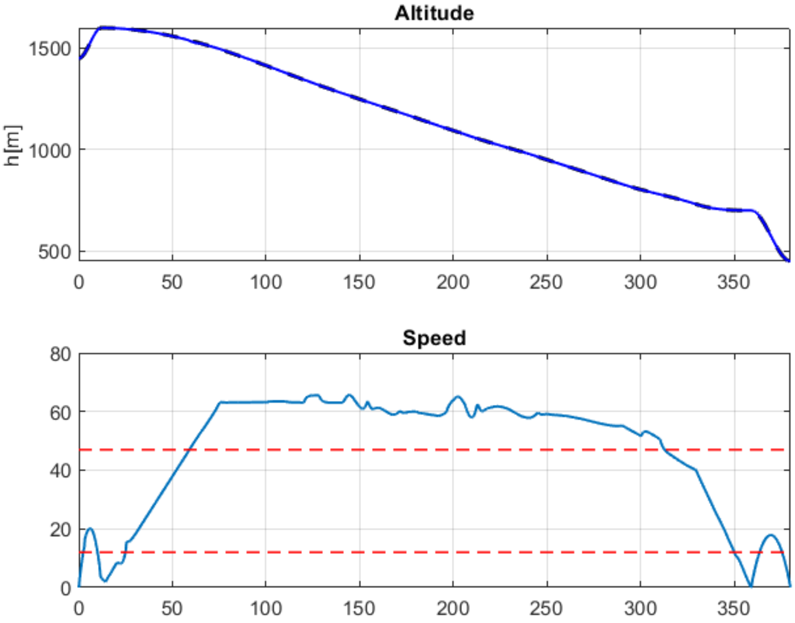


**Figure 8.12:** Altitude and speed for return mission

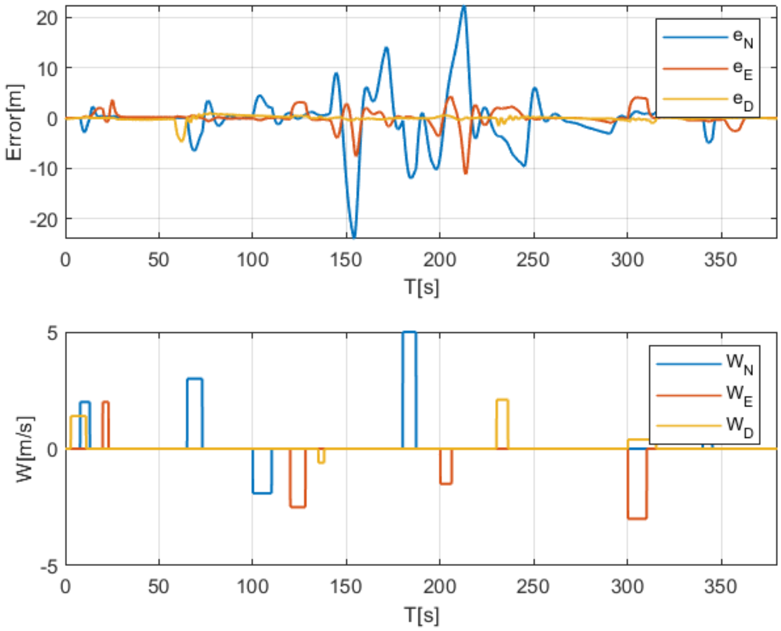


**Figure 8.13:** Position error and wind disturbances for retirn mission

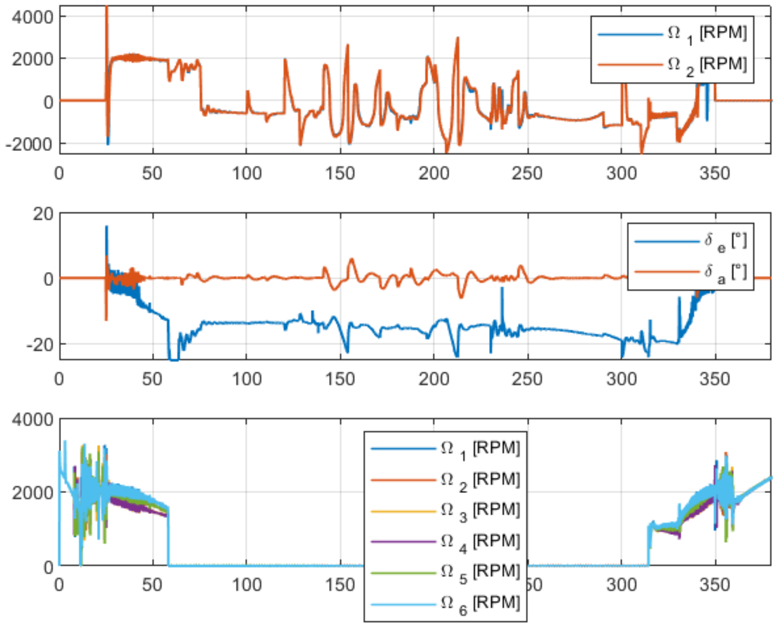


**Figure 8.14:** Control variables for return mission

The plot displaying the rotational speeds of the horizontal motors illustrates that the aircraft requests reverse thrust for most of the descent phase. This is a result of the flight path angle exceeding the aircraft's limiting $\gamma$ for this mission profile.

Lastly, an important aspect is to ensure that the aircraft can energetically sustain a round-trip mission. The consumed energy was calculated for both legs of the journey, resulting in an approximate value of 7.3E+7 J. This energy consumption corresponds to a battery mass of about 230 Kg, which is slightly more than 10% of the total take-off mass.

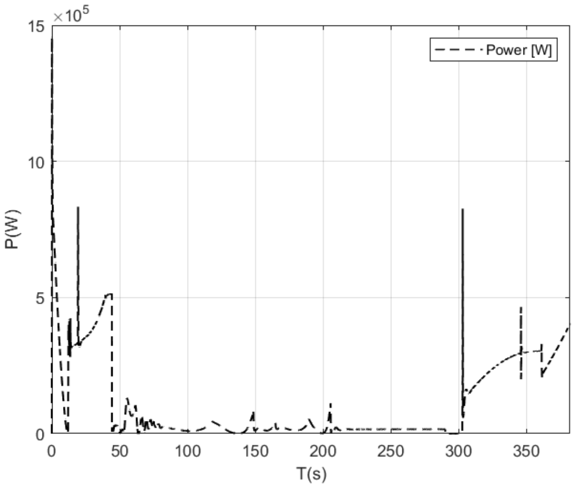


**Figure 8.15:** P(W).vs.t(s) - Balme-Ceres

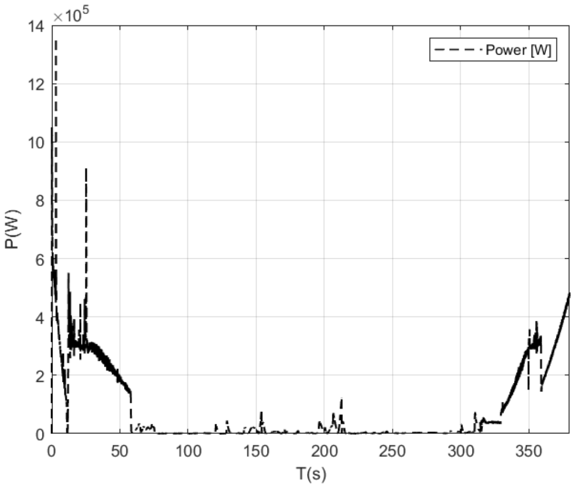


**Figure 8.16:** P(W).vs.t(s) - Ceres-Balme

# Chapter 9

# Conclusions

This work delves into the development of a control system tailored for an autonomous all-electric Fixed-Wing hybrid aircraft. Following a concise introduction to the realm of UAM and innovative hybrid concepts suitable for mid-range missions, Chapter 2 provided an in-depth exposition of the proposed patented TP technology as a viable solution. Subsequent chapters served as a foundation for the mathematical tools required to model the aircraft, presenting nonlinear equations of motion and the various forces and moments acting on the system. The mathematical model was further linearized to evaluate the aircraft's linear stability across different trim points and speeds.

The groundwork of control theory necessary for crafting the control system was introduced, with a particular emphasis on the novel INDI control technique. References to relevant literature that addresses challenges posed by INDI were also provided.

Subsequently, the control system was designed and validated for its tracking performance and resilience in the face of wind disturbances. Three distinct controllers were developed, each dedicated to a specific state of the aircraft. Chapters 8 and 9 witnessed the amalgamation of these controllers, enabling the simulation of two comprehensive missions. The efficacy of the state machine in transitioning seamlessly between controllers was assessed, revealing a satisfactorily smooth transition process.

Future developments will include:

- Enhancing and refining the TR mode controller. It is imperative to explore alternative methods of providing reference attitudes to the inner loop. This exploration aims to account for the significant coupling between velocity and altitude, due to the simultaneous action of lift motors, aerodynamic lift (elevator deflection), and thrust motors. One approach to achieving this is by investigating different methods of inverting equations through the formulation of optimization problems.

- Evaluating the impact of sensor characteristics and potential delays introduced by filtering processes. Given the profound reliance of INDI control on sensor measurements and their derivatives, it is essential to conduct simultaneous simulations of navigation and control. This simultaneous assessment can help determine whether complementary filtering techniques can effectively mitigate the sensitivity to time delays.
- Incorporating a PCH mechanism into the controller. This addition aims to prevent the controller from reaching saturation limits, ensuring smoother and more stable performance.
- Developing a model for the energy recovery system and assess aircraft performance for deceleration and descent phases properly and retuning gains if needed.

# Bibliography

[1] Raul Batlle. «The potential of deploying Urban Air Mobility services as an integrated transport system: a London area case study». PhD thesis. Aug. 2020. DOI: 10.13140/RG.2.2.29467.80163 (cit. on p. 1).

[2] *Continuing Urbanisation.* European Commission's Knowledge for Policy. Year of Access: 2023. URL: https://knowledge4policy.ec.europa.eu/continuing-urbanisation_en (cit. on p. 1).

[3] PWC Strategy&. *The path towards a mobility in the third dimension: How to create a National ecosystem for Advanced Air Mobility.* Insight Report. PWC Strategy&, 2021. URL: https://www.strategyand.pwc.com/it/en/assets/pdf/S&-path-towards-a-mobility-in-the-third-dimension.pdf (cit. on p. 2).

[4] *Urban Air Mobility (UAM).* European Union Aviation Safety Agency. Year of Access: 2023. URL: https://www.easa.europa.eu/en/domains/urban-air-mobility-uam#group-easa-related-content (cit. on p. 2).

[5] Plötner, Preis, Straubinger, and Shamiyeh. *Putting Urban Air Mobility into perspective.* Tech. rep. Bauhaus Luftfahrt, 2023 (cit. on p. 2).

[6] A. Lerro, L. Nanu, and P. Gili. «ThrustPod: a novel solution for vertical take-off and landing systems». In: *Proceedings of the DICUAM 2022 Conference* (Mar. 2022) (cit. on pp. 4, 5).

[7] Mao S, Gong Z., Ye Z., Wang Z., Guo T., and Zhang C. «Linear-Active-Disturbance-Rejection-Based Vertical Takeoff and Acceleration Strategy with Simplified Vehicle Operations for Electric Vertical Takeoff and Landing Vehicles». In: *Mathematics* (2022). DOI: 10.3390/math10183333 (cit. on pp. 10, 131).

[8] M. Moussid, A. Sayouti, and H. Medromi. «Dynamic Modeling and Control of HexaRotor using Linear and Nonlinear Methods». In: *International Journal of Applied Infomation Systems(IJAIS)* 9.5 (Aug. 2015). ISSN: 2249-0868 (cit. on pp. 20, 34).

[9] S. K. Phang, S. Z. Ahmed, and M. R. A. Hamid. «Design, Dynamics Modelling and Control of a H-Shape Multi-rotor System for Indoor Navigation». In: *Proceedings of the 1st International Conference on Unmanned Vehicle Systems(UVS)* (Feb. 2019) (cit. on p. 23).

[10] Maccotta M. *Multirotor UAVs for fugitive emissions detection: sizing, modeling and control system desgin*. 2018 (cit. on pp. 23, 25).

[11] L. Nanu. «ThrustPod Full-Scale eVTOL A/C propeller analysis». In: (June 2023) (cit. on p. 30).

[12] Brian L. Stevens, Frank L. Lewis, and Eric N. Johnson. *Aircraft Control and Simulation*. Wiley, 2006 (cit. on pp. 32, 93).

[13] *EMRAX 348 Datasheet*. EMRAX d.o.o. URL: `https://emrax.com/wp-content/uploads/2022/11/EMRAX_348_datasheet_A00.pdf` (cit. on p. 35).

[14] A. De Marco, E. L. Duke, and J. S. Berndt. «A General Solution to the Aircraft Trim Propblem». In: *AIAA Modeling and Simulation Technologies Conference and Exhibit* (Aug. 2007) (cit. on p. 41).

[15] The MathWorks, Inc. *MATLAB Documentation: fmincon*. URL: `https://it.mathworks.com/help/optim/ug/fmincon.html` (cit. on p. 41).

[16] D. S. Naidu and A. J. Calise. «Singular Perturbations and Time Scales in Guidance and Control of Aerospace Systems: A Survey». In: *Journal of Guidance, Control, and Dynamics* 24.6 (Nov. 2001) (cit. on p. 58).

[17] R. C. van't Veld. *Incremental Nonlinear Dynamic Inversion Flight Control: Stability and Robustness Analysis and Improvements*. 2016 (cit. on pp. 58, 59, 67).

[18] Schumacher C. and Khargonekar P. P. «Stability Analysis of a Missile Control System with a Dynamic Inversion Controller». In: *Journal of Guidance, Control, and Dynamics* 21.3 (1998), pp. 508–515. DOI: `10.2514/2.4266` (cit. on p. 58).

[19] Schumacher C., Khargonekar P. P., and McCamroch. «Stability Analysis of Dynamic Inversion Controllers Using Time-Scale Separation». In: *AIAA-98-4322* (1998) (cit. on p. 58).

[20] The MathWorks, Inc. *MATLAB Documentation: PID Controller*. URL: `https://it.mathworks.com/help/simulink/slref/pidcontroller.html?searchHighlight=pid%20simulink&s_tid=srchtitle_support_results_4_pid%20simulink` (cit. on p. 60).

[21] F. Sabatino. *Quadrotor Control: modeling ninlinear control design and simulation*. 2015 (cit. on p. 63).

[22] Slotine and Li. *Applied Nonlinear Control.* Pearson College Div., 1990 (cit. on p. 63).

[23] R. da Costa, Q. Chu, and J. Mulder. «Reentry Flight Controller Design Using Nonlinear Dynamic Inversion». In: *Journal of Spacecraft and Rockets* 40.1 (2003), pp. 64–71. DOI: `10.2514/2.3916` (cit. on p. 63).

[24] Smeur, Chu, and de Croon. «Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Air Vehicles». In: *Journal of Guidance, Control, and Dynamics* 39.3 (Mar. 2016). DOI: `10.2514/1.G001490` (cit. on pp. 64, 66, 76).

[25] Year of access: 2023. URL: `http://www.aerostudents.com/courses/advanced-flight-control/nonlinearDynamicInversion.pdf` (cit. on p. 64).

[26] Z. Liu, J. Guo, M. Li, S. Tang, and X. Wang. «VTOL UAV Transition Maneuver Using Incremental Nonlinear Dynamic Inversion». In: *International Journal of Aerospace Engineering* 2018 (). DOI: `10.1155/2018/6315856` (cit. on pp. 66, 81, 108, 122).

[27] T. M. L. De Ponti. *Incremental Nonlinear Dynamic Inversion Controller for a Variable Skew Quad Plane.* 2022 (cit. on p. 66).

[28] O. Pfeifle and W. Fichter. «Minimum Power Control Allocation for Incremental Control of Over-Actuated Transition Aircraft». In: *Journal of Guidance, Control, and Dynamics* (Nov. 2022) (cit. on p. 66).

[29] J. Chang, Z. Guo, R. De Breuker, and X. Wang. «Predictor-based Adaptive Incremental Nonlinear Dynamic Inversion for Fault-Tolerant Flight Control». In: *IFAC PapersOnLine* 55-6 (2022), pp. 730–736 (cit. on p. 66).

[30] P. Bhardwaj, S. Raab, J. Zhang, and F. Holzapfel. «Integrated Reference Model for a Tilt-rotor Vertical Take-off and Landing Transition UAV». In: *Applied Aerodynamics Conference* (June 2018) (cit. on p. 67).

[31] S. Raab, J. Zhang, P. Bhardwaj, and F. Holzapfel. «Consideration on Control Effector Dynamics and Saturations in an Extended INDI approach». In: *AIAA Aviation 2019 forum* (June 2019) (cit. on p. 67).

[32] S. Sieberling, Q. P. Chu, and J. A. Mulder. «Robust Flight Control Using Incremental Nonlinear Dynamic Inversion and Angular Acceleration Prediction». In: *Journal of Guidance, Control, and Dynamics* 33.6 (Nov. 2010). DOI: `10.2514/1.49978` (cit. on p. 69).

[33] R. Steffensen, A. Steinert, Z. Mbikayi, S. Raab, J. Angelov, and F. Holzapfel. «Filter and sensor delay synchronization in incremental flight control laws». In: *Aerospace Systems* 6 (2023), pp. 285–304. DOI: `10.1007/s42401-022-00186-2` (cit. on p. 70).

[34] The MathWorks, Inc. *MATLAB Documentation: Polynomial Trajectory.* URL: `https://it.mathworks.com/help/robotics/ref/polynomialtrajectory.html` (cit. on p. 72).

[35] Zhou L., Yang J., Strampe T., and Klingauf U. «Incremental nonlinear dynamic inversion based path-following control for a hybrid quad-plane unmanned aerial vehicle». In: *Int J Robust Nonlinear Control* (2022), pp. 1–24. DOI: `10.1002/rnc.6503` (cit. on pp. 87, 88, 102).

[36] *www.epectec.com.* URL: `https://www.epectec.com/batteries/cell-comparison.html` (cit. on p. 151).

[37] *www.meteoblu.com.* Accessed in 2023. URL: `https://www.meteoblue.com/it/tempo/historyclimate/climatemodelled/ceres_italia_3179099` (cit. on p. 158).

[38] *www.meteoblu.com.* Accessed in 2023. URL: `https://www.meteoblue.com/it/tempo/historyclimate/climatemodelled/balme_italia_3182446` (cit. on p. 158).