



TÉCNICO
LISBOA



Longitudinal Control of a Tail-Sitter UAV

Raquel Alexandra Maia Rocha

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisor(s): Prof. Alexandra Bento Moutinho
Prof. José Raul Carreira Azinheira

Examination Committee

Chairperson: Prof. Fernando Lau
Supervisor: Prof. José Raul Azinheira
Member of the Committee: Prof. Rita Cunha

October 2022

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

The completion and substantiation of this master's dissertation symbolizes the end of a long trek as a student of Academia da Força Aérea and Instituto Superior Técnico. However, it wouldn't be possible without the direct and indirect support of all those who accompanied me along this journey. To all of them, I deliver my most sincere thanks, and in particular to the following:

I am extremely grateful to Academia da Força Aérea, the institution I feel the greatest pride being a part of, for all the teachings, opportunities and life experiences it has provided me with for the last six years. All these livelihoods, from the best to the worst instances, have contributed in great measure to the person I am today.

I express my deepest appreciation to Professor José Azinheira and Professor Alexandra Moutinho of Instituto Superior Técnico, for their help and guidance in all phases of this project. I am deeply thankful, especially for all their availability, patience and excellent pedagogy that have allowed me to deepen my knowledge in this matter and for the freedom to trace my own path.

I am deeply indebted to Alexandra Athayde, doctoral student in Instituto Superior Técnico, for his important attendance and fundamental help in beginning this work. His advice and knowledge were instrumental to the writing of this work.

This endeavour would not have been possible without the funding of FCT – Fundação para a Ciência e Tecnologia, I.P., through IDMEC under project Eye in the Sky (PCIF/SSI/0103/2018), and under LAETA (project UIDB/50022/2020).

Many thanks to my speciality comrades António Oliveira and Tiago Nunes for the unwavering support, friendship and companionship along these six years of study and for sharing with me far beyond our courses. Special thanks to my comrade and friend Carolina Baptista, for your sensitivity and determination, which were always an inspiration to me. To all the conversations and moments which have allowed us both to grow and prosper.

I am grateful for my comrade and specialty sister Tirsa Arriscado who accompanied and supported me in all moments and stages of our academic, professional, sport and personal life. From joy to despair. From smiles to tears. For being by my side and lending me your hand whenever I needed it.

I extend my sincere thanks to Tiago, one of the most incredible people in my life, for motivating me and always bringing out the best in me, for all that you have taught me and all the adventures. You are an inspiration and thank you for all that words cannot express.

I would be remiss in not mentioning my family who have full heartedly lived this adventure with me, who have been my constant support and resolute incitement since my very first day. Thank you for helping me trace my path, for the freedom and responsibility you delivered upon me and for advising me even when I found myself buried in doubt and indecision. And a particular thanks, for all the help and support, to my sibling Ideal.

Lastly, I deliver a very special thanks to Sofia, my girlfriend, for all the love, kindness and understanding, from up close or from afar. Thank you for being my pillar and for supporting me in all my struggles and conquests, for believing in me and for giving me the strength to follow my dreams and my values.

To all of you, thank you!

Resumo

A presente dissertação detalha o modelo de um veículo aéreo não tripulado tail-sitter, e os desafios de projetar o seu controlador de voo. Sendo uma classe particular de aeronaves de asa fixa, os tail-sitters são capazes de realizar tanto voo aerodinâmico como manobras estacionárias, descolagem e aterragem vertical, bem como as transições entre as duas condições de operação. Neste trabalho, primeiramente, é apresentado o veículo tail-sitter escolhido para controlo e é detalhado o modelo matemático não-linear que descreve a dinâmica da aeronave, que é então implementado num simulador realista. Segundo, o modelo não-linear é linearizado e analisado para que possam ser utilizados métodos de controlo linear. Como as dinâmicas lateral e longitudinal da aeronave podem ser desacopladas, neste trabalho considera-se apenas o modelo longitudinal como objetivo de controlo. Utilizando técnicas de controlo clássico, nomeadamente controladores proporcional-integral-derivativo e reguladores lineares quadráticos, a estabilização em torno dos modelos linearizados, o seguimento da trajectória durante as transições e também as manobras de descolagem e aterragem autónomas são possíveis através de uma lógica de controlo supervisor. Este controlador supervisor é responsável por seleccionar o modo de operação do controlador dependendo da fase de voo real e da manobra desejada. Por fim, todo o controlador é testado através da simulação de duas missões diferentes: a primeira realiza apenas manobras de voo vertical, e a segunda inclui voo vertical e aerodinâmico. Os resultados das simulações são apresentados e o controlador avaliado.

Palavras-chave: *Tail-sitter*, Modelo Não-linear, Controlo Linear, Regulador Quadrático Linear, Transições Automáticas, Descolagem e Aterragem Vertical

Abstract

The present dissertation details a tail-sitter unmanned aerial vehicle model and the challenges of designing its flight controller. As a particular type of fixed-wing aircraft, tail-sitters are able to perform both aerodynamic flight and stationary maneuvers, vertical take-off and landing, as well as transitions between the two operating conditions. In this work, first, the tail-sitter vehicle chosen for control is presented, the mathematical nonlinear model describing the aircraft dynamics is detailed and a real-time simulator is implemented. Secondly, the nonlinear model is linearised and analysed so that linear control methods can be used. Since the aircraft's lateral and longitudinal dynamics can be decoupled, only the longitudinal model is considered for control purposes. Using classical control techniques, namely proportional–integral–derivative and linear quadratic regulator controllers, the stabilization about the linearized operating points, the trajectory tracking during transitions and also the autonomous take-off and landing maneuvers are possible by means of a supervisory control logic. This supervisory controller is responsible for selecting the controller operating mode depending on the actual flight stage and desired maneuver. Finally, the whole controller is tested by simulating two different missions: the first one performing only vertical flight maneuvers, and the second one including both vertical and aerodynamic flight. The simulations' results are presented and the controller is evaluated.

Keywords: Tail-sitter, Nonlinear Model, Linear Control, Linear Quadratic Regulator, Autonomous Transitions, Vertical Take-Off and Landing

Contents

| | |
|--------------------------------------------------|----------|
| Declaration | iii |
| Acknowledgments | v |
| Resumo | vii |
| Abstract | ix |
| List of Tables | xiii |
| List of Figures | xv |
| Nomenclature | xix |
| 1 Introduction | 1 |
| 1.1 Motivation | 1 |
| 1.2 Topic Overview | 2 |
| 1.2.1 Tail-sitter | 3 |
| 1.2.2 Tail-sitter Control Challenges | 4 |
| 1.3 Platform Presentation | 6 |
| 1.4 Objectives | 7 |
| 1.5 Thesis Outline | 7 |
| 2 Nonlinear Model | 8 |
| 2.1 Frames and Conventions | 8 |
| 2.1.1 Reference Frames | 8 |
| 2.1.2 Attitude Representation | 9 |
| 2.2 Actuators | 10 |
| 2.3 Aircraft Equations of Motion | 11 |
| 2.4 Aerodynamic Variables | 13 |
| 2.5 Tail-sitter Model | 14 |
| 2.5.1 Gravity | 14 |
| 2.5.2 Thrusters | 14 |
| 2.5.3 Wing Aerodynamics | 20 |
| 2.5.4 Structural Segments Aerodynamics | 24 |
| 2.5.5 Ground Contact Dynamics | 25 |
| 2.6 Aircraft Simulator | 26 |

| | | |
|----------|--------------------------------------------|-----------|
| 3 | Longitudinal Model Characterization | 28 |
| 3.1 | Flight Envelope | 29 |
| 3.2 | Aircraft Trimming | 32 |
| 3.3 | Linearization | 38 |
| 3.4 | Longitudinal Analysis | 39 |
| 3.4.1 | Stability | 39 |
| 3.4.2 | Controllability | 44 |
| 4 | Flight Control Design | 45 |
| 4.1 | LQR Theory | 45 |
| 4.2 | Control Strategy | 48 |
| 4.2.1 | Controller Inputs | 49 |
| 4.2.2 | Controller Structure | 50 |
| 5 | Simulation Flight Results | 58 |
| 5.1 | Response Analysis | 58 |
| 5.1.1 | Hover | 59 |
| 5.1.2 | Level | 63 |
| 5.1.3 | Transitions | 67 |
| 5.1.4 | Take-Off and Landing | 71 |
| 5.2 | Simulation Flight Mission | 73 |
| 5.2.1 | Mission 1 | 73 |
| 5.2.2 | Mission 2 | 75 |
| 6 | Conclusions | 77 |
| 6.1 | Summary | 77 |
| 6.2 | Observations | 77 |
| 6.3 | Future Work | 78 |
| | Bibliography | 79 |
| A | X-Vert Parameters | 82 |
| B | Quaternions | 85 |

List of Tables

| | | |
|-----|------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1 | Segments Parameters | 21 |
| 2.2 | Rod Segments' Parameters | 25 |
| 3.1 | Hover flight initial guess and trim | 37 |
| 3.2 | Transition flight initial guess and trim | 37 |
| 3.3 | Level flight (at 9 m/s) initial guess and trim | 37 |
| 3.4 | Hover Flight Longitudinal Eigenvalues | 40 |
| 3.5 | Hover Flight Longitudinal Eigenvectors | 41 |
| 3.6 | Transition Flight Longitudinal Eigenvalues | 42 |
| 3.7 | Level Flight Longitudinal Eigenvalues | 43 |
| 4.1 | Maximum state and input deviations from equilibrium | 55 |
| 4.2 | LQR complete gain matrix K for each operating mode | 56 |
| A.1 | Aircraft Body Parameters, values from [16]. | 82 |
| A.2 | Thruster Parameters, values from [16]. | 82 |
| A.3 | Electric Motor Parameters, values from [32]. | 83 |
| A.4 | Wing Geometric Parameters, values from [16]. | 83 |
| A.5 | Aerodynamic Parameters, values from [16]. | 83 |
| A.6 | Aerodynamics coefficients, values from [11]. | 84 |
| A.7 | Coordinates of Reference Points for Contact Force Calculations, given in the geometric reference frame G, in meters. | 84 |

List of Figures

| | | |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1.1 | Representation of the HAB + UAV solution for wildfire operations, from [1] | 2 |
| 1.2 | A tail-sitter take-off and landing transitions to and from level flight mode | 3 |
| 1.3 | Convair XFY-1 aircraft [8]. | 5 |
| 1.4 | E-flite X-Vert VTOL Horizon Hobby [17] | 6 |
| 2.1 | Coordinate frames, geometric dimensions and the four actuators [16] | 9 |
| 2.2 | Airspeed V_a , angle of attack α and angle of side slip β | 13 |
| 2.3 | Inflow velocity and angular speed on the left propeller, adapted from [22]. | 16 |
| 2.4 | Thrust and torque coefficients as a function of the advance ratio J , from [23]. | 17 |
| 2.5 | Slipstream variables | 18 |
| 2.6 | Induced velocity on a propeller, adapted from [21]. | 18 |
| 2.7 | Wing Segments | 20 |
| 2.8 | Aerodynamic center location and area of wing segments | 21 |
| 2.9 | Longitudinal aerodynamics forces and moments, adapted from [18]. | 22 |
| 2.10 | Angle of attack influence on lift coefficient (in black) and on drag coefficient (in red). Both coefficients are represented with elevons at minimal, zero and maximal deflection denoted as shown in the legend. Adapted from [16]. | 23 |
| 2.11 | Angle of attack influence on pitching moment coefficient. All coefficients are represented with elevons at minimal, zero and maximal deflection denoted as shown in the legend. Adapted from [16]. | 23 |
| 2.12 | Rod segments in a wind field $V_{a,j}$ producing a drag force D_j , adapted from [16] | 25 |
| 2.13 | Ground Contact Forces and the four selected point in red. Adapted from [16]. | 26 |
| 2.14 | Aircraft Simulator | 27 |
| 3.1 | Minimum Vertical Flight Mission | 28 |
| 3.2 | Minimum Horizontal Flight Mission | 29 |
| 3.3 | Vertical Flight Forces Diagrams | 30 |
| 3.4 | Horizontal Flight Forces Diagrams | 31 |

| | | |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 3.5 | The first graphic shows the vertical flight envelope $u - \theta$ (the trim values are represented by the black dots and the flight envelope limits by the grey dashed lines. The second and third graphics show the actuators δ_{E_0} and δ_{T_0} trim values. When the actuators trim value (represented by the black dots) overcome the actuators' limits (represented by the red dashed lines), the flight envelope limits are defined. | 32 |
| 3.6 | The first graphic shows the horizontal flight envelope $u - \theta$ (the trim values are represented by the black dots and the flight envelope limits by the grey dashed lines. The second and third graphics show the actuators δ_{E_0} and δ_{T_0} trim values. When the actuators trim value (represented by the black dots) overcome the actuators' limits (represented by the red dashed lines), the flight envelope limits are defined. | 33 |
| 3.7 | Trimming Algorithm | 36 |
| 3.8 | Vertical Eigenvalues Evolution Map. The first plot includes all the vertical flight envelope eigenvalues and the next pictures are zoomed over the grey areas of its previous plot. . . | 41 |
| 3.9 | Horizontal Eigenvalues Evolution Map: from $u_0 = 9$ m/s (in black) to $u_0 = 12$ m/s (in red). | 44 |
| 4.1 | Feedback with Variables in Servomechanism Block Diagram | 47 |
| 4.2 | Windup illustration: offset by errors in the other direction | 48 |
| 4.3 | Integral Windup Compensation Block Diagram | 48 |
| 4.4 | Controller input types | 49 |
| 4.5 | Controller inputs explanation | 50 |
| 4.6 | Controller Structure | 51 |
| 4.7 | Mode Selector Flowchart | 51 |
| 4.8 | References Generator Flowchart | 52 |
| 4.9 | Transition and Back Transition References | 54 |
| 4.10 | Altitude PID Controller Diagram | 54 |
| 4.11 | LQR Controller Diagram where pink boxes indicate variable flight conditions and corresponding gain matrices | 57 |
| 5.1 | Step in handle 1 during Hover (simulation visualization) | 59 |
| 5.2 | Step in handle 1 during Hover | 60 |
| 5.3 | Step in handle 2 during Hover (simulation visualization) | 61 |
| 5.4 | Step in handle 2 during Hover | 62 |
| 5.5 | Step in handle 1 during Level flight (simulation visualization) | 63 |
| 5.6 | Step in handle 1 during Level flight | 64 |
| 5.7 | Step in handle 2 during Level flight (simulation visualization) | 65 |
| 5.8 | Step in handle 2 during Level flight | 66 |
| 5.9 | Transition from Hover to Level flight (simulation visualization) | 67 |
| 5.10 | Transition from Hover to Level flight | 68 |
| 5.11 | Back transition from Level to Hover flight (simulation visualization) | 69 |
| 5.12 | Back Transition from Level to Hover flight | 70 |

| | |
|-------------------------------------------------------------|----|
| 5.13 Take-Off and Land (simulation visualization) | 71 |
| 5.14 Take-Off and Land | 72 |
| 5.15 Mission 1 (simulation visualization) | 73 |
| 5.16 Mission 1 | 74 |
| 5.17 Mission 2 (simulation visualization) | 75 |
| 5.18 Mission 2 | 76 |
| A.1 Flying Wing Dimensions | 84 |

Nomenclature

Acronyms

| | |
|--------------|----------------------------------|
| BEC | Battery Elimination Circuit |
| CA | Aerodynamic Center |
| CG | Center of Gravity |
| ESC | Electronic Speed Controller |
| GCS | Ground Control Station |
| GPS | Global Positioning System |
| HAB | High Altitude Balloon |
| IMU | Inertial Measurement Units |
| LIDAR | Light Detection and Ranging |
| LQR | Linear–Quadratic Regulator |
| MAV | Micro-Air-Vehicle |
| NED | North-East-Down |
| PID | Proportional-Integral-Derivative |
| PMB | Power Management Module |
| PWM | Pulse Width Modulation |
| RPV | Remotely Piloted Vehicle |
| UAV | Unmanned Aerial Vehicle |
| VTOL | Vertical Take-Off and Landing |

Greek symbols

| | |
|----------|---------------------|
| α | Angle of attack. |
| β | Angle of side-slip. |

| | |
|------------|-------------------------------|
| γ | Climb angle. |
| δ_E | Elevator deflection. |
| δ_A | Aileron deflection. |
| δ_R | Rudder deflection. |
| δ_T | Throttle actuation. |
| δ | Elevon deflection. |
| θ | Pitch Angle. |
| Λ | Wing sweep angle. |
| μ | Air dynamic viscosity. |
| ρ | Air density. |
| τ | Thruster actuation. |
| Φ | Euler angles vector. |
| ϕ | Roll Angle. |
| ψ | Yaw Angle. |
| Ω | Angular velocity. |
| ω | Euler angles velocity vector. |

Roman symbols

| | |
|-----------|----------------------------------|
| \bar{c} | Mean aerodynamic chord. |
| A | Dynamic matrix. |
| B | Input matrix. |
| D | State-space disturbances vector. |
| F | Forces vector. |
| F | Thrust torque. |
| I | Inertia matrix. |
| K | Gain matrix. |
| M | Moments vector. |
| p | Position vector. |
| Q | State weighting matrix. |

| | |
|------------------|---------------------------------|
| \mathbf{q} | Quaternion vector. |
| \mathbf{R} | Input weighting matrix. |
| \mathbf{U} | State-space input vector. |
| \mathbf{v} | Velocity vector. |
| \mathbf{X} | State-space state vector. |
| \mathbf{Y} | State-space output vector. |
| \mathbf{R}_a^b | Rotation matrix from a to b. |
| \mathbb{F} | Frame. |
| A | Disk area. |
| A_R | Wing aspect ratio. |
| b | Wing span. |
| B_m | Motor damping constant. |
| C_D | Coefficient of drag. |
| c_e | Elevon chord. |
| C_L | Coefficient of lift. |
| C_M | Coefficient of moment. |
| D | Drag force. |
| g | Acceleration of gravity. |
| h | Altitude. |
| J | Advance ratio. |
| J_m | Motor inertia moment. |
| K_e | Motor back-electromotive force. |
| k_p | Spring constant. |
| K_t | Motor torque constant. |
| k_v | Damper constant. |
| L | Lift force. |
| l | Rolling moment. |
| L_m | Motor inductance. |

m Mass.
 m Pitching moment.
 n Yawing moment.
 p, q, r Angular velocity components.
 p_N, p_E, p_D Position NED coordinates.
 q_0, q_1, q_2, q_3 Quaternion components.
 r Radius.
 R_m Motor resistance.
 Re Reynolds number.
 S Wing surface area.
 T Thrust force.
 t Time.
 u, v, w Velocity components.
 V Aispeed.
 V_m Motor voltage.
 V_{batt} Battery voltage.
 Y Side force.

Subscripts

i, j, k Computational indexes.
 x, y, z Cartesian components.
0 Initial condition
B Body Frame
BX Back transition mode
c Servomechanism controller term
e Equilibrium condition
G Geometric Frame
H Hover mode
I Inertial Frame

| | |
|------|------------------------------|
| i | Integrative controller term |
| L | Level mode |
| l | left side |
| lon | Longitudinal. |
| max | Maximum limit. |
| min | Minimum limit. |
| p | Proportional controller term |
| r | right side |
| ref | Reference condition. |
| W | Wind Frame |
| wake | Fully developed wake. |
| X | Transition mode |

Superscripts

| | |
|----|---------------------------------------|
| -1 | Inverse. |
| ~ | Deviation from equilibrium condition. |
| T | Transpose. |

Chapter 1

Introduction

This chapter briefly explains the motivation behind the work proposed for this work and gives an overview of the importance of Unmanned Aerial Vehicles (UAVs) flight and application regarding the Eye in the sky Project. The goals and contributions of this work are also presented, along with a description of the content covered in each of the subsequent chapters.

1.1 Motivation

The relevance of this dissertation stems from the Eye in the Sky project. This project involves the design and development of an aerial platform that will support wildfire operations [1].

The system aims to improve the situational perception and monitoring of the fire perimeter through the provision of geo-referenced images of high spatial resolution in real time and the improvement of emergency communications networks by transporting communications repeaters.

In order to collect and transmit images and to relay communications in support of wildfire management, the equipment needs to be risen and placed in altitude over the area of interest. The possibility of using unmanned aerial means to carry this equipment is clear and widely recognised. However, there are several limitations pointed to its use. The first is the possible interference with other aerial firefighting assets operating within the same altitude range. Another issue is the low autonomy of these devices, typically less than 30 minutes.

In the Eye in the Sky project, the solution considered is the development of a platform consisting of a High Altitude Balloon (HAB) and an UAV [1], as explained in Figure 1.1. The HAB will be responsible for ascending in altitude all the equipment, including the UAV. Then, when the platform starts moving away from the region of interest, the UAV is released. The UAV carries the payloads and is responsible for the guided descent and controlled release of probes in a predetermined location. For the described operation, the use of a flying wing with Vertical Take-Off and Landing (VTOL) is suggested since it does not require a landing runway and because of its landing precision capabilities.

One of the major benefits of the proposed platform is that, by being transported by the HAB, the operational capacity of the UAV increases due to the potential energy accumulated at high altitude. It

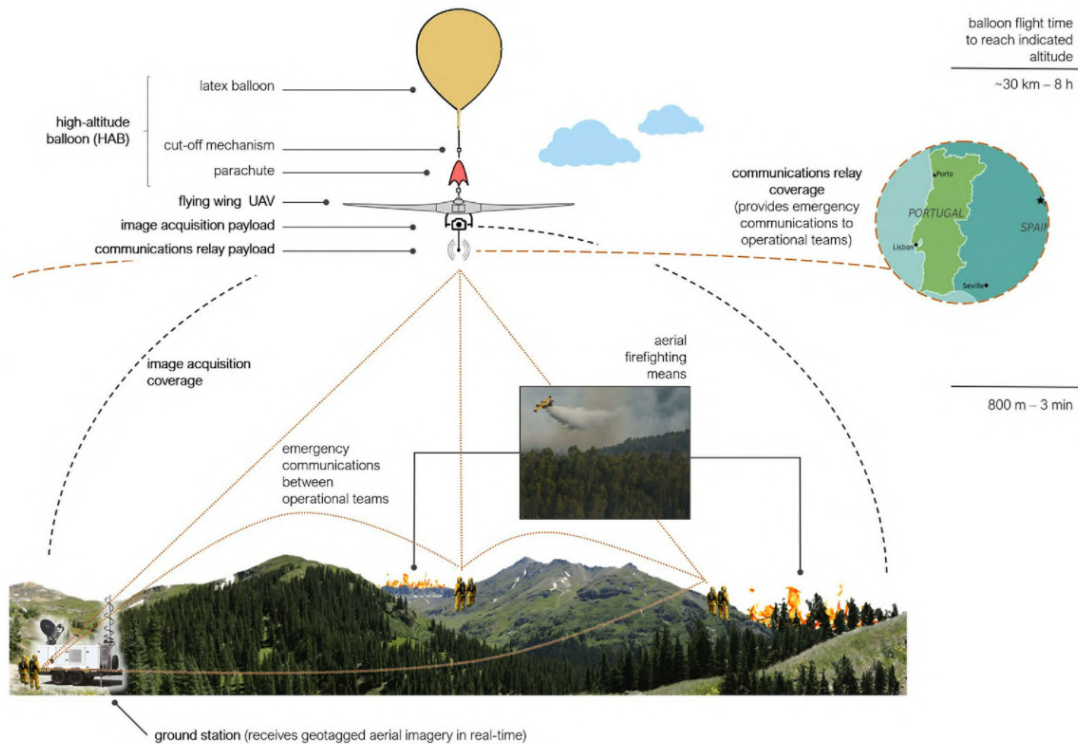


Figure 1.1: Representation of the HAB + UAV solution for wildfire operations, from [1]

will allow long-range flights with little or no power usage, converting altitude into ground cover. Another advantage is the ability to deploy the system in a short period of time, so that communication and aerial image transmission are immediately guaranteed.

This solution besides covering hundreds of kilometres at different heights with a considerable speed, will allow the acquisition of data otherwise unobtainable in a cost-effective manner. Similarly, to support wildfire operations, various monitoring applications can benefit from this complete platform and, in particular, from UAVs.

1.2 Topic Overview

UAVs are a class of aircraft that can fly without a human pilot onboard. These vehicles can be remotely controlled from the ground, which require reliable wireless communication, or controlled by autonomous flight controllers onboard. In both cases, aircraft systems require some sensor payload, such as cameras, Inertial Measurement Units (IMUs), LiDAR, and GPS to collect and transmit data in real time [2].

As referred in the previous section, UAVs have several applications, both in military and civilian fields, such as search and rescue, surveillance, precision agriculture, infrastructure inspection, wireless communications, package delivery, aerial photography, 3D mapping, among others [3]. As a result of the many different ways in which these systems can be used, the demand for this group of aircraft has increased in recent years. Numerous designs have been developed with different specifications for their size, weight, range, endurance, engine type, and configuration in order to meet the requirements

imposed by such applications. In addition to the different designs, several control techniques have been studied to improve both the stability of the UAV and the flight maneuvering. Besides, there are other concerns in a higher level of control that can be solved with the support of optimisation techniques, for instance, avoiding obstacles and potential collisions, the search for the shortest path, the savings in battery life, path planning, or flight formations.

One of the main obstacles UAVs currently face, due to their small dimensions and light weight, is maintaining control at high altitudes and in windy conditions. However, before beginning to study control methods, it is essential to select an appropriate type of UAV for the mission of interest.

In respect to their shape and configuration, UAVs are usually classified into two categories: fixed-wing aircraft, which have better performance in flight efficiency and long endurance missions; and rotor-craft, which are preferred for their precise maneuvering and hover capability. In order to combine these two capabilities, recent developments have been done in a third category: the fixed-wing VTOL aircraft, sometimes mentioned as hybrid aircraft [4], able to perform both hovering and efficient aerodynamic flight.

In this manner, there are several ways to perform VTOL manoeuvre, including tilting-rotor, quadplane, thrust-vectoring, tail-sitting, and others [5]. For the purpose of this dissertation, a flying wing tail-sitter is chosen as a case study of the modeling and control challenges associated with this unconventional configuration.

1.2.1 Tail-sitter

This platform, by definition, takes off and lands on its tail, which means, in a nose-up orientation. After taking off in this orientation, the entire system can tilt (pitches down about 90°) and shifts from hover to aerodynamic flight mode. Since a tail-sitter uses the same rotors for both hover and aerodynamic flight, no additional ones are required. For landing, the vehicle has to return to vertical flight, by pitching up about the same 90°, and then flying down again in the nose-up orientation until it reaches the ground. An illustration of the described movement is shown in Figure 1.2.

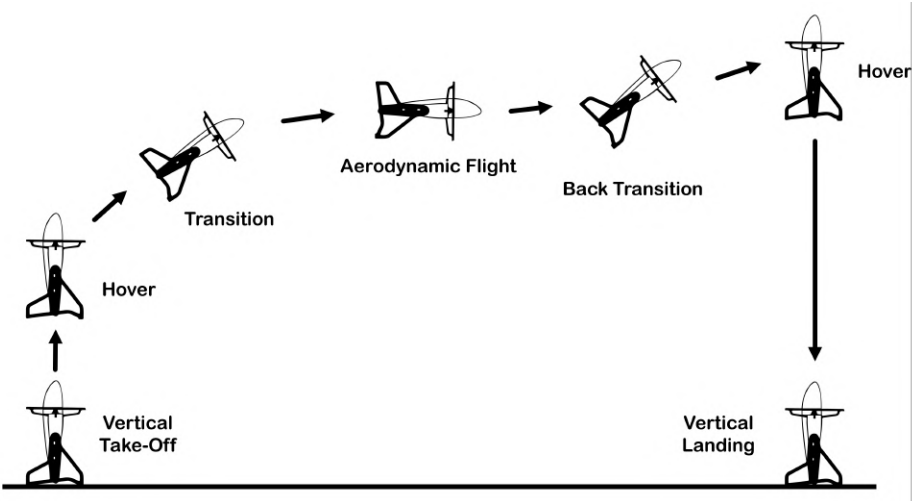


Figure 1.2: A tail-sitter take-off and landing transitions to and from level flight mode

Historically, one of the earliest tail-sitter VTOL aircraft design was the "Convair XFY-1", developed by the US Army in the 1950's, shown in Figure 1.3. This project was cancelled soon after its first flights due to the difficulty of the control and the pilot's limited view during landing. The main control problem was related to the transition from vertical to horizontal flight [6]. This maneuver requires strong pitching moments that could not always be created with the limited control authority of the aircraft ailerons. In an effort to solve these difficulties, there were some models created involving more complex systems, such as more rotors or more actuators. Beyond the increasing weight and complexity of the equipment involved, these newer designs were preferred as they could favour the pilot's work.

Nevertheless, because aerial robots do not have these pilot's view problems, several tail-sitter VTOL aerial robots have been researched and developed in recent years [7]. The main difficulty in their design has been achieving robust transitions between hover and leveled flights, and so the first tail-sitter UAV models designed were over-engineered when compared with some newer options. Naturally, with the increasing sensor precision as well as control strategies, more optimized tail-sitting prototypes started to be developed.

In their simplest form, one of the advantages of tail-sitters is that, unlike the tilting-rotor, quadplane and thrust-vectoring, they do not need extra actuators to perform the vertical manoeuvres. The main requirements of a tail-sitter are a high thrust-to-drag ratio (greater than 1, as compared to 0.3 for conventional aircraft) and large control surfaces with large deflections, which allows them to achieve vertical and hover flight and the desired high levels of aerobatic maneuverability. These platforms can have one or more propellers, always pointing in the longitudinal direction on the body frame. Since its propellers cannot tilt, the system has fewer mechanisms and actuators. The minor number of components reduces the probability of failures and increases the system's reliability.

When comparing tail-sitters with tilt-rotors aircraft, which is one of the most similar configuration solutions for the exposed requirements, some more advantages can be indicated. Tilt-rotors maintain the same body orientation in both vertical and aerodynamic flight, and therefore, these aircraft undergo high drag forces especially during take off and landing. With large drag forces, the aircraft will have to apply much more thrust in these flight stages than tail-sitters which reduces the aircraft efficiency. During vertical flight and hovering, tail-sitters are also considered to have greater maneuverability. In this orientation their control surfaces should have more control authority than tilt-rotors since larger forces and moments can be created by elevon deflections downstream the propellers.

1.2.2 Tail-sitter Control Challenges

A control system can be simply defined as a system which controls other systems response in order to achieve a desired state. Among all the existing control types, they can be generally categorized in two groups: linear control systems and nonlinear control systems. As the designation suggest, a linear control system is based on models governed by linear differential equations and a nonlinear control system is based on models often governed by nonlinear differential equations.

The simplicity of linear models structure gives them a great amount of advantages as they can be analyzed using very powerful known mathematical tools and they are less demanding during computer



(a) Horizontal Flight



(b) Vertical Flight

Figure 1.3: Convair XFY-1 aircraft [8].

simulations. The problem is that, because most real-world systems are highly nonlinear, linear models are never exact, and therefore of low interest for the controllers development. However, in cases where mainly solutions near a stable operating point are of interest, nonlinear systems can often be linearized by approximating them by a linear system. In this way, linear control techniques can be used in nonlinear systems.

The challenge of controlling a tail-sitter, in comparison with conventional aircraft, arises from the very different aircraft dynamics between hover and leveled flights. Such differences may require distinct controllers for each operating mode. To overcome this difficulty, several successful control strategies for unmanned tail-sitters have been reported and tested. A plausible solution suggests a *supervisory control* switching logic choosing between hover and level flight linear controllers. The problem in this control structure is the add of a discrete behaviour to the continuous UAV model when the controller switches during operating mode transitions. This creates a new layer of complexity which must be dealt appropriately. The use of the same control structure but different gains for different flight modes is a very well known technique used in the aeronautic industry, which is commonly referred as *gain scheduling*. In [9], the use of a PID (Proportional-Integral-Derivative) controller is explored for each flight mode. A PID is a control loop mechanism employing feedback and it is built by analysing a specific transfer function. These developed controllers are based on transfer functions that describe approximately the main aerodynamic behavior of the longitudinal and lateral modes. In this work, the transfer functions were identified analysing data from an experienced pilot executed excitation maneuvers from trim conditions.

Continuing in linear control, in [10], [11] and [12] optimal control techniques, namely LQR (Linear-Quadratic Regulator), were successfully implemented during both flight conditions and transitions, in a conventional configuration aircraft, in a flying wing and in a tail-sitter, respectively. The theory of optimal control is concerned with operating a dynamic system at minimum cost. Therefore, the LQR algorithm is essentially an automated way of finding an appropriate state-feedback controller which reduces the amount of work done by the control systems engineer to optimize the controller.

More complex linear control techniques are also experimented by the research community. Controllers based on *adaptive control* [13] or *hybrid control* [14] are tested on tail-sitters.

In [15], a nonlinear control technique was used. In this case, a backstepping controller applied on an adaptive dynamics model estimated by an on board *recursive least squares* was able to control the aircraft during flight tests.

In [16], a single quaternions based controller, with a cascaded architecture, is developed for the very same tail-sitter propose for this work, the X-VERT VTOL Horizon Hobby, and is proven through experimental flight tests to be able to perform the transitions needed.

1.3 Platform Presentation



Figure 1.4: E-flite X-Vert VTOL Horizon Hobby [17]

Having an overall perspective about the tail-sitter UAV configuration, it is time to select an appropriate platform to model and control. Aligned with the motivational project where this work is inserted, some requirements for the required system are set:

- The system should be able to communicate with a GCS (Ground Control Station) in order to share flight state information and onboard sensors data in real-time, tune parameters while a vehicle is in flight, and other useful functionalities.
- The system should perform autonomous flight and execute pre-planned flight missions, such as taking off, hovering in a steady position and then landing, in a first instance. In further project phases, the goal should be to perform more complex pre-planned flight missions or even to follow references generated in real time, including position, altitude and velocity.
- The system should also enable manual control of the aircraft, as a matter of redundancy and necessity for initial flight setup.

Since the focus of this work is not to design an aircraft itself, working with a readily available UAV will streamline the process. Therefore, the proposed platform was the E-flite X-Vert VTOL tail-sitter, made by Horizon Hobby, shown in Figure 1.4.

Its geometry can be described as a tailless blended wing-body aircraft (flying wing) with a span of approximately 0.5 m. As a result, instead of having ailerons on its wings and elevators on its tail, it has a single control surface on the trailing edge on each side, known as elevons. The X-Vert has two propellers, aligned with the control surfaces, and fixed in a forward direction in the body frame.

Being a commercial tail-sitter vehicle, the X-Vert can be bought ready to fly together with a *Spektrum* Transmitter. With a custom hardware and attitude controller, this platform is capable of performing stable hover and level flight as well as performing transitions between these two flight modes. On its packaging and attached manual, it states that, with its proprietary controllers, the X-Vert is able to fly horizontally in “moderately windy”, but hover and flying vertically should only be attempted in “very light or no wind”.

The receiver, sensors (IMU and barometer), ESCs (Electronic Speed Controllers), and controller are all assembled in a unique board. However, as the source code of this flight controller is not available, this entire board will need to be replaced by new components. Analysing some possible controller boards and verifying available compatible sensors, the X-Vert platform can fulfill all the stated requirements.

1.4 Objectives

The purpose of the present dissertation resides over the development of a flight controller for the X-Vert tail-sitter and its validation in a simulated software environment. Hence, the main objectives are:

1. To understand and report the detailed nonlinear model of the tail-sitter platform provided in *Matlab* and implement it in a real-time simulator suitable for control design and testing purposes.
2. To focus on the longitudinal stability analysis of the nonlinear model by making use of linearization and linear control theory analysis.
3. To design a longitudinal flight controller based on linear control techniques, namely classic and optimal control techniques for stabilization and reference tracking.
4. To demonstrate through simulations the autonomous ability of the controller to perform stable hover and level flights, as well as transition between these two types of flights, and also to perform autonomous take-off and landing.
5. To simulate two minimum flight missions which would include all the critical phases and verify the smooth operation of the designed controller. The first mission should include only vertical flight, and the second mission should include transitions and aerodynamic flight. These missions would also serve as starting points for future real environment flight experiences.

1.5 Thesis Outline

After this introductory chapter, the dissertation outline is divided into four parts. First, the platform nonlinear mathematical model and the simulator implementation are detailed in Chapter 2. In Chapter 3, the trimming and linearization methods used are described along with the longitudinal model characterization within the flight envelope. In Chapter 4, the LQR control theory and variations implemented are briefly explained, as well as the final control strategy and architecture implemented to meet the proposed objectives. In Chapter 5, the simulation flight results are presented and analysed. Finally, the dissertation conclusions are presented in Chapter 6 along with recommendations for future works.

Chapter 2

Nonlinear Model

The current chapter describes a nonlinear model of a tail-sitter Micro-Air-Vehicle (MAV). This model was provided by Professor José Raul Azinheira for simulation purposes, which was in turn adapted from a experimentally validated model presented in [16]. Beside of the model described in the present work has been provided without experimental validation as would be desirable, because it was adapted from a validated model, it is expected to represent the behavior of the X-Vert vehicle in an acceptable way and therefore it was used for control purposes. The simulator is implemented in *Matlab/Simulink*.

The description of the nonlinear model is divided into five main sections. The first section describes the essential frames and conventions for the model development. In the second one, the equations of motion of a rigid aircraft are presented. The third section details how the airspeed and other essential aerodynamic variables in formulating the equations of motion for a MAV are defined. The fourth section presents the mathematical expressions of the forces and moments acting on the body. Lastly, the implemented aircraft simulator is presented.

Throughout this chapter, the following notation is used in the aircraft model equations:

- **scalar values** are symbolized by uppercase and lowercase letters;
- **vectors** are symbolized by boldface lowercase letters;
- **matrices** are symbolized by boldface uppercase letters;
- **coordinate frames** are symbolized by blackboard-bold letters.

2.1 Frames and Conventions

2.1.1 Reference Frames

To derive and understand the aircraft dynamic behaviour, this section presents the various coordinate systems used to characterize the position and orientation of the aircraft, the relation between these coordinate systems as defined in [18], and some conventions and nomenclature for the aircraft model development. The reference frames of interest are:

- **The inertial frame** $\mathbb{F}_I = [x_I, y_I, z_I]^T$: is fixed on the surface of the Earth and is chosen as NED coordinates (North-East-Down). The ground is assumed to be locally flat and non-rotating for the

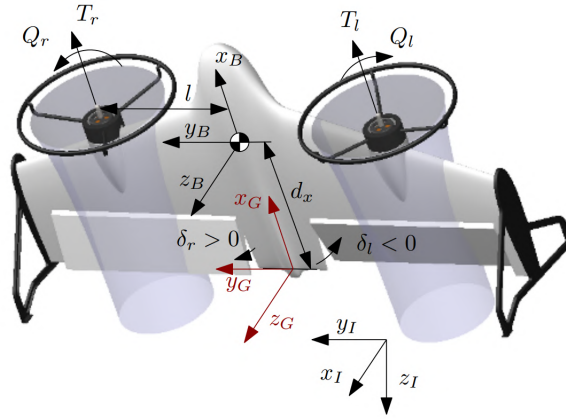


Figure 2.1: Coordinate frames, geometric dimensions and the four actuators [16]

current application, considering the distance and duration of the flight.

- **The geometric frame** $\mathbb{F}_G = [x_G, y_G, z_G]^T$: is fixed on the trailing edge of the flying wing, laterally centered and inside the chord line. The geometric frame G is used for geometric references independent of the center of mass location.
- **The body frame** $\mathbb{F}_B = [x_B, y_B, z_B]^T$: is fixed on the aircraft and located at its center of gravity CG. The center of mass is assumed to be laterally centered, inside the chord line and located at a distance d_x from the trailing edge. The x_B is chosen so that it is pointing forward with the longitudinal reference line, the y_B is aligned with the right wing of the UAV and the z_B is placed pointing down, completing the right-handed set.
- **The wind frame** $\mathbb{F}_W = [x_W, y_W, z_W]^T$: fixed on the same origin as the body axes, but differs in orientation. Instead of associated with aircraft physical assembly, the wind axes system is aligned to correspond to its motion. The x_W points into the incoming air stream, the z_W points down and lies in the plane symmetry and the y_W is chosen perpendicular to the other two axes.

The need to distinguish the inertial and body frames arise since, as it is shown in the next sections, the Newton's equations of motion are derived relative to a fixed inertial frame while forces and moments that act on the aircraft body are easily represented in the body frame.

2.1.2 Attitude Representation

The transformation between the body frame \mathbb{F}_B and the inertial frame \mathbb{F}_I is obtained by rotating them certain angles about their three axes. These angles are commonly referred to as Euler angles, namely Roll (ϕ), Pitch (θ) and Yaw (ψ). This representation is preferably used because it provides intuitive means for characterizing the orientation of a body in three dimensions. The transformation from the inertial frame \mathbb{F}_I to the body frame \mathbb{F}_B , represented by the rotation matrix \mathbf{R}_I^B is:

$$\mathbf{R}_I^B = \mathbf{R}_{ZYX} = \mathbf{R}_X(\phi) \mathbf{R}_Y(\theta) \mathbf{R}_Z(\psi) \quad (2.1)$$

This matrix is composed by a sequence of three rotations which can be ordered in several different ways. Since their order will influence the resulting matrix, the rotation sequence 'ZYX' is used as it is the

most common choice in aircraft applications [18].

Thus, the transpose of \mathbf{R}_I^B gives the transformation from the body frame \mathbb{F}_B to the inertial frame \mathbb{F}_I and is represented by \mathbf{R} as follows:

$$\mathbf{R} = \mathbf{R}_B^I = \begin{bmatrix} c_\theta c_\psi & (-c_\phi s_\psi + s_\phi s_\theta c_\psi) & (s_\phi s_\psi + c_\phi s_\theta c_\psi) \\ c_\theta s_\psi & (c_\phi c_\psi + s_\phi s_\theta s_\psi) & (-s_\phi c_\psi + c_\phi s_\theta s_\psi) \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \quad (2.2)$$

in which s_ϕ and c_ϕ represent the sine, $\sin(\phi)$, and cosine, $\cos(\phi)$, operators, respectively. During the computational implementation it is used the *Matlab* function *eul2rotm*.

Although the physical interpretation of Euler angles is clear and used widespread, it is also known that they have a mathematical singularity, commonly referred to as gimbal lock. For the Euler angle sequence chosen, when the pitch angle approaches $\theta = \pm \pi/2$, there is a singularity, which means that the yaw and roll angle are not unique and computational instabilities may happen. One possible solution to this problem could go through switch between sets of Euler angles during flight. However, considering the wide flight envelope of the aircraft under study (compared to conventional aircraft) there is no Euler set that will be suitable for all expected orientations. Instead, the quaternion q attitude representation is used whenever possible despite of its lack of intuitive appeal. Quaternion attitude definition, relations and properties are given in Appendix B.

Similarly to the rotation matrix defined in (2.2) with Euler angles, the transformation from the body frame \mathbb{F}_B to the inertial frame \mathbb{F}_I from a quaternion representation is given by:

$$\mathbf{R} = \mathbf{R}_B^I = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (2.3)$$

During the computational implementation, the *Matlab* function *quat2rotm* is used to compute the desired rotation matrix.

2.2 Actuators

Actuators are controllable surfaces or components that modify the forces and moments acting to the body. The aircraft in study contains four unconventional actuators, namely:

- **The left thruster:** which is driven by the left thruster signal $\tau_l (-)$ that goes between 0 and 1. The left thruster creates a force T_l in the longitudinal direction and a torque Q_l about the propeller axis of rotation, both applied to the aircraft. This torque is opposite to the direction of the propeller rotation.
- **The right thruster:** which is driven by the right thruster signal $\tau_r (-)$ that also goes between 0 and 1. The right thruster creates a longitudinal force T_r and a torque Q_r about the propeller axis of rotation. To cancel the left thruster torque, the right propeller rotates counterclockwise.

- **The left elevon:** which is a control surface located at the trailing edge of the vehicle, intentionally located behind the left thruster, in the slipstream, to ensure it maintains effectiveness at low speeds. Its deflection is represented by δ_l (rad) and its maximum deflection is 39° for each side.
- **The right elevon:** which is similarly located behind the right thruster and its deflection is represented by δ_r (rad). By convention, a control surface deflection increases the camber, so both elevon deflections follows are defined by the right hand rule along the y_B axis, i.e. a positive deflection means the elevon deflects down.

The elevons deflection divert the airflow over them (originated from forward velocity or induced airflow created by the propellers) and generate a force. Since this force is not applied at the CG, a *positive* deflection will cause a *negative* moment.

Because most of the equations that describe the aerodynamic forces and moments in books are expressed in terms of standard actuators configuration, it is useful to relate the listed actuators to the conventional control surfaces (**elevator** δ_E , **aileron** δ_A and **rudder** δ_R) in addition to **thrust signal** δ_T . So in this aircraft configuration, driving the elevons differentially has the same effect as ailerons, producing a torque about x_B axis, while driving elevons together has the same effect as an elevator, causing a torque about y_B axis. On the other hand, the rudder-like effect can be produced by differential thrusts. The longitudinal absolute force results from the combination of both thrusters. These relations are expressed in the following equations:

$$\begin{aligned} \delta_E &= \frac{\delta_r + \delta_l}{2} & \delta_A &= \frac{\delta_r - \delta_l}{2} \\ \delta_R &= \frac{\tau_r - \tau_l}{2} & \delta_T &= \frac{\tau_r + \tau_l}{2} \end{aligned} \quad (2.4)$$

When in nose up attitude, the tail-sitter is able to perform hover in a fixed position, climb and descend in a vertical line or even perform low speed displacements in the horizontal plane. The movement in the z_B axis direction is achieved by lightly pitching about the y_B axis, produced by the elevons equal deflection. The movement in the y_B axis direction is achieved by a gentle tilt about the z_B axis which can be produced by differential thrusts. The heading, in this situation, can be controlled by differential elevon deflections producing a moment in the x_B axis.

All the terms detailed in this section are shown in Figure 2.1.

2.3 Aircraft Equations of Motion

To describe the aircraft behaviour, thirteen state variables will be introduced. These are:

- Three **velocity states** $\mathbf{v} = [u, v, w]^T$: translation velocity of the aircraft CG in the body frame \mathbb{F}_B , also called body velocity. Expressed in [m/s].
- Three **angular velocity states** $\boldsymbol{\omega} = [p, q, r]^T$: rotational velocity of the aircraft in the body frame \mathbb{F}_B , also called body rates. Expressed in [rad/s].
- Three **position states** $\mathbf{p} = [p_N, p_E, p_D]^T$: the NED position of the CG defined relative to the inertial frame \mathbb{F}_I . Expressed in [m]. Altitude will be denoted as $h = -p_D$.

- Four **angular position states** $\mathbf{q} = [q_0, q_1, q_2, q_3]^T$: attitude of the aircraft defined in terms of quaternions. This representation does not have unit, and so it is symbolized by [-].

The rigid body equations for the kinematic relationships of the aircraft as well as its dynamics can be found in any aircraft textbook such as [18].

Kinematics

Besides, the aircraft velocity is commonly expressed in the body frame \mathbb{F}_B , the aircraft position is usually measured and expressed in the inertial frame \mathbb{F}_I . Consequently, to relate these two variables the rotational transformation is required.

$$\dot{\mathbf{p}} = \mathbf{R}(\mathbf{q}) \mathbf{v}_B \quad (2.5)$$

The quaternion attitude relation with the body rates is more complex and is represented by the Ω matrix, shown in (2.7).

$$\dot{\mathbf{q}} = \frac{1}{2} \Omega \mathbf{q} \quad (2.6)$$

$$\Omega = \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \quad (2.7)$$

During implementation, when propagating (2.6) the quaternion unitary is loosed. In order to maintain this property a second term is added to the previous equation ([18]):

$$\dot{\mathbf{q}} = \frac{1}{2} \Omega \mathbf{q} + \lambda \mathbf{q} \quad \text{where,} \quad \lambda = 1 - \|\mathbf{q}\|^2 \quad (2.8)$$

Dynamics

The aircraft dynamic equations (2.9) are derived from Newton's Second Law to rigid bodies and result into the following equations that express the relationship between the forces \mathbf{F}_B and moments \mathbf{M}_B acting on the aircraft and the resulting accelerations. These equations are then integrated to generate the aircraft states \mathbf{p} , \mathbf{q} , \mathbf{v}_B , $\boldsymbol{\omega}_B$.

$$\begin{aligned} \dot{\mathbf{v}}_B &= \frac{1}{m} \mathbf{F}_B - \boldsymbol{\omega}_B \times \mathbf{v}_B \\ \dot{\boldsymbol{\omega}}_B &= \mathbf{I}^{-1} (\mathbf{M}_B - \boldsymbol{\omega}_B \times \mathbf{I} \boldsymbol{\omega}_B) \end{aligned} \quad (2.9)$$

where m is the mass of the aircraft, \times is the 3D cross product, and \mathbf{I} is the inertia matrix of the aircraft about the body frame \mathbb{F}_B axes centered at the CG. The null elements in this matrix assume a symmetry that does not exist in the real model and that the CG is perfectly located over the x_B axis. These assumptions should be seen as an approximation.

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{xz} & 0 & I_{zz} \end{bmatrix} \quad (2.10)$$

The values of the aircraft parameters m and I are presented in Table A.1.

2.4 Aerodynamic Variables

During 2.1.1 some reference frames of interest, such as the wind frame, were defined. The utility of the wind frame \mathbb{F}_W is that it simplifies the calculation of aerodynamic forces since stability derivatives are measured in this axis system. The air velocity $\mathbf{v}_a = [u_a, v_a, w_a]^T$ is the velocity of the aircraft relative to the surrounding air. When in no wind conditions, the air velocity \mathbf{v}_a and the velocity relative to the body reference frame \mathbf{v}_B are the same. When in wind condition, to guarantee an appropriate modelling, a careful distinction between these velocities must be taken into account. The wind velocity \mathbf{v}_w is the wind velocity relative to the inertial frame \mathbb{F}_I , represented in the body frame \mathbb{F}_B . The relation between these three vector velocities is:

$$\mathbf{v}_a = \mathbf{v}_B - \mathbf{v}_w \quad (2.11)$$

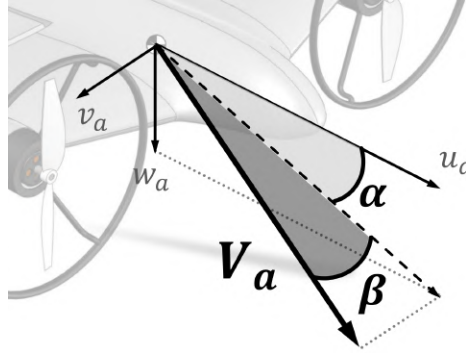


Figure 2.2: Airspeed V_a , angle of attack α and angle of side slip β

Besides of the vector representation, the air velocity can also be expressed in terms of the airspeed V_a , the angle of attack α and the angle of side slip β , which is easier to directly use in aerodynamic forces and moments equations. To better visualise this, the listed variables are drawn in Figure 2.2 and are defined as followed:

$$\begin{aligned} V_a &= \sqrt{u_a^2 + v_a^2 + w_a^2} \\ \alpha &= \text{atan2}(w_a, u_a) \\ \beta &= \text{asin}(v_a, V_a) \end{aligned} \quad (2.12)$$

In the aerodynamics model and thrusters model, instead of using the air velocity of CG, the velocity at different points on the aircraft can be evaluated by accounting for its position in relation to the CG and the angular velocities. That said, the air velocity at any point i , located at a position \mathbf{r}_i on the aircraft, can be found from the following equation:

$$\mathbf{v}_{a,i} = \mathbf{v}_a + \boldsymbol{\omega}_B \times \mathbf{r}_i \quad (2.13)$$

At each point of interest, 2.12 can also be applied.

2.5 Tail-sitter Model

In 2.3, $\mathbf{F}_B = [F_x, F_y, F_z]^T$ and $\mathbf{M}_B = [L, M, N]^T$ were referred as the total force and moment acting at the CG, as shown in Equations 2.14. In this model, five types of forces were considered to be involved:

- The **gravitational force** F_g represents the aircraft weight which, by definition, is applied at the CG, and so it does not produce torque.
- The **thrust** F_{th} and the **torque** M_{th} which result from the left and right operating propellers.
- The **aerodynamic forces** F_a and **moments** M_a which are obtained through the aerodynamics effects on the wing body and elevons.
- The **rod aerodynamic forces** F_r and **moments** M_r which are purely resistive and originate from structural segments.
- The **ground contact forces** F_c and **moments** M_c which represent the vehicle interacting with the ground.

$$\begin{aligned} \mathbf{F}_B &= \mathbf{F}_g + \mathbf{F}_t + \mathbf{F}_a + \mathbf{F}_r + \mathbf{F}_c \\ \mathbf{M}_B &= \mathbf{M}_t + \mathbf{M}_a + \mathbf{M}_r + \mathbf{M}_c \end{aligned} \quad (2.14)$$

All the geometric values, thruster parameters and aerodynamic coefficients used to describe the X-Vert model along this chapter are given in Appendix A.

2.5.1 Gravity

The gravity's effect on the aircraft is modeled as a force proportional to the mass m and the gravitational acceleration g applied at the CG. In inertial frame \mathbb{F}_I this force always act on z_I axis direction, so it can be easily transformed into body frame \mathbb{F}_B , only by applying the transposed of rotation matrix defined in (2.3).

$$\mathbf{F}_g = \mathbf{R}^T(\mathbf{q}) \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (2.15)$$

The CG location is assumed to be in the vertical plane of symmetry of the aircraft and in the same horizontal plane as the thrusters axes of rotation and the cord line of the wings. The distance from the trailing edge is denoted as d_x and is given in Table A.1.

2.5.2 Thrusters

The purpose of this section is to adequately formulate a model that describes the propulsion system of the tail-sitter, which has to result in the thrusters total force F_t and torque M_t generated from the throttle signals τ_l and τ_r .

To define the thrusters force, only the longitudinal forces component are considered. This means that, even when the free stream flow is not normal to the propellers disk plane, the appearing non-

longitudinal forces will be neglected since they are much smaller than the aerodynamic forces acting on the aircraft body. Therefore, the left and right thrusters forces T_l and T_r will be considered normal to the propellers disk and positive aligned with the x_B axis direction. So, the total thrust is:

$$\mathbf{F}_t = \begin{bmatrix} T_l + T_r \\ 0 \\ 0 \end{bmatrix} \quad (2.16)$$

The thrusters are placed at a distance $l = y_{p,G}$ along the y_B axis from the CG of the aircraft but in the same horizontal plane. That said, even though they do not produce moments in the y_B direction, when the thrust forces are not the same, they generate a yawing moment about the z_B axis. Additionally, each propeller also generates a torque Q_l and Q_r . Besides rotating in opposite directions when these torques are not equal, they also generate a rolling moment about the x_B axis. Both contributions are expressed in the first term of the following equation:

$$\mathbf{M}_t = \begin{bmatrix} Q_r - Q_l \\ 0 \\ l(T_l - T_r) \end{bmatrix} + I_{th}(\Omega_l - \Omega_r) \begin{bmatrix} 0 \\ -r \\ q \end{bmatrix} \quad (2.17)$$

The second term from the previous equation derives from the gyroscopic moments. The aircraft has to overcome the propellers' resisting torques and also the ones from the motors moving parts. Although these effects result in a small contribution, they are not negligible. In (2.17), I_{th} is the rotational inertia of the spinning parts of both thrusters, about x_B axis. Ω_l and Ω_r are the left and right thruster angular velocities and are both positive.

As this tail-sitter is equipped with two thrusters, left and right, each of the thruster unit consists of a propeller coupled with a brushless DC (Direct Current) electric motor, in turn driven by an Electronic Speed Controller (ESC). During the next sections it is described how T_l, T_r, Q_l and Q_r are obtained from input signals τ_l and τ_r . First, there are the equations that describe how the ESC and the electric motor induce the rotation Ω_p in the propeller from the throttle signal τ and the battery voltage V_{batt} . Second, the equations that describe the propeller are represented relating the angular velocity Ω_p of the propeller and the inflow velocity u_{in} to generate a thrust force T and a torque Q .

ESC and Electric Motor

The speed of the motor-propeller is controlled by the ESC. The ESC is connected to the battery with an available voltage V_{batt} and varies the voltage provided to the motor V_m . Although ESCs as electronic systems have their own dynamics, these dynamics are fast and can be neglected. Thus, they will be described only as a proportion to the controller's commanded Pulse-Width-Modulation (PWM) signal τ , normalized between zero and one.

$$V_m = V_{batt} \tau \quad (2.18)$$

Since the ESC is run by a small LiPo battery whose power drains quickly, the actual voltage applied

to the electric motor will drop over time even for a constant signal command. However, in the present work, this will not be taken into account. A possible model of the battery could be found in [19].

About the electric motor, a simple model for a DC (Direct Current) motor from [20] is implemented. It is derived from the Kirchhoff's current law and Newton's second law applied to the electrical and mechanical system, respectively. This results in a second order model, including the motor current I and the torque generated by the propeller Q , which is represented by the following equation to obtain the motor angular velocity Ω :

$$\begin{aligned} \dot{I} &= \frac{1}{L_m} (V_m - R_m I - K_e \Omega) \\ \dot{\Omega} &= \frac{1}{J_m} (K_t I - Q - B_m \Omega) \end{aligned} \quad (2.19)$$

where J_m is the inertial moment, L_m is the inductance, R_m is the motor resistance, B_m is the damping constant, K_e is the back-electromotive force, and K_t is the torque constant. Note that the inclusion of the torque Q on the motor model, which will be explained in Section 2.5.2, effectively represents an internal feedback.

Propeller

In this section, a model of the propeller aerodynamics is presented based in Momentum Theory and Bernoulli's Equation, adapted from [21] and [22].

First, the inflow velocity u_{in} , drawn in Figure 2.3 is the longitudinal component of the velocity computed from the (2.11) taken at the propeller location r_{prop} from (A.3). Note that the left propeller should have negative y coordinate from the CG.

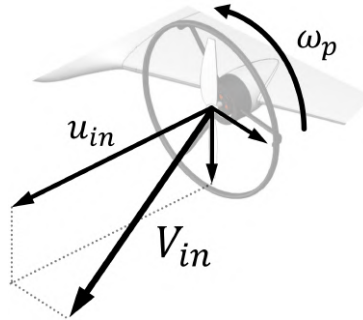


Figure 2.3: Inflow velocity and angular speed on the left propeller, adapted from [22].

Knowing the propeller angular speed Ω and the inflow velocity u_{in} , it is possible to calculate the advance ratio J , in $[1/rev]$, as follows:

$$J = \frac{\pi u_{in}}{\Omega r_p} \quad (2.20)$$

where, r_p is the radius of the propeller.

Then, according to the UIUC data site [23], for the *E-flite 130x75mm* propeller the values of the thrust and torque coefficients C_T and C_P can be numerically obtained from Figures 2.4 as a function of the

advance ratio J .

$$C_T = C_T(J), \quad C_Q = C_P(J) \quad (2.21)$$

Actually, the coefficients also depend on the Reynolds Number. However, because the variation between each of the Reynolds lines is relatively small, the mean line between the maximum and minimum value for each advance ratio J is used.

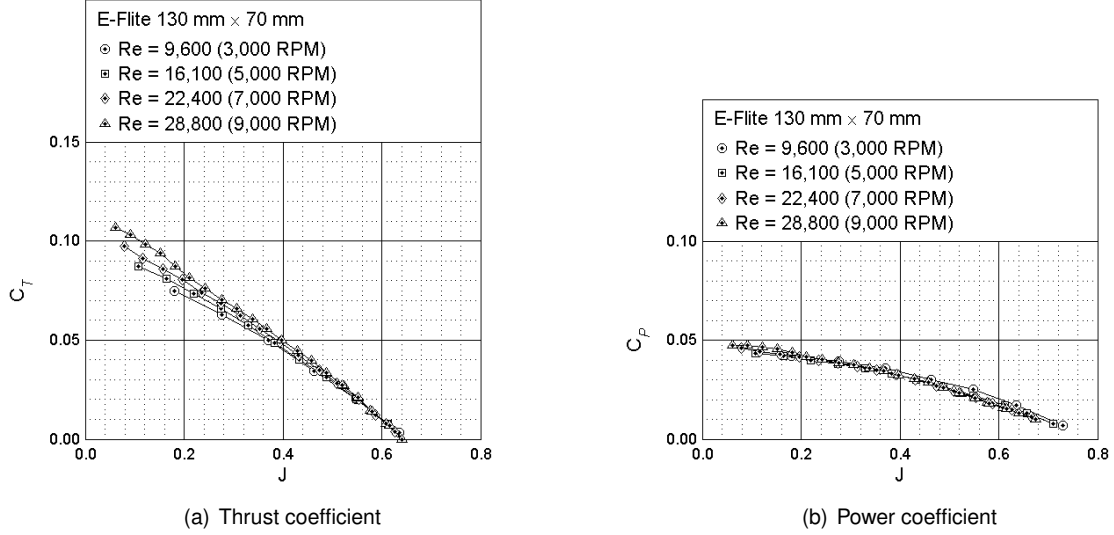


Figure 2.4: Thrust and torque coefficients as a function of the advance ratio J , from [23].

Finally, thrust force T and torque Q for any given angular speed Ω can be obtained from:

$$T = \frac{4}{\pi^2} \rho \Omega^2 r_p^4 C_T, \quad Q = \frac{4}{\pi^3} \rho \Omega^2 r_p^5 C_P \quad (2.22)$$

The last equations, where both T_l, T_r, Q_l and Q_r are computed as magnitudes, are then taken part in (2.17) that accounts for the direction of these quantities.

Before proceeding to the wing aerodynamics, further insight about the airflow behind the propeller must be provided, since it will be used in the next section for modelling the slipstream effects on the elevons.

Slipstream

From Momentum Theory [21] and [22], as it is deduced below, one can reasonably predict the velocity downstream of the propeller disk.

Once the propeller is moving, the incoming air stream normal to the propeller V_{in} is accelerated and a thrust force T is generated in the opposite direction. Its value is given by:

$$T = \underbrace{\rho A V_{prop}}_{\dot{m}} \underbrace{(V_{wake} - V_{in})}_{2 w_{ind}} \quad (2.23)$$

where $A = \pi r_p^2$ is the disk area, \dot{m} is the mass rate flow through the propeller, V_{prop} and V_{wake} represent the air velocity immediately below the propeller and the air velocity in the fully developed wake

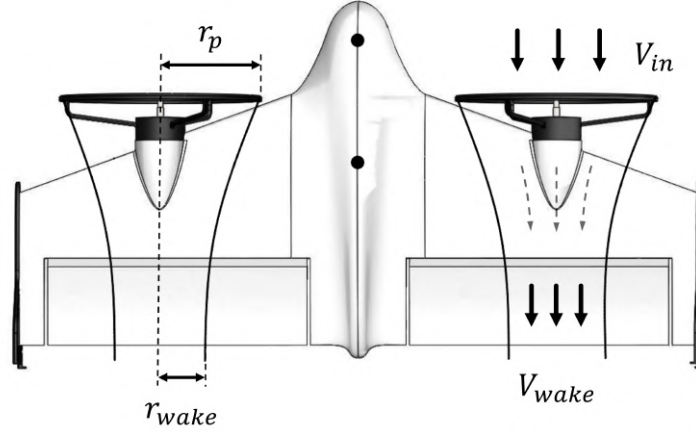


Figure 2.5: Slipstream variables

respectively, as illustrated in Figure 2.5. From this theory, V_{prop} is the average of V_{in} and V_{wake} , and the difference between V_{prop} and V_{in} is called the induced velocity w_{ind} . The airflow is assumed to be fully developed when the wake velocity reaches two times the induced velocity w_{ind} , which occurs approximately at a distance of two propeller's radii downstream of the propeller disk.

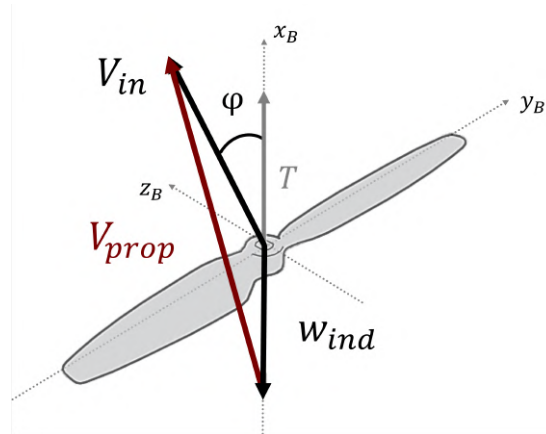


Figure 2.6: Induced velocity on a propeller, adapted from [21].

Keeping in mind that the induced airflow covers some of the control surfaces, it becomes important to calculate w_{ind} . From Figure 2.6, the resulting velocity at the disk propeller V_{prop} can be obtained from the sum of the undisturbed airspeed V_{in} and the induced velocity w_{ind} as:

$$V_{prop} = \sqrt{(V_{in} \cos \varphi + w_{ind})^2 + (V_{in} \sin \varphi)^2} \quad (2.24)$$

where $\varphi = \arccos(u_{in}, V_{in})$ represents the angle between V_{in} and the axis of rotation of the propeller. Combining (2.23) and (2.24) leads to the following fourth order equation in w_{ind} :

$$w_{ind}^4 + 2 \cos(\varphi) V_{in} w_{ind}^3 + V_{in}^2 w_{ind}^2 = \left(\frac{T}{2 \rho A} \right)^2 \quad (2.25)$$

Once the thrusts T have been calculated for both propellers, the absolute value of the induced

velocity w_{ind} in each one can be determined solving (2.25). Due to its high non-linearity, the equation is solved numerically.

In this same theory [21] and [22], assuming mass conservation and incompressible flow, as the air behind the propeller continues to accelerate, the area of the accelerated flow contracts. Thus, the radius of the wake at this point can be obtained from:

$$r_{wake} = \sqrt{r_p^2 \left(\frac{V_{in} + w_{ind}}{V_{in} + 2w_{ind}} \right)} \quad (2.26)$$

If fully developed wake is assumed, it can also be approximated to a simple relation with the propeller radius:

$$r_{wake} = \sqrt{2} r_p \quad (2.27)$$

Thus, the area of the elevon affected by the propeller's wake is given by:

$$S_{wake} = \underbrace{2 r_{wake}}_{d_{wake}} c_e \quad (2.28)$$

where d_{wake} is the wake diameter and c_e is the elevon chord.

The velocity vector in the fully developed wake $\mathbf{v}_{wake} = [u_{wake}, v_{wake}, w_{wake}]^T$ can be determined by:

$$\mathbf{v}_{wake} = \mathbf{v}_{in} + \begin{bmatrix} 2 w_{ind} \\ 0 \\ 0 \end{bmatrix} \quad (2.29)$$

As defined in Section 2.4, the Equations 2.12 can also be employed for the wake vector \mathbf{v}_{wake} to determine the specific angle of attack α_{wake} and side slip angle β_{wake} which will be useful to define wing aerodynamics. These angles are obtained from:

$$\begin{aligned} V_{wake} &= \sqrt{u_{wake}^2 + v_{wake}^2 + w_{wake}^2} \\ \alpha_{wake} &= \text{atan2}(w_{wake}, u_{wake}) \\ \beta_{wake} &= \text{asin}(v_{wake}, V_{wake}) \end{aligned} \quad (2.30)$$

Finally, the lateral position of the middle of the wake over the elevon is obtained from:

$$y_{mid\ wake} = y_{p,G} - (x_{p,G} - x_{e,G}) \tan(\beta_{wake}) \quad (2.31)$$

where $x_{p,G}, y_{p,G}$ and $x_{e,G}$ are the propeller and elevon coordinates given in the geometric frame \mathbb{F}_G .

Once more, note that, since the angular velocities at each propeller may differ, the resulting values from this wake model will not be necessarily equal for both sides, and should therefore be determined separately.

2.5.3 Wing Aerodynamics

Aerodynamic forces, as the name suggests, are the forces exerted on a body by the air passing around it. They are the result of the relative motion between these two elements and so, they can be interpreted acting in the wind frame \mathbb{F}_W and will depend on the true airspeed V_a , angle of attack α , and side slip angle β of the entire platform, i.e. evaluated for the CG. The rotation from the wind frame \mathbb{F}_W to the body frame \mathbb{F}_B can be done by applying the following rotation matrix:

$$\mathbf{R}_a = \mathbf{R}_W^B = \begin{bmatrix} c_\alpha c_\beta & -c_\alpha s_\beta & -s_\alpha \\ s_\beta & c_\beta & 0 \\ s_\alpha c_\beta & -s_\alpha s_\beta & c_\alpha \end{bmatrix} \quad (2.32)$$

in which s_ϕ and c_ϕ represent the sine ($\sin\phi$) and cosine ($\cos\phi$) operators, respectively. During the computational implementation the *Matlab* function *eul2rotm* is used.

Wing Segments

To calculate the aerodynamic forces and moments, the wing is decomposed into four segments, as demonstrated in Figure 2.7. The segments 1 and 2 represent the area of the half wing and elevon that is not affected by the fully developed wake on each side, left and right, respectively; and the segments 3 and 4 represent the area of the elevon under the propeller wake, also for the left and right side. This separation is important to ensure the effectiveness of the control surfaces mainly during hover flight.

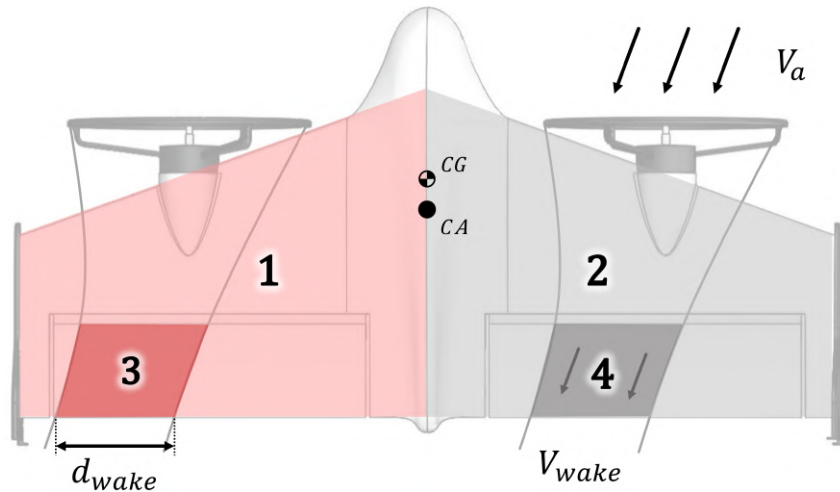


Figure 2.7: Wing Segments

The four segments are described by the position coordinates of their aerodynamic center $r_{i,G} = [x_{i,G}, y_{i,G}, z_{i,G}]$, given in the geometric frame \mathbb{F}_G , and surface S_i . The longitudinal position $x_{i,G}$ is known as located approximately at a quarter-chord from the leading-edge of the profile. In the case of the half wing segments, the mean aerodynamic chord \bar{c} is used and, in the case of under wake segments, the elevon chord c_e is used. On the other hand, the lateral position of the first segments is given approximately by the lateral coordinate of the trapeze center $y_{trapeze}$, from (A.6) and the lateral position of the under wake segments is estimated by the (2.31), for left and right side. See Figure 2.8.

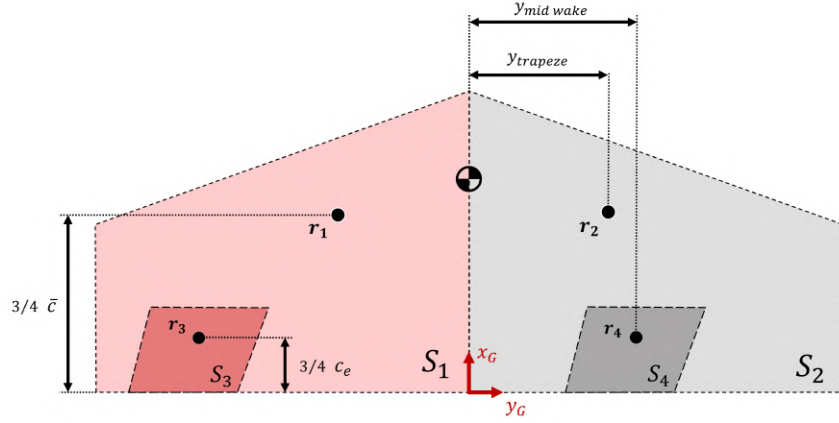


Figure 2.8: Aerodynamic center location and area of wing segments

Each element has its own air velocity $V_{a,i}$, angle of attack α_i , and side slip β_i . The first two are evaluated using Equations 2.12 and 2.13, and the other two using Equations 2.29 and 2.30 derived in the slipstream model for the respective position in the body frame $r_i = r_{i,G} - r_{CG}$. δ_i represents the control surface deflection on the specific segment. The Table 2.1 summarizes all the segments' parameters mentioned above.

Table 2.1: Segments Parameters

| Element | S_i | \bar{c}_i | $x_{i,G}$ | $y_{i,G}$ | $z_{i,G}$ | $V_{a,i}$ | α_i | β_i | δ_i |
|---------|--------------------|-------------|---------------|---------------------|-----------|--------------|-------------------|------------------|------------|
| 1 | $S/2 - S_{wake,l}$ | \bar{c} | $3/4 \bar{c}$ | $-y_{trapeze}$ | 0 | $V_{a,l}$ | α_l | β_l | δ_l |
| 2 | $S/2 - S_{wake,r}$ | \bar{c} | $3/4 \bar{c}$ | $y_{trapeze}$ | 0 | $V_{a,r}$ | α_r | β_r | δ_r |
| 3 | $S_{wake,l}$ | c_e | $3/4 c_e$ | $-y_{mid\ wake\ l}$ | 0 | $V_{wake,l}$ | $\alpha_{wake,l}$ | $\beta_{wake,l}$ | δ_l |
| 4 | $S_{wake,r}$ | c_e | $3/4 c_e$ | $y_{mid\ wake,r}$ | 0 | $V_{wake,r}$ | $\alpha_{wake,r}$ | $\beta_{wake,r}$ | δ_r |

Segment Forces and Moments

In an airfoil, as illustrated in Figure 2.9, the longitudinal aerodynamic force can be decomposed into two causes: the shear force in the flow direction due to the viscosity of the air, also known as drag D ; and the normal force due to the pressure difference on its surface, known as lift L , both acting through a point called center of pressure. However, because the position of this point changes over the chord line, when the angle of attack varies, its analysis is chosen to be done at a geometrically fixed point, called the Aerodynamic Center AC (which is not the same as the CG). The translation of the lift and drag forces into this point produces a pitching moment m in the y_B axis.

Therefore, each segment i , evaluated using $V_{a,i}$ on its surface S_i , creates its own lift force L_i , drag force D_i , and pitching moment m_i about its aerodynamic center, from the following equations:

$$\begin{aligned}
 L_i &= \frac{1}{2} \rho V_{a,i}^2 S_i C_L(\alpha_i, \delta_i) \\
 D_i &= \frac{1}{2} \rho V_{a,i}^2 S_i C_D(\alpha_i, \delta_i) \\
 m_i &= \frac{1}{2} \rho V_{a,i}^2 S_i \bar{c}_i C_m(\alpha_i, \delta_i)
 \end{aligned} \tag{2.33}$$

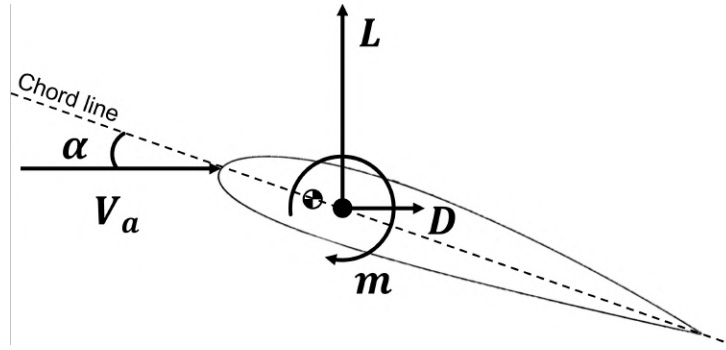


Figure 2.9: Longitudinal aerodynamics forces and moments, adapted from [18].

where lift coefficient C_L , drag coefficient C_D , and pitching moment coefficient C_m are functions of the local angle of attack α_i and elevon deflection δ_i , which are described later.

Since the aerodynamic center of the segments is not the same as the center of gravity CG of the aircraft, the segment's aerodynamic forces and moments have to be translated to this point. Although the lift force L_i , the drag force D_i , and the pitching moment m_i all belong to the longitudinal motion, because they may differ from the left and right side, lateral moments such as rolling moment l_i and yawing moment n_i will be generated. Their values can be determined by the following equations:

$$\begin{aligned}
 Y_i &= 0 \\
 l_i &= -L_i y_i \\
 n_i &= D_i y_i
 \end{aligned}
 \tag{2.34}$$

Full Envelope Longitudinal Aerodynamics

As referred before, the angle of attack α is the variable that has the most significant impact on the generated aerodynamic forces and moments. A tail-sitter, by definition, is an aircraft configuration that ends up operating in a full envelope of α . However, its relation with the aerodynamic coefficient is not constant through its full range and must be modeled using separated ranges. In [16], [24], and [19], the lift coefficient is separated into two main parts: the linear range for low angles of attack and the nonlinear range when the angle of attack increases and the airflow starts separating from the airfoil surfaces causing a loss of lift, also known as "stall". Although the airfoils of the tail-sitter are not flat, this linear model is developed based on a theoretical flat plate profile and the high angle of attack model is retrieved from an empirical database listing it for different flat plate aspect ratios. Naturally, the control surface deflections also modify the aerodynamic forces and moments. Their effect can be represented as an increase in the airfoil camber and a reduction in its effective chord length.

Therefore, in this work, for the $C_L(\alpha, \delta)$, $C_D(\alpha, \delta)$, and $C_m(\alpha, \delta)$, high ordered nonlinear functions which approximate very closely the full envelope shown in [16] are used. These functions are presented in Figures 2.10 and 2.11.

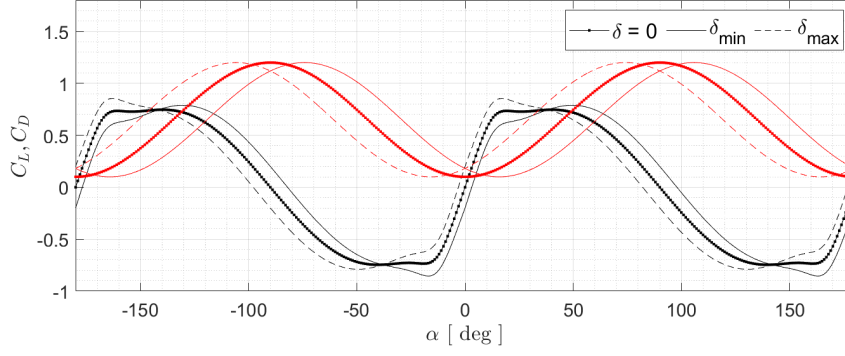


Figure 2.10: Angle of attack influence on lift coefficient (in black) and on drag coefficient (in red). Both coefficients are represented with elevons at minimal, zero and maximal deflection denoted as shown in the legend. Adapted from [16].

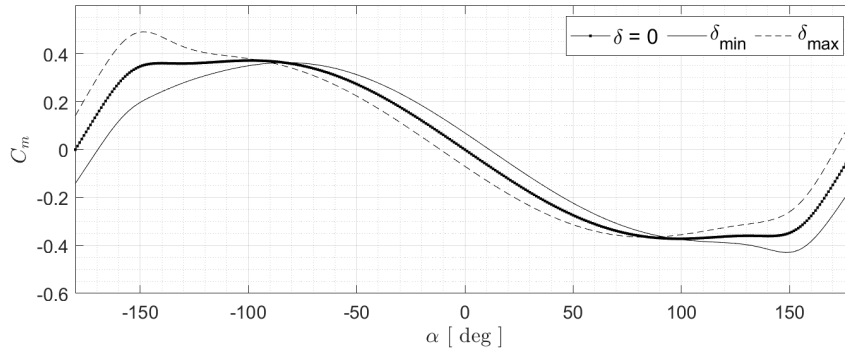


Figure 2.11: Angle of attack influence on pitching moment coefficient. All coefficients are represented with elevons at minimal, zero and maximal deflection denoted as shown in the legend. Adapted from [16].

Full Wing Aerodynamics

Until here, only longitudinal forces due to the angle of attack effect were evaluated, so the lateral side force Y_i in each segment is assumed to be zero. However, to calculate the resultant forces and moments, along with the sum of all segments' contributions, more terms can be added with respect to other variables.

$$\begin{aligned} \mathbf{F}_a &= \mathbf{R}_a \left(\sum \mathbf{F}_i + \mathbf{F}_w \right) \\ \mathbf{M}_a &= \sum \mathbf{M}_i + \mathbf{M}_w \end{aligned} \quad (2.35)$$

where $\mathbf{F}_i = [-D_i, Y_i, -L_i]$ and $\mathbf{M}_i = [l_i, m_i, n_i]$. The terms \mathbf{F}_w and \mathbf{M}_w represent the effect of the relevant state variables evaluated with respect to the full wing, as follows:

$$\mathbf{F}_w = \frac{1}{2} \rho V_a^2 S \begin{bmatrix} C_{D_w} \\ C_{Y_w} \\ C_{L_w} \end{bmatrix} \quad \text{and} \quad \mathbf{M}_w = \frac{1}{2} \rho V_a^2 S \bar{c} \begin{bmatrix} C_{l_w} \\ C_{m_w} \\ C_{n_w} \end{bmatrix} \quad (2.36)$$

where V_a is the airspeed at CG, S is the wing area, and \bar{c} is the aerodynamics mean chord. The variables considered in longitudinal aerodynamics are the derivative of the angle of attack $\dot{\alpha}$ and the pitch rate q , and in lateral aerodynamics they are the side slip angle β , the roll rate p and yaw rate r . To keep the model simple and because these variables have a minor effect in the overall forces and moments compared to the angle of attack α , the aerodynamic coefficients are considered constant and thus the wing coefficients are considered to change linearly with the variation of each state variable, see Equations 2.37. Therefore, their values, shown in Table A.6, were taken [11] corresponding to a flying wing model. As they were not measured for this tail-sitter platform, they can be assumed as approximations to add realism to the aerodynamic model.

$$\begin{aligned}
C_{D_w} &= 0 \\
C_{Y_w} &= C_{Y_\beta} \beta + \frac{b_w}{2V_a} (C_{Y_p} p + C_{Y_r} r) \\
C_{L_w} &= \frac{c_w}{2V_a} (C_{L_\alpha} \dot{\alpha} + C_{L_q} q) \\
C_{l_w} &= C_{l_\beta} \beta + \frac{b_w}{2V_a} (C_{l_p} p + C_{l_r} r) \\
C_{m_w} &= \frac{c_w}{2V_a} (C_{m_\alpha} \dot{\alpha} + C_{m_q} q) \\
C_{n_w} &= C_{n_\beta} \beta + \frac{b_w}{2V_a} (C_{n_p} p + C_{n_r} r)
\end{aligned} \tag{2.37}$$

2.5.4 Structural Segments Aerodynamics

In modeling an aircraft, in addition to the induced drag (the side effect of the production of lift by the wing), another important parcel of drag forces must be taken into account. When an object moves through the air, regardless of whether it is capable of generating lift, a parasitic drag is generated, which depends on its form, skin friction, and fluid properties. Thus, according to [25], the absolute value of the drag force produced can be modeled by:

$$D_j = \frac{1}{2} \rho V_{a,j}^2 S_j C_{D_j}(Re_j) \tag{2.38}$$

Each structural segment j has a surface S_j , a local air velocity vector $v_{a,j}$ evaluated through (2.13) and a specific drag coefficient C_{D_j} which is function of the Reynolds number Re_j .

$$C_{D,j} = 1 + 10Re_j^{-2.3} \quad \text{and} \quad Re_j = \frac{\rho V_{a,j} d_j}{\mu} \tag{2.39}$$

where $\mu = 1.81 \times 10^{-5}$ [kg/(m s)] is the dynamic viscosity of air at 15°C, and d_j is the characteristic linear dimension of the structure segment. Reminder that the air density and dynamic viscosity can be assumed to be constant since their variations in the altitude range conditions is negligible.

Thereby, the structural parts are divided into six individual segments, each with its CG located at $r_j = r_{j,G} - r_{CG}$. The first two, modeled as circular rods, represent each of the propellers' guards. Their length l_j is equal to the perimeter of a circumference with radius r_{pg} . The other four segments represent the right and left landing gear structures, modeled as two straight rods on each side. The

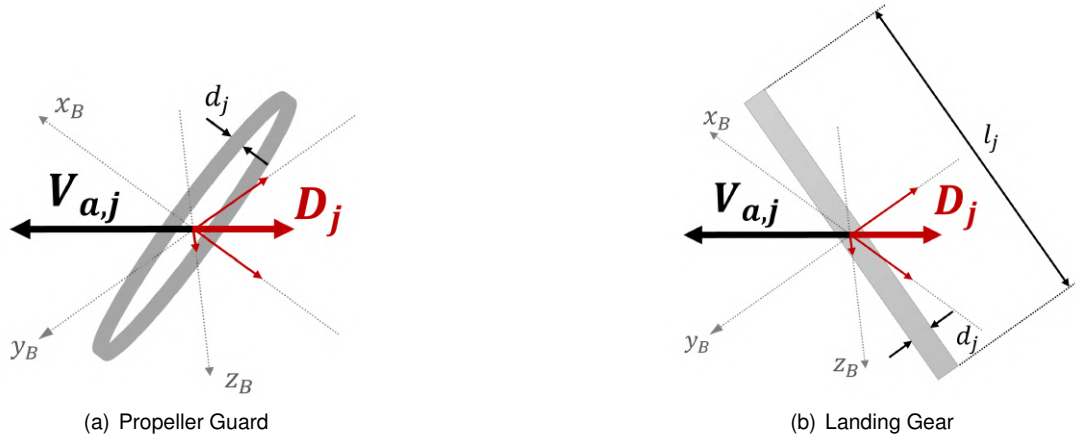


Figure 2.12: Rod segments in a wind field $V_{a,j}$ producing a drag force D_j , adapted from [16]

surface of each segment is defined by $S_j = d_j l_j$. All rod segments' parameters are described in Table 2.2.

Table 2.2: Rod Segments' Parameters

| Element | d_j | l_j | $x_{i,G}$ | $y_{i,G}$ | $z_{i,G}$ |
|---------|-------|---------------|-----------|------------|-----------|
| 1 | 0.003 | $2\pi r_{pg}$ | $x_{p,G}$ | $y_{p,G}$ | $z_{p,G}$ |
| 2 | 0.003 | $2\pi r_{pg}$ | $x_{p,G}$ | $-y_{p,G}$ | $z_{p,G}$ |
| 3 | 0.007 | 0.063 | -0.011 | -0.252 | 0.063 |
| 4 | 0.007 | 0.063 | -0.011 | -0.252 | -0.063 |
| 5 | 0.007 | 0.063 | -0.011 | 0.252 | 0.063 |
| 6 | 0.007 | 0.063 | -0.011 | 0.252 | -0.063 |

Having the same orientation but opposite direction of the local air velocity, the absolute value D_j is then multiplied by the local air velocity unit vector in the body frame \mathbb{F}_B to represent all rod segment forces in the same coordinate frame orientation. Finally, these forces are translated to the center of gravity and summed as expressed in the following equations:

$$\mathbf{F}_r = \sum D_j \mathbf{u}_j, \quad \mathbf{M}_r = \sum \mathbf{r}_j \times D_j \mathbf{u}_j \quad (2.40)$$

2.5.5 Ground Contact Dynamics

To simulate a full flight mission, the ground contact dynamics during take off and landing must also be taken into account. To model the contact forces, a spring-damper analogy is used in four strategic selected exterior points to detect the contact with the ground. Since in a tail-sitter configuration the takeoff and landing maneuvers are performed in the vertical position, pointing upwards, the selected points are located in the lower and outer parts of the lateral structures as shown in Figure 2.13.

Considering the inertial frame \mathbb{F}_I and using the spring-damper analogy, the spring term acts on the vertical z_I axis proportional to the displacement d_k of the point k penetrating the ground. On the other

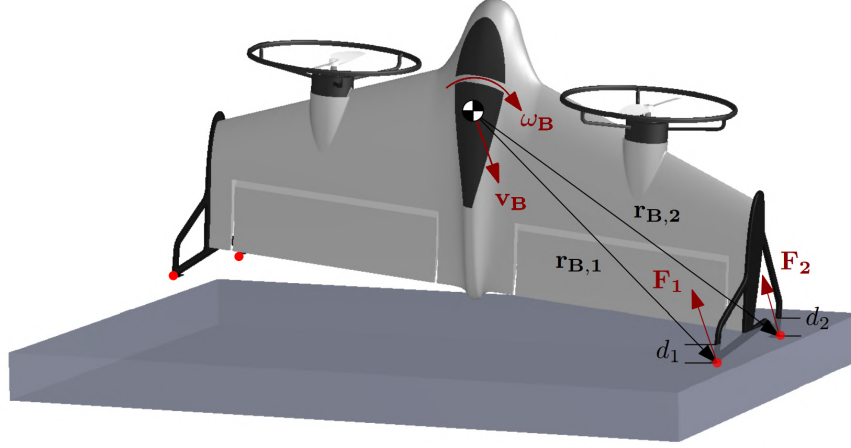


Figure 2.13: Ground Contact Forces and the four selected point in red. Adapted from [16].

hand, the damper term is proportional to the point's inertial velocity $v_{I,k}$ and can act on the three axis directions depending on its velocity vector direction. These three values are calculated as follows:

$$\begin{aligned} d_k &= -z_{I,k} \\ \mathbf{r}_{I,k} &= \mathbf{R}(\mathbf{q}) \mathbf{r}_k + \mathbf{p} \\ \mathbf{v}_{I,k} &= \mathbf{R}(\mathbf{q})(\mathbf{v}_B + \boldsymbol{\omega}_B \times \mathbf{r}_{I,k}) \end{aligned} \quad (2.41)$$

where $\mathbf{r}_k = \mathbf{r}_{k,G} - \mathbf{r}_{CG}$ and $\mathbf{p} = [p_N, p_E, p_D]^T$ is the three state position of the CG in the inertial frame \mathbb{F}_I . The location of the points in the geometric frame \mathbb{F}_G is presented in Table A.7.

Consequently, the contact force $F_{I,k}$ produced at each respective point location is modeled as:

$$\mathbf{F}_{I,k} = \begin{bmatrix} 0 \\ 0 \\ -m k_p d_k \end{bmatrix} - m k_v \mathbf{v}_{I,k} \quad (2.42)$$

In this model, $k_p = 400 \text{ s}^{-2}$ and $k_v = 3.6 \text{ s}^{-2}$ are used as the spring and damper constants, respectively, which provide a realistic stopping time when the aircraft reaches the ground. Note that, as a ground contact force, it should only appear when d_k is lower than zero.

Finally, these inertial forces are then translated into the body frame \mathbb{F}_B and summed to generate the total contact force \mathbf{F}_c and contact moment \mathbf{M}_c :

$$\mathbf{F}_c = \sum \mathbf{R}(\mathbf{q})^T \mathbf{F}_{I,k}, \quad \mathbf{M}_c = \sum \mathbf{r}_k \times \mathbf{R}(\mathbf{q})^T \mathbf{F}_{I,k} \quad (2.43)$$

2.6 Aircraft Simulator

The objective of this entire chapter was to develop and implement a nonlinear model that would approximately describe the aircraft dynamics. These dynamics depend on the actual state variables and

are influenced by the system's actuators, and can also be perturbed by some unpredicted disturbances such as wind conditions.

The actual state can be represented by a full state vector X of 13 variables:

$$X = [u, v, w, p, q, r, p_N, p_E, p_D, q_0, q_1, q_2, q_3]^T \quad (2.44)$$

The actuators are represented by the input vector U of 4 variables:

$$U = [\delta_l, \delta_r, \tau_l, \tau_r]^T \quad (2.45)$$

The wind effects are represented by the disturbances vector D of the 3 input variables, in the in the inertial frame \mathbb{F}_I :

$$D = [w_N, w_E, w_D]^T \quad (2.46)$$

To easily read the state variables, the quaternion attitude was first converted to Euler angles and the output of the implemented model is represented by the output vector Y of 12 variables:

$$Y = [u, v, w, p, q, r, p_N, p_E, p_D, \phi, \theta, \psi]^T \quad (2.47)$$

That said, the dynamic equations that describe the system can be described as follows:

$$\dot{X} = f(X, U, D) \quad (2.48)$$

To close the dynamic system loop, the recursive information about the states variables X is given by the integration of the state derivatives \dot{X} resulting from the dynamic system equation (2.48).

Finally, after the aircraft model implementation, to better comprehend how the platform is behaving and moving, a 3-D graphic simulator in *Simulink* was created. See Figure 2.14.

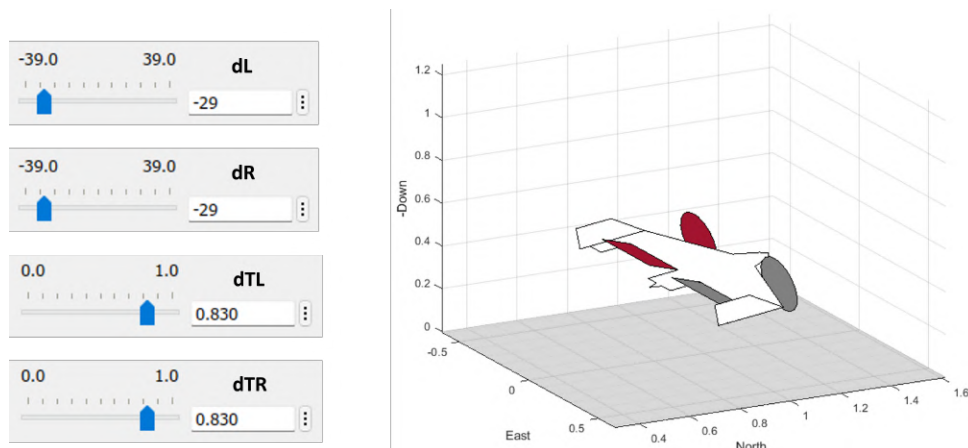


Figure 2.14: Aircraft Simulator

Chapter 3

Longitudinal Model Characterization

In 2, the tail-sitter simulator was presented based on a nonlinear model capable of performing both longitudinal and lateral maneuvers. The purpose of the present chapter is to understand and characterize its dynamics. In the study of dynamical systems, linearization is a powerful tool and is used as a method for assessing the local stability of an equilibrium point of a nonlinear differential equations system. However, before determining the type of analysis and flight limits of interest, the two minimum flight missions proposed for this work as control objectives should be explained and defined:

- First, an exclusively **vertical flight mission**, where the controller must guarantee stabilization during take-off (vertical climb), hover (maintaining vertical orientation with zero climb velocity), move forward and backward (without making use of aerodynamic flight), and finally landing (vertical descent until the aircraft touches the ground). See Figure 3.1.

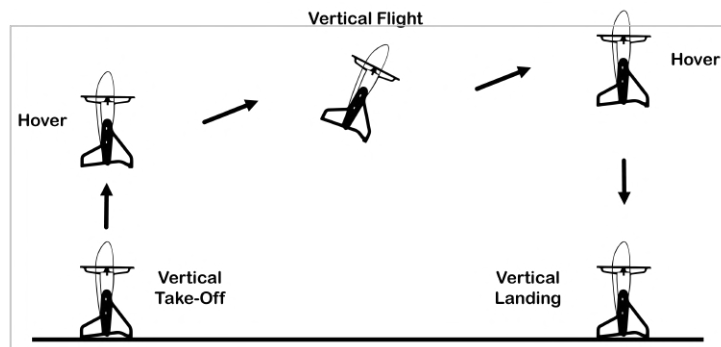


Figure 3.1: Minimum Vertical Flight Mission

- Second, a minimum **horizontal flight mission**, which, in addition to the required maneuvers for the first mission, must be able to transition from hover to horizontal flight, perform stable horizontal flight, and then transition back to hover. See Figure 3.2.

That said, this chapter is divided into four sections. During the first section, the flight envelope is defined for the simulation, i.e. the limits where the aircraft is capable of finding equilibrium points. Then, in the second section, the implementation of the trimming function to determine the equilibrium points is explained. In the third section, the linearization method is presented. Finally, during the fourth section, the linear systems are analyzed and used to characterize the aircraft dynamics which is the starting

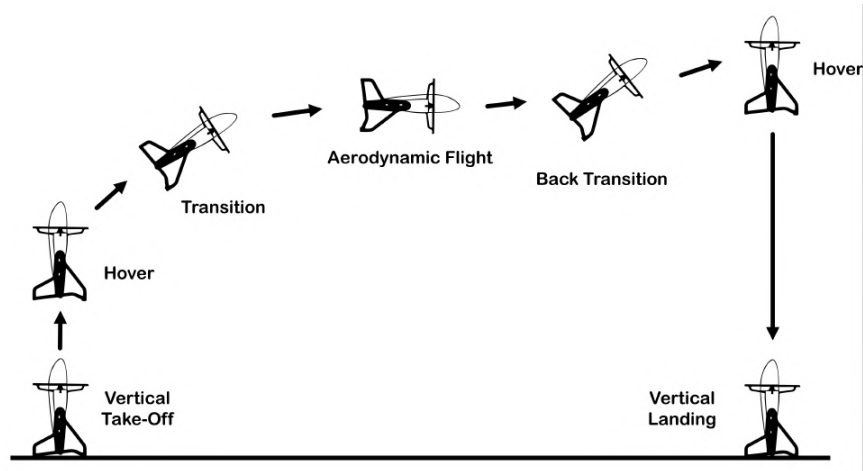


Figure 3.2: Minimum Horizontal Flight Mission

point for the development of the stabilization controller.

Note that, from now on, the conventional representation of the actuators $\mathbf{U} = [\delta_E, \delta_A, \delta_R, \delta_T]^T$ is chosen over the direct representation because its more intuitive way helps the understanding of the behaviour and comparison with conventional aircraft analysis.

3.1 Flight Envelope

To control a flight system it is really important to be aware of and understand the limits of its operation. Usually, a flight envelope graphically refers to the capabilities of the aircraft in terms of airspeed and load factor or airspeed and atmospheric density. On the one hand, the load factor awareness helps to keep the aircraft in its structural safety margins during maneuvers. On the other hand, the atmospheric density control is important to set the maximum altitude where the aircraft is able to generate enough lift to perform a level controlled flight. However, because in this work study no large altitude variations nor structurally aggressive maneuvers are intended, only the airspeed limits will be defined according to the two main flight orientations: vertical and horizontal.

Vertical Flight

For the vertical flight, i.e. nose pointing up, there are three different movements that can be performed. First, the aircraft can stay still in the air, also known as hover. Second, the aircraft can move in the vertical axis, both climbing and descending. And lastly, the aircraft can move forward and backwards, by slightly pitching over the y_B axis.

For each flight condition, the maximum and minimum velocity could be approximately deduced by understanding the aircraft dynamic limits. The Figure 3.3 illustrates the forces applied on the aircraft and helps explaining the cause of each limit.

- About the **hover**, since its definition implies zero velocity, there is no limit to be set. However, if a wind model is introduced in future works, the maximum wind velocity to perform a safe hover (which will differ depending on the wind direction) should be calculated.
- The **maximum climb velocity** V_{max} is limited by the propeller maximum thrust. Subtracting the

aircraft weight the the maximum thrust, the maximum drag that the platform can overcome is obtained. Then, inverting the drag formula and having the aircraft total drag coefficient, the approximated maximum velocity could be found.

$$D_{max} = T_{max} - mg \quad \rightarrow \quad D_{max} = \frac{1}{2} \rho S C_D V_{max}^2 \quad \rightarrow \quad V_{max} \quad (3.1)$$

- The **maximum descent velocity** V_{min} could be limited by the minimum slipstream velocity $V_{wake, min}$ over the elevons to guarantee their authority and the aircraft stabilization and control. $V_{wake, min}$, considering fully developed wake and that the aircraft is descending, can be calculated as two times the induced velocity for the produced thrust force minus the absolute value of the aircraft velocity vector. In this case, the thrust is the minimum value that summed with the drag at this descent velocity equals the aircraft weight. To calculate this, a numerical convergent function should be implemented. Based on [16], the value of $V_{wake, min} = 10$ m/s could be assumed. Additionally, from Momentum Theory [21], it is known that the instability of the aircraft increases with the increase of the descent velocity due to turbulence around the rotor and the aircraft body. Therefore, even if the elevons authority is guaranteed, and reasonable limit should be imposed.

$$V_{min} \quad \rightarrow \quad T = mg - D(V_{min}) \quad \rightarrow \quad V_{wake, min} = 2 w_{ind}(T) - V_{min} \quad (3.2)$$

- The **maximum forward velocity** V_{max} is limited by the maximum angle of inclination allowed from the nose up position ($\theta = \pi/2$). Since the aerodynamic drag force is too big is during this movement, to maintain the altitude, the maximum forward velocity would be too slow. Therefore, in order to achieve a reasonable forward velocity the thrust must be increased and the aircraft will likely start climbing. The approximated maximum deviation achieved during simulation without destabilization was about 25° . From the forces equilibrium, a numerical convergent function could be implement to estimate the maximum forward velocity.

$$\begin{cases} mg = T \sin(\alpha) + L \\ T \cos(\alpha) = D \end{cases} \quad (3.3)$$

where T , L , and D are functions of V_{max} and α .

- The **maximum backwards velocity** V_{min} is limited by the same assumptions that the maximum forward velocity. Therefore, since the wing aerodynamics are based on a flat plate module, the wing dynamics do not depend on the way that the aircraft is moving in the longitudinal plane, and the absolute value of the forward and backwards velocity is the same.

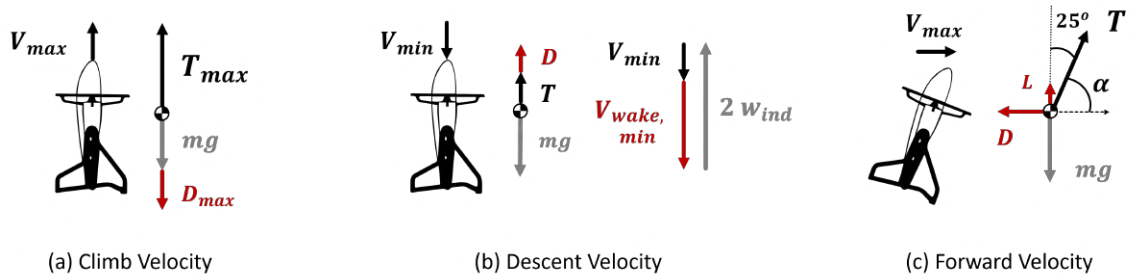


Figure 3.3: Vertical Flight Forces Diagrams

Horizontal Flight

For the horizontal flight, i.e. level aerodynamic flight where the force that counteracts the aircraft weight should rely mainly on the wing lift, there are only two velocity limits to be calculated: the maximum level velocity and the minimum level velocity. The Figure 3.4 illustrates the forces applied on the aircraft and helps understanding the cause of each limit.

- The **maximum level velocity** V_{max} , similarly to the maximum climb velocity, this is limited by the propeller maximum thrust. In this case, the weight is counterbalanced by the lift force, and the maximum thrust should equal the total drag at the maximum velocity. Since both drag and thrust depend on the velocity, a numerical function could be implemented to find this value.

$$V_{max} \rightarrow L = \frac{1}{2} \rho S C_L V_{max}^2 = mg \rightarrow C_L(\alpha) \rightarrow \alpha$$

$$D_{max}(V_{max}, \alpha) = T_{max}(V_{max}, \alpha) \quad (3.4)$$

- The **minimum level velocity** V_{min} is limited by the stall velocity at the maximum elevon deflection. However, because the stall angle of attack is quite high, the thrust force has a vertical part which will also contribute to counteracts the aircraft weight. Therefore, the calculus to find this speed limit are not so linear and, again, a numerical function could be implemented.

$$\begin{cases} mg = T \sin(\alpha) + L \\ T \cos(\alpha) = D \end{cases} \quad (3.5)$$

where T , L , and D are functions of V_{min} and α .

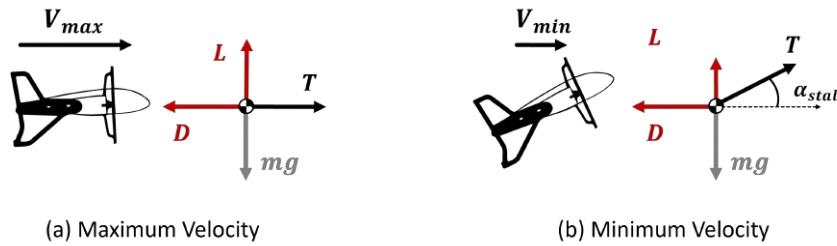


Figure 3.4: Horizontal Flight Forces Diagrams

Flight Envelope Results

Since these theoretical limits would only be raw approximations and would likely diverge from the real limits, instead of using theoretical limits, the trimming algorithm (explained in the next section) is expected to reveal by itself where the aircraft can or cannot find feasible equilibrium points. Therefore, for both vertical and horizontal limit approximation, a large range of velocity values were tried to trim and then validated in the aircraft simulator.

For the vertical flight, the trim was tested for $u_0 = [-5, 5]$ m/s, see Figure 3.5. Inspecting the results, the maximum climb velocity is limited by the thrust signal that saturates approximately at $V_{max} = 3$ m/s, which was expected in the previous theoretical analysis. However, because the thrust and aircraft aerodynamics do not account for the slipstream turbulence, the descent velocity is not limited in the

simulation. In order to consider these effects, the maximum descent velocity is arbitrary imposed as the symmetric value of the maximum climb velocity, i.e. $V_{min} = -3$ m/s.

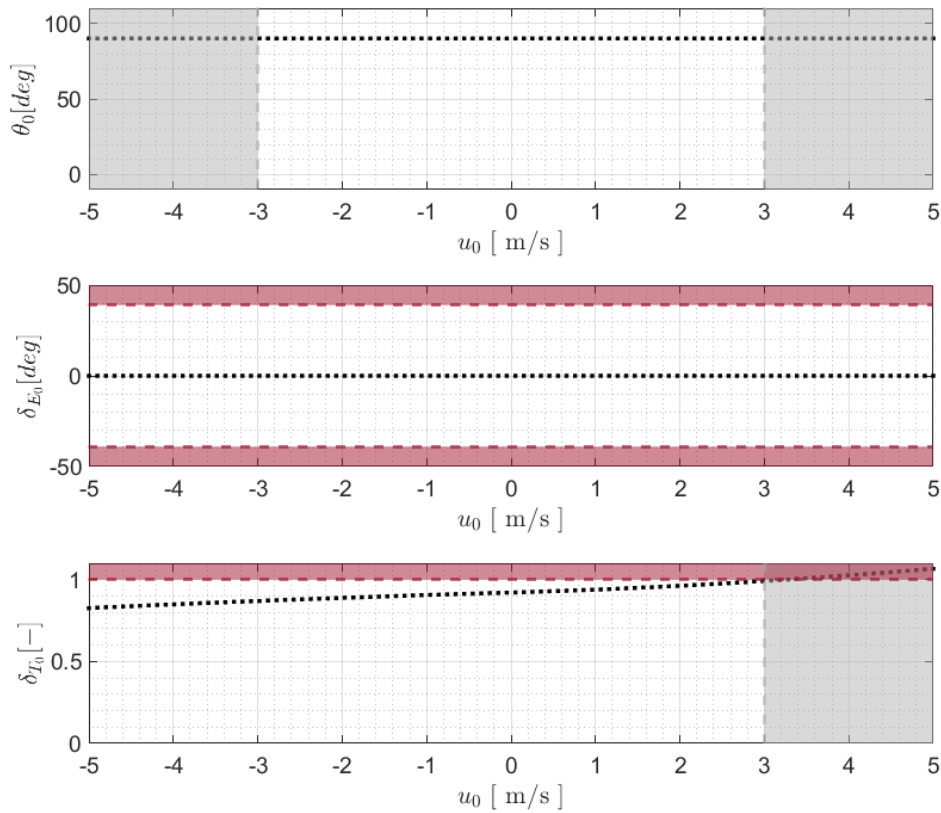


Figure 3.5: The first graphic shows the vertical flight envelope $u - \theta$ (the trim values are represented by the black dots and the flight envelope limits by the grey dashed lines). The second and third graphics show the actuators δ_{E_0} and δ_{T_0} trim values. When the actuators trim value (represented by the black dots) overcome the actuators' limits (represented by the red dashed lines), the flight envelope limits are defined.

In the same flight orientation, the aircraft trim was also tested for forward and backwards movement, however it was not possible to achieve reasonable velocities maintaining the aircraft altitude. Besides the aircraft instability during these movements, they were later tested with the developed flight controller and the following backwards and forward velocities (velocities in the north direction of the inertial frame \mathbb{F}_I , dp_N/dt) were achieved: $V_{min} = -0.6$ m/s.

For the horizontal flight, the trim was tested for $u_0 = [-7, 12]$ m/s, see Figure 3.6. Inspecting the results, the maximum level velocity is limited by the thrust signal that saturates at $V_{max} = 11$ m/s, and the minimum level velocity is limited by the elevons signal that saturates at $V_{min} = 8$ m/s.

3.2 Aircraft Trimming

The theory behind the linearization process is finding a linear approximation of a nonlinear system which is valid in a small region around an operating point. In aircraft stability and control analysis trim

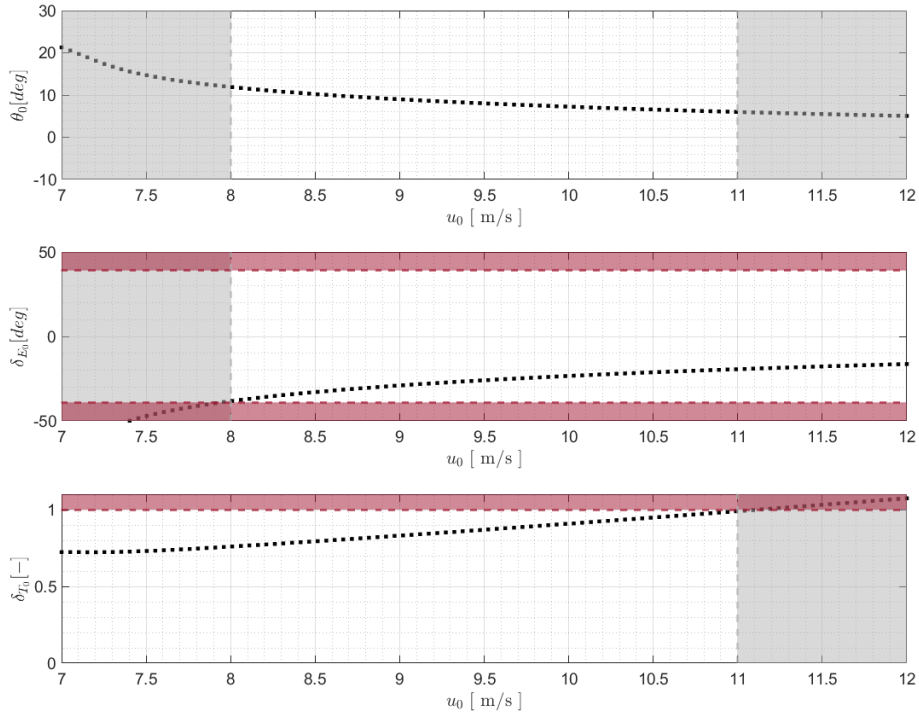


Figure 3.6: The first graphic shows the horizontal flight envelope $u - \theta$ (the trim values are represented by the black dots and the flight envelope limits by the grey dashed lines). The second and third graphics show the actuators δ_{E_0} and δ_{T_0} trim values. When the actuators trim value (represented by the black dots) overcome the actuators' limits (represented by the red dashed lines), the flight envelope limits are defined.

points, also known as equilibrium points, are usually chosen as points of interest.

Mathematically, a trim point is a point where the system's state derivatives equal zero, which in this system model means to assure that the linear and angular accelerations are null and that the inputs provide an equilibrium point for the aircraft. Formally:

$$\dot{\mathbf{X}} = f(\mathbf{X}, \mathbf{U}, \mathbf{D}) = 0 \quad (3.6)$$

However, the solution of simultaneous nonlinear equations is not easy to find and, when the dynamic systems are more complex, it can often be found only by numerical methods. The method chosen to find the trim points is known as cost function minimization, where the cost function can be defined as:

$$\mathcal{J} = \dot{u}^2 + \dot{v}^2 + \dot{w}^2 + 10\dot{p}^2 + 10\dot{q}^2 + 10\dot{r}^2 \quad (3.7)$$

To ensure that the cost function's final value accounts for the different tolerances between the linear and angular accelerations, note that each angular acceleration is multiplied by a weight factor of 10 since they are represented in radians. Ideally, at trim, \mathcal{J} should be zero.

Constraints

Beyond the cost function definition, depending on the desired flight conditions, the constraints that the scalar function $f(\mathbf{X}, \mathbf{U}, \mathbf{D})$ will be subject to must be set. The first two desired trim points for this work are the trim for straight level flight and the trim for hover flight. Therefore, the constraints for each of these flights are:

$u = u_d$, where u_d is the desired longitudinal velocity which must be within the flight envelope for the longitudinal flight or, in case of hover, must equal zero;

$\theta = \theta_d$, where θ_d should be equal to α in case of longitudinal level flight or 90° in case of hover flight;

$v = 0$, since no lateral velocity is desired;

$q_1 = q_3 = 0$, to ensure an exclusively longitudinal orientation;

$q_0^2 + q_2^2 = 1$, the quaternion constraint;

$p = q = r = 0$, to guarantee steady flight;

$\delta_l = \delta_r$, which means $\delta_A = 0$ and;

$\tau_l = \tau_r$, which means $\delta_R = 0$, since these cases only require longitudinal equilibrium and any differential deflection would destabilize the lateral equilibrium;

where the angle of attack α (only in level flight), the pitch angle θ (which directly depends on q_0 and q_2), the left and right elevon deflections and the left and right propeller signals (which can be represented by δ_E and δ_T , respectively) are the free variables that will ensure the equilibrium between all the modeled forces and moments in order to obtain null accelerations.

With all the constraints defined, it should now be possible to start searching for the minimization problem solution. However, because multiple feasible trim solutions for a given trim condition may exist, it is first necessary to provide an initial state approximation to the algorithm. Different guesses may lead to different solutions and if they are not precise enough the trim value can be quite poor. The reason is that gradient descent based nonlinear optimization methods often need a good initial guess for the cost function to converge to a reasonable value.

Vertical Flight Initial Approximation

For the hover flight initial point, because the aircraft is pointing up, the attitude initial condition is equal to the Euler to quaternion conversion of $\pi/2$ radians in the pitch angle. Additionally, since the aircraft should be still in the air, which means zero airspeed, the aerodynamic forces are assumed to be zero. Therefore, the only forces applied in the aircraft are its weight and the thrust force in the opposite direction. Knowing that, each propeller should sustain half of the aircraft weight and by inverting the propeller model, one can obtain the initial thrust signal. Thus, if propellers are perfectly aligned with the

CG plane, no moments are expected and both elevon deflections can be guessed as zero. Resuming, the full state and control initial vectors for hover flight are:

$$\mathbf{X}_0 = [0, 0, 0, 0, 0, 0, -, -, -, q_{0_0}, 0, q_{2_0}, 0]^T \quad (3.8)$$

$$\mathbf{U}_0 = [\delta_{E_0}, 0, 0, \delta_{T_0}]^T$$

Note that position states are not relevant. They only depend on where the aircraft is located when the simulation is initialized.

Horizontal Flight Initial Approximation

The level flight initial point vector is more complex to calculate since aerodynamic forces and moments are included. In aerodynamic flight, the propellers are responsible for the aircraft velocity which will then, depending on the angle of attack, generate the necessary lift to equalize the aircraft weight. Assuming that during this flight mode the wing remains in the linear regime, the lift equation can be inverted to obtain the initial angle of attack.

$$L_0 = m g \quad \rightarrow \quad \alpha_0 = \frac{L_0}{\frac{1}{2} \rho u_0^2 S C_{L_\alpha}} \quad \rightarrow \quad w_0 = u_0 \tan(\alpha_0) \quad (3.9)$$

where $u_0 = u_d$ is the desired trim level flight velocity, and C_{L_α} is the lift coefficient slope in the linear range. The velocity vertical component w_0 is determined by inverting (2.12).

Then, considering that the pitching moment is also both linearly influenced by the angle of attack and the elevon deflection, and that it should be zero to maintain equilibrium, an initial approximation of the control deflection can be calculated.

$$C_{m_0} = C_{m_\alpha} \alpha_0 + C_{m_{\delta_E}} \delta_{E_0} = 0 \quad \rightarrow \quad \delta_{E_0} = -\frac{C_{m_\alpha} \alpha_0}{C_{m_{\delta_E}}} \quad (3.10)$$

where the values of C_{m_α} and $C_{m_{\delta_E}}$ are interpolated from $C_m(\alpha, \delta)$ at α_0 .

To estimate the thrust input signals, besides of the importance of structural segments drag, this one is neglected and only the wing aerodynamic drag is considered to simplify the problem. Therefore, knowing α_0 and δ_{E_0} , the thrust required is given by the drag formula in (2.33) dividing by two, to keep both propellers symmetric. Then, once again, the propeller model is inverted and the thrust input signal guess is calculated.

$$D_0 = \frac{1}{2} \rho u_0^2 S C_{D(\alpha_0, \delta_{E_0})} \quad \rightarrow \quad T_0 = \frac{D_0}{2} \quad \rightarrow \quad \delta_{T_0}(T_0, u_0, \alpha_0) \quad (3.11)$$

To conclude, in order to maintain the altitude during level flight, the pitch angle should equal the angle of attack. Therefore, the attitude initial guess can be calculated using the Euler to quaternion conversion of α_0 radians in the pitch angle. The full state and control initial vectors for level flight are:

$$\mathbf{X}_0 = [u_0, 0, w_0, 0, 0, 0, -, -, -, q_{0_0}, 0, q_{2_0}, 0]^T \quad (3.12)$$

$$U_0 = [\delta_{E_0}, 0, 0, \delta_{T_0}]^T$$

Constrained Nonlinear Optimization Algorithm

The implementation of cost function minimization trimming method in *Matlab* is done by executing the *fmincon* command. This command starts from an initial point $Z_0 = [X_0; U_0]^T$ and searches, using a sequential quadratic programming algorithm (identified as *sqp - legacy* in its options), until it finds a minimizer point $Z_e = [X_e; U_e]^T$ of the cost function J that respects the defined constraints.

Besides the local minimum vector solution, it also returns the value of the cost function J . Since this is a numerical based model, it is expected that the equilibrium point found do not necessarily reduce the state variables to zero, but instead to an acceptable tolerance. For each considered state derivatives for minimization, which are the linear and angular accelerations, a value below 10^{-5} can be considered zero. The difference in the units between the linear and angular accelerations was already accounted by applying a weight factor of 10 in the cost function J . Thus, summing the six variables, the maximum final value should be less than 6×10^{-5} . If J satisfies the maximum tolerance, the solution is chosen as the trim point. If not, the *fmincon* command should be ran in loop, using the remaining trial points Z_e as the new guesses until one fits the limit value. This procedure is illustrated in Figure 3.7.

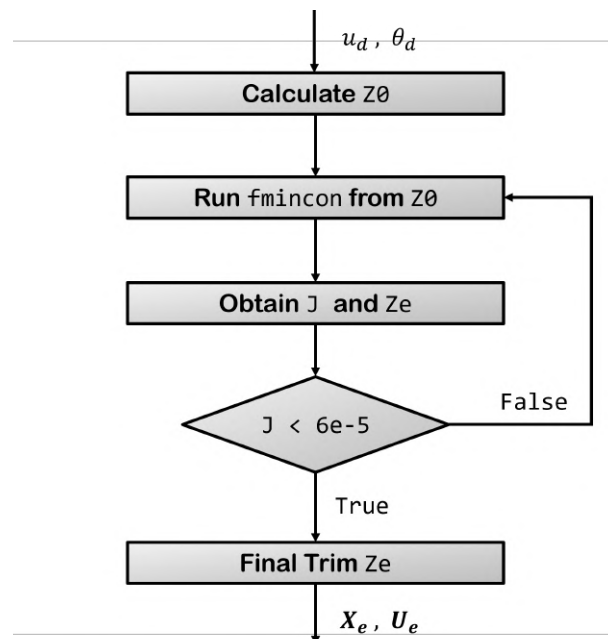


Figure 3.7: Trimming Algorithm

To avoid the loop getting stuck on a local minimum without satisfying the maximum tolerance, the number of loops must be limited, and different initial conditions should be tested. Reasonable trim results were found with the following *fmincon* options:

- Termination step tolerance (TolX): 1×10^{-15}
- Maximum number of function evaluations (MaxFunEvals): 1×10^{30}
- Maximum number of iterations (MaxIter): 1×10^5

Trim Results

The trim implemented function was tested for high range of initial vertical and horizontal velocities. Then, in order to validate the results, they were tested in the simulator presented in Section 2.6. These tests helped to design the flight envelop.

For analysis and control design purposes, the selected operating points for each flight mode were $u_0 = 0$ m/s (hover) for the vertical flight, and a low velocity level flight operating point $u_0 = 9$ m/s for the horizontal flight. These two lower velocities conditions for each flight mode were chosen because they both require additional control effort when comparing with higher velocity flights.

The transition flights between these two flight operating modes do not have an equilibrium point in the same sense as the level or hover flight but linearization around some flight condition is required in order to obtain the transition controller, as it will be explained in further sections. Therefore, based on [10], the chosen operating point is in vertical orientation as hover flight but with some climb velocity. Different values within the vertical flight envelope were tried, however observing that the differences in the simulations results were not critical, $u_0 = 1$ m/s was chosen. The guess values used for this flight condition are the same as the ones used to the hover flight, except the longitudinal velocity u_0 , as the trim results will be almost the same.

The guess and trim values for hover, transition and level flight are presented in Tables 3.1, 3.2 and 3.3.

Table 3.1: Hover flight initial guess and trim

| | q_0 (-) | q_2 (-) | δ_E (rad) | δ_T (-) |
|--------------------|-----------|-----------|------------------|----------------|
| Guess Z_0 | 0.7071 | 0.7071 | 0.0000 | 0.9184 |
| Trim Z_e | 0.7071 | 0.7071 | 0.0000 | 0.9219 |

Table 3.2: Transition flight initial guess and trim

| | u (m/s) | q_0 (-) | q_2 (-) | δ_E (rad) | δ_T (-) |
|--------------------|-----------|-----------|-----------|------------------|----------------|
| Guess Z_0 | 1.0000 | 0.7071 | 0.7071 | 0.0000 | 0.9184 |
| Trim Z_e | 1.0000 | 0.7071 | 0.7071 | 0.0000 | 0.9375 |

Table 3.3: Level flight (at 9 m/s) initial guess and trim

| | u (m/s) | w (m/s) | q_0 (-) | q_2 (-) | δ_E (rad) | δ_T (-) |
|--------------------|-----------|-----------|-----------|-----------|------------------|----------------|
| Guess Z_0 | 9.0000 | 1.3034 | 0.9974 | 0.0723 | -28.840 | 0.7129 |
| Trim Z_e | 8.9999 | 1.4239 | 0.9969 | 0.0784 | -29.019 | 0.8304 |

3.3 Linearization

Linearization is the process of finding a linear approximation of a nonlinear system with respect to all variables that is valid in a small region around an operating point. The solution of the dynamic function (2.48), that describes the system, for a specific point is called the particular solution. The approximated linear model can be found based on a first order Taylor expansion about this point of interest:

$$\dot{\mathbf{X}} \approx \mathbf{f}(\mathbf{X}_0, \mathbf{U}_0) + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{X}} \right|_{\mathbf{X}=\mathbf{X}_0} (\mathbf{X} - \mathbf{X}_0) + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{U}} \right|_{\mathbf{U}=\mathbf{U}_0} (\mathbf{U} - \mathbf{U}_0) \quad (3.13)$$

where $\mathbf{f}(\mathbf{X}_0, \mathbf{U}_0)$ is the particular solution and the other two terms are the first derivative of the solution with respect to each state and input variable, respectively, at the particular solution multiplied by the deviation of that variable (i.e. the real value of the variable minus its value on the operating point). To represent the linearized model, new variables centered about the operating point are defined:

$$\begin{aligned} \tilde{\mathbf{X}} &= \mathbf{X} - \mathbf{X}_0 \\ \tilde{\mathbf{U}} &= \mathbf{U} - \mathbf{U}_0 \end{aligned} \quad (3.14)$$

The state space representation for linearized model in terms of these deviations can be obtain from the following expression:

$$\dot{\tilde{\mathbf{X}}} = \mathbf{A}\tilde{\mathbf{X}} + \mathbf{B}\tilde{\mathbf{U}} \quad (3.15)$$

where \mathbf{A} and \mathbf{B} represent the derivative of each state with respect to all state and input variables, i.e. the system matrix and the input matrix, respectively. Nonetheless, this method is based in an analytical study of the dynamic model, which in a complex nonlinear model is hard to solve. To overcome this problem, a small disturbance method is implemented. This consists of disturbing each state or input variable at its equilibrium value and applying it to the dynamic system, so that the derivative between these two points can be estimated. That said, the elements of the \mathbf{A} and \mathbf{B} matrices can be expressed as:

$$A_{ij} = \frac{\partial f_i}{\partial X_j} \approx \frac{f_i(X_{j,0} + \Delta X_j) - f_i(X_{j,0})}{X_{j,0} + \Delta X_j} \quad (3.16)$$

$$B_{ij} = \frac{\partial f_i}{\partial U_j} \approx \frac{f_i(U_{j,0} + \Delta U_j) - f_i(U_{j,0})}{U_{j,0} + \Delta U_j}$$

where f_i is the dynamic solution of state i , and ΔX_j is the perturbation of the state or input j . Each element of \mathbf{A} and \mathbf{B} matrices relate how the current state or input j affects the change of state i . If the state change is not dependent on the current state or input its values will be zero.

The dimension of each state or input perturbation ΔX_j depends on its units and must guarantee that the system remains in the linear range about the equilibrium point, and so, it may also vary depending on the selected point. That said, the value of the perturbation for each variable is 10^{-3} , in SI units, for the level flight equilibrium point, and 10^{-4} for the hover flight, an order of magnitude below, since this flight mode has a smaller linear region. Besides, a particular attention is given to the attitude states. Since

a linear increase in a quaternion state does not represent a linear deviation in space, two coordinate transformations need to be employed. First, the equilibrium quaternion states are converted to the Euler equilibrium angles, using (B.3), and the specific perturbation, in radians, is applied. Then, the perturbed Euler states are back converted to quaternion representation, using (B.2), and the state derivatives are obtained.

Additionally, note that the attitude state derivatives resulting from the system dynamics are also in a quaternion form. However, in order to obtain a linearized set of Euler angles in the A and B matrices, the quaternion derivatives need to be converted to Euler angles derivatives. The benefit of Euler angles is that the resulting mathematical model is very easy to decouple since each angle only describes movement on a single plane. The quaternion derivatives to Euler derivatives conversion is explained in Appendix B.

In aeronautics, when the matrices are written in the Euler form, if the lateral variables do not affect the longitudinal ones, and vice versa, i.e. the correspondent matrix element is zero or close to zero, its analysis is typically decoupled into two separate models: the longitudinal model, which expresses the aircraft characteristics about the vertical and longitudinal motion, and the lateral model, which expresses the aircraft characteristics about the lateral motion. Since this work is focused on the development of a longitudinal controller, and the models decoupling is feasible, only the states and input variables of interest are analysed. Therefore, the longitudinal variables are:

$$\mathbf{X}_{lon} = [u, w, q, \theta, h]^T \quad \mathbf{U}_{lon} = [\delta_E, \delta_T]^T \quad (3.17)$$

where $h = -p_D$ represents the altitude. Although p_N is also a longitudinal state, this state is not relevant for the model characterization.

3.4 Longitudinal Analysis

In this section the longitudinal linear model is assessed. The full flight envelope will be analysed, in a matter of stability, controllability and observability, through a particular analysis of the three selected operating points (hover, transition and level flight). The system and control matrices for each of these points are presented in the following section.

3.4.1 Stability

Stability, as one of the most important performance specification of a control system, can be described as what does the system do as the time goes to infinity [26]. A system is considered unstable if, when perturbed from its equilibrium point, it does not return to its initial condition.

In a multi state system, if at least one of its state variables is unstable, the full system itself is said to be unstable. In continuous time control theory, this means that at least one of the eigenvalues of the state space A matrix has real part greater than zero, or that there are more than one pole at the origin.

For the stability analysis two operating points for each flight orientation were object of study in order

to understand the system stability evolution with the change of velocity within the flight envelope.

Vertical Flight

First, for the vertical flight, the hover operating point is analysed. The state-space representation about this equilibrium point is:

$$\dot{\tilde{\mathbf{X}}}_{lon} = \underbrace{\begin{bmatrix} -0.4011 & 0.0001 & 0 & 0.0049 & 0.0009 \\ 0 & -315.14 & -6.3485 & -9.7996 & 0 \\ 0 & -6.9281 & 0.1340 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -1.0000 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A}_{hover}} \tilde{\mathbf{X}}_{lon} + \underbrace{\begin{bmatrix} 0 & 21.2101 \\ -1.2861 & 0 \\ -22.2886 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{\mathbf{B}_{hover}} \tilde{\mathbf{U}}_{lon}$$

Inspecting the longitudinal system matrix \mathbf{A}_{hover} , the hover dynamics are characterized by a very small set of parameters. It has more zeros than the well known typical longitudinal system matrix for aerodynamic flight. About the longitudinal control matrix \mathbf{B}_{hover} , one should notice that a variation in elevon equals deflections $\tilde{\delta}_E$ does not influence the change of longitudinal acceleration $\dot{\tilde{u}}$ (in body frame \mathbb{F}_B), and that a variation in throttle signal $\tilde{\delta}_T$ does not influence the change of vertical acceleration $\dot{\tilde{w}}$ or the pitching rate derivative $\dot{\tilde{q}}$. As expected, both input variations do not impact directly the pitch angle derivative $\dot{\tilde{\theta}}$ nor the altitude derivative $\dot{\tilde{h}}$.

The \mathbf{A}_{hover} matrix eigenvalues and respective eigenvector are shown in Tables 3.4 and 3.5.

Table 3.4: Hover Flight Longitudinal Eigenvalues

| Eigenvalues (rad/s) | Damping (ζ) (-) | Frequency (ω_n) (rad/s) |
|-----------------------|-------------------------|----------------------------------|
| $-3.99 \cdot 10^{-1}$ | 1 | $3.99 \cdot 10^{-1}$ |
| $-2.35 \cdot 10^{-3}$ | 1 | $2.35 \cdot 10^{-3}$ |
| $-3.15 \cdot 10^{+2}$ | 1 | $3.15 \cdot 10^{+2}$ |
| $-3.47 \cdot 10^{-1}$ | 1 | $3.47 \cdot 10^{-1}$ |
| $6.20 \cdot 10^{-1}$ | -1 | $6.20 \cdot 10^{-1}$ |

Looking at these results, some topics can be highlighted:

- All eigenvalues are fully damped which means that the system behaviour over this operating point does not oscillate.
- The last eigenvalue has a positive real part which dictates the system instability. Analysing the respective eigenvector, its instability is mainly related with the pitch angle θ and pitching rate q .
- The third eigenvalue (mainly associated with the w velocity state) is located far to the left of the real axis and so is considered to respond rapidly. Because its value is at least 3 orders higher than the others, it can be neglected when studying the steady state response.

Table 3.5: Hover Flight Longitudinal Eigenvectors

| Eigenvectors | v_1 | v_2 | v_3 | v_4 | v_5 |
|--------------|---------|---------|---------|---------|---------|
| u | -0.3704 | -0.0024 | -0.0000 | -0.0873 | 0.0041 |
| w | 0 | 0 | 0.9998 | 0.0220 | -0.0369 |
| q | 0.0000 | 0.0000 | 0.0220 | 0.3162 | 0.5266 |
| θ | 0 | 0 | -0.0001 | -0.9104 | 0.8493 |
| h | -0.9289 | -1.0000 | -0.0000 | -0.2512 | -0.0065 |

- Eigenvalues which are more near to origin are called dominant eigenvalues. The steady response of a system is dominated by these ones.
- The second eigenvalue is almost over the origin and so it can be seen as an integrator. This eigenvalue is associated with the altitude state variable.
- Because one eigenvalue is unstable, the whole system is considered to be unstable. Consequently, a control system is demanded to stabilize the aircraft vertical longitudinal motion.

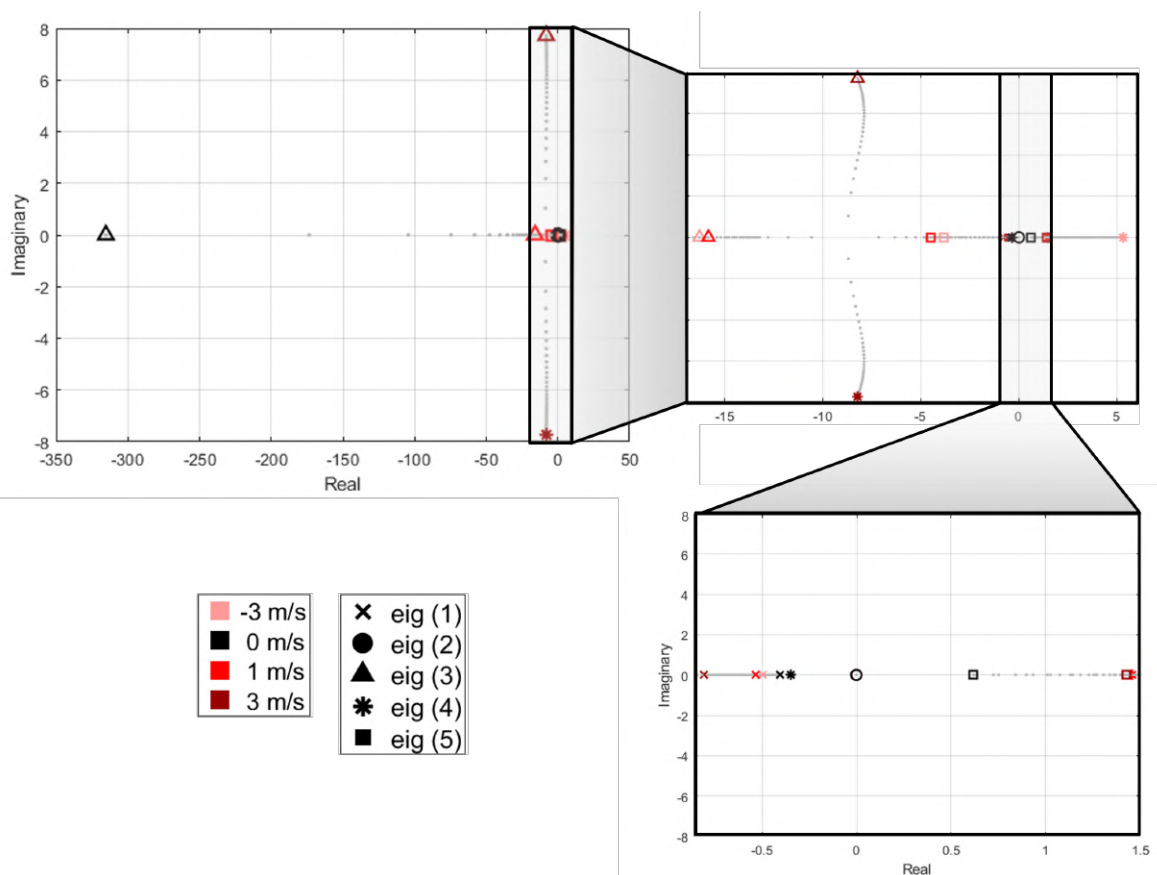


Figure 3.8: Vertical Eigenvalues Evolution Map. The first plot includes all the vertical flight envelope eigenvalues and the next pictures are zoomed over the grey areas of its previous plot.

In addition to hover eigenvalues, the eigenvalues of the full vertical flight envelope, which means for $u_0 = [-3, 3]$ m/s, are plotted in Figure 3.8. The evolution of the system eigenvalues between these two operating points is represented by the grey dots. Each symbol represent a different eigenvalue, and

the colors represent a specific velocity. The black symbols represent the hover eigenvalues, $u_0 = 0$ m/s, shown in Table 3.4. The red symbols represent the eigenvalues for $u_0 = 1$ m/s, and the pink and bordeaux symbols show the eigenvalues for $u_0 = -3$ m/s and $u_0 = 3$ m/s, respectively. Note that, from hover, as the longitudinal velocity increase or decrease, the fastest eigenvalue moves to the right, and the unstable eigenvalues move even more to the right making the instability more accentuated which will cost more control actuation to stabilize the system. When the velocity is higher than 1.3 m/s, the third and fourth eigenvalues (represented by a triangle and a star, respectively) start to compose a conjugate pair and the system starts having some oscillation. The second eigenvalue, related with the altitude state, stays at the origin for all velocities within the flight envelope.

Transition Flight

Although this flight condition is included in the vertical flight analysis, as it will be further explained, a linear model for the transition flight controller is needed. Hence, the state-space representation about the transition equilibrium point $u_0 = 1$ m/s is:

$$\dot{\tilde{\mathbf{X}}}_{lon} = \underbrace{\begin{bmatrix} -0.5349 & 0.0001 & 0 & 0 & 0.0009 \\ 0 & -16.776 & 0.6061 & -9.7989 & 0 \\ 0 & -10.558 & -2.0834 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -1.0000 & 0 & 0 & 0 & 0 \end{bmatrix}}_{A_{trans}} \tilde{\mathbf{X}}_{lon} + \underbrace{\begin{bmatrix} 0 & 21.2565 \\ -1.3672 & 0 \\ -23.6944 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{B_{trans}} \tilde{\mathbf{U}}_{lon}$$

As expected, it is noticeable that the systems' matrices slightly resembles to the hover linear model. The A_{trans} matrix eigenvalues are shown in Table 3.6.

Table 3.6: Transition Flight Longitudinal Eigenvalues

| Eigenvalues (rad/s) | Damping (ζ) (-) | Frequency (ω_n) (rad/s) |
|-----------------------|-------------------------|----------------------------------|
| $-5.33 \cdot 10^{-1}$ | 1 | $5.33 \cdot 10^{-1}$ |
| $-1.76 \cdot 10^{-3}$ | 1 | $1.76 \cdot 10^{-3}$ |
| $-1.58 \cdot 10^{+1}$ | 1 | $1.58 \cdot 10^{+1}$ |
| $-4.48 \cdot 10^{+0}$ | 1 | $4.48 \cdot 10^{+0}$ |
| $1.46 \cdot 10^{+0}$ | -1 | $1.46 \cdot 10^{+0}$ |

Horizontal Flight

The state-space representation for horizontal flight about its equilibrium point at $u_0 = 9$ m/s is:

$$\dot{\tilde{X}}_{lon} = \underbrace{\begin{bmatrix} -1.1666 & 2.2109 & -1.1612 & -9.6570 & 0.0002 \\ -0.7345 & -6.9164 & 6.5293 & -1.6658 & -0.0009 \\ 6.2661 & -41.2418 & -20.7876 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -0.1695 & 0.9855 & 0 & -9.1320 & 0 \end{bmatrix}}_{A_{level}} \tilde{X}_{lon} + \underbrace{\begin{bmatrix} 1.7381 & 14.9578 \\ -4.9784 & -2.8722 \\ -111.8187 & 10.7099 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{B_{level}} \tilde{U}_{lon}$$

The longitudinal system matrix A_{level} has the typical structure of a conventional aircraft platform, and so the longitudinal aircraft modes can be identified [27]. The A_{level} matrix eigenvalues are shown in Table 3.7.

Table 3.7: Level Flight Longitudinal Eigenvalues

| Eigenvalues (rad/s) | Damping (ζ) (-) | Frequency (ω_n) (rad/s) | Aircraft Mode |
|-----------------------|-------------------------|----------------------------------|---------------|
| $-13.9 \pm 15.2 i$ | $6.76 \cdot 10^{-1}$ | $2.06 \cdot 10^{+1}$ | Short-Period |
| $-0.543 \pm 1.12 i$ | $6.20 \cdot 10^{-2}$ | $7.90 \cdot 10^{-1}$ | Phugoid |
| $-5.94 \cdot 10^{-4}$ | 1 | $5.94 \cdot 10^{-4}$ | - |

A typical longitudinal model has two pairs of complex poles and, because five state variable were considered, an additional one at the origin:

- The first pair of complex poles, the ones with a higher negative real part correspond to the short-period mode which describes a rapid pitching from the aircraft about its center of gravity. Generally a feedback of the pitch rate q through the "elevator" deflection δ_E helps to reduce the oscillation and period time.
- The other pair corresponds to the phugoid mode which represents the slow interchange of kinetic energy and potential energy as the aircraft tries to re-establish an equilibrium flight condition. To improve the phugoid response, generally a feedback of the pitch angle θ through the "elevator" deflection δ_E is used.
- The pole at the origin, as in hover, works as an integrator and is associated with the altitude state variable.

In addition to the level flight eigenvalues, at $u_0 = 9$ m/s, the eigenvalues of the full horizontal flight envelope, which means for $u_0 = [8, 11]$ m/s, are plotted in Figure 3.9. The evolution of the system eigenvalues between these two operating points is represented by the grey dots. Each symbol represent a different eigenvalue, and the colors represent a specific velocity. The black symbols represent the level eigenvalues for $u_0 = 9$ m/s, shown in Table 3.4. The pink, red and bordeaux symbols show the

eigenvalues for $u_0 = 8$ m/s, $u_0 = 10$ m/s and $u_0 = 11$ m/s, respectively. In opposition to the vertical flight, in this flight orientation, the eigenvalues structure remains the same. With the velocity increase, the poles tend to move left and closer to the real axis, which means that the response tend to get faster and less oscillatory. Therefore, higher the speed, more stable is the system, and so more control actuation is needed to maneuver the system.

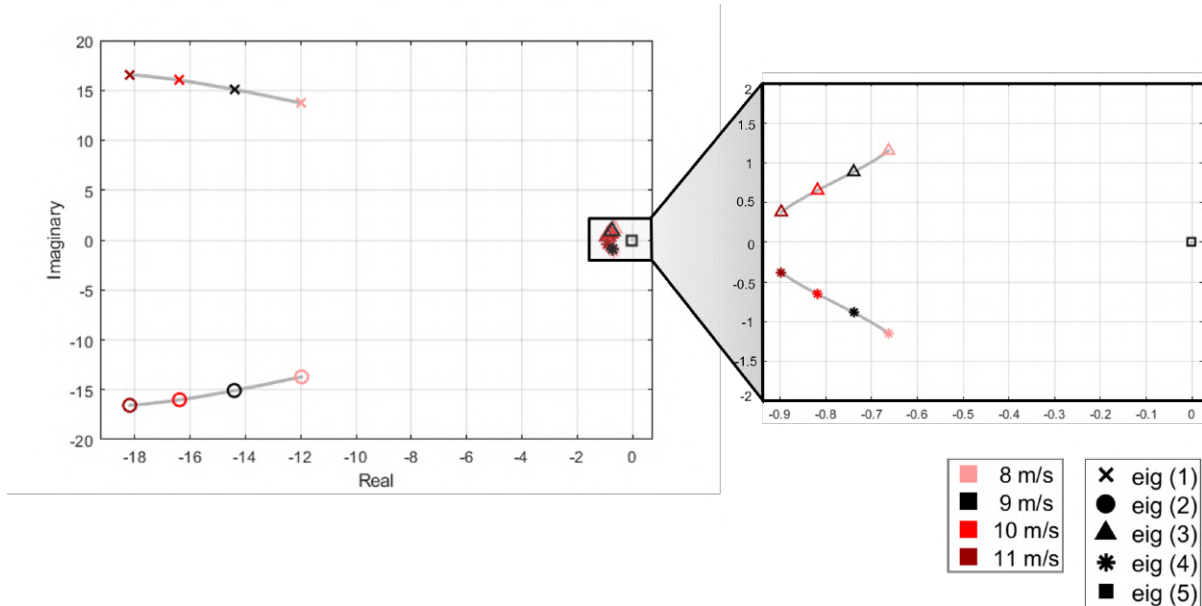


Figure 3.9: Horizontal Eigenvalues Evolution Map: from $u_0 = 9$ m/s (in black) to $u_0 = 12$ m/s (in red).

3.4.2 Controllability

Before deciding what control strategy apply to a system, its controllability should be analysed. The concept of controllability denotes the possibility of forcing a system into a particular state using only certain admissible manipulations, and can be verified by looking into the dynamic and input matrices. If one of the system's eigenvalues is not controllable, that part of the dynamics will remain untouched in the closed-loop system, and therefore the desired output is not reachable.

Mathematically, controllability, i.e number of uncontrollable states in a system equals zero, is represented by:

$$n - \text{rank} [B \ AB \ A^2B \ \dots \ A^{n-1}B] = 0 \quad (3.18)$$

where n is the number of state variables, and rank is the number of linearly independent rows in a matrix. Making use of the *Matlab* commands *ctrb* and *rank* to implement this equation, both hover and level flight linear system models, with $n = 5$, are controllable. The same evaluation was also done to flight envelope velocity limits and successful results were obtain. Thus, given that the structure of these matrices within each flight mode does not critically change with airspeed, the controllability of the full flight envelope is implied.

Chapter 4

Flight Control Design

In 3 the linearized aircraft systems were derived for both vertical and horizontal flight. This section covers the controller design including some introductory theory, the presentation of the followed strategy and the description of each step of its implementation.

Control system design has the main goal of constructing a system that has a desirable response to specific inputs. A perfectly defined noiseless system could be controlled using open-loop control, but, unfortunately, the immense complexity of real life systems makes this never the case. Thus, the need for closed-loop control arises from the various unknowns entering the system. Closed-loop systems are designed to automatically achieve and maintain the desired output condition by comparing it with the actual condition. Since the desired output or control action are dependent on the output itself, these feedback control systems make possible to control a system containing an acceptable amount of unknowns.

During control design, one should have in mind two response stages: transition and steady-state [28]. A desirable transient response should be sufficiently fast and without excessive oscillations. A desirable steady-state response should be capable of following the desired output with sufficient accuracy. In order to find a good control solution that suits the desired requirements, several techniques from classic to more modern control can be applied. In the next section, an optimal control strategy called Linear Quadratic Regulator (LQR) is presented.

Since this work is focused on the development of the longitudinal control, throughout the following sections, the lateral motion is assumed to be non-existent. This simplification constitutes a reasonable approximation of an additional lateral stabilization, which is not subject of this work.

4.1 LQR Theory

Linear Quadratic Regulator (LQR) is an optimal state variable feedback control strategy which attempts to find a controller that offers the best performance based on a specific measurement method [29]. Fundamentally, the LQR algorithm tries to find a balance between the amount of energy related with the state error and the energy consumed by the control actuation. For instance, to achieve a small

state error a large control signal is required, but to decrease the control signal energy a larger state error will result.

Mathematically, the LQR design is performed by minimising a quadratic cost function. The cost function is constructed by assigning a certain weighting to each state and input corresponding to their relative importance. The quadratic cost function is described as followed:

$$J = \int_0^{\infty} [\mathbf{X}^T \mathbf{Q} \mathbf{X} + \mathbf{U}^T \mathbf{R} \mathbf{U}] dt \quad (4.1)$$

where \mathbf{Q} and \mathbf{R} represent the state and input control weighting matrices respectively. These matrices must be positive and are usually diagonal.

$$\mathbf{Q} = \text{diag} [Q_u \quad Q_w \quad Q_q \quad Q_\theta \quad Q_h] \quad (4.2)$$

$$\mathbf{R} = \text{diag} [Q_{\delta_E} \quad Q_{\delta_T}] \quad (4.3)$$

Because it is difficult to intuitively understand the effect of the weightings, a first estimation for these coefficients can be given by the Bryson's rule. According to this method, each term of the diagonal matrix is the square of the inverse of the maximum deviation expected for the corresponding variable during the maneuver:

$$Q_i = \frac{1}{(\text{max deviation in } i \text{ from steady state})^2} \quad (4.4)$$

Besides of being used as a starting point for the weighting matrices, other advantage of this method is the units compensation through scaling. Nevertheless, caution must be taken when refining the state weightings by inspection alone. Apparent performance, especially during manoeuvres, can be greatly improved by tweaking the state weightings which can lead to a loss of insight and is strongly discouraged. A better approach is to find a valid reason for changing weightings whenever possible.

The solution to the cost function minimization takes the form of a difference equation known as the matrix Ricatti equation, and the solution to this equation is suited to feedback control since it has the form of full state feedback:

$$\tilde{\mathbf{U}} = -\mathbf{K} \tilde{\mathbf{X}} \quad (4.5)$$

Using the *Matlab* command *lqr*, a $m \times n$ matrix \mathbf{K} is obtained, where n is the number of state variables and m is the number of input control variables.

Besides the positive and symmetrical \mathbf{Q} and \mathbf{R} matrices, other requirements must be fulfilled to use LQR control [30]:

- As a linear control method, in order to use this control solution the system must be linear. The stabilization of nonlinear systems is feasible within a neighbourhood of the equilibrium point which is the reason why the tail-sitter model was linearized.
- Additionally, the linear state-space model is required to be controllable and observable, which from the analysis presented in the previous chapter it is conclude that both linear systems (vertical and horizontal) satisfy them.

From this control method, some variation in its implementation can be done in order to meet better results and specifications. Some theory and explanations of variations used in the developed controller are given below.

Feedback with Variables in Servomechanism

The final feedback matrix K can be applied in some different ways [27]: it can be placed either in the feedback loop, in order to stabilize the system, or in the direct loop, to control (not cancel) the error between the value of the states and the references.

With feedback with variables in servomechanism, the final actuation signal U is given by:

$$U = K_c(X_{ref} - X_c) - K_p\tilde{X}_p \quad (4.6)$$

where K_p is the feedback matrix composed by the column vectors of K correspondent to the complementary states X_p (inner feedback loop), and K_c is composed by the column vectors of K correspondent to the output/desired states X_c (direct loop). Its implementation is represented by the block diagram in Figure 4.1.

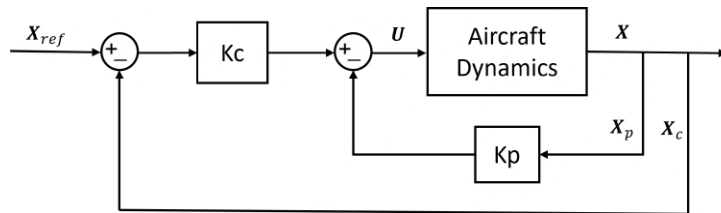


Figure 4.1: Feedback with Variables in Servomechanism Block Diagram

Feedback with Integrative Variables

Even though the closed-loop dynamic response is stable, the resulting response may not be as desired. As a PI controller in classic control introduce integrative terms in order not only to stabilize states or outputs but also to force its cancellation with constant entries, in optimal control this output error cancellation is also possible. Therefore, to cancel a variable X_i under stationary condition, its integrated variable can be introduce as a supplementary state:

$$\dot{X}_{n+1} = X_i \quad (4.7)$$

The full state gain matrix can be separated into two gain matrices: one for the request in the outer loop, and the other for the state stabilization in the inner loop of the controller. Therefore, the controller input in time domain takes the following form:

$$U = -K\tilde{X} + K_i \int_0^t E(t) dt \quad (4.8)$$

In theory, the number of the request values (or integrative variables) must not exceed the number of

actuator inputs available.

Integral Windup Compensation

Since the linear controller is designed to operate within a linear range, ignoring the actuator nonlinearities, such as saturation, will cause the integrator to windup which will significantly deteriorate the closed-loop performance. Integral windup is the process of accumulating the integral term of the controller output [31]. If the reference command $r(t)$ do not saturate the actuator the rising time is small and the integral error $e(t)$ is quickly annulled, see Figure 4.2(a). However, if the reference command applied causes the control variable to attain the actuator limit during the transient response, the control error decreases more slowly and therefore the integral term becomes much larger, see Figure 4.2(b). As this accumulated error is unwound, the consequences are an excessive overshooting and long settling time which may cause the instability of the system, see Figure 4.2(c).

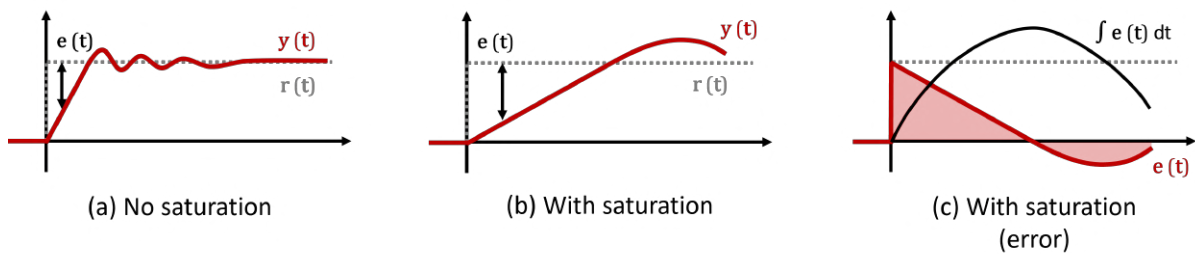


Figure 4.2: Windup illustration: offset by errors in the other direction

In order to avoid this behaviour, the controller integral term is kept constant during periods when the control action is saturated. This is implemented using a switch before the integrative action controlled by a saturation flag. Its implementation is represented by the block diagram in Figure 4.3.

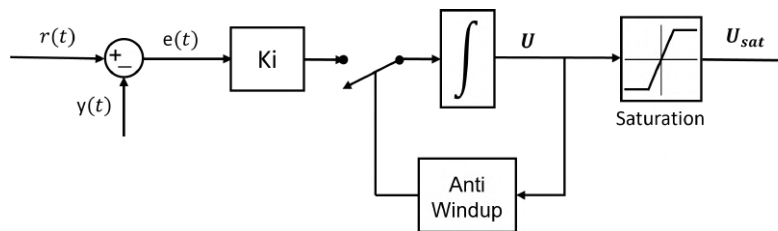


Figure 4.3: Integral Windup Compensation Block Diagram

4.2 Control Strategy

Before making some design choices, it is important to determine the exact goals and some output specifications. Therefore, taking into account the proposed minimum flight missions, the main requirements for this controller are to perform:

- vertically and horizontally stable flight;
- climb and descend in both flight modes;
- move forward and backwards in vertical flight mode;

- control longitudinal velocity in horizontal flight;
- perform autonomous transitions between flight modes;
- perform autonomous take off and landing.

There are also suggested some quantitative performance limits. However, considering that the present work is the first iteration of this flight controller, these should not be considered satisfactory for a real implementation scenario. That said, based on similar works [10], [11], [12], [16], the following limits are assumed to be acceptable:

- 2 seconds in the step response settling time for each input request during vertical or horizontal flight modes;
- 4 seconds to complete both transition maneuvers;
- 1 degree error in the pitch angle request (a restrict constrain is imposed because this variable is critical to the aircraft stability);

To achieve these requirements, this section is divided as follows: First, the controller input types (or channels) are defined and it is explained how they are supposed to influence the aircraft behaviour. Second, the overall control structure is presented in detail. The next sections describe how each flight mode operates.

4.2.1 Controller Inputs

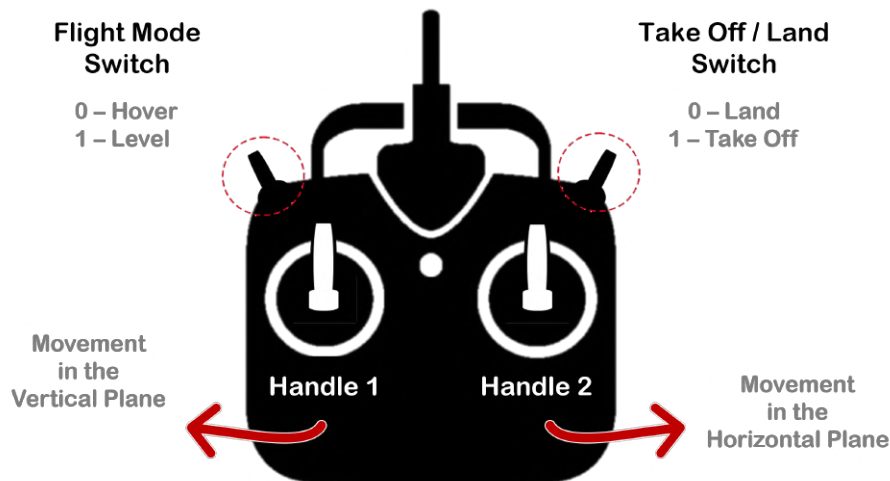


Figure 4.4: Controller input types

Associating the software inputs implementation with a with a UAV remote transmitter used to manually flight the aircraft, the following inputs are used: two switches and two handles. Only the up and down movement of the handles is considered, the left and right movement is left for a future lateral controller. These four inputs are illustrated in Figure 4.4 and are explained below.

- **Flight Mode Switch:** this input has two positions that indicate the desired flight mode:
 - Position 0 - Vertical Flight Mode** which will also be denoted as Hover, since it is flown about this operating point;

Position 1 - Horizontal Flight Mode which will also be denoted as Level, since it is flown about the level operating point referred in the trimming section.

- **Take-Off / Land Switch:** this input switch allows the autonomous take off and land. The aircraft must be initialized in the "Land" switch position and, once it is changed to the "Take-Off" switch position, it will autonomously climb until it reaches the pre-established take-off altitude. Then the aircraft is free to receive any other type of control inputs and fly. To autonomously land the aircraft, the aircraft must be in the vertical flight mode and only then the "Land" switch position can be activated. Then the vehicle is going to safely descend and reduce its velocity until it reaches the ground at almost zero velocity.

Position 0 - Land Mode

Position 1 - Take-Off Mode

- **Handle 1:** varying between -1 and 1, this input signal controls the aircraft movement in the vertical plane. During Vertical Flight Mode, this movement is done by controlling the aircraft longitudinal velocity u , and in Horizontal Flight Mode it is done by controlling the pitch angle θ which will consequently vary the aircraft climb angle $\gamma = \theta - \alpha$.
- **Handle 2:** varying between -1 and 1, this input signal controls the aircraft movement in the horizontal plane. During Vertical Flight Mode, this movement is done by controlling the pitch angle θ allowing the aircraft to move forward and backwards, and in Horizontal Flight Mode it is done by controlling the longitudinal velocity u , increasing or decreasing the aircraft velocity around the equilibrium velocity.

The presented handles logic is resumed in Figure 4.5.

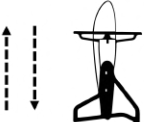
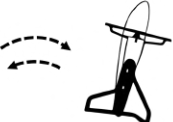

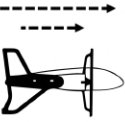
| | Handle 1 | Handle 2 |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| Vertical Flight Mode |  <p><i>Climb and Descend</i> (u_{ref})</p> |  <p><i>Forward and Backward</i> (θ_{ref})</p> |
| Horizontal Flight Mode |  <p><i>Climb and Descend</i> (θ_{ref})</p> |  <p><i>Faster and Slower</i> (u_{ref})</p> |

Figure 4.5: Controller inputs explanation

4.2.2 Controller Structure

After receiving these input signals, the objective of the control system is to generate actuators signals $\delta_L, \delta_R, \tau_L, \tau_R$ to achieve a reference behavior. The block diagram of the controller is shown in Figure 4.6. Each block is then explained below.

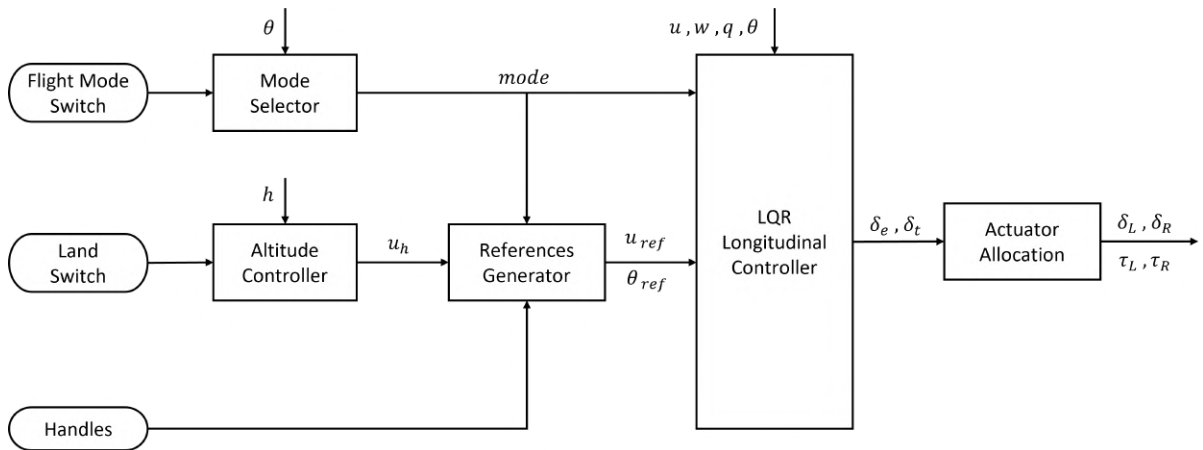


Figure 4.6: Controller Structure

Mode Selector:

This block reads whether flight mode switch is in Vertical Flight Mode (H: Hover) or Horizontal Flight Mode (L: Level). On the one hand, during stable flight, if there is no variation detected between the last mode and the actual mode the outputted flight mode remains the same as the input mode. However, if variation is detected the outputted mode becomes the corresponding transition mode (X: Transition or BX: Back Transition). On the other hand, to end a transition mode the aircraft actual attitude is compared to the equilibrium attitude of the desired flight condition, i.e. the Transition mode ends once the $\theta < \theta_L$, and the Back Transition ends once $\theta > \theta_H$. This logic is illustrated in Figure 4.7.

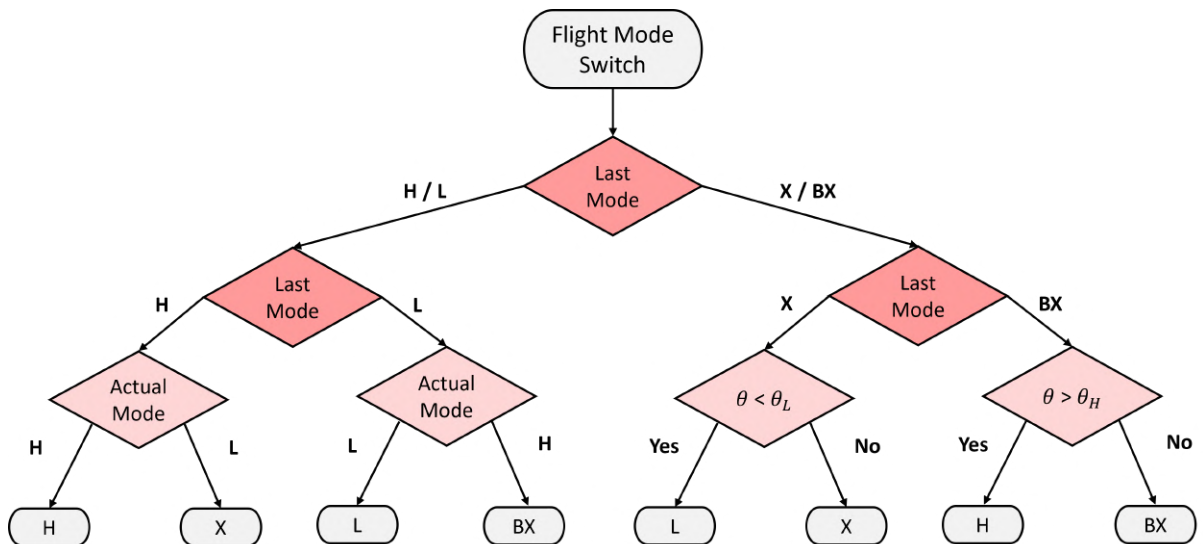


Figure 4.7: Mode Selector Flowchart

References Generator:

This block outputs the reference states u_{ref} and θ_{ref} that follow to the LQR controller. Depending on the flight mode printed by the Mode Selector block, this module manages the controller inputs in two separate ways, see Figure 4.8.

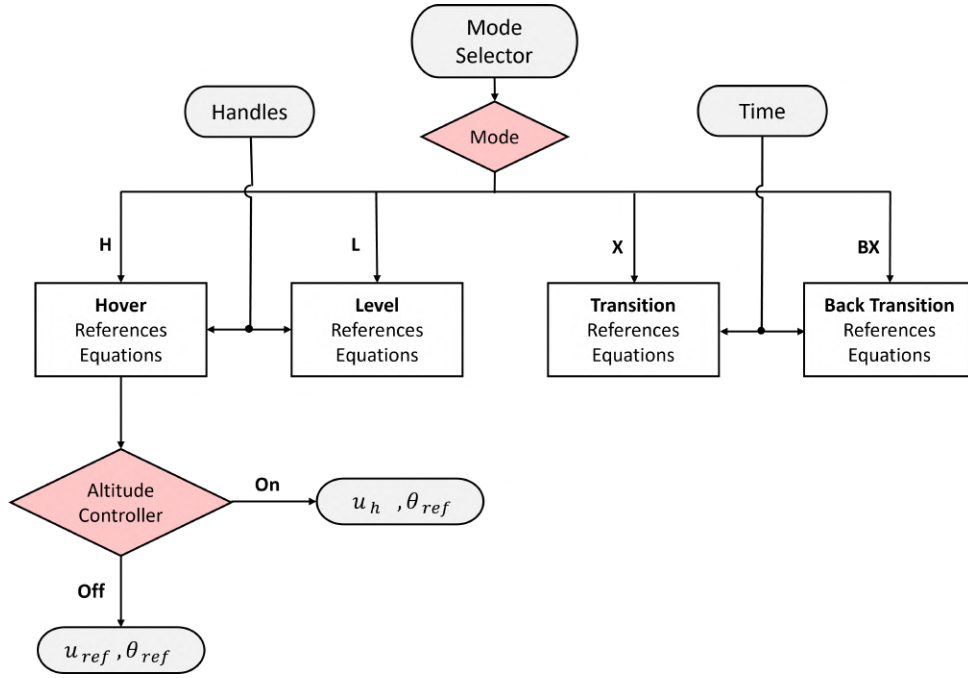


Figure 4.8: References Generator Flowchart

During Hover or Level, this block computes the handles inputs and prints the desired references to the LQR controller in order to lead the aircraft to behave as pretended. Since the inputs from the controller handles are limited between -1 and 1, to ensure that the aircraft is kept within the flight envelope and the linear regime in each flight mode, the input values are then processed in specific ways. The corresponding references equations for Hover and Level are presented below.

- **Hover:** The climb and descend velocity are limited to ± 3 m/s from the equilibrium condition which is $u_H = 0$ m/s. The forward and backwards movement results from a pitch angle deviation limited to $\pm 25^\circ \approx \pm 0.436 \text{ rad}$ from the hover equilibrium attitude which is $\theta_H = 90^\circ$. The minus sign is because a positive handle 1 signal means moving forward and so the pitch angle must reduce from its equilibrium position to redirect the aircraft into the moving direction.

$$u_{ref} = 3 \times handle_2 \quad (4.9a)$$

$$\theta_{ref} = \theta_H - 0.436 \times handle_1 \quad (4.9b)$$

- **Level:** The longitudinal velocity is limited to ± 1 m/s from the level equilibrium velocity $u_L = 9$ m/s to ensure that the aircraft remains within the flight envelope. The pitch angle is limited to an amplitude equals to the level equilibrium attitude $\theta_L \approx 9^\circ$, which means a variation of $\frac{\theta_L}{2}$ up or down.

$$u_{ref} = u_L + 1 \times handle_1 \quad (4.10a)$$

$$\theta_{ref} = \theta_L + \frac{\theta_L}{2} \times handle_2 \quad (4.10b)$$

During Transition and Back Transition, the output references do not depend on the handles input. The

u_{ref} and θ_{ref} trajectories are previously defined and therefore only depend on time from the beginning of the maneuver. The corresponding references equations for Transition and Back Transition are discussed below and were based on [10] and [12].

- **Transition:** The aircraft longitudinal velocity during the transition changes from u_H to u_L . Initially a step function between this two values was used, but, to decrease the altitude gain during the maneuver, the reference was adapted so that the response of the thrust was improved. The second attempt was to use a ramp which already improved a lot its performance. Lastly the smooth curves represented by the following equations were tested and obtained the best response. The variable $T = 3$ seconds is the duration of the Transition and was selected iteratively until it reaches the minimum feasible maneuver time. The same reasoning was use to the pitch angle reference. These reference equations are plotted in Figure 4.9.

$$u_{ref} = u_H + \frac{u_L - u_H}{2} \left(1 - \cos \frac{\pi t}{T} \right) \quad (4.11a)$$

$$\theta_{ref} = \theta_L - \frac{\theta_L - \theta_H}{2} \left(1 + \cos \frac{\pi t}{T} \right) \quad (4.11b)$$

- **Back Transition:** During this maneuver, the longitudinal velocity u should go to u_H and the pitch angle θ should reach θ_H as fast as possible. To do this, the elevons actuation should saturate and the thruster signal decrease. However, in order to boost the generated pitching momentum and reduce the Back Transition duration, the elevons authority can be amplified by increasing the slipstream velocity through the thruster actuation too. Therefore, two steps were chosen for both state references which will immediately saturate both actuation signals and perform a faster Back Transition. These reference equations are also plotted in Figure 4.9.

$$u_{ref} = u_L \quad (4.12a)$$

$$\theta_{ref} = \theta_H \quad (4.12b)$$

When the Take-Off / Land Switch is used and the altitude controller is activated (which, as explained before, can only occur during hover flight), u_{ref} is bypassed by the u_h from the Altitude Controller block, and just the θ_{ref} from the Hover references equations is used to guarantee stability.

Altitude Controller

The altitude tracking is achieved by generating the longitudinal reference velocity u_h from the altitude error signal h_{error} . In order to automatically take-off and land, a PID (Proportional Integral Differential) classic controller was used, see Figure 4.10. The integral part assures the precise altitude reference tracking but usually also introduces some overshooting to the response. Since this controller is responsible for the autonomous landing, an overshooted response would crash the aircraft into the ground which is not desirable. That said, the derivative term is added to guarantee that the response is overdamped and that the aircraft will touch the ground with almost zero velocity. The proportional, integral

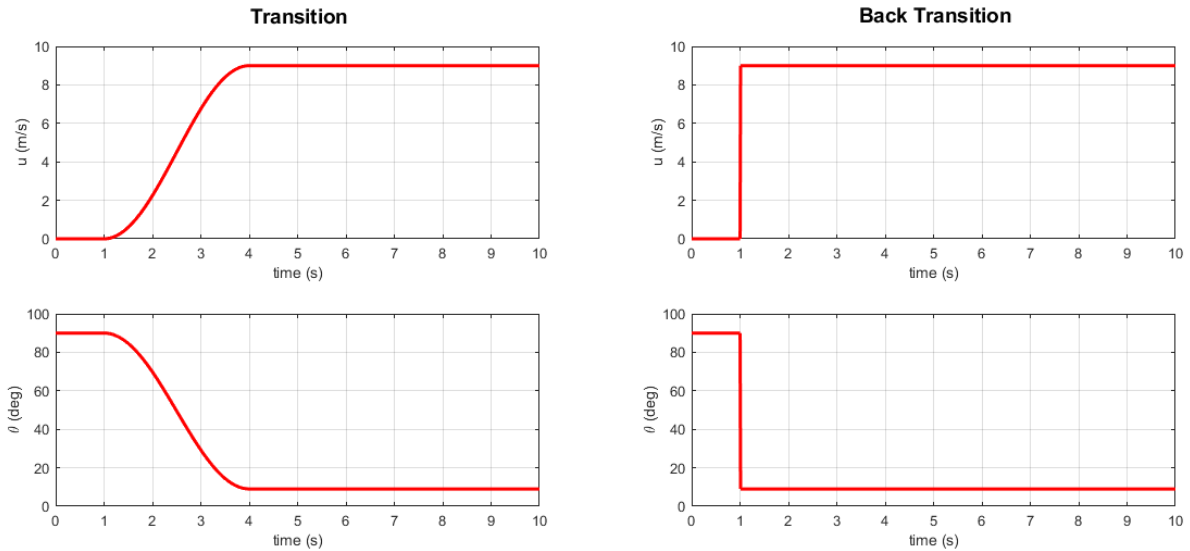


Figure 4.9: Transition and Back Transition References

and derivative gains used to obtain a suitable response were $K_{ph} = 1 \text{ s}^{-1}$, $K_{ih} = 0.01 \text{ s}^{-1}$ and $K_{dh} = 2 \text{ s}^{-1}$.

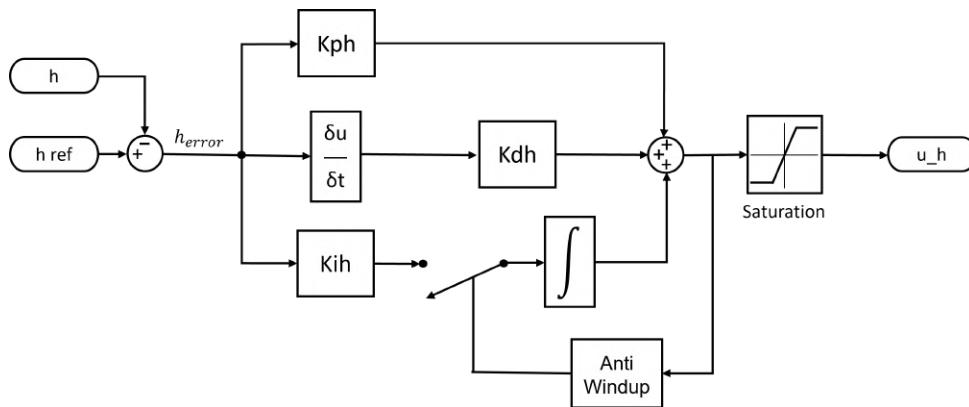


Figure 4.10: Altitude PID Controller Diagram

LQR Longitudinal Controller

The longitudinal variables to be controlled in this block are only the u , w , q and θ . The altitude h is controlled in the outer-loop altitude PID controller as explained earlier.

As requirement, the LQR longitudinal controller must stabilize the aircraft and regulate its velocity u and pitch angle θ accordingly to the given references. However, only one of these variables can be tracked at a time. The aerodynamic (or horizontal) aircraft dynamics relate the velocity u and the pitch angle θ . Therefore, they are not independent and one has to compensate and trim the aircraft for a request in the other. Between these two, it was chosen to precisely track the θ because it is seen as a more critical variable to ensure the aircraft stabilization during all its flight modes.

That said, the LQR control method was applied, as described in Section 4.1, considering the four u , w , q and θ longitudinal variables plus an integrative θ_i . The velocity u will be controlled using the servomechanism method.

As a linear control technique, the LQR control method only works around the equilibrium point it was design for. This means that not only the dynamic and input matrices A and B to calculate the gain matrix depend on the operating mode, but also the LQR state and input weighting to find the initial Q and R matrices by making use of Bryson's method. In addition, also the state trim values that are subtracted to the actual state values and the input trim values that are added by the end vary with the operating mode. The Hover, Transition and Level state and input vectors along with the corresponding A and B matrices are provided in the previous chapter. For the Back Transition there are used the same vectors and matrices used for the Transition.

Weighting Selection

The considered maximum deviations from equilibrium states for each operating mode, which are applied in Bryson's method, are listed in Table 4.1. The explanation of each selected values is given below.

Table 4.1: Maximum state and input deviations from equilibrium

| | u (m/s) | w (m/s) | q (rad/s) | θ (rad) | θ_i (rad) | δ_E (rad) | δ_T (-) |
|------------------------|-----------|-----------|-------------|----------------|------------------|------------------|----------------|
| Hover | 0.1 | 1 | 0.1 | 0.001 | 0.0001 | 0.001 | 0.01 |
| Level | 10 | 1 | 1 | 0.01 | 0.001 | 0.01 | 0.01 |
| Transition | 10 | 10 | 1 | 0.01 | 0.001 | 0.001 | 0.01 |
| Back Transition | 0.1 | 10 | 1 | 0.01 | 0.001 | 0.001 | 0.01 |

- **Longitudinal Velocity u :** In Hover, the longitudinal velocity u is not decisive for its stabilization and it has no integrative variable. The chosen velocity deviation is constricted to help on the reference tracking. In Level flight, the longitudinal velocity u is loosened when compared with Hover because its precise tracking is not possible with the implemented design. The reason is that the pitch angle is being tracked with an integrator, if the velocity is increased and the pitch angle is maintained, the aircraft starts to climb and so will not perform a level flight. During Transition, the objective is to follow the pitch angle reference. The velocity is also loosened which will prevent the aircraft to over accelerate and gain altitude. On the other hand, during Back Transition, the velocity is constricted to help the thrust actuator to saturate and increase the elevons authority. That leads to a smaller maneuver time duration.
- **Downward Velocity w :** In Hover, comparing to the longitudinal velocity u , the downward velocity w is not so constricted. This allows the system to follow the pitch angle references. Contrarily, in Level flight, the downward velocity w should be more accurately tracked than the longitudinal velocity u . This should prevent the aircraft from leaving the Level operating mode. In both transitions, the downward velocity w tracking is loosened when comparing with Hover and Level.
- **Pitching Rate q :** In Hover and Level, this variable is more closely tracked since the objective is for its value to remain zero. During transitions, the pitching rate maximum deviation can be loosened when compared with the other two modes.

- **Pitch Angle θ** : The maximum deviations pretended during Level and transitions modes are in the order of one degree which in radians are almost two orders below. In Hover, its maximum deviations are tightened because in order to guarantee that the system remains in the linear regime the pitch angle limits are more restrict.
- **Pitch Angle Integrator θ_i** : The maximum deviation of the integrator state is ten times lower than the original state to allow a faster reference tracking.
- **Elevons δ_E** : During Level, the maximum deviation pretended in elevon's deflection is in the order of one degree. In Hover, the elevon's deflection is constricted because this mode is more sensitive to disturbances and naturally unstable. During transitions, a big amplitude of requests is done to the controller. Therefore, the maximum deviation is also tightened, when comparing to Level mode, to avoid the actuation to easily saturate.
- **Thrusters δ_T** : The maximum deviation selected for the thruster input is one percent of its maximum value for all operating modes.

Gain Matrix

Using the values from Table 4.1 in (4.4), the matrices Q and R are calculated. Then, using the *Matlab* command $lqr(A,B,Q,R)$, the gain matrix K is obtained for each flight mode. These results are presented in Table 4.2.

Table 4.2: LQR complete gain matrix K for each operating mode

| | | u (m/s) | w (m/s) | q (rad/s) | θ (rad) | θ_i (rad) |
|------------------------|------------------|-----------|-----------|-------------|----------------|------------------|
| Hover | δ_E (rad) | 0.0000 | 0.0204 | -0.9648 | -10.1039 | -1.0000 |
| | δ_T (-) | 0.0872 | 0.0000 | -0.0000 | 0.0000 | 0.0000 |
| Level | δ_E (rad) | -0.0497 | 0.1884 | -0.0824 | -2.8389 | -9.8530 |
| | δ_T (-) | 0.0515 | -0.0963 | 0.0257 | 1.2157 | 1.7086 |
| Transition | δ_E (rad) | 0.0000 | 0.1179 | -0.2167 | -1.0624 | -1.0000 |
| | δ_T (-) | 0.0755 | 0.0000 | -0.0000 | -0.0000 | 0.0000 |
| Back Transition | δ_E (rad) | 0.0000 | 0.1179 | -0.2167 | -1.0624 | -1.0000 |
| | δ_T (-) | 0.1164 | 0.0000 | -0.0000 | -0.0000 | 0.0000 |
| | | K_c | | K_p | | K_i |

- K_c is a [2x1] vector which corresponds to the first column (u) of K ;
- K_p is a [2x3] matrix which corresponds to the second (w), third (q), and forth (θ) columns of K ;
- K_i is a [2x1] vector which corresponds to the last column (θ_i) of K .

LQR Structure

The structure of the LQR controller block is illustrated in Figure 4.11. The velocity u is controlled using servomechanism where the velocity error u_{error} is multiplied by the K_c vector. The pitch angle θ is tracked making use of an error integrator where the pitch angle error θ_{error} is multiplied by the K_i

vector. Note that an Anti-Windup mechanism is applied, i.e. when the actuation saturates the integrator output remains constant instead of continuously growing. The other longitudinal variables w , q and θ are stabilized by multiplying the variables deviations from the equilibrium point $\mathbf{X0}(w, q, \theta)$ with the feedback \mathbf{K}_p matrix. Finally, all terms are summed and then added to the actuation equilibrium point $\mathbf{U0}(\delta_E, \delta_T)$ and the final control signals are obtained.

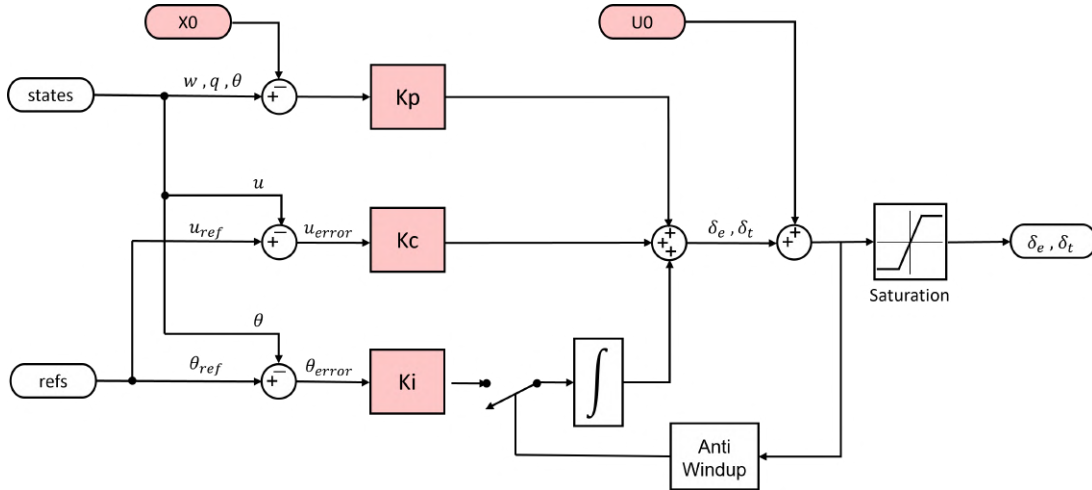


Figure 4.11: LQR Controller Diagram where pink boxes indicate variable flight conditions and corresponding gain matrices

The values of the equilibrium states and inputs vectors, $\mathbf{X0}$ and $\mathbf{U0}$, and the gain matrices, \mathbf{K}_c , \mathbf{K}_p , \mathbf{K}_i , change depending on the output of the Mode Selector block as they vary with the actual operating mode. This technique, of controlling nonlinear systems with dynamics changing from one operating condition to another, is commonly called *Gain Scheduling*. In Figure 4.11, the changing vectors and matrices are colored in pink.

Actuator Allocation

The last module of the controller is the Actuator Allocation. Its objective is to translate the desired conventional control signals δ_E , δ_A , δ_R and δ_T into actuator signals, namely the left and right elevons deflections δ_l , δ_r , and the left and right throttle signals τ_l , τ_r . For this purpose, the equations (4.13) are inverted:

$$\begin{aligned} \delta_l &= \delta_E + \delta_A & \delta_r &= \delta_E - \delta_A \\ \tau_l &= \delta_T + \delta_R & \tau_r &= \delta_T - \delta_R \end{aligned} \quad (4.13)$$

Note that, once only longitudinal dynamics are being controlled, δ_A and δ_R are null. Therefore, the left and right signals (both in elevon's deflection and throttle) are always equal.

Chapter 5

Simulation Flight Results

The following chapter is divided into two parts. First, the inputs are individually tested and the controller response is analysed. Second, the two minimum flight missions are simulated and their results are presented.

5.1 Response Analysis

This section analyses the controller initial reliability, investigates the handles step responses in both operating modes and also the transitions and take-off and land performances.

In Hover and Level mode, a step in both left and right handles are tested separately. To test Transition, the aircraft is initialized in Hover and then the Mode Selector Switch is shifted. The handles are left still in zero/null position. To test the Back Transition, the opposite is done. The aircraft is initialized in Level flight and the Mode Selector Switch is shifted. Finally, to test the altitude controller, the aircraft is initialized in vertical orientation on the ground and the Take-Off/Land Switch is shifted. After the aircraft achieving the reference take-off altitude, the switch is again shifted and the aircraft autonomously lands.

The simulations' results are presented in the following sections. For each scenario, all simulation state and input variables are plotted separately. The blue lines in each graph represent the simulation results. The red lines represent the actuators' limits. The green lines in the velocity u and pitch angle θ plots show the reference values given to the LQR controller, and the green line in the altitude h plot shows the reference value given to the altitude controller when the Take-Off/Land Switch is used.

Additionally, an image of the aircraft simulated environment for each scenario is inserted to help the aircraft trajectory and attitude changing visualization.

5.1.1 Hover

In both Hover simulations the aircraft is initialized with the aircraft pointing up, facing the North direction, above the ground. Flight Mode Switch starts in position 0 i.e. Hover mode, and Take-Off/Land Switch starts in position 1 i.e. Take-Off mode.

Step in Handle 1

In the first case, at $t = 1$ s a step input is given in handle 1 which produces a step request to the pitch angle θ_{ref} . This request results in an elevon response δ_E and a small decrease in the throttle signal δ_T . Although the aircraft tends to a steady-state condition, the system does not achieve it in the given time. The system is capable of following the attitude reference with less than 1° of error and move forward, however, during this maneuver some aerodynamic forces appear and the aircraft starts gaining altitude. Then, at $t = 7$ s a negative input step is given in handle 1 to return the aircraft to hover conditions. The aircraft takes near 5 s to achieve the initial steady condition which is higher than the stated requirements previously. Between $t = 5$ s and $t = 10$ s the aircraft gains 1 m in altitude per 3 m in the north direction. Since this movement should only occur in the horizontal plane, a smaller altitude gain should be attempted. The settling time for the second input request is about 5 or more seconds which does not fulfill the stated requirements.

To determine the steady-state velocities u and w for a unit step in handle 1, other simulation was run with a higher stop time. To track the pitch angle reference, the aircraft stabilizes at $u = 4.5$ m/s and $w = 0.73$ m/s with a settling time about 70 s. The elevon and throttle signals stabilize at $\delta_E = -19.8^\circ$ and $\delta_T = 88\%$. Besides of the settling time high deviation from the stated requirements, this flaw is not considered to be critical since vertical flight is a really unstable flight condition and the aircraft is not supposed to flight much time in this mode.

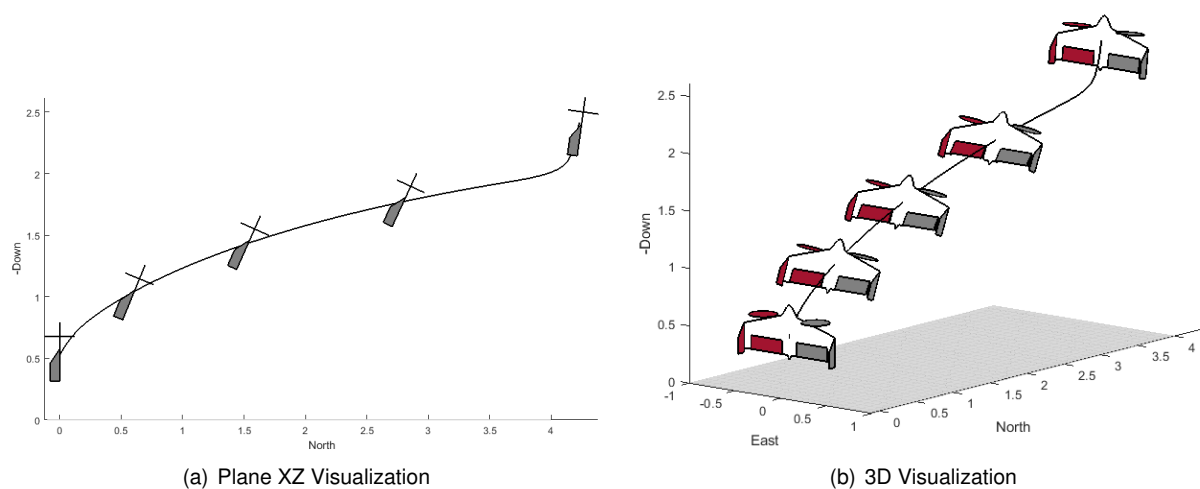


Figure 5.1: Step in handle 1 during Hover (simulation visualization)

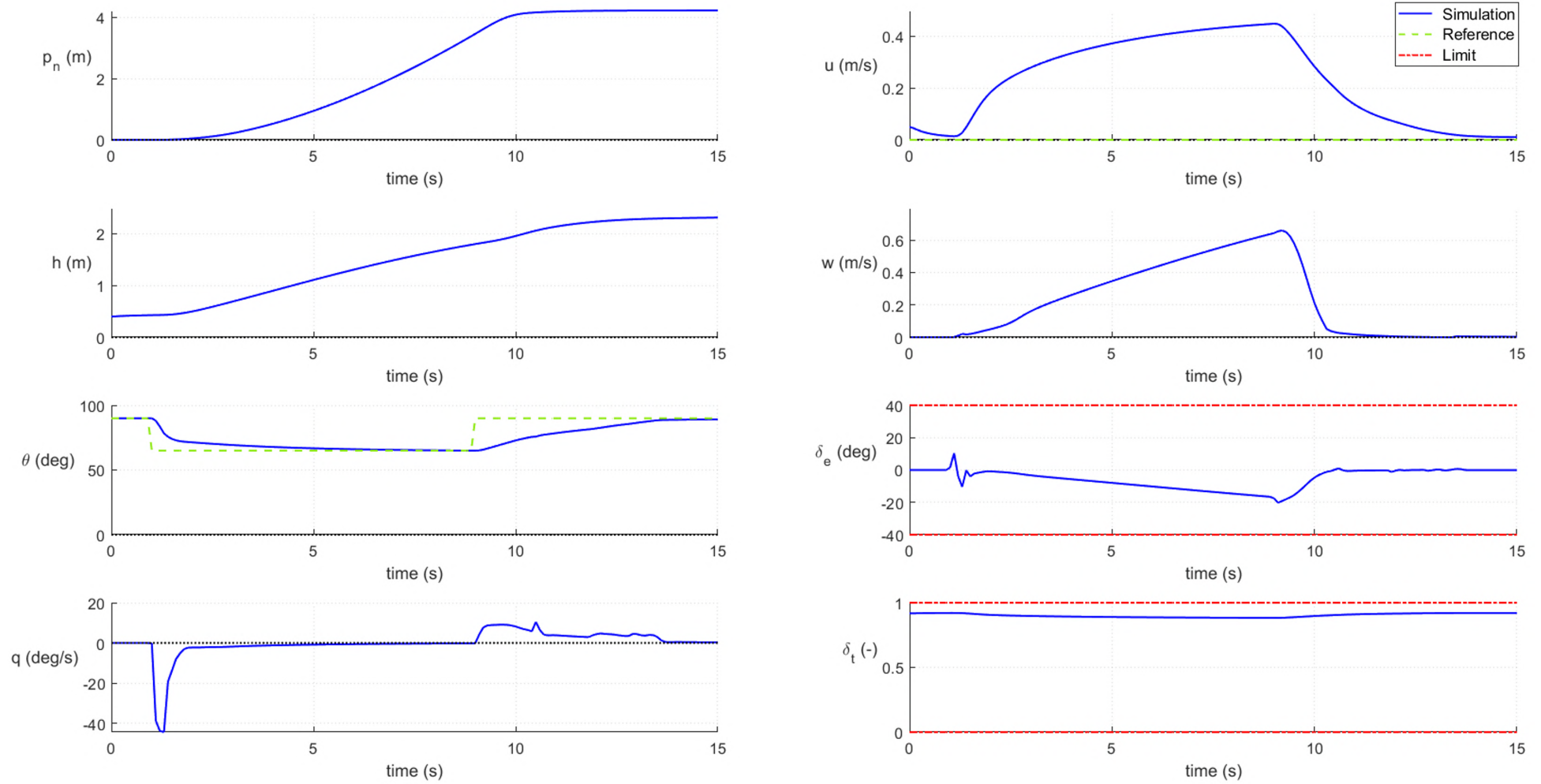


Figure 5.2: Step in handle 1 during Hover

Step in Handle 2

In the second case, at $t = 1$ s a step input is given in handle 2 which produces a velocity step request u_{ref} . This request results in a change of the throttle signal δ_T , which saturates until the velocity reaches its final value, while the elevon's signal remains null. The aircraft altitude continues to increase and, as intended, the aircraft north position does not modify. Beside of the $u_{ref} = 3$ m/s request, the velocity stabilizes at a minor value of $u = 2.4$ m/s. This result was expected since this velocity error is not controlled with an integrator. At $t = 6$ s a negative input step is given in handle 2 and the aircraft returns to its hover trim condition. Both positive and negative step requests only took about 2 s to stabilize which is assumed to be acceptable considering the stated requirements. Note that, as no pitch angle request was made, the variables p_N, θ, w, q and δ_E stay null, which was expected.

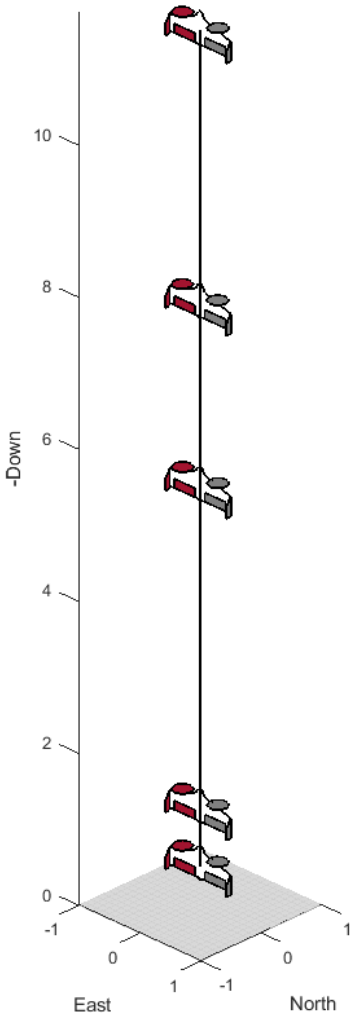


Figure 5.3: Step in handle 2 during Hover (simulation visualization)

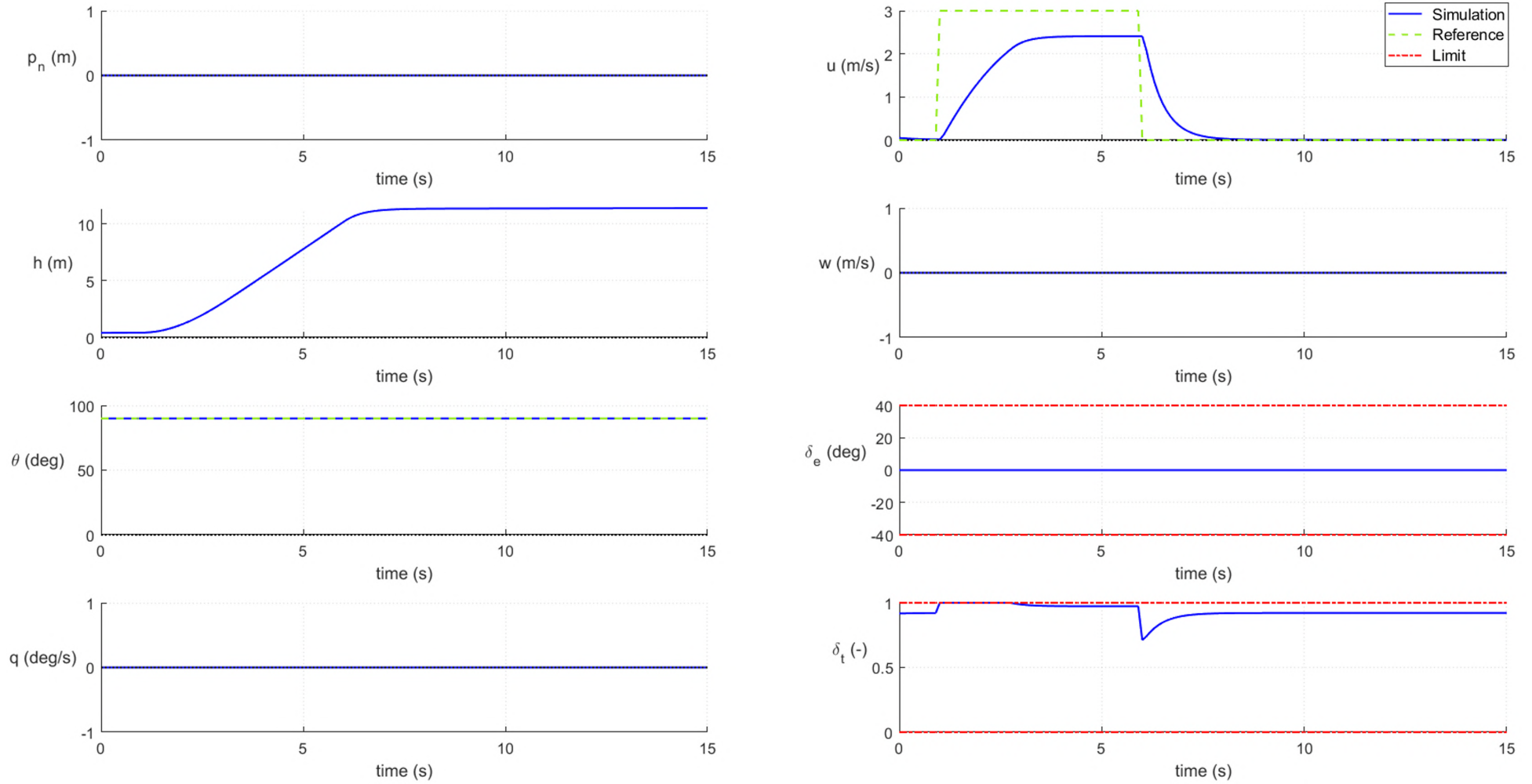


Figure 5.4: Step in handle 2 during Hover

5.1.2 Level

In both Level simulations the aircraft is initialized at level trim conditions, facing the North direction, above the ground. Flight Mode Switch starts in position 1 i.e. Level mode, and Take-Off/Land Switch starts in position 1 i.e. Take-Off mode.

Step in Handle 1

In this case, at $t = 1$ s a step input is given in handle 1 which produces a velocity step request u_{ref} of 1 m/s over the trim velocity. Although the aircraft can trim at that velocity, its precise tracking is not possible because this variable is controlled with a servomechanism direct loop feedback and not with an integrator. The velocity only increases 0.2 m/s from its equilibrium velocity which results in an error of about 80% and is not acceptable. When the velocity request changes, the pitch angle oscillates less than 1 s and then stabilizes at the initial trim value. Since this value θ is higher than the one needed to trim the aircraft at its new velocity, the aircraft climb angle γ increases, which is not desired. The u and θ states are not independent from each other in the aircraft dynamics. The pitch angle reference θ_{ref} should be adjusted when a u_{ref} is requested to prevent the aircraft climbing. When the negative step is given in handle 1, at $t = 4$ s, the aircraft easily retrieves to its initial condition. In both steps, the system takes less than 1 s to stabilize at a steady state condition and the pitch angle error is less than 1° . These two parameters fulfil the stated requirements.

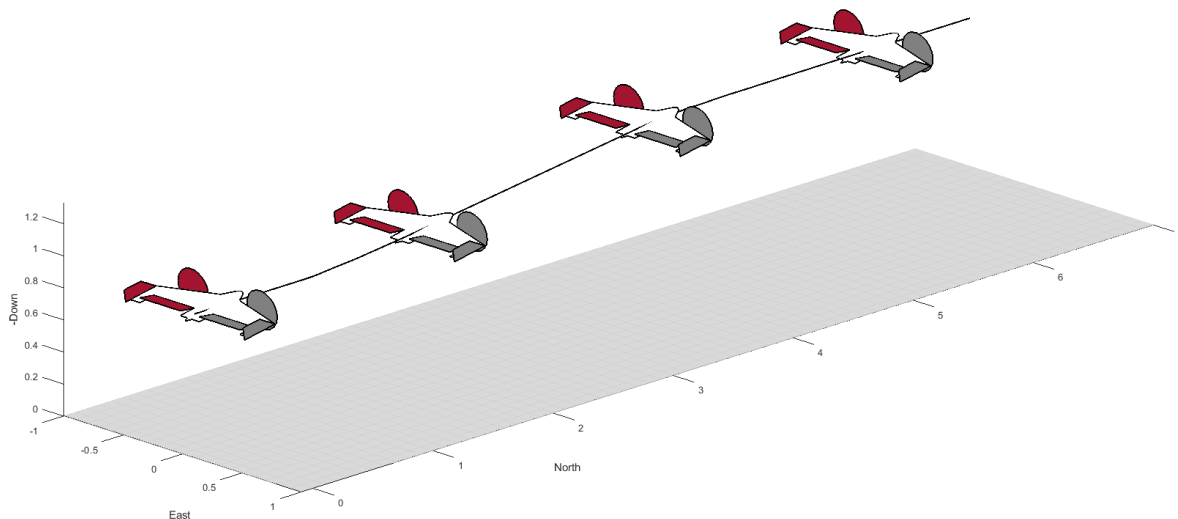


Figure 5.5: Step in handle 1 during Level flight (simulation visualization)

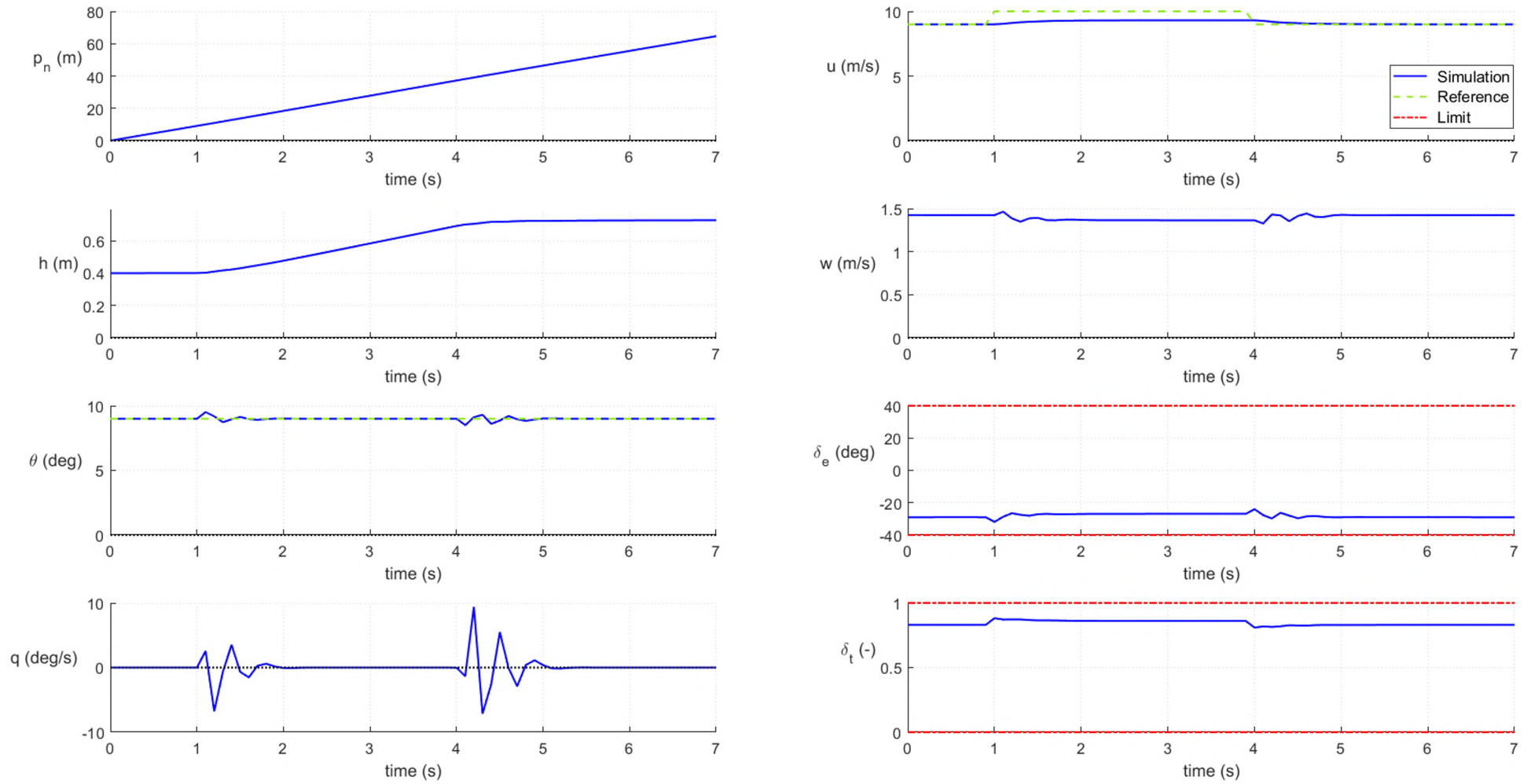


Figure 5.6: Step in handle 1 during Level flight

Step in Handle 2

In this case, at $t = 1$ s a step input is given in handle 2 which produces a pitch angle step request θ_{ref} . In response, the elevon signal δ_E increases in magnitude and the pitch angle θ quickly achieves the reference value with less than 1° of pitch angle error and just a light overshoot. The throttle signal δ_T responds with a small decrease and results in an aircraft u velocity reduction that was not requested and should be fixed. As intended, the aircraft climb velocity γ increases and the aircraft starts gaining altitude. At $t = 4$ s a negative step is given in handle 2 and the aircraft goes back to its initial condition. Note that a big impulse happens in the elevon signal δ_E which results in a big loss in the pitch angle. Because this impulse is quickly compensated and it results in almost no altitude loss, this is considered to be acceptable.

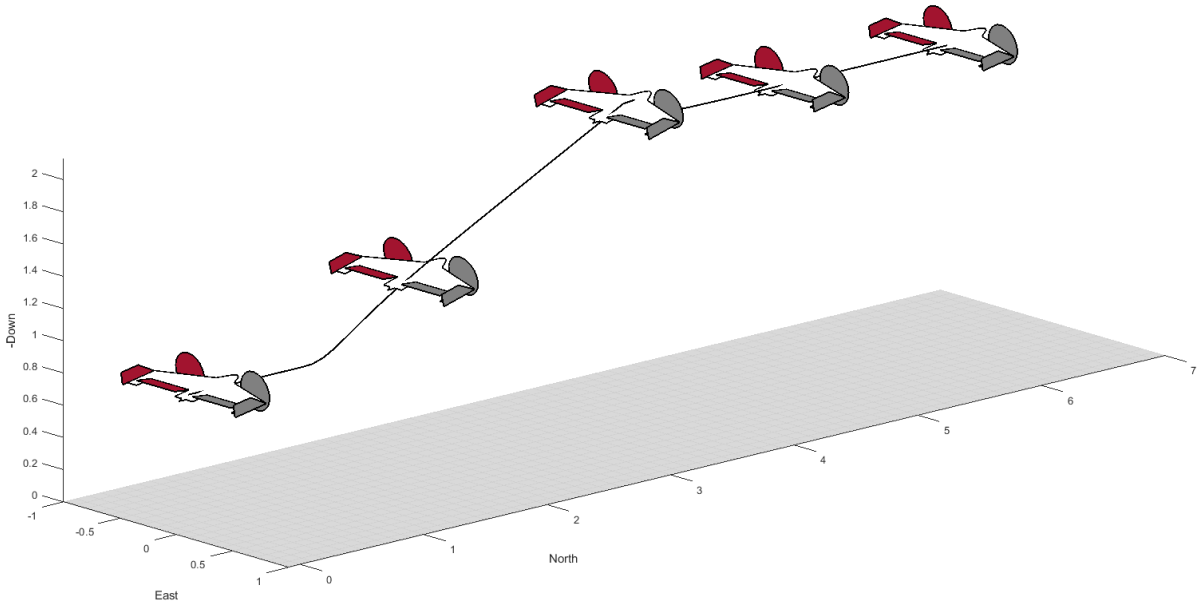


Figure 5.7: Step in handle 2 during Level flight (simulation visualization)

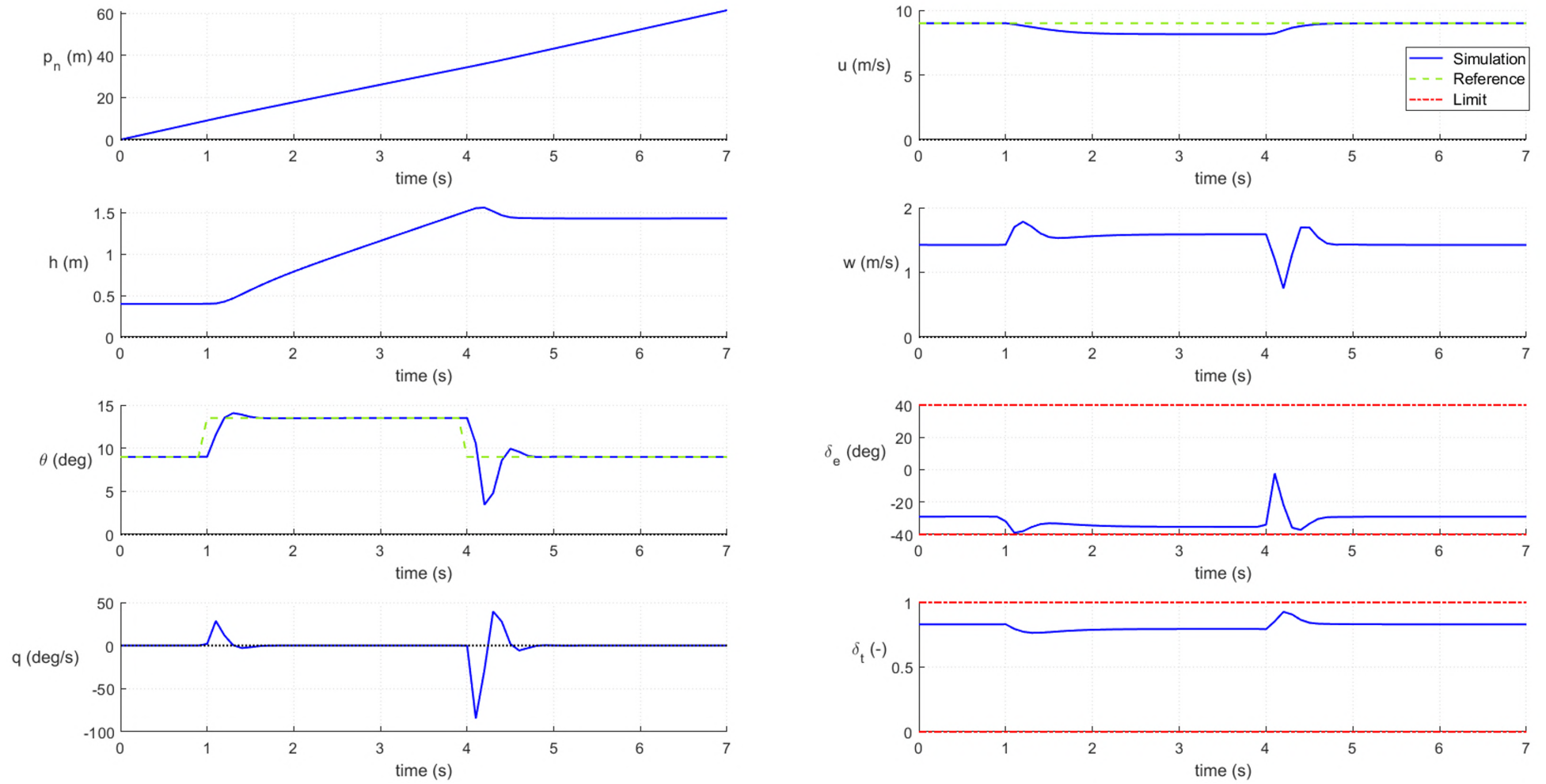


Figure 5.8: Step in handle 2 during Level flight

5.1.3 Transitions

Transition

In this simulation the aircraft is initialized in hover trim conditions, facing the North direction, above the ground. Flight Mode Switch starts in position 0 i.e. Hover mode, and Take-Off/Land Switch starts in position 1 i.e. Take-Off mode.

At $t = 1$ s the Mode Selector Switch is shifted and the Transition begins. The velocity and pitch angle references, u_{ref} and θ_{ref} , are not expected to be precisely followed, so the errors in the references tracking during the maneuver are accepted. The aircraft is able to perform the transition in 3 s with no oscillations or overshoot. This is possible due to the implementation of the anti-windup technique which allows the aircraft not to destabilize, besides of the actuator saturation of the throttle signal δ_T . During this maneuver, the aircraft gains about 2.5 m in altitude, which is considered to be acceptable, and takes less then 10 m in the north direction to reach the Level flight condition.

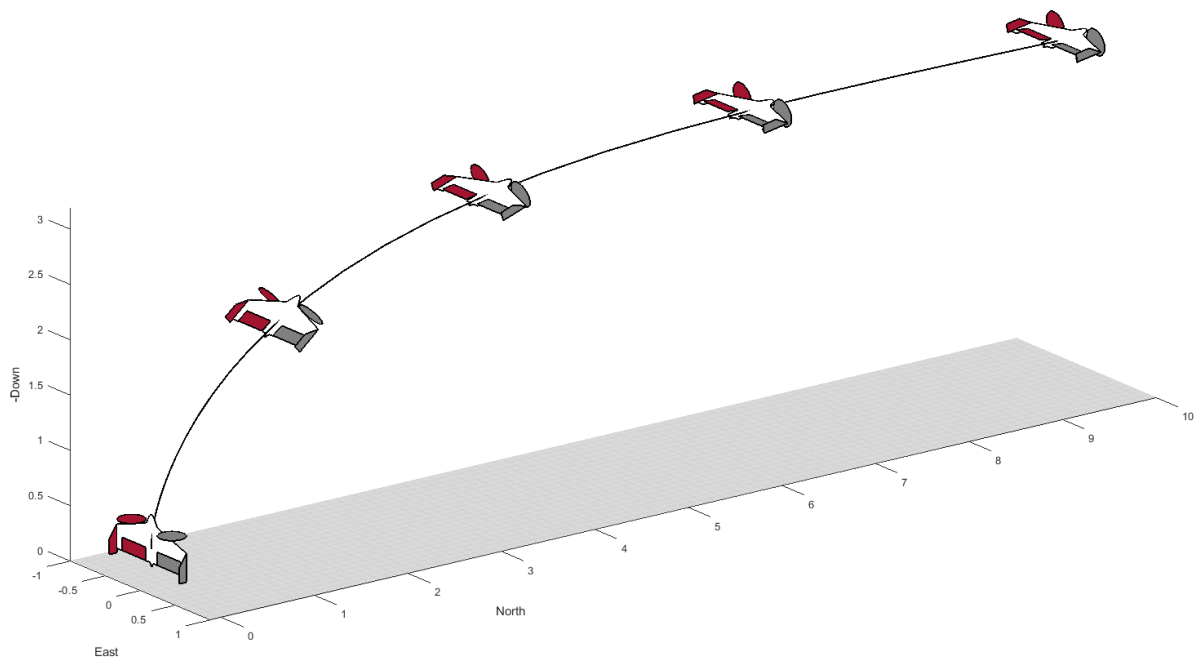


Figure 5.9: Transition from Hover to Level flight (simulation visualization)

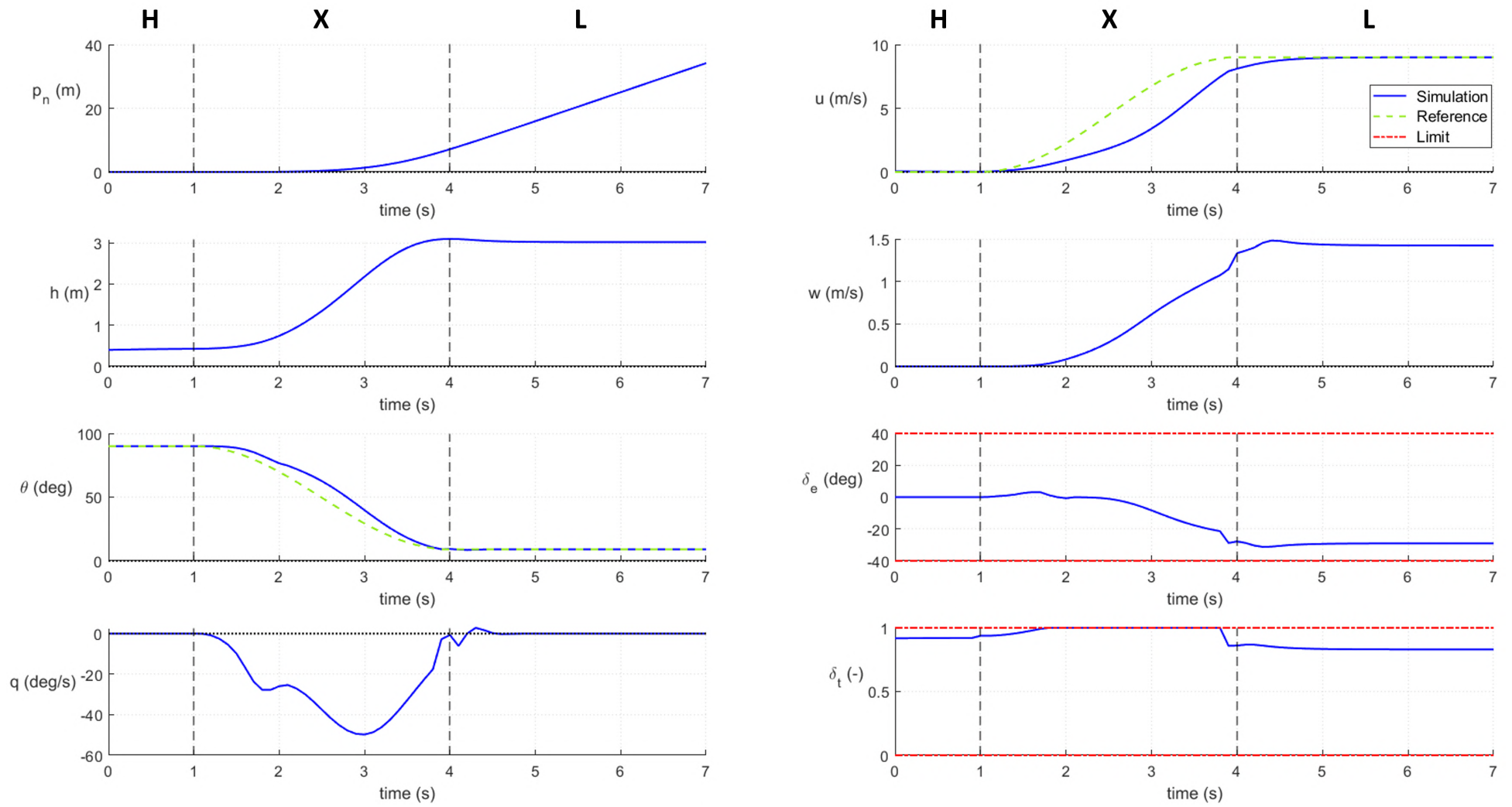


Figure 5.10: Transition from Hover to Level flight

Back Transition

In this simulation the aircraft is initialized in level trim conditions, facing the North direction, above the ground. Flight Mode Switch starts in position 1 i.e. Level mode, and Take-Off/Land Switch starts in position 1 i.e. Take-Off mode.

At $t = 1$ s the Mode Selector Switch is shifted and the Back Transition begins. The big step in the attitude request θ_{ref} immediately saturates the elevon signal δ_E until almost the end of the maneuver. The velocity reference u_{ref} makes the throttle signal δ_T also saturate as intended to increase the elevons' authority during the maneuver. By the end of the Back Transition, the pitch angle θ is at the hover trim condition, but the u velocity is much higher, about 5 m/s. The Hover controller quickly decreases this value to the desired velocity. As explained before, the actuators' saturation did not destabilized the aircraft since an anti-windup technique is implemented. Finally, the aircraft is capable of performing the back transition in less than 3 seconds which is within the previously defined requirements' limits.

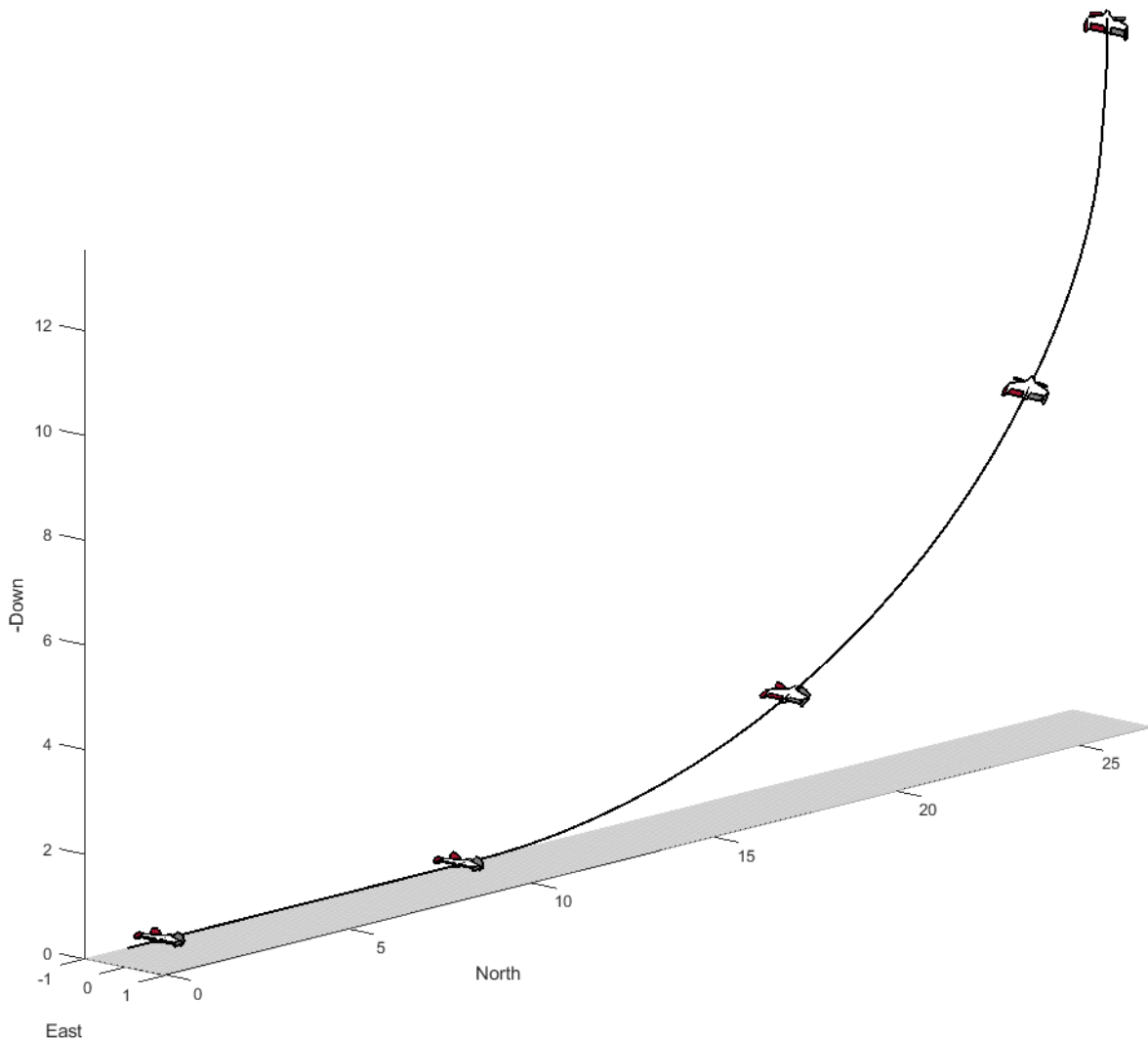


Figure 5.11: Back transition from Level to Hover flight (simulation visualization)

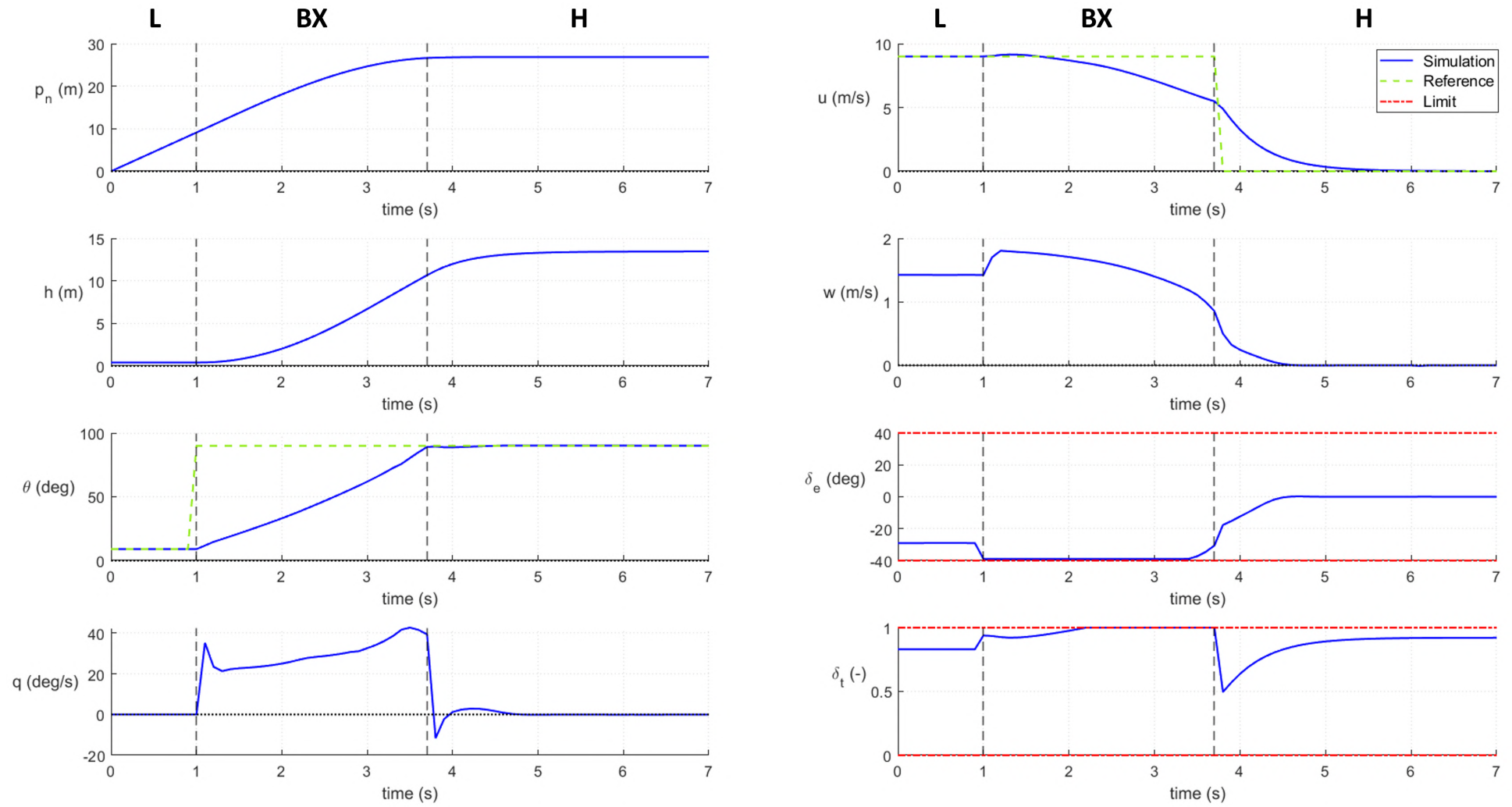


Figure 5.12: Back Transition from Level to Hover flight

5.1.4 Take-Off and Landing

In this simulation the aircraft is initialized in hover trim conditions, facing the North direction, touching the ground. Flight Mode Switch starts in position 0 i.e. Hover mode, and Take-Off/Land Switch starts in position 0 i.e. Landing mode.

At $t = 1$ s the Take-Off/Land Switch is shifted and the altitude controller is activated with an altitude reference of $h_{ref} = 1$ m. This controller produces a change in the velocity reference u_{ref} based on the altitude error while the pitch angle reference θ_{ref} remains constant. When the aircraft reaches the reference altitude, the altitude controller is turned off. At $t = 6$ s the Take-Off/Land Switch is shifted again and the altitude controller is activated with an altitude reference of $h_{ref} = 0.2$ m. This landing altitude reference value is chosen to avoid the aircraft crashing into the ground. The aircraft CG is located at $d_x = 0.13$ m and so there is some space for sensor or control errors. Note that, as intended, the aircraft smoothly reaches the reference altitude with no overshoots.

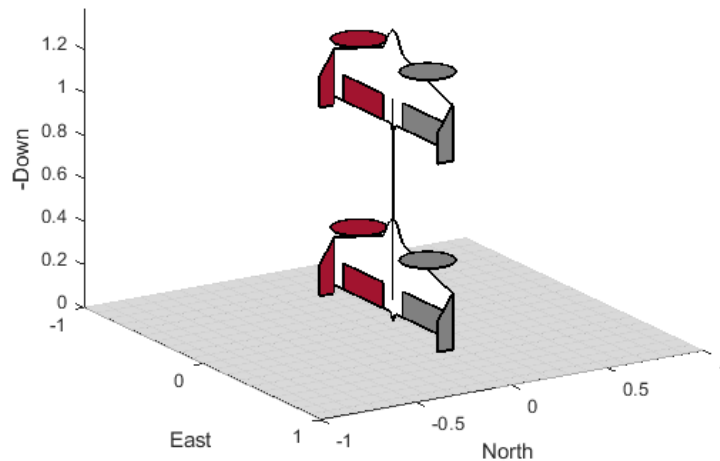


Figure 5.13: Take-Off and Land (simulation visualization)

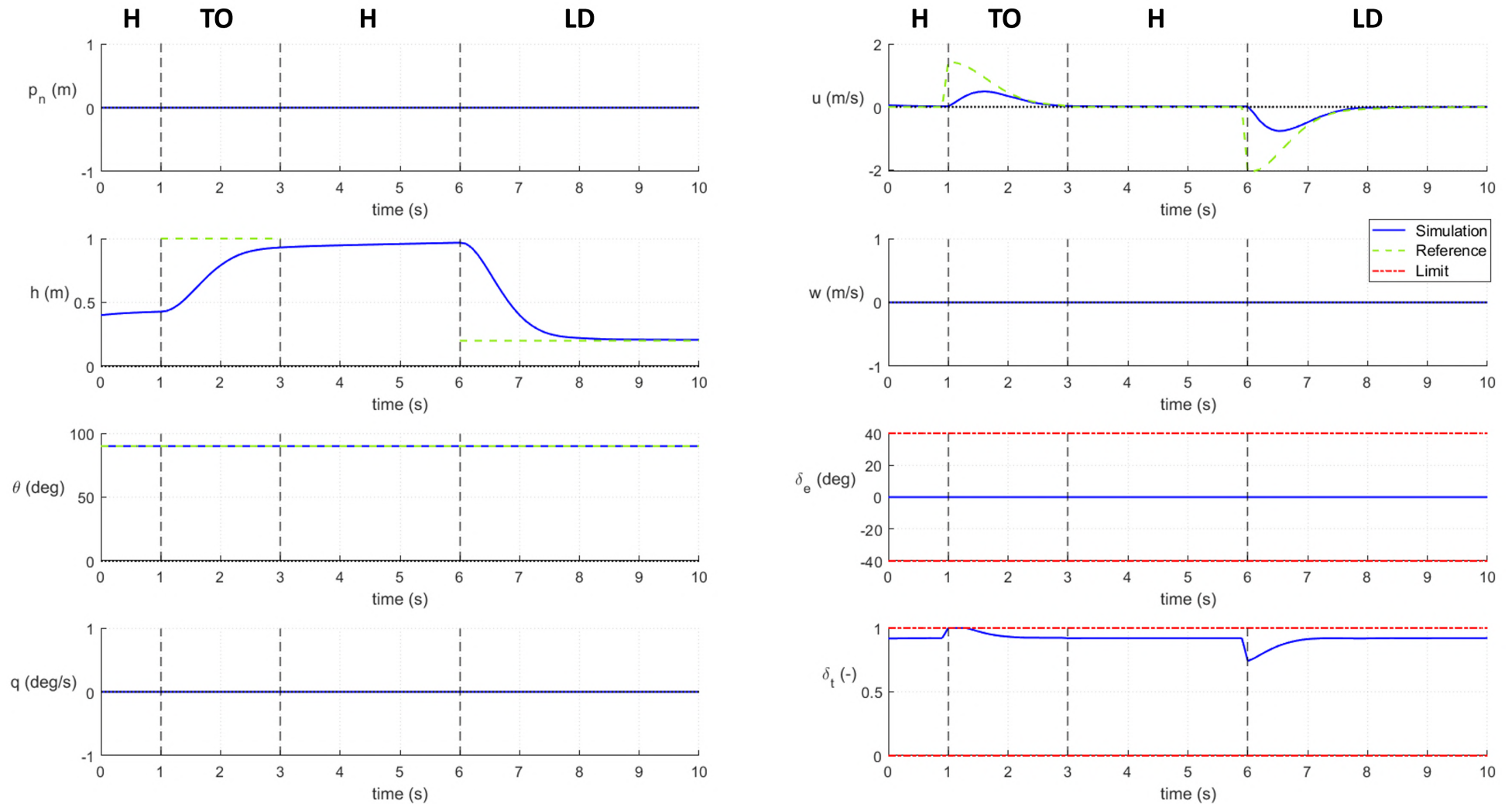


Figure 5.14: Take-Off and Land

5.2 Simulation Flight Mission

With all the controller blocks tested and the key flight phases analysed, the two minimum flight missions, described in 3.1, are run in the simulator. The states are simulated without noise and in no wind conditions. Both missions are initialized in hover trim conditions, facing the North direction, touching the ground. Flight Mode Switch starts in position 0 i.e. Hover mode, and Take-Off/Land Switch starts also in position 0 i.e. Land mode. Because each individual flight phase was already analysed, during this section, only some additional observations about the evolution of the different state requests and controller modes are made.

5.2.1 Mission 1

In the first mission, a Take-Off is commanded at $t = 1$ s and, at $t = 3$ s, when the aircraft is already at the take-off altitude reference, an unitary handle 1 signal is applied making the aircraft move forward with a vertical attitude, during 5 s. Then, the handle 1 signal is set to zero again and, at $t = 9$ s, the Landing is commanded. During the first two phases of this mission (take-off and step in handle 1), the aircraft behaves as analysed individually in the previous section. Then, when the negative handle 1 step input is given, the aircraft starts to stabilize in Hover position but the Landing command is given before it reaches steady-state condition. Although the aircraft starts landing in an unsteady condition, the controller is capable of stabilizing the aircraft during this flight phase. Lastly, the aircraft smoothly touches the ground and the simulated mission is considered to be succeeded.

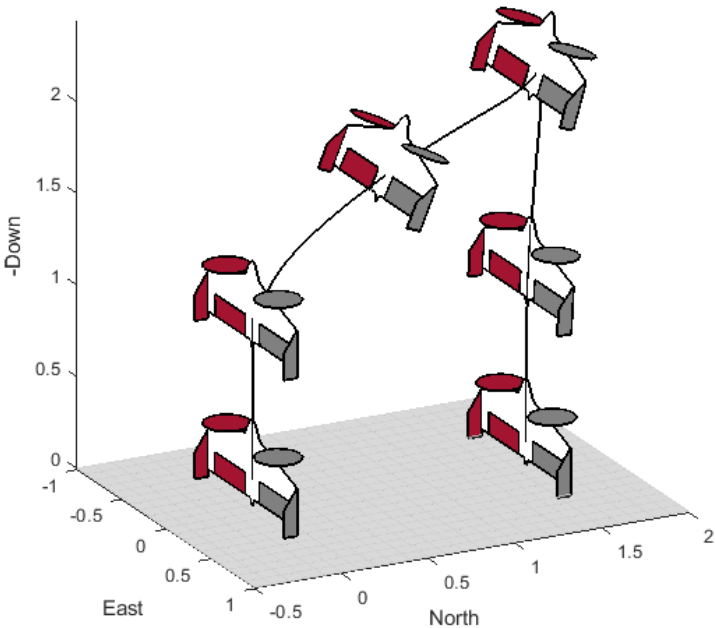


Figure 5.15: Mission 1 (simulation visualization)

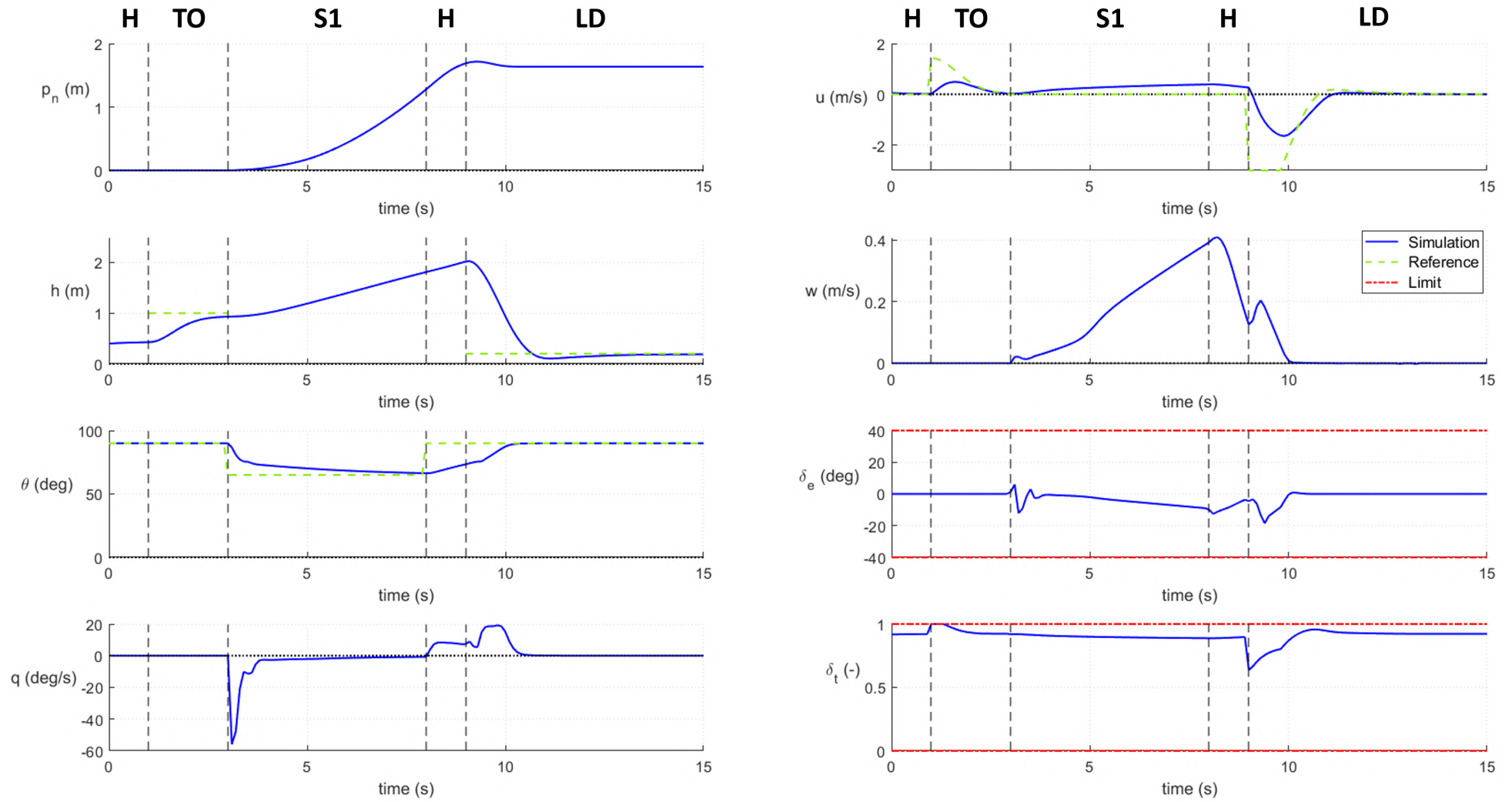


Figure 5.16: Mission 1

5.2.2 Mission 2

In the second mission, a Take-Off is commanded at $t = 1$ s and, at $t = 3$ s the Transition is requested, which develops as described in the transition analysis. At $t = 6$ s the stable level flight is achieved. After 1 second in Level mode, the Back Transition is also requested which ends in Hover flight at $t = 10$ s. After 2 s in this steady position, the Landing mode is activated, beginning at an altitude of 16 m, the aircraft takes almost 8 s to touch the ground.

The flight control system proved not only that each individual mode controller worked properly but also that the overall controller structure was able to change smoothly between flight phases.

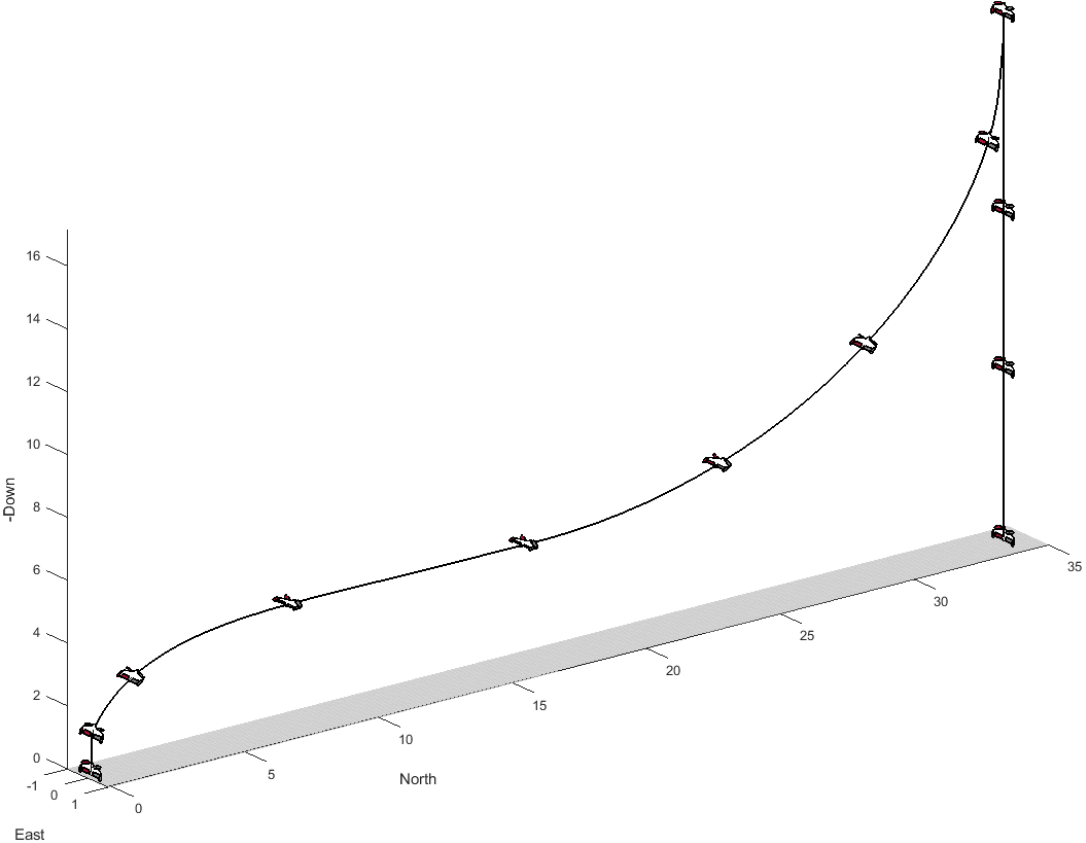


Figure 5.17: Mission 2 (simulation visualization)

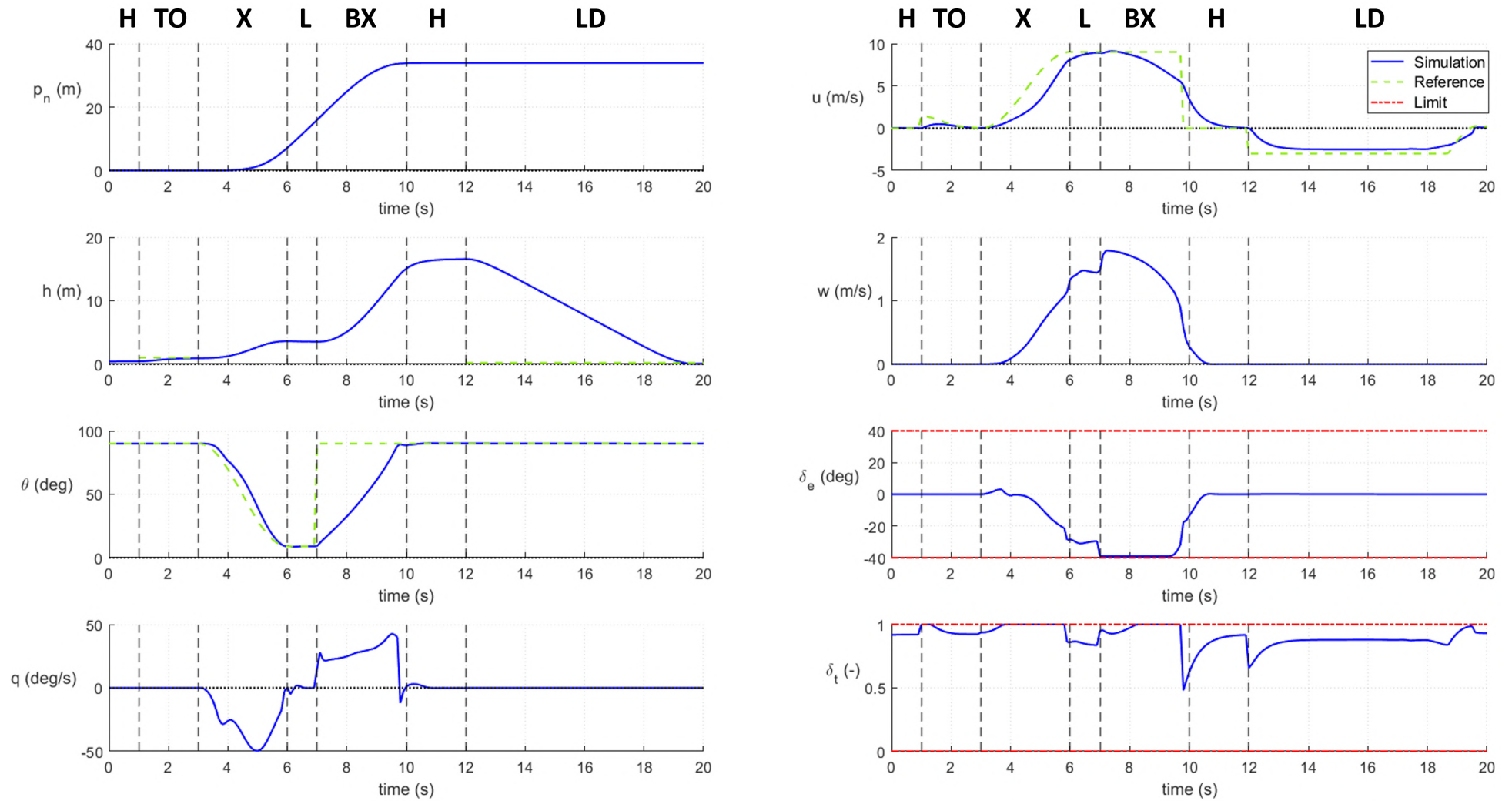


Figure 5.18: Mission 2

Chapter 6

Conclusions

6.1 Summary

This work detailed the design and simulation of the flight control system capable of performing autonomous take-off, landing and transition between flight modes of a particular tail-sitter UAV. The research undertaken was separated into three main tasks. First, it was developed a complete dynamic model of the chosen platform, which was implemented in a *Matlab/Simulink* simulator. Secondly, the understanding of the aircraft behaviour around the main operating points was studied based on linearization analysis techniques. Then, the definition of the aircraft flight envelope was useful to establish its operation limits. Finally, the longitudinal flight controller based on linear control techniques was developed and tested in several simulations. The main conclusion and achievements of this work are presented below, as well as some suggestions for future works that can be developed from this work.

6.2 Observations

The observations and comments made in this section cover the objectives stated for this dissertation:

- As proposed, a complete nonlinear model of the aircraft was implemented and describes the platform behaviour at an adequate level for a first iteration of the controller design. However, some experimental validation must be done and inaccuracies must be identified and corrected. One should remember that no sensors were modeled and, therefore, the real-time simulator implemented uses the direct state variables with no noise added which represents an unrealistic simulation.
- The longitudinal characterization and analysis proved to be extremely relevant in understanding the aircraft stability and differences between the two main flight operating modes. This comprehension enabled a better insight of aircraft control needs, which then, in the controller design, allowed a faster selection of LQR weightings and controller parameters.
- About the control design and structure, the choice of using linear control techniques proved to be feasible even against the challenges related to the aircraft unstable configuration in vertical flights and its poor performance in aerodynamic flights.

- Evaluating the simulations results, overall, the objectives were met. However, there is much room for improvement in most of the flight phases and reference tracking. The most critical cases are: the forward and backwards movement in Hover, since the altitude gain should be reduced and the time response could be smaller. During Level flight, when a velocity request is given, the aircraft does not maintain the level flight and starts climbing which should be corrected. The Transition was performed reasonably well, yet a smaller distance and altitude gain should be attempted. The Back Transition simulation demonstrated that the aircraft is easily capable of executing this maneuver. However, it revealed a big altitude gain which must be reduced. About autonomous Take-Off and Land, the altitude controller showed good results as the aircraft was able to track the references as intended.
- To conclude the controller testing, the two minimum flight missions that included all the critical phases of interest for this work were successfully completed in simulation. Therefore, it proved not only that each individual mode controller worked properly but also that the overall controller structure worked well and that it was able to change smoothly between flight phases.

6.3 Future Work

Base on the observations and conclusions of this work, some suggestions for future works on this platform control study are presented below:

- A model verification and more accurate parameter estimation should be done with real world testing. An accurate model simulator and a reliable platform for controller design, verification and validation are crucial for future control optimization.
- In order to make real experimental flights, sensors should be select and added to the aircraft platform. Therefore, they should be also modelled and included in the simulator.
- With the purpose of introducing more realism to simulations, a wind model could also be added, and controller analysis should be done in the presence of wind disturbances.
- To improve the aircraft weak performances detailed above, one feasible solution to be analysed is to add the altitude state in the LQR feedback controller, which could prevent altitude gains when they are not desired.

Bibliography

- [1] Eye In The Sky. Using High-Altitude Balloons for Decision Support in Wildfire Operations, 2020. URL <https://adai.pt/eyeinthesky/>. Accessed: 2022-03-21.
- [2] R. G. L. Narayanan and O. C. Ibe. Joint Network for Disaster Relief and Search and Rescue Network Operations. *Wireless Public Safety Networks 1: Overview and Challenges*, 2015. doi: 10.1016/B978-1-78548-022-5.50006-6.
- [3] Civilian applications: the challenges facing the UAV industry. *Air & Space Europe*, 1(5/6), 1999. doi: 10.1016/S1290-0958(00)88873-5.
- [4] W. Green and P. Oh. A MAV that flies like an airplane and hovers like a helicopter. *IEEE, International Conference on Advanced Intelligent Mechatronics*, 2005. doi: 10.1109/aim.2005.1511063.
- [5] D. Caverly. Aerodynamic Characterization and Control of a Tail-Sitter Aircraft Taking Off in a Crosswind. Master's thesis, McGill University, Montreal, Canada, April 2020.
- [6] W. F. Chana and J. F. S. Coleman. *World's First VTOL Airplane Convair/Navy XFY-1 Pogo*, volume 105. SAE International, 1996. doi: 10.4271/962288.
- [7] K. Kita, A. Konno, and M. Uchiyama. Hovering Control of a Tail-Sitter VTOL Aerial Robot. *Journal of Robotics and Mechatronics*, 21:277–283, 2009. doi: 10.20965/jrm.2009.p0277.
- [8] Convair XFY-1 Aircraft, 2010. URL https://upload.wikimedia.org/wikipedia/commons/thumb/8/84/Convair_XFY-1_Pogo_2.jpg/523px-Convair_XFY-1_Pogo_2.jpg. Accessed: 2022-03-23.
- [9] V. Myrand-Lapierre, A. Desbiens, E. Gagnon, F. Wong, and É. Poulin. Transitions between level flight and hovering for a fixed-wing mini aerial vehicle. *Proceedings of the 2010 American Control Conference, ACC 2010*, June 2010. doi: 10.1109/acc.2010.5530875.
- [10] P. Casau. Autonomous Transition Flight for a Vertical Take-Off and Landing aircraft. Master's thesis, Instituto Superior Tecnico, Universidade de Lisboa, September 2011.
- [11] J. Carlos and L. Parreira. Modelling, Simulation, Analysis and Stabilization of a Flying Wing. Master's thesis, Instituto Superior Tecnico, Universidade de Lisboa, November 2019.
- [12] S. C. Kriel. A Comparison of Control Systems for the Flight Transition of VTOL Unmanned Aerial Vehicles. Master's thesis, University of Stellenbosch, Matieland, South Africa, March 2008.

- [13] Y. Jung and D. H. Shim. Development and application of controller for transition flight of tail-sitter UAV. *Journal of Intelligent and Robotic Systems: Theory and Applications*, February 2011. doi: 10.1007/s10846-011-9585-1.
- [14] P. Casau, D. Cabecinhas, and C. Silvestre. Hybrid Control Strategy for the Autonomous Transition Flight of a Fixed-Wing Aircraft. *IEEE Transaction on Control Systems Technology*, 21(6), November 2013.
- [15] N. B. Knoebel and T. W. McLain. Adaptive Quaternion Control of a Miniature Tailsitter UAV. *American Automatic Control Council*, June 2008.
- [16] R. Chiappinelli. Modeling and Control of a Tailsitter UAV. Master's thesis, McGill University, Montreal, Canada, April 2018.
- [17] Horizon Hobby. E-flite X-VERT VTOL BNF Basic, 504mm, 2022. URL <https://www.horizonhobby.com/>. Accessed: 2022-03-25.
- [18] T. W. M. Randal W. Bread. *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012. ISBN 9772081415.
- [19] W. Khan. *Dynamics Modeling of Agile Fixed Wing Unmanned Aerial Vehicles*. PhD thesis, McGill University, Montreal, Canada, April 2016.
- [20] A. Moutinho and J. Azinheira. Drone-X Simulacao e Controlo de Drones. Instituto Superior Tecnico, 2018.
- [21] J. Leishman. *Principles of Helicopter Aerodynamics*. Cambridge Aerospace Series, December 2006. ISBN 9781107013353.
- [22] B. W. McCormick. *Aerodynamics, Aeronautics and Flight Mechanics*. John Wiley & Sons Inc, September 1994. doi: 10.1177/095441009721100102.
- [23] Department of Aerospace Engineering of University of Illinois at Urbana-Champaign. UIUC Propeller Data Site, 2022. URL <https://m-selig.ae.illinois.edu/props/propDB.html>. Accessed: 2022-05-4.
- [24] W. Khan and M. Nahon. Modeling Dynamics of Agile Fixed-Wing UAVs for Real-Time Applications. *IEEE International Conference on Unmanned Aircraft Systems*, June 2016.
- [25] S. F. Hoerner. *Fluid Dynamic Drag*. Hoerner, Sighard. F., 1965. doi: 10.1093/benz/9780199773787.article.b00088376.
- [26] O'Reilly. Concept of Stability and Routh-Hurwitz Criterion, 2022. URL <https://www.oreilly.com>. Accessed: 2022-08-1.
- [27] J. Azinheira. Controlo de Voo. Instituto Superior Tecnico, Universidade de Lisboa, February 2020.

- [28] G. Ellis. Chapter 13 - Model Development and Verification. In *Control System Design Guide*, pages 261–282. Butterworth-Heinemann, fourth edition edition, 2012. ISBN 978-0-12-385920-4. doi: 10.1016/B978-0-12-385920-4.00013-8.
- [29] B. D. O. Anderson and J. B. Moore. *Optimal Control: Linear Quadratic Methods*. Prentice-Hall International, Inc., 1989. ISBN 9780486457666.
- [30] J.-J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice-Hall International, Inc., 1991.
- [31] A. Visioli. *Practical PID Control*. Springer, 2006. doi: 10.1007/1-84628-586-0.
- [32] Flash RC. BL280 Brushless Outrunner Motor, 2600Kv, 2022. URL https://www.flashrc.com/e_flite/24231-bl280_brushless_outrunner_motor_2600kv.html. Accessed: 2022-05-1.
- [33] J. Sola. Quaternion kinematics for the error-state Kalman filter. November 2017.
- [34] B. L. Stevens, F. L. Lewis, and E. N. Johnson. *Aircraft Control and Simulation*, chapter The Kinematics and Dynamics of Aircraft Motion. John Wiley & Sons, 1992. ISBN 9781118870983.

Appendix A

X-Vert Parameters

This appendix resumes the X-Vert parameters gathered from different references.

Table A.1: Aircraft Body Parameters, values from [16].

| Parameter | Symbol | Value | Unit |
|--------------------------------|----------|------------------------|-----------|
| Mass | m | 0.21 | kg |
| Moments of Inertia | I_x | $3.002 \cdot 10^{-3}$ | $kg\ m^2$ |
| | I_y | $6.245 \cdot 10^{-4}$ | $kg\ m^2$ |
| | I_z | $3.538 \cdot 10^{-3}$ | $kg\ m^2$ |
| Non-zero Products of Inertia | I_{xz} | $-14.03 \cdot 10^{-6}$ | $kg\ m^2$ |
| CG Position from Trailing Edge | d_x | 130 | mm |

The position of the CG in the geometric frame \mathbb{F}_G is:

$$\mathbf{r}_{CG} = [d_x, 0, 0]' \quad (\text{A.1})$$

Table A.2: Thruster Parameters, values from [16].

| Parameter | Symbol | Value | Unit |
|-----------------------------|------------|-----------------------|---------|
| Battery Voltage | V_{batt} | 7.4 | V |
| Thruster Rotational Inertia | I_{th} | $1.626 \cdot 10^{-6}$ | $kg\ m$ |
| Propeller Radius | r_p | 62.5 | mm |
| | $x_{p,G}$ | 173 | mm |
| Propeller Position | $y_{p,G}$ | 145 | mm |
| | $z_{p,G}$ | 0 | mm |

The position of the right propeller in the geometric frame \mathbb{F}_G is:

$$\mathbf{r}_{prop,G} = [x_{p,G}, y_{p,G}, z_{p,G}]' \quad (\text{A.2})$$

The left propeller has a negative y coordinate. The propellers position in the body frame \mathbb{F}_B is:

$$\mathbf{r}_{prop} = \mathbf{r}_{prop,G} - \mathbf{r}_{CG} \quad (\text{A.3})$$

Table A.3: Electric Motor Parameters, values from [32].

| Parameter | Symbol | Value | Unit |
|--------------------------|--------|----------------------|-----------------|
| Inductance | L_m | $3.6 \cdot 10^{-4}$ | H |
| Resistance | R_m | 0.032 | Ω |
| Back-electromotive Force | K_e | $9.68 \cdot 10^{-2}$ | $V/(rad/s)$ |
| Rotational Inertia | J_m | $1.4 \cdot 10^{-5}$ | $kg\ m^2$ |
| Torque Constant | K_t | $2.47 \cdot 10^{-2}$ | $(N\ m)/A$ |
| Damping Constant | B_m | $2.9 \cdot 10^{-4}$ | $(N\ m\ s)/rad$ |

Table A.4: Wing Geometric Parameters, values from [16].

| Parameter | Symbol | Value | Unit |
|------------------------|-----------|--------|-------|
| Wing Area | S | 0.0798 | m^2 |
| Wing Span | b | 0.5 | m |
| Mean Aerodynamic Chord | \bar{c} | 0.16 | m |
| Wing Aspect Ratio | A_R | 3.13 | - |
| Delta Wing Sweep Angle | Λ | 19.8 | deg |

The mean aerodynamic chord is:

$$\bar{c} = S/b \quad (\text{A.4})$$

The wing aspect ratio is:

$$A_R = b^2/S \quad (\text{A.5})$$

Table A.5: Aerodynamic Parameters, values from [16].

| Parameter | Symbol | Value | Unit |
|---------------------------------|----------------|----------------------|-----------|
| Control Derivative Coefficients | c_x | $9.91 \cdot 10^{-4}$ | m^3/rad |
| | c_y | $4.74 \cdot 10^{-4}$ | m^3/rad |
| | b_x | $9.37 \cdot 10^{-4}$ | m^3/rad |
| | b_y | $3.48 \cdot 10^{-4}$ | m^3/rad |
| Maximum Elevons Deflection | δ_{max} | 39 | deg |
| Skin Friction Coefficient | $C_{D,0}$ | 0.02 | |
| Oswald's Efficiency Factor | k_0 | 0.87 | |

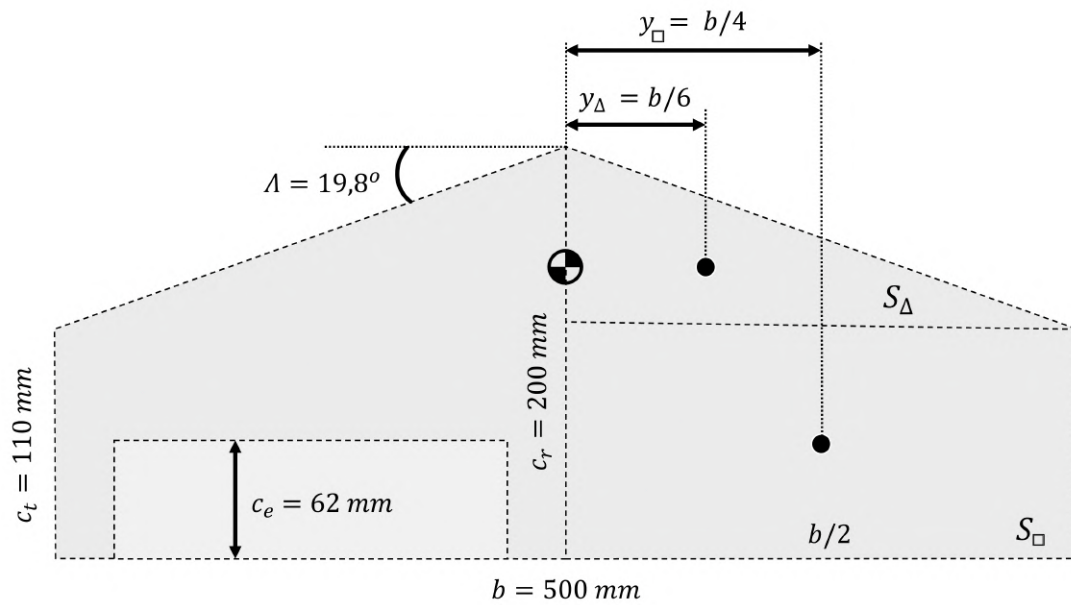


Figure A.1: Flying Wing Dimensions

The half wing lateral position is given by:

$$y_{trapeze} = \frac{y_{\Delta} S_{\Delta} + y_{\square} S_{\square}}{S_{\Delta} + S_{\square}} \quad (\text{A.6})$$

Table A.6: Aerodynamics coefficients, values from [11].

| | | | | | |
|------------------------|------------|-----------|----------|-----------|-----------|
| $C_{L_{\dot{\alpha}}}$ | 0 | C_{L_q} | 3.1851 | - | - |
| $C_{m_{\dot{\alpha}}}$ | -0.4897 | C_{m_q} | -2.4487 | - | - |
| $C_{Y_{\beta}}$ | -0.0024922 | C_{Y_p} | 0.26198 | C_{Y_r} | -0.06725 |
| $C_{l_{\beta}}$ | -0.16039 | C_{l_p} | -0.45055 | C_{l_r} | 0.31074 |
| $C_{n_{\beta}}$ | 0.03903 | C_{m_p} | -0.18904 | C_{n_r} | 0.0028225 |

Table A.7: Coordinates of Reference Points for Contact Force Calculations, given in the geometric reference frame G, in meters.

| $x_{k,G}$ | $y_{k,G}$ | $z_{k,G}$ |
|-----------|-----------|-----------|
| -0.110 | -0.251 | 0.071 |
| -0.110 | -0.251 | -0.071 |
| -0.110 | 0.251 | 0.071 |
| -0.110 | 0.251 | -0.071 |

Appendix B

Quaternions

In this appendix, all the following definitions and relations are according to [18], [33] and [34].

B.1 Definition

A quaternion is a four elements number used in three-dimensional rotations and orientations. A quaternion number is represented in the form:

$$\mathbf{q} = q_0 + q_1i + q_2j + q_3k = q_0 + \mathbf{q}_v^T \begin{bmatrix} i \\ j \\ k \end{bmatrix}$$

where i, j, k are unit imaginary numbers. As shown, it is composed by a scalar part q_0 and a vector part \mathbf{q}_v , which may be thought of as a rotation of an angle Θ about a three-dimensional unit vector \mathbf{u} .

$$\mathbf{q} = \begin{bmatrix} \cos(\Theta/2) \\ \mathbf{u} \sin(\Theta/2) \end{bmatrix}$$

Furthermore, when a quaternion is used to represent a rotation, it is required that to be a unit quaternion, which means the Euclidean norm of the quaternion must be 1 at all time:

$$\|\mathbf{q}\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1$$

Also, a quaternion \mathbf{q} is mostly represented as a 4-dimensional vector:

$$\mathbf{q} = \begin{bmatrix} q_0 \\ \mathbf{q}_v \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

B.1.1 Quaternion Product

The quaternion product \otimes of two quaternions p and q represents the rotation by p followed by a rotation by q and is defined as:

$$p \otimes q = \begin{bmatrix} p_0q_0 - p_1q_1 - p_2q_2 - p_3q_3 \\ p_0q_1 + p_1q_0 + p_2q_3 - p_3q_2 \\ p_0q_2 - p_1q_3 + p_2q_0 + p_3q_1 \\ p_0q_3 + p_1q_2 - p_2q_1 + p_3q_0 \end{bmatrix} \quad (\text{B.1})$$

Quaternions obey the normal laws of algebra, except that multiplication is not commutative, which means:

$$p \otimes q \neq q \otimes p$$

During the computational implementation it is used the Matlab function $quatprod = quatmultiply(p,q)$ to compute the quaternion product.

B.2 Conversion between Euler Angles and Quaternions

In simulation and control, it is often desired to keep track of a certain attitude. Due to Euler angles simpler interpretation, to initialize a quaternion usually it is easier to convert it from an Euler attitude representation. Directly, the quaternion can be obtained from standard roll-pitch-yaw (ϕ, θ, ψ) Euler angles as:

$$\begin{aligned} q_0 &= \cos \frac{\psi}{2} \cos \frac{\theta}{2} \cos \frac{\phi}{2} + \sin \frac{\psi}{2} \sin \frac{\theta}{2} \sin \frac{\phi}{2} \\ q_1 &= \cos \frac{\psi}{2} \cos \frac{\theta}{2} \sin \frac{\phi}{2} - \sin \frac{\psi}{2} \sin \frac{\theta}{2} \cos \frac{\phi}{2} \\ q_2 &= \cos \frac{\psi}{2} \sin \frac{\theta}{2} \cos \frac{\phi}{2} + \sin \frac{\psi}{2} \cos \frac{\theta}{2} \sin \frac{\phi}{2} \\ q_3 &= \sin \frac{\psi}{2} \cos \frac{\theta}{2} \cos \frac{\phi}{2} - \cos \frac{\psi}{2} \sin \frac{\theta}{2} \sin \frac{\phi}{2} \end{aligned} \quad (\text{B.2})$$

On the other hand, although not explicitly necessary for simulation purposes, Euler angles can be retrieved from unit quaternion as:

$$\begin{aligned} \phi &= \text{atan2}(2q_2q_3 + 2q_0q_1, q_3^2 - q_2^2 - q_1^2 + q_0^2) \\ \theta &= \text{asin}(2q_0q_2 - 2q_1q_3) \\ \psi &= \text{atan2}(2q_1q_2 + 2q_0q_3, q_0^2 + q_1^2 - q_2^2 - q_3^2) \end{aligned} \quad (\text{B.3})$$

During the computational implementation there are used the Matlab functions $eul2quat$ and $quat2eul$ for the specified conversions. Nevertheless, as explained in the previous section, the quaternion multiplication is not commutative. The 'ZYX' sequence is already the default sequence and there is no need to specified as a parameter of the Matlab function.

B.2.1 Euler pitch angle in hover

Although the quaternion formulation provides more computationally efficient and lack of singularity problems, because of quaternion unitary constraint (quaternion Euclidean norm must be 1) their effect is highly non-linear. A quaternion increasing from 0.9 to 1.0 may represent a much larger change in angle than an increase from 0.0 to 0.1. Thence, for linear control purposes quaternions will deliver inconsistent results and Euler angle representation should be used. However, because the tail-sitter configuration includes the discontinuity point in its flight envelop, a way of contouring this issue about the vertical position should be deduced.

Therefore, if one considers a pure longitudinal movement ($\phi = \psi = 0$), converting Euler to quaternions gives q_v and, in the hovering condition ($\text{com } \theta_0 = \pi/2$) it equals to q_{v0} , as in the following expressions.

$$q_v = \begin{bmatrix} \cos(\theta/2) \\ 0 \\ \sin(\theta/2) \\ 0 \end{bmatrix} \quad \text{and} \quad q_{v0} = \begin{bmatrix} \sqrt{2}/2 \\ 0 \\ \sqrt{2}/2 \\ 0 \end{bmatrix}$$

Linearization around this condition gives:

$$\delta q_v = q_v - q_{v0} = \frac{\partial q_v}{\partial \theta} \delta \theta = \frac{1}{2} \begin{bmatrix} -\sin(\theta_0/2) \\ 0 \\ \cos(\theta_0/2) \\ 0 \end{bmatrix} \delta \theta = \frac{1}{2} \begin{bmatrix} -\sqrt{2}/2 \\ 0 \\ \sqrt{2}/2 \\ 0 \end{bmatrix} \delta \theta$$

From q_2 , $\delta \theta$ can be obtained continuously:

$$\delta \theta = 2\sqrt{2} \left(q_2 - \frac{\sqrt{2}}{2} \right) \quad (\text{B.4})$$

valid for θ close to $\pi/2$. It is defined as the linear zone, in other words, the limits for the applications of this linearisation, only small variations around $\theta_{v0} = \pi/2$ that allows an error of less than 1° during the conversion.

$$\theta_v = \theta_{v0} + \delta \theta_v \quad \rightarrow \quad q(\psi, \theta_v, \phi) \quad \rightarrow \quad \delta \theta$$

$$\text{error} = \delta \theta_v - \delta \theta$$

Consequently, $\delta \theta_v \in [-15, 15]^\circ$.

B.3 Conversion between Quaternion and Euler angles derivatives

In order to obtain a linearized decoupled mathematical model, the quaternion derivatives from the systems dynamics are converted to Euler angles derivatives making use of the following transformation:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T_q^{\dot{\Phi}} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \quad (\text{B.5})$$

where $T_q^{\dot{\Phi}}$ is the transformation matrix from quaternion derivatives to Euler angles derivatives. Since this transformation is not easy to directly deduce, two sequential transformations are applied:

$$T_q^{\dot{\Phi}} = T_\omega^{\dot{\Phi}} T_q^\omega \quad (\text{B.6})$$

First, the quaternion derivatives are converted into angular rates using T_q^ω , and then the angular rates are converted into Euler angles derivatives using $T_\omega^{\dot{\Phi}}$.

$$T_\omega^{\dot{\Phi}} = \begin{bmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) \sec(\theta) & \cos(\phi) \sec(\theta) \end{bmatrix} \quad T_q^\omega = 2 \begin{bmatrix} -q_1 & q_0 & q_3 & q_2 \\ -q_2 & -q_3 & q_0 & q_1 \\ -q_3 & -q_2 & -q_1 & -q_0 \end{bmatrix} \quad (\text{B.7})$$