



**Politecnico
di Torino**

Politecnico di Torino

Master's Degree in Aerospace Engineering

Definition of Requirements for Horizontal Landing of a Re-entry Vehicle

Supervisors

Prof. Lorenzo Casalino

Ing. Martins Sudars

Ing. Andrea Musacchio

Candidate

Marco Siragusa

Academic year 2024-2025

Contents

List of Figure	3
Abstract	5
1 Introduction	7
2 Flight Dynamics Concepts and Optimization	9
2.1 Principles of Flight Mechanic	9
2.1.1 Coordinates frames	10
2.1.2 Reference Frames	11
2.1.3 Aerodynamic forces	12
2.1.4 Aerodynamic moments	13
2.1.5 Damping moments	15
2.1.6 Control surfaces and moments	15
2.2 Guidance Navigation and Control Theory	16
2.2.1 Architecture of GNC System	17
2.2.2 The Closed-Loop	18
2.2.3 PID Controller	18
2.3 Optimization	19
2.3.1 Parallelization and MATLAB Parallel Computing Toolbox . .	22
3 The Landing Problem	24
4 Optimization Approach	27
4.1 Chosen Strategy	27
4.2 Mathematical Model	28
4.2.1 Forces and Moments	32
4.2.2 Control System	34
4.3 Optimizer configuration	34
4.3.1 Constraints	36
5 Experimental Analysis	37
5.1 Simulation environment	37

5.1.1	Hypothesis and features	37
5.1.2	Descent Guidance black-box	39
5.1.3	Descent Control black-box	40
5.1.4	Landing black box	43
5.2	Optimization environment	45
5.2.1	Optimizer Choice	45
5.2.2	Rationale for the Chosen Method	47
5.3	Initial Conditions, Constraints and Cost Function	48
5.3.1	Descent Guidance Optimization	49
5.3.2	Descent Control Optimization	51
5.3.3	Landing Optimization	51
5.4	First Results	53
6	Overall Requirements Definition and Future Development	63
6.1	Results	64
6.2	Future Developments and Reflections on Research	66
7	Conclusions	70
	Bibliography	72

List of Figures

2.1	Wind (W) and body (B) frames [7]	10
2.2	Wind, Body and Earth fixed body frames	11
2.3	Scheme of aerodynamic and gravity forces [9]	13
2.4	Forces and Moments scheme	15
2.5	Example of body flap and elevons in Space Shuttle [10]	16
2.6	GNC system [12]	18
2.7	PID controller scheme [15]	19
2.8	Optimization algorithm scheme [14]	21
3.1	HLV's landing problem	25
4.1	Division in <i>Descent</i> and <i>Landing</i> simulations	29
4.2	Guidance Descent Blackbox scheme	30
4.3	Control Descent and Landing Blackbox scheme	30
4.4	Descent optimization approach	35
4.5	Landing optimization approach	35
5.1	Simulink <i>Descent Guidance</i> simulation	40
5.2	Simulink <i>Descent Control</i> simulation	43
5.3	Simulink <i>Landing</i> simulation	44
5.4	Initial Conditions and landing site	48
5.5	$z(t)$	53
5.6	$v_x(t)$ e $v_z(t)$	54
5.7	$\theta(t)$	54
5.8	$\alpha_{id}(t)$ vs $\alpha(t)$	55
5.9	$\alpha_{id}(t)$ vs $\alpha(t)$, first seconds	56
5.10	$\alpha_{id}(t)$ vs $\alpha(t)$, flare maneuver and last seconds	56
5.11	$FPA(t)$	57
5.12	$\delta_b(t)$	57
5.13	$\delta_e(t)$	58
5.14	$x(t)$	59
5.15	$v_x(t)$	60
5.16	$\theta(t)$	60

List of Figures

5.17	$q(t)$	61
5.18	$\delta_b(t)$	61
5.19	$\delta_e(t)$	62
6.1	Final Results	65

Abstract

The growing interest in reusable space transportation systems has renewed the focus on horizontal landing capabilities for re-entry vehicles. While parafoil-assisted recovery remains a widely studied and applied method, particularly for small-scale or experimental platforms, current trends in aerospace engineering and spaceflight operations emphasize the need for autonomous runway landings to improve operability, reusability, and mission flexibility. Within this context, lifting-body and winged configurations are increasingly regarded as viable solutions, although the minimum aerodynamic and performance requirements for their final descent and landing remain insufficiently defined.

This thesis addresses this gap by developing a methodology to identify the fundamental requirements that a re-entry vehicle must satisfy—expressed in terms of lift-to-drag ratio (L/D), lift-to-weight ratio (L/W), and other key aerodynamic performance metrics—to successfully perform a controlled horizontal landing on a runway without the need for parafoil systems. To support the analysis, a simplified automatic guidance model was implemented, enabling the simulation of the terminal descent and landing phase under various aerodynamic configurations and constraints.

A central component of this research is the implementation of an optimization framework to evaluate vehicle behavior and validate the feasibility of safe runway landing for representative design cases. Optimization methods are a cornerstone in aerospace design, as they allow the systematic treatment of nonlinear dynamics, operational constraints, and multi-objective trade-offs. In this work, the *Sequential Quadratic Programming* (SQP) algorithm, as implemented in the *Sparse Nonlinear OPTimizer* (SNOPT), was adopted. SNOPT is particularly suited for large-scale constrained optimization problems, where sparsity in the constraint Jacobian can be exploited to achieve computational efficiency. Its robustness and flexibility make it a standard tool in trajectory optimization and guidance problems, enabling the incorporation of realistic aerodynamic models, vehicle dynamics, and operational constraints within a unified optimization environment.

In the proposed formulation, the optimization parameters are defined as the time-dependent deflection angles of two control surfaces, which directly influence the aerodynamic forces and moments during the final descent. By optimizing these control inputs, the guidance model is capable of adapting the trajectory to meet

runway-landing requirements while satisfying aerodynamic and dynamic constraints. The optimization process was carried out within a three-degree-of-freedom (3-DOF) simulation environment, which provided a computationally efficient yet sufficiently accurate representation of the vehicle's motion during the terminal phase. The underlying model was initially inspired by the Space Shuttle and subsequently generalized to encompass generic winged-body and lifting-body configurations, thereby ensuring the applicability of the methodology to a broader class of reusable re-entry vehicles.

The outcome of this research is a comprehensive mapping of requirements for the final re-entry and landing phase, offering a set of guidelines that can inform the preliminary design of future reusable vehicles. By combining aerodynamic modeling, automatic guidance, and advanced optimization tools such as SNOPT, the thesis contributes to the broader development efforts toward more versatile and fully autonomous re-entry systems, thereby supporting the long-term vision of cost-effective and operationally flexible space transportation.

1. Introduction

The purpose of this thesis is to define the fundamental requirements for horizontal re-entry vehicles, positioning itself within a rapidly expanding and dynamic domain of the aerospace sector. In light of the growing global interest in reusable re-entry systems, the work aims to contribute to the existing body of knowledge by identifying key design considerations and establishing preliminary guidelines to be employed during the conceptual and early design phases. This topic is of considerable relevance not only from an academic perspective but also in relation to industrial and commercial applications. Major aerospace companies, such as *Thales Alenia Space*, have demonstrated a strong interest in the development of reusable vehicles and associated technologies, recognizing their strategic value in terms of cost reduction, operational flexibility, and sustainability. Beyond the definition of requirements, the thesis seeks to provide a structured framework for analyzing the main challenges and opportunities associated with horizontal landing strategies. By integrating theoretical insights with methodological approaches, the research aspires to support the development of more efficient, reliable, and sustainable aerospace solutions. In doing so, it also intends to offer a useful reference for both future academic investigations and industrial initiatives in the field of reusable space transportation systems. By addressing the landing problem as a sequential optimization task, the thesis contributes to the broader effort of defining feasible aerodynamic configurations and control strategies that ensure mission success. The methodology and results are of direct relevance to aerospace companies engaged in the development of next-generation reusable vehicles, which require robust frameworks to validate landing feasibility during the early design phases.

This research addresses these challenges by developing a methodology to define the requirements necessary for a safe and controlled horizontal landing of a re-entry vehicle, with particular emphasis on aerodynamic parameters, guidance strategies, and optimization frameworks.

The work is structured into several chapters. In Chapter 2, the fundamental concepts of flight dynamics and optimization are introduced. The overview also includes the definition and composition of reference frames, the characterization of aerodynamic forces, and the description of the various moments generated by aerodynamic effects or movable control surfaces. The theoretical framework of optimization is

discussed through the presentation of its standard formulation and a survey of common optimization algorithms. In addition, a concise introduction to the theoretical background of Guidance, Navigation, and Control (GNC) systems is provided, with particular reference to typical architectures and feedback loops.

Chapter 3 addresses the landing problem itself, detailing the physical phases of the maneuver and decomposing the problem into descent with flare and ground roll-out. The challenge is also formulated as an optimal control problem, and several hypothetical approaches are proposed. Furthermore, specific guidance objectives and methodologies are discussed.

In Chapter 4, the adopted optimization methodology is described in detail. The mathematical model underlying the analysis is introduced, together with its interactions with the optimization functions. The methodological approach of performing two sequential optimizations, each relying on distinct black-box formulations, is also presented.

Chapter 5 illustrates the simulation environment and the assumptions adopted to replicate realistic operating conditions. The **Simulink** platform, used for the implementation of the dynamical equations, is thoroughly described. Moreover, the optimization environment **SQP SNOPT** is presented, with the rationale for its selection, and subsequently applied to the case study. A baseline optimization process is performed, and the obtained results are analyzed and discussed.

An extension of the optimization process to a broader set of scenarios is introduced in Chapter 6, leading to the definition of the requirements for horizontal landing and to the identification of potential perspectives for further development. The results of the analysis are summarized in the form of a requirements map.

Finally, in Chapter 7, all findings are synthesized and critically discussed. The contribution of the proposed methodology to the understanding and solution of the landing problem is evaluated, and the main implications of the study are highlighted. Concluding remarks emphasize both the strengths and limitations of the adopted approach, while outlining promising directions for future research and potential applications in aerospace engineering.

A general bibliography of horizontal landing and related topics has been considered, including seminal works on the Space Shuttle landing system [?], flight dynamics and control [?, ?, ?], atmospheric re-entry dynamics [2], and modern optimization methods applied to aerospace systems [3]. These references provide both the historical background and the technical foundation upon which this thesis builds.

2. Flight Dynamics Concepts and Optimization

Guiding a generic Horizontal Landing Vehicle (HLV) through the final descent, flare, and touchdown phases presents several significant challenges. First, the development of a three-degree-of-freedom (3-DOF) longitudinal flight dynamics model is essential. Such a model must capture rigid-body motion, aerodynamic forces, and atmospheric effects, thereby ensuring that the navigation system operates within a framework of realistic vehicle behavior. Building upon this foundation, the subsequent task involves the implementation of a robust guidance framework. In this context, the navigation logic enforces landing constraints by means of an iterative optimization process, dynamically adjusting the control commands to guarantee a safe touchdown within prescribed operational limits. Another major challenge concerns the validation of requirements, which is achieved by extending the optimization problem to a broader set of HLV configurations characterized by varying aerodynamic and structural properties.

Addressing these challenges requires a solid understanding of Flight Dynamics, Flight Mechanics, Guidance, Navigation and Control (*GNC*) theory [4] [5], as well as Optimization Algorithms. Accordingly, this section provides an introduction to the fundamental principles of Flight Mechanics and *GNC*, while the discussion of optimization methods is presented in detail in subsequent chapters.

2.1 Principles of Flight Mechanic

Flight mechanics is the discipline that describes and predicts an aircraft's motion by analyzing the balance of forces and moments acting on it. In a HLV's descent and landing phase, only two aerodynamic forces (lift and drag) and weight govern a vehicle's behavior: lift opposes weight to sustain altitude, drag resists forward motion, and weight pulls it toward Earth. HLV motion is also described by moments, generated by aerodynamic forces (acting on the aerodynamic center (*AC*), quite far from the center of gravity (*CG*) of the vehicle) and control surfaces like body flap and elevons.

Forces and moments feed into Newton's second law to yield the rigid-body equations of motion. By solving these coupled nonlinear differential equations, we can simulate

trajectories, design stability-augmenting control laws, and ensure that the vehicle will respond predictably to pilot inputs or guidance commands. Due to the fact that the 3-DOF model describes longitudinal dynamics, we will focus on forces and moments that only affect this behavior.

2.1.1 Coordinates frames

Central to this process are coordinate frames (such as the body axis and Earth-fixed frames) with which are calculated the state variables (attitude, velocity, position), that provide a complete mathematical description of the instantaneous flight performance and behavior. At the heart of these descriptions lies the **Body-Fixed frame**, a right-handed Cartesian system whose origin is typically at the aircraft's center of gravity. Its axes are conventionally aligned with the fuselage: the x -axis points forward toward the nose, the y -axis extends out through the right wing, and the z -axis points downward.

Complementing the Body frame is the **Wind frame**, which is defined by the instantaneous velocity vector of the aircraft relative to the airflow. Its longitudinal axis lies along the relative wind direction, while the lateral axis remains orthogonal in the plane of symmetry, and the normal axis completes the right-handed set. Lift and Drag forces are applied in this frame and their projections on the body frame are made using α and β angles that represent, respectively, a rotation along y_w and z_w axes.

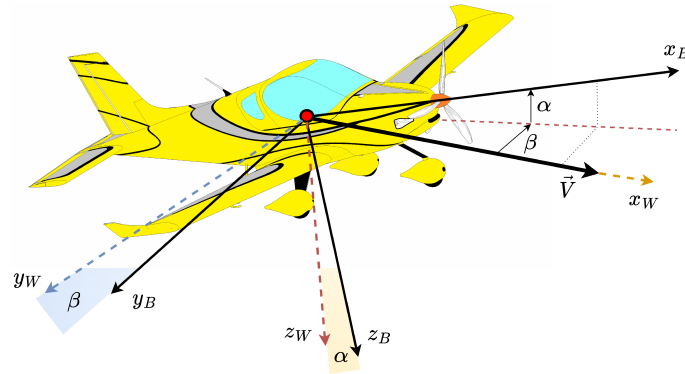


Figure 2.1: Wind (W) and body (B) frames [7]

For navigation and mission-level trajectory design, Earth-fixed frames are essential. One of the most widely used is the **North-East-Down (NED) frame**, in which the x -axis points toward true North, the y -axis toward East, and the z -axis downward toward the Earth's center. This choice aligns well with aviation conventions for latitude, longitude and altitude, and simplifies the integration of inertial-navigation and GPS measurements.

$$R_{wind-body} = \begin{bmatrix} \cos(\alpha)\cos(\beta) & \sin(\alpha)\sin(\beta) & -\sin(\alpha) \\ -\sin(\beta) & \cos(\beta) & 0 \\ \sin(\alpha)\cos(\beta) & \sin(\alpha)\sin(\beta) & \cos(\alpha) \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} \quad (2.1)$$

$$R_{body-NED} = \begin{bmatrix} \cos(\theta)\cos(\psi) & \cos(\theta)\sin(\psi) & -\sin(\theta) \\ \sin(\phi)\sin(\theta)\cos(\psi) - \cos(\phi)\sin(\psi) & \sin(\phi)\sin(\theta)\sin(\psi) + \cos(\phi)\cos(\psi) & \sin(\phi)\cos(\theta) \\ \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\psi)\sin(\phi) & \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\psi)\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix} \quad (2.2)$$

$$R_{body-NED} = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (2.3)$$

$$R_{NED-ENU} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (2.4)$$

It's important to notice that also the angular velocities in the body frame (p, q, r) should be calculated from angular velocities in the ENU frame ($\dot{\phi}, \dot{\theta}, \dot{\psi}$) through a rotation matrix but, due to the fact that only q velocity is present, this matrix is simplified as an identity matrix. In fact:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = R_w \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.5)$$

$$R_w = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi)\cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

2.1.3 Aerodynamic forces

The HLV is subjected to two aerodynamic forces that arise from its interaction with the surrounding airflow.

Aerodynamic lift is the force perpendicular to the oncoming flow that supports the weight of the vehicle; it is most commonly expressed by the well-known relation

$$L = \frac{1}{2} \rho V^2 S C_L \quad (2.7)$$

where ρ is the air density, V the true airspeed, S the wing surface, and C_L the lift coefficient. The coefficient C_L itself is determined by the wing's shape, angle of attack (α) and Mach number, and is typically obtained from wind tunnel tests, computational fluid dynamics simulations, or empirical databases.

Aerodynamic drag acts parallel to the flow and resists forward motion. Its standard form is

$$D = \frac{1}{2} \rho V^2 S C_D \quad (2.8)$$

As C_L , also C_D depends on wing's shape, α and Mach number.

In flight dynamics simulations, these forces are resolved into body-axis components via angle of attack and sideslip angle.

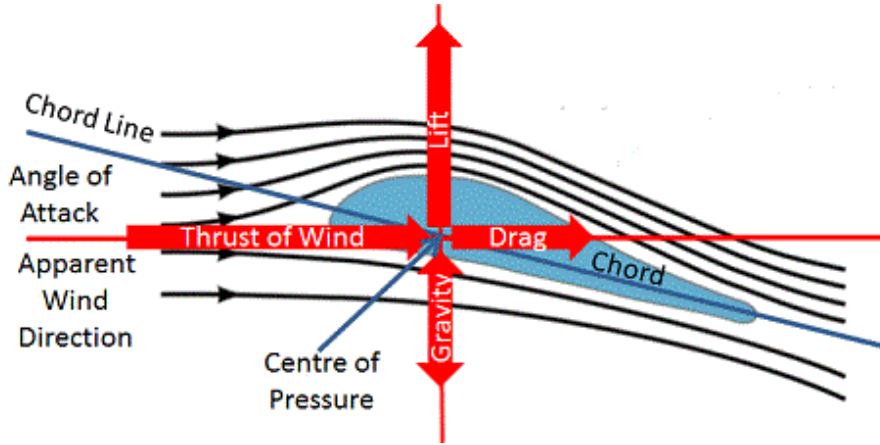


Figure 2.3: Scheme of aerodynamic and gravity forces [9]

2.1.4 Aerodynamic moments

Aerodynamic moments arise from the non-uniform pressure and shear distribution over an aircraft's surfaces, generating torques about its center of gravity. There are three principal moments (M , L and N), but only the first one affects the longitudinal motion:

- **Pitching moment** (M) about the lateral (y) axis, which tends to rotate the nose up or down.

$$M_a = \frac{1}{2} \rho V^2 S \bar{c} C_m \quad (2.9)$$

where $\bar{c}$ is the mean aerodynamic chord. The pitching moment comes from lift and drag distribution (thus from Mach number and α), plus contributions from control deflections.

- **Rolling moment** (L) about the longitudinal (x) axis, which induces wing-tip up or down.

$$L_a = \frac{1}{2} \rho V^2 S b C_L \quad (2.10)$$

with b the wing span. Roll moments come from aileron deflections, wing dihedral effects, and asymmetric lift or drag distribution.

- **Yawing moment** (N) about the vertical (z) axis, which yaws the nose left or right.

$$N_a = \frac{1}{2} \rho V^2 S b C_n, \quad (2.11)$$

yawing moments result from rudder inputs, sideslip-induced side forces on the vertical tail, and differential drag.

In the context of aircraft aerodynamics (so also HLV's winged body vehicles), the lift and drag forces do not generally act through the center of gravity (CG) of the vehicle. Instead, their resultant is conventionally assumed to act at the aerodynamic center (AC), which for subsonic airfoils is located approximately at the quarter-chord position. In the case of HLV, the AC positioning could be slightly different due to the fact that the wings are built differently to deal with supersonic and hypersonic flows. The fact that the line of action of these aerodynamic forces does not pass through the center of gravity gives rise to an aerodynamic moment about the CG. This moment can be decomposed into two contributions: one associated with the lift force, which acts perpendicular to the freestream velocity, and another associated with the drag force, which acts in the direction opposite to the motion. Together, these forces produce a net pitching moment whose magnitude and sign depend on the relative position of the CG with respect to the AC.

$$M_f = (\vec{r}_{cg} - \vec{r}_{ac}) \times (\vec{L} + \vec{D}) \quad (2.12)$$

where $\vec{r}_{cg}$ is the coordinate vector of cg and $\vec{r}_{ac}$ the vector of aerodynamic center. From a stability perspective, this pitching moment is of central importance. If the CG lies ahead of the AC, the aerodynamic moment tends to be restoring, contributing to the static longitudinal stability of the aircraft. Conversely, if the CG is located behind the AC, the resulting moment may destabilize the configuration. For this reason, the interplay between the center of gravity and the aerodynamic center is a fundamental consideration in the design and operational performance of any flight vehicle.

lateral, longitudinal, and vertical axes, respectively.

In particular, these are the most used control surfaces for longitudinal dynamics:

- **Body Flaps:** Large deflectable surfaces near the fuselage, used during reentry to generate pitching moments and help stabilize the vehicle at high angles of attack.

$$M_b = \frac{1}{2} \rho V^2 S \bar{c} C_{m,b} \quad (2.14)$$

- **Elevons:** Used for pitch and roll control, especially in the subsonic phase, are usually two, situated in every wing. Simultaneous deflections produce pitch moment, while opposite deflections produce roll moment.

$$M_e = \frac{1}{2} \rho V^2 S \bar{c} C_{m,e} \quad (2.15)$$

Both values of $C_{m,b}$ and $C_{m,e}$ are scheduled as a function of Mach number, α and the deflection of, respectively, body flap (δ_b) and elevon (δ_e). Positive deflection is considered when the control surface performs a positive rotation around the y axis. This is an example of Space Shuttle's control surfaces:

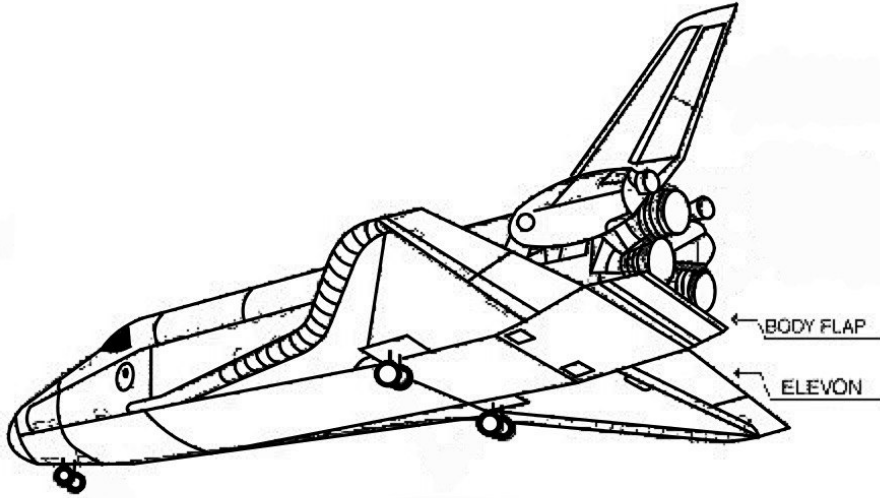


Figure 2.5: Example of body flap and elevons in Space Shuttle [10]

2.2 Guidance Navigation and Control Theory

The theory of Guidance, Navigation and Control (*GNC*) of flight vehicles addresses the fundamental challenge of ensuring that a vehicle follows a desired trajectory while maintaining stability and robustness against disturbances. In this framework,

Guidance provides the reference trajectory and mission objectives, such as targeting a landing site or optimizing energy management, *Navigation* defines the state and position of the vehicle, while *Control* ensures that the vehicle can track these commands by actuating control surfaces or propulsion elements to regulate its attitude, velocity, and position.

For conventional aircraft, guidance and control systems are designed around relatively stable aerodynamic conditions, with linear control methods often sufficient to guarantee performance. However, for reentry vehicles, the dynamics are far more complex. Reentry vehicles encounter rapidly changing atmospheric density, strong aerodynamic forces, and stringent thermal and structural constraints, which demand advanced nonlinear and adaptive control strategies [4] [5] [11].

Unlike purely ballistic capsules, which land vertically using parachutes or retro-thrusters, HLVs are designed to reenter the atmosphere and perform a controlled, airplane-like landing on a runway. This configuration offers significant advantages in terms of reusability, landing precision, crew safety, and operational flexibility.

The guidance of HLVs typically involves multi-phase strategies: an initial hypersonic entry phase, where trajectory shaping is crucial to manage heating and g-forces; a transition phase, where lift is used to maneuver toward the target landing corridor; and a final descent phase, which resembles conventional aircraft guidance with glide slope and flare maneuvers. Control systems must therefore cope with a wide range of flight regimes, from hypersonic reentry down to subsonic aerodynamics near landing, combining space access with aircraft-like behavior. *GNC* systems rely on robust trajectory optimization, adaptive feedback control, and integration of inertial navigation with GPS to achieve high precision.

2.2.1 Architecture of GNC System

The GNC system can be divided into three functional layers, as just seen:

- Navigation – Estimates the vehicle’s current state (position, velocity, attitude, angular rates) using sensors like Inertial Measurement Units (IMUs), GPS, star trackers (at higher altitudes), radar altimeters (near landing) and more. Outputs: full state vector $(x, y, z, V, \phi, \theta, \psi)$.
- Guidance – Determines the desired trajectory or reference profile that the vehicle should follow. In early reentry manages heating, load factors, and footprint constraints, instead in terminal phase it provides lateral alignment, glide slope commands, and flare initiation cues. Outputs: reference flight path angle γ_{ref} , reference reference velocity V_{ref} , heading χ_{ref} and altitude profile h_{ref} .
- Control – Tracks guidance commands by generating actuator signals. Implements feedback control laws (PID, LQR, gain-scheduled controllers, or more advanced adaptive controllers) and compensates disturbances such as gusts,

error between the reference command and the measured state; the integral term K_i , which accumulates the past error over time to eliminate steady-state offsets and ensure accurate tracking; and the derivative term K_d , which predicts the future trend of the error by evaluating its rate of change, thus providing anticipatory damping against oscillations. Mathematically, the control law can be expressed as

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}, \quad (2.16)$$

where $u(t)$ denotes the control input, $e(t)$ the tracking error, and K_p, K_i, K_d the proportional, integral, and derivative gains, respectively. Here is a PID controller scheme:

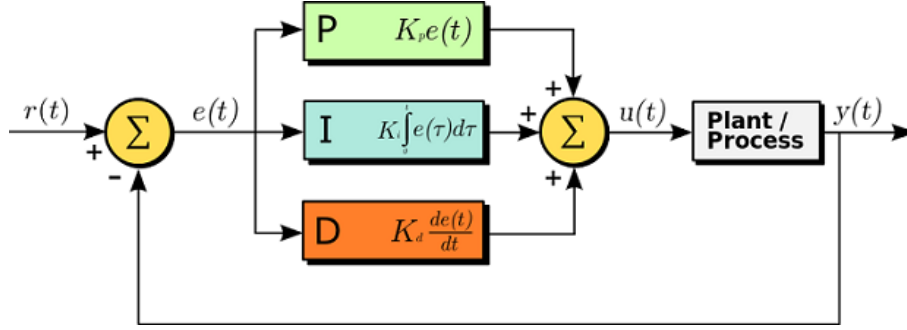


Figure 2.7: PID controller scheme [15]

The tuning of these gains determines the overall performance of the closed-loop system, balancing fast response, accuracy, and robustness. In the context of reentry vehicles, PID controllers may be employed in specific flight regimes or subsystems where the dynamics can be approximated as linear and time-invariant, providing a reliable and computationally efficient control solution.

2.3 Optimization

First-order optimization algorithms represent a fundamental class of numerical methods aimed at efficiently solving problems subject to specific objectives and constraints [13]. Formally, an optimization method is referred to as *first-order* when its update rules depend exclusively on information derived from the first derivative (the gradient) of the objective function, without requiring access to higher-order derivatives such as the Hessian. This reliance on gradient information ensures relatively low computational cost per iteration, which makes such methods particularly attractive in high-dimensional settings.

A general first-order update rule can be expressed as:

$$x_{k+1} = x_k - \eta_k \nabla f(x_k) \quad (2.17)$$

where $x_k \in R^n$ denotes the current iterate, $\eta_k > 0$ is the step size (or learning rate), and $\nabla f(x_k)$ is the gradient of the objective function $f : R^n \rightarrow R$ evaluated at x_k . Variants of this formulation introduce modifications to the step size, gradient scaling, or direction, in order to improve convergence speed, robustness, or stability. These algorithms are extensively applied in engineering, applied mathematics, and computational sciences, as they provide systematic procedures to improve performance, accuracy, and efficiency in complex systems. Despite their effectiveness, first-order approaches may suffer from sensitivity to local minima, saddle points, or ill-conditioning. To mitigate these issues, several families of first-order algorithms have been developed, which can be broadly classified as follows:

- **Gradient-descent methods**

Gradient Descent is a fundamental first-order optimization algorithm designed to minimize a differentiable objective function. It operates as an iterative procedure that seeks a local minimum by updating the current solution in the direction opposite to the gradient, which corresponds to the direction of steepest descent. The efficiency of Gradient Descent depends critically on the choice of η_k . A step size that is too large may cause divergence, while an excessively small value may lead to slow convergence. Under appropriate assumptions, such as convexity and Lipschitz continuity of the gradient, Gradient Descent is guaranteed to converge to a global minimum; in the non-convex case, it generally converges to a local minimum or a stationary point.

- **Stochastic methods**

In large-scale optimization, the computation of full gradients can be prohibitively expensive. Stochastic approaches, such as *Stochastic Gradient Descent (SGD)* and its mini-batch variants, approximate the gradient using randomly sampled subsets of data. While this introduces variance into the optimization trajectory, the stochasticity can be beneficial, as it enhances exploration and facilitates escape from shallow local minima. Such methods have become indispensable in modern machine learning and data-intensive optimization tasks, where problem dimensionality and dataset size render full-gradient methods impractical.

- **Evolutionary and population-based methods**

Beyond purely gradient-driven schemes, evolutionary algorithms—most notably *Genetic Algorithms*—provide an alternative optimization paradigm. Inspired by the principles of natural selection and genetic evolution, these methods operate on populations of candidate solutions that evolve through mechanisms such as selection, crossover, and mutation. Since they do not require differentiability of the objective function, evolutionary approaches are applicable to highly nonlinear, discontinuous, or multimodal landscapes. Although typically more computationally demanding than gradient-based methods, their

feasible region of the problem. Constraints are mathematical conditions, often formulated as equalities or inequalities, that encode the physical, technical, or practical requirements of the system under study. They ensure that the solutions produced by the optimization process are not only mathematically valid but also practically admissible. Without them, the optimization would explore the entire solution space, potentially yielding results that are theoretically optimal yet infeasible in real applications.

It is important to highlight that optimization problems can be classified as either constrained or unconstrained. In unconstrained formulations, only the objective function is considered, and the search extends over the entire domain of the decision variables. While mathematically simpler, such problems rarely capture the complexity of real-world scenarios. Conversely, constrained optimization explicitly accounts for limitations, and the solution sought must satisfy all imposed conditions. In this case, the optimal solution is the one that yields the best value of the objective function among the feasible alternatives.

The procedure begins with an initial guess for the decision variables, which serves as the starting point of the search. At each iteration, the algorithm evaluates the objective function to determine the quality of the current solution.

Based on this evaluation, a new candidate is generated by applying a search strategy. In gradient-based methods, this typically involves exploiting derivative information to move in a direction of improvement, while derivative-free or heuristic approaches rely on rules of exploration, sampling, or probabilistic mechanisms to identify promising regions of the search space. If constraints are present, the algorithm ensures that the candidate solutions remain feasible, either by directly enforcing the conditions or by penalizing violations within the objective function.

The process continues iteratively, updating the solution at each step, until one or more stopping criteria are satisfied. These may include convergence of the objective function value, satisfaction of optimality conditions, or the exhaustion of computational resources such as time or iteration limits.

2.3.1 Parallelization and MATLAB Parallel Computing Toolbox

In the context of computational optimization, parallelization represents a crucial strategy to enhance performance and scalability, particularly when dealing with large-scale problems or computationally intensive algorithms. The fundamental principle of parallel computing lies in decomposing a complex task into smaller, independent subtasks that can be executed simultaneously across multiple processing units. This approach significantly reduces overall computation time, facilitates the exploration of multiple solution candidates in parallel, and enables the efficient utilization of available hardware resources such as multicore CPUs, GPUs, and distributed clusters.

MATLAB provides native support for parallel computing through the **Parallel Computing Toolbox**, which offers a comprehensive framework for executing operations concurrently. This toolbox allows users to define and manage *parallel pools*—collections of MATLAB workers that operate as independent computational engines. Once a parallel pool is established, parallel constructs – such as **parfor** loops and distributed arrays – can be employed to distribute workloads automatically across available workers. For instance, optimization algorithms involving population-based methods (e.g., Genetic Algorithms or Particle Swarm Optimization) can assign the evaluation of individual candidate solutions to distinct workers, thereby achieving substantial acceleration without altering the core algorithmic logic.

By leveraging these capabilities, optimization procedures can exploit concurrency at multiple levels, achieving enhanced efficiency and robustness while maintaining numerical accuracy and reproducibility. The integration of parallel computing within MATLAB thus constitutes an essential asset for addressing the computational demands of modern optimization tasks.

3. The Landing Problem

The first step in addressing the objective of this thesis consists in the analysis of the reference landing problem.

The landing problem for a horizontal landing reentry vehicle is a complex task that extends from the final segment of atmospheric flight to ground operations. It typically begins at altitudes on the order of 15 km, where the vehicle transitions from high-speed, high-altitude flight to a controlled descent toward the runway. In this phase, the vehicle must manage the residual kinetic and potential energy accumulated during reentry, while ensuring that trajectory and attitude remain within controllable limits [22, 23].

From the entry interface of the terminal descent, the vehicle is required to execute a precise glide trajectory that guarantees both runway alignment and adequate margins of controllability. Aerodynamic forces dominate at these altitudes, and the vehicle relies on lift-to-drag management to balance the competing needs of range extension and controllability. Guidance objectives include regulating flight path angle and airspeed in order to establish favorable conditions for the final descent.

As the vehicle descends below a few kilometers of altitude, additional constraints emerge. The flare maneuver must ensure a progressive reduction of vertical speed in order to achieve a safe touchdown without exceeding structural or load factor limits. Upon ground contact, the dynamics change fundamentally: ground reaction forces, braking systems, and tire dynamics replace aerodynamic control as the dominant mechanisms for deceleration. This ground roll phase must be completed within the available runway length, while maintaining stability against disturbances such as crosswinds or surface irregularities.

Overall, the landing problem for horizontal reentry vehicles is characterized by the coupling of high-energy flight mechanics with the requirements of conventional runway operations. It involves a continuous sequence of phases — high-altitude glide, descent, flare, and rollout — each governed by distinct physical phenomena and operational constraints, yet seamlessly connected to achieve a safe and reliable termination of the mission.

From a mathematical perspective, the landing problem can be formulated as an optimal control problem. The system dynamics are described by nonlinear equations of motion, subject to phase-dependent boundary conditions and path constraints. The objective function may reflect multiple criteria, such as minimization of runway

To implement these functions, guidance algorithms rely on integrated navigation equations. Control laws during this phase resemble those of advanced flight control systems in aircraft; however, they must also account for the unique aerodynamic characteristics of HLVs, which are optimized for reentry rather than for efficient low-speed flight.

While gliding, the guidance attempts to set the vehicle in a fixed position, similar to a trimmed position for aircraft, until it reaches a certain altitude (initial flare height h_{flare}). Past this point, the guidance should start to make the vehicle flare and set it to the landing position until touchdown.

In addition to trajectory shaping and guidance design, the landing problem must also address the identification of the aerodynamic parameter combinations that enable a feasible and robust descent. The vehicle's characteristics — primarily m , S , L/D — directly determine its ability to regulate flight path angle, dissipate velocity, and maintain controllability during the terminal descent. Consequently, the landing analysis is not solely a matter of optimal control under given dynamics, but also of determining the minimal set of aerodynamic properties necessary to achieve the maneuver.

The rationale behind this focus lies in the inherent design trade-off for horizontal landing reentry vehicles. On one hand, high aerodynamic performance facilitates controllability and widens the feasible landing envelope; on the other hand, excessive aerodynamic requirements may lead to over-designed or impractical vehicle configurations. The problem, therefore, involves a systematic search for the lowest yet sufficient values of aerodynamic coefficients that still permit a safe glide, controlled flare, and stable touchdown.

From an optimization perspective, this can be approached through a systematic analysis and mapping of the aerodynamic parameter space, in which different combinations of lift, drag, and moment coefficients are evaluated with respect to trajectory feasibility, dynamic constraints, and safety margins. Such a mapping enables the identification of regions where landing is feasible, highlighting how different aerodynamic configurations influence controllability and energy management. In this way, it becomes possible to select the most convenient set of aerodynamic parameters according to mission priorities — for example, favoring robustness, minimizing control effort, or reducing structural demands. This perspective not only clarifies the aerodynamic requirements for safe landing but also provides a flexible decision-making framework that adapts to the specific objectives of vehicle design and operational needs.

In summary, the landing problem must be regarded as a dual challenge: designing an optimal sequence of control inputs for descent and flare while simultaneously determining the optimal and minimal aerodynamic parameter set that ensures mission success. This dual formulation provides a more integrated perspective, where guidance, control, and aerodynamic design converge in the definition of feasible landing strategies.

4. Optimization Approach

After clearly defining the problem, an optimization framework is established with the objective of ensuring a successful landing under the baseline configuration. This chapter presents in detail the mathematical model governing the system dynamics and provides a systematic discussion of the rationale underlying the adopted optimization methodology. Particular attention is devoted to the formulation of the equations of motion, the assumptions introduced to simplify the analysis, and the choices that define the optimization process. In addition, the chapter outlines the logical structure of the solution strategy, highlighting how the chosen approach balances accuracy with computational efficiency.

4.1 Chosen Strategy

An initial strategy conceived for the resolution of the landing problem was structured as follows. The problem was divided into two sequential optimization stages as the (*Descent* and *Landing*), where the solution obtained from the first was employed as the initial state for the second. This initial guess encompassed several elements, including the total simulation time T , a set of parameters describing both the flare maneuver and the glide phase, as well as the time discretization of the angular deflections of the two control surfaces. The discretization process was performed over a considerable number N of time segments (50 for each control surface), interpolated across the total flight duration. The optimization procedure thus acted directly on the dynamic control system in order to satisfy the set of constraints typically associated with the landing phase.

Despite the initially satisfactory results, during the optimization analyses a significant computational burden was observed, primarily arising from the structure of the problem formulation itself. In the described setup, the optimizer received as input a vector of 103 elements. The first three entries corresponded to the total flight time T , the altitude at which the flare maneuver was initiated, and the equilibrium flight path angle. The subsequent 50 elements represented the time-discretized angular deflections of the body flap, while the final 50 elements corresponded to the angular deflections of the elevons. This input configuration proved to be excessively demanding in terms of computational effort, as it introduced a large number of op-

timization variables. For each optimization run, the solver SNOPT was required to compute a derivative matrix, consisting of the Jacobian of the constraints with dimension 9×103 and the Hessian of the Lagrangian with dimension 103×103 . As a consequence, optimization times often extended to several hours. Given that two such optimizations were executed in sequence, and that several tens of runs were required, the computational effort quickly became prohibitive. Furthermore, the interpolation of control variables, including the total simulation time, added additional complexity to the problem. As a result, the total flight time remained rarely modified during the optimization process, effectively constraining the vehicle to land within the time frame imposed by the initial guess.

The issues highlighted in these analyses led to the development of a new simulation approach aimed at reducing the number of variables and, consequently, alleviating the overall computational cost of the process. To this end, the workflow of the initial black-box was restructured into two separate black-box modules, representing respectively the **Guidance** and the **Control** during the descent of the vehicle. These two modules are hereafter referred to as *Descent Guidance* and *Descent Control*. As the terminology suggests, the first module defines the guidance strategy in terms of the commanded angle of attack α , while the second module attempts to follow this guidance law through the tuning of a controller. By adopting this strategy, the number of optimization variables was significantly reduced, from 103 down to $4 + 3$, which in turn resulted in considerably smaller Jacobian and Hessian matrices. As will be shown in the following sections, the optimization problem associated with *Descent Control* does not even require explicit constraints, relying solely on a cost function, thereby further reducing the overall computational effort and the duration of the process.

The previously described optimization strategy was instead employed for the runway *Landing* phase, where an explicit guidance law was not strictly required, and it was therefore considered more appropriate to act directly on the control inputs.

4.2 Mathematical Model

The simulator describes the longitudinal dynamics of a typical horizontal re-entry vehicle, representing forces that act along two body axes (X and Z) and moments around the third one (Y) [17] [18]. For clarity of analysis, the study of the vehicle motion was divided (like in the problem description) into two distinct phases: the *Descent phase*, extending until the instant of landing gear contact with the runway, and the *Landing phase*, encompassing the deceleration and roll-out until the vehicle comes to a complete stop. This separation was required because the two phases are subject to different boundary conditions and constraints. In particular, the transition to ground contact introduces specific dynamic effects, such as pitching moments induced by the landing gear, which are absent in the airborne phase.

4.2. Mathematical Model

In each step, the simulator resolves two vectorial equations of motion (in X and Z axes) and one attitude equation (around Y):

$$\begin{cases} \frac{d\vec{x}}{dt} = \vec{v} \\ \frac{d\vec{v}}{dt} = \frac{\vec{L}}{m} + \frac{\vec{D}}{m} + \vec{g} \\ I_y \ddot{\theta} = M_y \end{cases} \quad (4.3)$$

It's important to notice that in this study Coriolis and inertial effects are neglected. In order to solve these equations, the simulator calculates $\vec{v}$, $\vec{L}$, $\vec{D}$ and M_y from of the state vector given as input.

The sum of the forces defines the total accelerations applied to the vehicle. These forces are only Lift, Drag and Weight due to the fact that HLVs usually perform a gliding landing, so the Thrust is not present. From the acceleration equation, we can integrate it in the longitudinal axis (x and z) to obtain velocities and positions that describe the longitudinal linear behavior of the HLV [6].

$$\vec{F} = m\vec{g} + \vec{L} + \vec{D} \quad (4.4)$$

$$\vec{a} = \frac{\vec{L}}{m} + \frac{\vec{D}}{m} + \vec{g} \quad (4.5)$$

by integrating and splitting in axes components:

$$V_x(t_i + \Delta t) = V_x(t_i) + \int_{t_i}^{t_i + \Delta t} a_x dt \quad (4.6)$$

$$V_z(t_i + \Delta t) = V_z(t_i) + \int_{t_i}^{t_i + \Delta t} a_z(t) dt \quad (4.7)$$

Same for positions x and z :

$$x(t_i + \Delta t) = x(t_i) + \int_{t_i}^{t_i + \Delta t} V_x(t) dt \quad (4.8)$$

$$z(t_i + \Delta t) = z(t_i) + \int_{t_i}^{t_i + \Delta t} V_z(t) dt \quad (4.9)$$

The sum of all the moments defines the total moment applied to the vehicle:

$$M_y = M_f + M_a + M_q + M_b + M_e \quad (4.10)$$

that is fed directly into the rotational equation of motion (simplifying by considering only longitudinal rotational behavior):

$$I_y \ddot{\theta} - (I_z - I_x)rp = M_y \quad (4.11)$$

$$I_y \ddot{\theta} = M_y \quad (4.12)$$

where q is the pitch rate and I_y is one of the principal inertia moments. As for position dynamics, we can integrate to obtain rotational behavior:

$$q(t_i + \Delta t) = q(t_i) + \int_{t_i}^{t_i + \Delta t} \ddot{\theta}(t) dt \quad (4.13)$$

$$\theta(t_i + \Delta t) = \theta(t_i) + \int_{t_i}^{t_i + \Delta t} q(t) dt \quad (4.14)$$

As said before, for *Guidance* simulation, part of the rotational behavior is imposed as an input. This prescribed function is the angle of attack α_{id} , and the dynamic propagation is composed as follows:

$$\theta = \alpha_{id} + \gamma \quad (4.15)$$

4.2.1 Forces and Moments

In the simulator, forces acting on the X and Z body axes are implemented as Lift, Drag and Weight. L and D generate moments across the Y axis, executed in the simulation as well. Other moments are also considered, such as the damping moment and the control moment actuated by the body flap and the elevons.

In order to calculate the aerodynamic forces, it is first necessary to determine the dynamic pressure, defined as

$$Q = \frac{1}{2} \rho V^2 \quad (4.16)$$

where ρ denotes the air density and V the true airspeed. The static pressure, and consequently the air density, are obtained through the International Standard Atmosphere (ISA) model, by using as input the altitude, which corresponds to the z -coordinate of the state vector. This model has been implemented in **Simulink** using the *Aerospace Tool Blockset*. The true airspeed perceived by the vehicle is then derived from the velocity components in the body axis system, namely V_{xb} and V_{zb} , from which the magnitude of the airspeed is obtained as

$$V = \sqrt{V_{xb}^2 + V_{zb}^2} \quad (4.17)$$

This formulation ensures the consistent evaluation of both the atmospheric properties and the velocity components required for the aerodynamic force calculation.

Another essential step in obtaining the aerodynamic forces at each simulation step is the use of aerodynamic databases, which provide tabulated values of the aerodynamic coefficients as functions of key flight parameters, such as the Mach number, the angle of attack α , and the deflection angles of control surfaces (δ). This methodology avoids the prohibitive computational cost of real-time CFD evaluations while

still reproducing the nonlinear aerodynamic behavior over the entire flight envelope. Formally, the aerodynamic coefficients can be expressed as multidimensional functions of the relevant flight variables, e.g.

$$C_X = f_x(M, \alpha, \delta) \quad (4.18)$$

where C_X denotes the aerodynamic force coefficients along the body axes, and the functional dependence is obtained from the *Space Shuttle Aerodynamic Database* [?]. This database served as the baseline for the initial simulation and optimization; subsequently, the model parameters were systematically varied to explore alternative configurations.

L and D are calculated as follows:

$$L = \frac{1}{2} \rho V^2 S C_L \quad (4.19)$$

$$D = \frac{1}{2} \rho V^2 S C_D \quad (4.20)$$

where C_L and C_D are functions of Mach number M and angle of attack α .

L and D generate moments about the cg (named M_f), calculated as follows:

$$M_f = (\vec{r}_{cg} - \vec{r}_{ac}) \times \vec{F}_a \quad (4.21)$$

where $\vec{F}_a$ is the vector sum of $\vec{L}$ and $\vec{D}$.

In the *Landing* simulation, an additional force is considered, representing the ground reaction. This force generates a negative pitching moment with respect to the center of gravity (CG) until the nose landing gear touches the ground:

$$M_{ground} = \begin{cases} (\vec{r}_{cg} - \vec{r}_{wheel}) \times (\vec{F}_a + \vec{W}) & \text{if } \theta > 0 \\ 0 & \text{if } \theta = 0 \end{cases} \quad (4.22)$$

where $\vec{r}_{wheel}$ denotes the position vector of the rear landing gear and $\vec{F}_a + \vec{W}$ is the reaction force.

Aerodynamic moment is calculated as follows:

$$M_a = \frac{1}{2} \rho V^2 S C_{ma} \quad (4.23)$$

Damping moment is calculated as follows:

$$M_q = \frac{1}{2} \frac{\rho S c^2 q}{2V} C_{mq} \quad (4.24)$$

c is the Mean Aerodynamic Chord, also named MAC .

4.2.2 Control System

The control system is implemented through two primary movable aerodynamic surfaces, the body flap and the elevons [21]. They are calculated as follows:

$$M_b = \frac{1}{2} \rho S c C_{mb} \quad (4.25)$$

$$M_e = \frac{1}{2} \rho S c C_{me} \quad (4.26)$$

C_{mb} and C_{me} also depend on M , b and e , which are the deflection angles of the body flap and elevons.

At each discretization step, the attitude equation reported in (4.10) governs the evaluation of the total aerodynamic moment acting on the vehicle. The control surfaces generate additional aerodynamic moments that modify this resultant moment by introducing other contributions with respect to the uncontrolled aerodynamic moments. The magnitude of these control contributions is a direct function of the deflection angles of the movable surfaces. Through this mechanism, the total moment acting on the vehicle can either be canceled, when equilibrium conditions are required, or deliberately modified to achieve the desired attitude response. The overall compensating effect produced in this way is conventionally referred to as the *Control Moment*.

4.3 Optimizer configuration

To address the problem under consideration, a sequential triple-optimization framework (2 for the Descent phase and 1 for the Landing phase) has been adopted. In this scheme, the optimizer operates through interconnected black-box models, where the outcome of the first optimization serves as the input for the second.

The first black-box, denoted as *Descent Guidance*, receives as input the value of the flight path angle during the glide phase, together with a set of angle of attack (α) and time values that are used to parameterize both the glide segment and the subsequent flare maneuver. The optimization is constrained by typical landing requirements in terms of position, velocity, and attitude, while aiming to minimize the time needed to achieve a stable condition.

Once the guidance solution has been obtained, a second optimization is performed through the *Descent Control* black-box. This module takes as input the parameters of the controller (a PID controller has been chosen) and operates on the movable control surfaces with the objective of minimizing the error between the ideal angle of attack α_{id} and the actual controlled one. It should be recalled that, in this black-box, the attitude dynamics are fully included, allowing the movable surfaces to effectively control the vehicle's motion.

Once the optimized solution has been obtained, *Landing* acts as a third and last

4.3. Optimizer configuration

	Guidance	Control	Landing
Initial conditions	$\vec{X}_0$	$\vec{X}_0$	$\vec{X}_{0,L} = \vec{X}_f$
Final conditions	$\vec{X}_f$	$\vec{X}_f$	$\vec{X}_{f,L}$
Initial guess	$\alpha_1, \alpha_2, T_1, T_2, FPA_0$	K_p, K_i, K_d	$\delta_{b,L}, \delta_{e,L}, t_{f,L}$

4.3.1 Constraints

The final descent and landing of an HLV is governed not only by the need to track a reference trajectory but also by a set of constraints that ensure safety, structural integrity, and mission success. These constraints arise from aerodynamics, vehicle performance limits, environmental factors, and operational requirements. Specifically for 3-DOF longitudinal landing:

1. Geometric and Runway Constraints

- Touchdown point: The vehicle must reach the touchdown point with sufficient margin for deceleration.
- Runway length: The vehicle must stop in a distance shorter than the total length of the runway.
- Maximum touchdown velocity: The vertical touchdown velocity must not be too elevated to avoid impact damage to the vehicle.

2. Kinematic and Dynamic Constraints

- Velocity bounds: The velocity must remain into two margins, lower margin to avoid stall velocity and upper margin to prevent excessive aerodynamic or thermal loads.
- Normal load factor: The normal load factor must not exceed the maximum value allowed.
- Angular rate limits: Pitch/yaw/roll rates must stay within actuator capabilities.

5. Experimental Analysis

In order to validate the proposed methodology and assess its effectiveness, a comprehensive experimental analysis has been conducted. This section provides a detailed description of the computational environment, the assumptions adopted, the numerical data employed, and the optimization framework applied. The simulation environment has been developed in **MATLAB**, with the dynamic model of the vehicle implemented in **Simulink**, which enables the integration of the system equations and the evaluation of control strategies under various configurations.

The analysis is carried out under a set of simplifying yet representative assumptions, ensuring both computational tractability and physical relevance. The input data, together with their admissible ranges, are systematically defined to reproduce realistic operating conditions. Furthermore, the optimization process relies on the **SNOPT** (Sparse Nonlinear OPTimizer) algorithm, whose choice is described below in the chapter.

The following subsections will present the hypotheses, numerical values, parameter ranges, and optimization settings in detail, thereby establishing a transparent framework for the subsequent interpretation of the results.

5.1 Simulation environment

The complete 3-DOF dynamics of HLV's vehicle has been implemented in **Matlab Simulink**. This software provides very simple and user-friendly interfaces for the application of forces and moment, and can immediately provide an overview of the logical flow of information and the relationships among the formulas.

5.1.1 Hypothesis and features

Baseline Configuration

As previously discussed, an initial baseline model of the HLV was developed, inspired by the Space Shuttle. From this model, essential parameters such as mass, wing surface area, moment of inertia and aerodynamic coefficients were obtained. The baseline configuration was employed to establish a preliminary reference for the

expected behavior of reentry vehicles and to provide a means of internal validation for the simulator.

Space Shuttle parameters:

- Dry mass $m = 85230 \text{ kg}$
- Surface Area $S = 249.91 \text{ m}^2$
- Moment of inertia $I_y = 9.68 \cdot 10^6 \text{ kg} \cdot \text{m}^2$
- CG coordinates: $CG = [21.303, 0, -0.635] \text{ m}$
- AG coordinates: $AC = [21.460, 0, 0] \text{ m}$

Subsequently, several modifications were introduced in order to obtain a simulator that is consistent and independent of the physical characteristics of the Space Shuttle, while still being able to represent, in a first approximation, a reentry vehicle of arbitrary size and shape. Many of the parameters available in the baseline configuration, but not directly defined for alternative designs—such as the mean aerodynamic chord (MAC), the moments of inertia, and the distance between the center of gravity and the aerodynamic center—were scaled according to the dimensions of the configuration under consideration. In particular, the scaling was primarily based on the vehicle mass and wing surface area.

Databases

In the **Simulink** environment, the aerodynamic databases discussed in 4.2.1 are implemented using *3-D Lookup Tables*. A *3-D Lookup Table* is an interpolation tool that outputs the value of a dependent variable given three independent inputs, by referencing pre-computed multidimensional arrays. In this application, the lookup tables return the aerodynamic coefficients corresponding to the instantaneous Mach number, angle of attack, and control surface deflections. This framework ensures a balance between accuracy and computational efficiency, which is crucial for both real-time simulations and iterative trajectory optimization of aerospace vehicles.

Control Surfaces

The control surfaces considered are the body flap δ_b and the elevons δ_e , implemented only in the *Control Descent* and *Landing* phase black-boxes, that generate the sufficient moment in order to follow the guidance trajectory. It has been hypothesized that surfaces also have these limitations:

	Body flap δ_b	Elevons δ_e
Range of motion [°]	$[-25 ; 25]$	$[-40 ; 25]$
Rate Limiter [°/s]	± 4.5	± 15

Table 5.1: Control surfaces

5.1.2 Descent Guidance black-box

The *Descent Guidance* black-box defines an open-loop guidance profile for the considered vehicle. Specifically, the loop is “opened” with respect to the attitude dynamics, by prescribing an optimal angle of attack α that ensures the success of the glide phase, the flare maneuver, and the final landing. The angle of attack is imposed to remain constant at a value α_1 until a prescribed time T_1 , at which the maneuver is initiated. From this point, it is gradually increased until reaching a value α_2 at time T_2 . The temporal evolution of α , hereafter referred to as the α_{ideal} within this guidance law, is therefore defined by interpolating these four key parameters.

This ideal guidance profile then serves as the reference input for the *Descent Control* black-box, which is responsible for tracking the prescribed α_{ideal} through the appropriate controller tuning.

Simulation Workflow

The simulator is described following a step-by-step approach. The input variables are provided from the `Matlab` Workspace. It is worth noting that, at this stage, aerodynamic characteristics and geometric dimensions are not treated as simulation inputs, but rather assumed as constant values. Inputs:

- Initial state vector X_0 ;
- Parameters for α_{id} plot: $ALPHA = [\alpha_1, \alpha_2]$ and $TIME = [T_1, T_2]$;

The components of vectors $ALPHA$ and $TIME$ are interpolated in order to obtain the α_{id} angle as a function of time.

At each time step ($\Delta t = 0.5$ s), the simulation proceeds as follows:

1. Vehicle data are initialized and the state vector is decomposed into its components. Instantaneous value of $\alpha_{id}(t)$ is extracted from the plot.
2. Velocities in the *ENU* reference frame are transformed into the *body* frame using the appropriate rotation matrix.
3. Intermediate aerodynamic variables are computed, namely the flight path angle FPA , dynamic pressure Q , Mach number M , and gravitational acceleration g . The latter is considered to vary slightly with altitude.
4. Aerodynamic coefficients are obtained as functions of α and M , making use of aerodynamic databases based on Space Shuttle data. In particular, C_L , C_D are computed.

5.1. Simulation environment

5. From the aerodynamic coefficients, aerodynamic forces L and D are computed in the *wind* frame and then transformed into the *body* frame. Performance indicators such as L/D and L/W ratios are also evaluated.
6. Total forces in the *ENU* frame, F_{ENU} , are obtained by transforming the aerodynamic forces from the *body* frame and adding the weight contribution.
7. The derivative of the state vector is then computed. Velocities in the *ENU* frame provide $\dot{x}$ and $\dot{z}$. Accelerations $\dot{v}_x$ and $\dot{v}_z$ are obtained by dividing F_{ENU} by the vehicle mass.
8. The state vector is updated through numerical integration of its derivative, starting from the initial condition X_0 . The pitch angle θ is computed as sum of FPA and α_{id} and normalized within the interval $[-180^\circ, 180^\circ]$. The resulting state vector is then used as input for the next iteration.

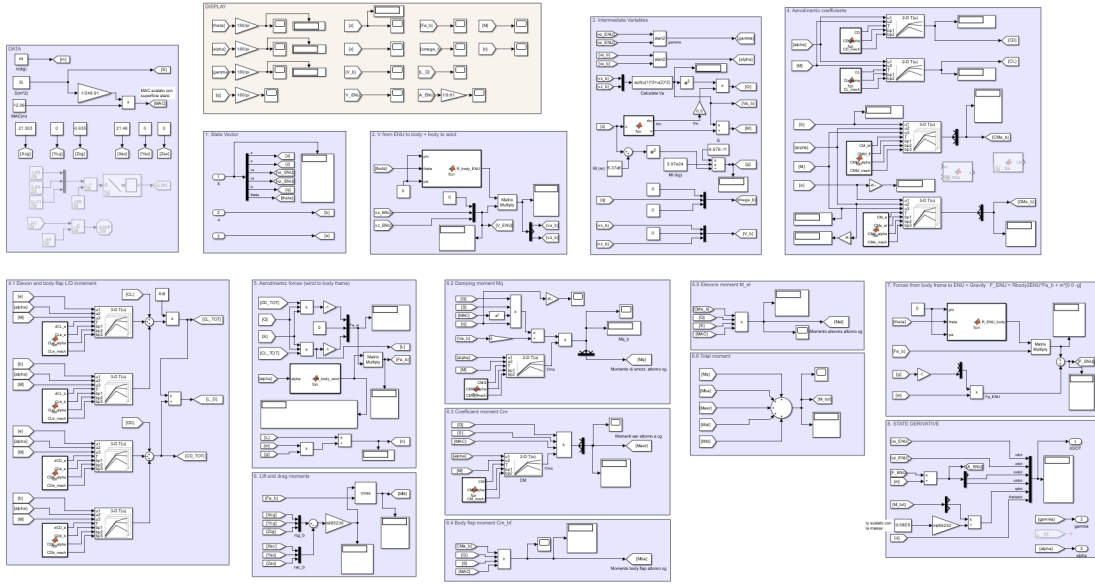


Figure 5.1: Simulink *Descent Guidance* simulation

The simulation terminates when the vertical position z reaches the airport reference altitude (94 m).

5.1.3 Descent Control black-box

The *Descent Control* black-box is therefore tasked with tracking the guidance profile defined in the previous phase. In this case, a PID controller is employed to command the movable control surfaces of the vehicle directly. Unlike the open-loop guidance,

the dynamic loop here is closed, with the angle of attack α entering the equations of motion and the propagation-integration process. By subtracting the “ideal” angle of attack α_{id} , as defined by the guidance module, from the current α , the PID controller computes a corrective control action aimed at minimizing this error. As outlined in Chapter 2, this control action is achieved by appropriately tuning the proportional, integral, and derivative gains, K_p , K_i , and K_d , which determine the response characteristics and stability of the closed-loop system.

The output of the PID controller is a continuously varying parameter, bounded in the interval $[-1, 1]$. This parameter is subsequently scaled to generate the control inputs for the body flap deflection δ_b and the elevon deflections δ_e , which are actuated simultaneously. It is important to account for the fact that the deflection ranges of these surfaces differ, as described in the previous section. In particular, positive deflections are applied symmetrically to both surfaces, while negative deflections exhibit slight differences.

The schematic formulation of the control law must also consider potential errors in position and timing caused by the tracking of the guidance profile. The main risk is that the flare maneuver may be executed at the same commanded instant as in the guidance law, but with the vehicle having accumulated positional deviations that place it in a location different from that assumed during guidance design. To mitigate this issue, the control of the angle of attack is therefore not imposed as a function of time but rather as a function of altitude. In this way, the flare maneuver is triggered when the vehicle actually reaches the prescribed altitude, regardless of the exact time at which this occurs.

Simulation Workflow

Inputs:

- Initial state vector X_0 ;
- Guidance law $\alpha_{id}(z)$;
- PID parameters K_p , K_i , K_d .

Steps

1. Vehicle data are initialized and the state vector is decomposed into its components. K_p , K_i and K_d fit into the PID to define control deflections of mobile surfaces δ_b and δ_e .
2. Velocities in the *ENU* reference frame are transformed into the *body* frame using the appropriate rotation matrix.
3. Intermediate aerodynamic variables are computed, namely the flight path angle *FPA*, the angle of attack α , dynamic pressure Q , Mach number M , and

gravitational acceleration g . The latter is considered to vary slightly with altitude.

4. Aerodynamic coefficients are obtained as functions of α , M , δ_b and δ_e , making use of aerodynamic databases based on Space Shuttle data. In particular, C_L , C_D , $C_{M,b}$, $C_{M,e}$ are computed, as well as the incremental contributions of body flap and elevons: $\Delta C_{L,b}$, $\Delta C_{L,e}$, $\Delta C_{D,b}$, $\Delta C_{D,e}$.
5. From the aerodynamic coefficients, aerodynamic forces L and D are computed in the *wind* frame and then transformed into the *body* frame. Performance indicators such as L/D and L/W ratios are also evaluated.
6. Aerodynamic forces generate moments around the center of gravity (CG), computed as the cross product between the forces and the distance vector between CG and aerodynamic center (AC).
7. Aerodynamic moments are calculated using the aerodynamic databases: C_M , C_{Mq} , $C_{M,b}$, $C_{M,e}$. The total moment M_{tot} is obtained as their sum.
8. Total forces in the *ENU* frame, F_{ENU} , are obtained by transforming the aerodynamic forces from the *body* frame and adding the weight contribution.
9. The derivative of the state vector is then computed. Velocities in the *ENU* frame provide $\dot{x}$ and $\dot{z}$. Accelerations $\dot{v}_x$ and $\dot{v}_z$ are obtained by dividing F_{ENU} by the vehicle mass. The angular acceleration $\dot{q}$ is calculated as the ratio between the total moment and the moment of inertia I_Y (noting that all moments act along the Y axis). The angular velocity q corresponds to $\dot{\theta}$.
10. The state vector is updated through numerical integration of its derivative, starting from the initial condition X_0 . The pitch angle θ is normalized within the interval $[-180^\circ, 180^\circ]$. The resulting state vector is then used as input for the next iteration.

5.1. Simulation environment

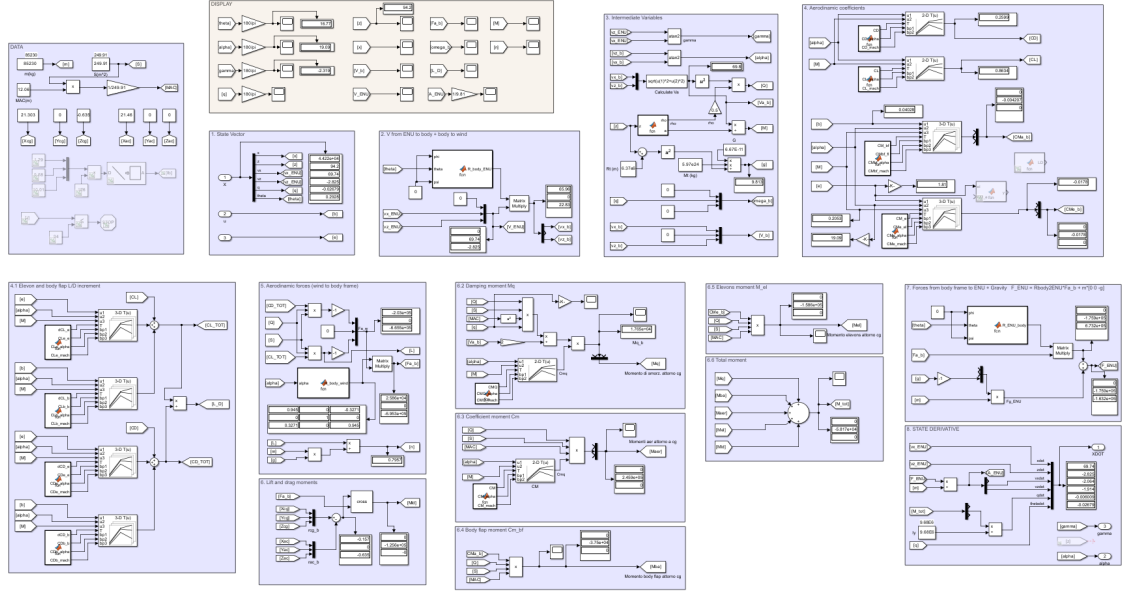


Figure 5.2: Simulink *Descent Control* simulation

Also this simulation terminates when the vertical position z reaches the airport reference altitude (94 m).

5.1.4 Landing black box

Although the input parameters and initial conditions differ, the structural differences between the *Descent Control* and *Landing* simulations are limited. The main distinctions can be summarized as follows.

Inputs:

- Initial state vector $X_{0,L}$;
- Average total simulation time T ;
- Discretization parameter N ;
- Body flap deflections vector B ;
- Elevons deflections vector E .

The components of vectors B and E are interpolated (considering actuator rate limitations and maximum deflection constraints) in order to obtain continuous control surface deflection functions δ , thus avoiding discontinuous step profiles.

- In this simulation, no PID controller is implemented to track a guidance profile; instead, the control parameters are provided as input in the form of time-dependent interpolations.

5.1. Simulation environment

- Vehicle data are initialized and the state vector is decomposed into its components. Current values of body flap and elevon deflections are extracted from interpolation of B and E .
- The pitch angle θ is constrained within the range $[0^\circ, 90^\circ]$, while the pitch rate q is considered only until both main landing gear wheels are in contact with the ground.
- A reaction force, acting at the rear landing gear, is introduced to represent the ground reaction. This force generates a moment about the center of gravity (CG), which remains active until the nose gear also makes contact with the ground. The moment is computed as the cross product between the total aerodynamic forces expressed in the *body* axes and the distance vector between the CG and the rear landing gear.
- To correctly evaluate the reaction moment, the weight vector is expressed in the *body* reference frame.
- Braking on the runway is modeled by introducing an additional force, F_{brake} . This force acts in the direction opposite to the velocity in the ENU frame and is defined as a fraction of the vehicle's weight. Both this braking force and the corresponding deceleration are included in the computation of the derivative state vector.

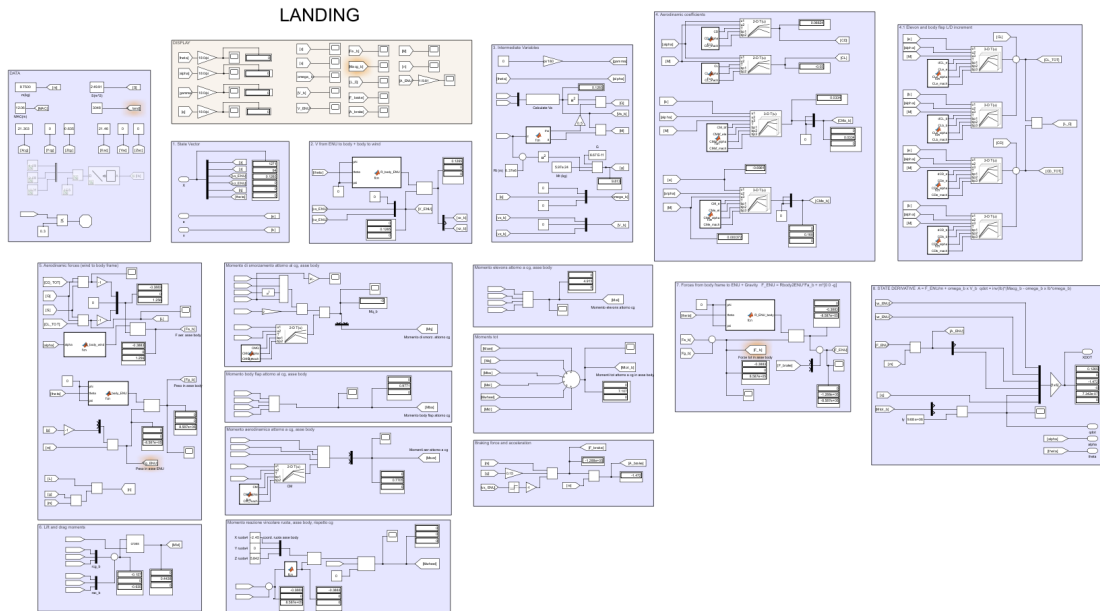


Figure 5.3: Simulink *Landing* simulation

5.2 Optimization environment

This section provides a detailed description of the methodological procedure adopted for the optimization process and the subsequent verification of the project requirements. The study involved a critical assessment of different optimization approaches and algorithms to identify the most suitable and efficient configuration for the application under consideration.

In particular, the selected optimization strategy is based on the *Single Shooting* method, in which the entire time horizon is integrated as a single initial value problem. This approach simplifies the problem formulation and reduces the number of optimization variables, while still ensuring a consistent propagation of the system dynamics throughout the trajectory. Although it may be more sensitive to initial guesses compared to multi-segment techniques, it offers computational efficiency and straightforward implementation, making it well suited for the present application.

5.2.1 Optimizer Choice

A wide range of optimization algorithms exists, each characterized by distinct methods and underlying principles. Consequently, a preliminary comparative study was carried out to identify the most appropriate approach for the specific problem under investigation. The analysis considered three main categories of optimizers: deterministic algorithms, genetic algorithms, and stochastic algorithms. Among the deterministic methods, both the `fmincon` solver available in `MATLAB` and the `SNOPT` optimizer were tested. These gradient-based techniques aim to efficiently converge to a local optimum through sequential quadratic programming and constrained nonlinear optimization schemes. In parallel, a genetic algorithm was implemented to assess the performance of evolutionary, population-based methods, which explore the design space through selection, crossover, and mutation operators. Furthermore, a stochastic particle swarm optimization (PSO) routine was developed and evaluated to investigate the capability of swarm-intelligence strategies in achieving global convergence. Although all these algorithms pursue the same optimization objective, they exhibit markedly different exploration and convergence behaviors, providing complementary insights into the performance and robustness of the proposed optimization framework.

***SNOPT* - Deterministic algorithm**

The deterministic gradient-based mode in `SNOPT` is particularly efficient when the optimization problem is smooth, differentiable, and characterized by a sparse constraint structure. In such cases, the algorithm can exploit accurate gradient information to rapidly converge to a local optimum with a high level of numerical precision. This makes the method especially suitable for trajectory optimization and guidance problems, where strict feasibility requirements and computational efficiency are of

primary importance. However, its reliance on local gradient information also represents its main limitation: the algorithm is inherently a local search method and is therefore prone to converge to suboptimal solutions in the presence of multiple minima. Moreover, the quality of the final solution is strongly influenced by the initial guess, which may hinder robustness when prior knowledge of the problem is limited.

fmincon function

In addition to **SNOPT**, another widely used optimizer is **Matlab's fmincon**, which is part of the Optimization Toolbox and is often adopted for constrained nonlinear problems. Its main advantage lies in accessibility: the solver is fully integrated within **Matlab**, supported by extensive documentation, and offers multiple algorithmic variants (such as SQP, interior-point, and trust-region methods) that can be selected according to the problem structure. When analytical gradients and, if needed, Hessians are provided, **fmincon** can be computationally efficient and accurate, while maintaining flexibility in the treatment of linear, nonlinear, and bound constraints. Nevertheless, **fmincon** shares the typical drawbacks of gradient-based deterministic methods. Its performance strongly depends on the initial guess, and convergence is usually limited to local optima, making it less suitable for highly non-convex landscapes. In the absence of analytical derivatives, it relies on finite differences, which may introduce additional computational cost and numerical inaccuracies.

Genetic algorithm

By contrast, the genetic optimization mode available in **Matlab** offers a global search capability that is not constrained by differentiability requirements. Instead of following a single path of descent guided by local gradients, genetic algorithms maintain and evolve a population of candidate solutions. This stochastic mechanism allows the exploration of wide regions of the design space, thereby reducing the risk of premature convergence to local minima. Genetic optimization is therefore advantageous in highly non-convex or poorly characterized problems, where global exploration is essential. Nevertheless, this broader search comes at the cost of computational efficiency: genetic algorithms typically require a large number of function evaluations, converge more slowly than deterministic methods, and often provide solutions that are less precise in terms of strict optimality. In addition, their performance is highly sensitive to the tuning of algorithmic parameters, such as population size and mutation rates. It is worth noting that genetic optimization methods do not inherently account for explicit constraints, but operate exclusively on the basis of the cost function. To ensure that the search is restricted to feasible regions of the design space, the constraint equations are reformulated as penalty terms within the cost function. These terms are multiplied by a sufficiently large weighting parameter, such that

any violation of the constraints is heavily penalized, thereby discouraging infeasible solutions and guiding the optimization process toward admissible trajectories.

Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is another population-based stochastic optimization technique inspired by the collective behavior of social organisms, such as bird flocks or fish schools. In PSO, a set of particles—each representing a potential solution—moves through the search space according to both its individual experience and the experience of neighboring particles. This mechanism allows PSO to dynamically balance exploration and exploitation, adapting the search behavior based on collective intelligence. Compared to classical genetic algorithms, PSO requires fewer control parameters and typically exhibits smoother convergence behavior. However, its performance is still sensitive to parameter tuning and it may suffer from premature convergence when the cost landscape features narrow valleys or multiple local minima. In the context of the present study, PSO was implemented as a benchmark to evaluate its capability in handling nonlinear, multi-variable optimization under aerodynamic and guidance constraints. While it demonstrated promising global search behavior and robustness against poor initial guesses, its computational cost was significantly higher than that of deterministic gradient-based methods. Consequently, PSO was primarily used for exploratory analyses and parameter sensitivity studies, rather than for routine trajectory optimization.

5.2.2 Rationale for the Chosen Method

For the present work, the deterministic gradient-based mode of SNOPT was selected as the primary optimization tool. This choice is motivated by the specific requirements of an autonomous guidance system for the horizontal landing of a reentry vehicle, where computational efficiency, robustness, and precision are of paramount importance. The problem formulation is characterized by smooth and differentiable dynamics, together with a sparse constraint structure that SNOPT can exploit effectively. Under these conditions, the deterministic approach ensures rapid and reliable convergence to feasible solutions while satisfying strict aerodynamic and operational constraints.

Although preliminary tests were also carried out with alternative optimizers, including the genetic mode of SNOPT, the Particle Swarm Optimization (PSO) routine, and Matlab's `fmincon`, the deterministic SNOPT implementation consistently demonstrated superior robustness and efficiency. PSO and genetic approaches exhibited stronger global search capabilities, but their higher computational cost and slower convergence made them less suitable for the real-time or near real-time guidance applications targeted in this study.

However, given the relatively small number of decision parameters in the present optimization problem, the PSO method proved particularly effective for performing

a preliminary convergence analysis. Its stochastic exploration allowed the identification of promising regions within the solution space without being overly affected by the initial guess. Therefore, a hybrid optimization strategy was adopted: the PSO algorithm was first employed to obtain an approximate global solution, which served as the initial guess for the deterministic SNOPT refinement. This sequential approach effectively combined the global exploration capability of PSO with the local precision and efficiency of the gradient-based optimizer, leading to faster convergence and improved numerical stability while ensuring that all dynamic and aerodynamic constraints were satisfied.

5.3 Initial Conditions, Constraints and Cost Function

After defining the optimization function, the next important step is the definition of *Initial Conditions*, *Constraints* and *Cost Function*, that completely define the problem.

Santa Maria Airport has been identified as the landing site. Santa Maria is a Portuguese airport in the autonomous region of the Azores. Its coordinates are $36^{\circ} 58' 24.17'' N$ $25^{\circ} 09' 55.14'' W$ and its altitude is 94 m ASML .

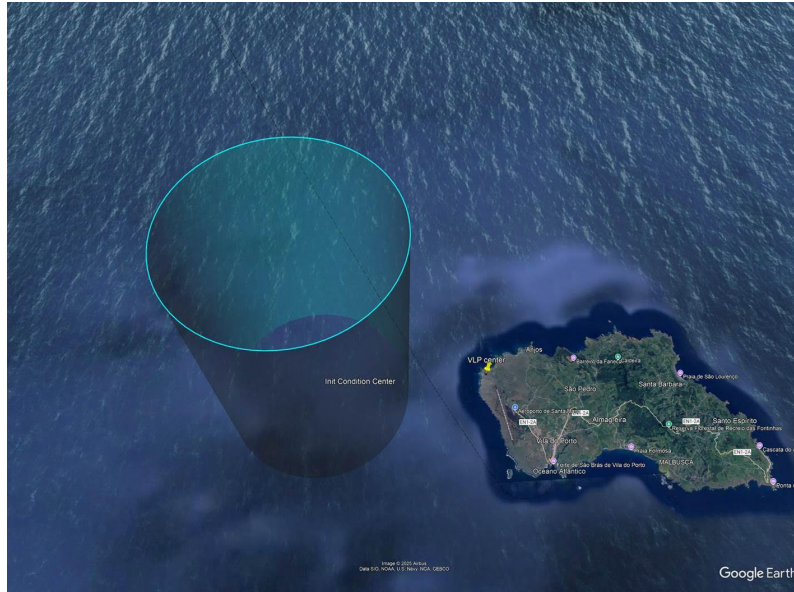


Figure 5.4: Initial Conditions and landing site

The initial and final conditions were determined during the study phase. For instance, the state vector for the landing simulation was either pre-defined or derived from the final state vector of the preceding simulation. The constraints were chosen

to ensure a successful landing and braking on the runway, all while adhering to the structural limitations of the vehicle and the scenario.

It is important to note that, in the following text, the constraints are presented without distinction; however, in optimization, they are typically classified into equality and inequality constraints. Equality constraints require that a given function of the decision variables be exactly satisfied, thereby restricting the feasible set to lie on a lower-dimensional manifold. Inequality constraints, by contrast, impose upper or lower bounds on functions of the decision variables, thus defining regions of admissibility within the search space.

5.3.1 Descent Guidance Optimization

Initial Conditions and Guess

The starting point is located at an altitude of $z = 15000 \text{ m}$. Initial velocity has been chosen to be at $M = 0.8$ with a flight path angle of $FPA = \gamma = -50^\circ$. From these parameters, the corresponding velocity components along the x and z axes can be derived.

$$\begin{cases} \sqrt{V_x^2 + V_z^2}/a = M = 0.8 \\ \tan^{-1}(V_z/V_x) = \gamma = -50^\circ \end{cases} \quad (5.1)$$

We obtain $V_x = 151.7338 \text{ m/s}$ and $V_z = -180.8291 \text{ m/s}$.

Initial attitude angle θ has been settled to -40° , while angular velocity $q = 0$.

The initial state vector is complete:

$$X_0 = \begin{cases} x(t_0) = 0 \text{ m} \\ z(t_0) = 15000 \text{ m} \\ V_x(t_0) = 151.7338 \text{ m/s} \\ V_z(t_0) = -180.8291 \text{ m/s} \\ q(t_0) = 0 \\ \theta(t_0) = -40^\circ \end{cases} \quad (5.2)$$

It's important to note that angles and angular velocities in the simulator are transposed in *rad* and *rad/s*, and linear velocities are reported in *ENU*.

As outlined in the preceding chapter, the initial guess is defined by a limited set of parameters, namely the values of α and the corresponding times associated with the ideal trajectory, together with the glide flight path angle FPA_0 (as reported in 5.1.2):

$$U_0 = [ALPHA, TIME, FPA_0] \quad (5.3)$$

The explicit numerical values of the initial guess are not provided, as they are subject to variation at each iteration of the optimization process.

Constraints

The vehicle is required to perform the landing maneuver under specific conditions. In particular, it must touch down at an altitude of $z = 94 \text{ m}$ (corresponding to the elevation of Santa Maria Airport) with a low vertical velocity, i.e., $-4 < V_z(t_f) < 0 \text{ m/s}$, in order to remain within the structural limits of the vehicle [26]. The longitudinal velocity at touchdown, $V_x(t_f)$, is constrained by the available runway length. Furthermore, both the attitude angle and the angle of attack must be positive at the moment of touchdown to ensure the execution of the flare maneuver. Additional limitations are imposed throughout the glide phase.

The constraints can be categorized according to their time of application, namely those enforced during the entire simulation and those applied only at the final time t_f :

$$\forall t \in [t_0; t_f] \quad C = -10^\circ \leq \alpha(t) \leq 45^\circ \quad (5.4)$$

It's worth noticing that an altitude limitation would have been redundant because, as described in Chapter 4, the simulation ends when the vehicle reaches the airport altitude. The final-time constraints are defined as follows:

$$C_f = \begin{cases} \alpha(t_f) \geq 0^\circ \\ \theta(t_f) \geq 0^\circ \\ z(t_f) \leq 95 \text{ m} \\ V_x(t_f) \leq 100 \text{ m/s} \\ V_z(t_f) \leq -4 \text{ m/s} \end{cases} \quad (5.5)$$

Cost Function

The primary objective in selecting the Cost Function J was to achieve a stable vehicle attitude as rapidly as possible and to maintain it throughout the glide phase, until the vehicle begins the flare maneuver, at T_1 . This stable attitude is defined by a Flight Path Angle (FPA) set at a target value of $FPA_0 = -20^\circ$. To accomplish this goal, the following Cost Function was established:

$$J = \int_0^{T_1} |FPA(t) - FPA_0| dt \quad (5.6)$$

The time-based integration term compels the optimizer to converge to the desired FPA_0 in the shortest possible time while the integrating term ensures maintaining this angle until the initiation of the flare maneuver. As shown in the previous chapter, FPA_0 and T_1 have initial values but are not fixed and may change during the optimization process.

5.3.2 Descent Control Optimization

Initial Conditions and Guess

The initial conditions are identical to those used in the *Guidance Descent* simulation:

$$X_0 = \begin{cases} x(t_0) = 0 \text{ m} \\ z(t_0) = 15000 \text{ m} \\ V_x(t_0) = 151.7338 \text{ m/s} \\ V_z(t_0) = -180.8291 \text{ m/s} \\ q(t_0) = 0 \\ \theta(t_0) = -40^\circ \end{cases} \quad (5.7)$$

It is important to note that the reference profile of α_{id} is also provided as an input, but it is interpolated as a function of altitude rather than time, as previously specified.

The initial guess, on the other hand, is different, as it is defined in terms of the proportional–integral–derivative (PID) controller parameters:

$$U_0 = [K_p, K_i, K_d] \quad (5.8)$$

Constraints

As described in Chapter 4, this simulation is performed without any explicit constraints, in order to allow the optimizer to focus solely on minimizing the cost function. In fact, if the cost function reaches a sufficiently small value, it implies that the constraints are inherently satisfied.

Cost Function

The cost function is defined as the integral of the module of the error $e(t)$ between the commanded guidance angle of attack (α_{ideal}) and the actual response:

$$J = \int_0^T |e(t)| dt = \int_0^T |\alpha(t) - \alpha_{ideal}(t)| dt \quad (5.9)$$

The use of the integral aims to minimize the tracking error over the shortest possible time horizon, thereby improving both accuracy and responsiveness of the control loop.

5.3.3 Landing Optimization

Initial Conditions and Guess

Certain initial conditions for the braking phase simulation are derived from the end of the preceding simulation. Specifically, the horizontal velocity and the pitch angle

(θ) are taken from the values the vehicle has just as it touches down, while the altitude remains fixed at the landing altitude. The vertical velocity is converted into an angular velocity based on the conservation of angular momentum law:

$$q_t = V_{xf,Descent} \cdot \frac{r}{r^2 + I_y/m} \quad (5.10)$$

The initial state vector is composed as follows:

$$X_0 = \begin{cases} x(t_0) = 0 \text{ m} \\ z(t_0) = 94 \text{ m} \\ V_x(t_0) = V_{xf,Descent} \\ V_z(t_0) = 0 \text{ m/s} \\ q(t_0) = q_t \\ \theta(t_0) = \theta_{f,Descent} \end{cases} \quad (5.11)$$

The guess is composed with the control vector B and E , and the total simulation time T :

$$U_0 = [T, \ B, \ E] \quad (5.12)$$

Final Constraints

The vehicle must be capable of landing on a 3,048-meter runway (Santa Maria runway length). Considering a typical margin of error for reentry vehicles ($\pm 15\%$), the effective distance available for deceleration is 2,629 meters [25] [27]. Other constraints may include reaching an attitude angle $\theta = 0$ and ensuring that the horizontal velocity is equal to zero. The latter constraint has been slightly modified by setting the final velocity to a value slightly above zero, in order to prevent the simulator from reaching negative velocities during the final propagation steps, which would compromise the validity of the simulation.

$$C_f = \begin{cases} x(t_f) \leq 2629 \text{ m} \\ \theta(t_f) = 0^\circ \\ V_x(t_f) \leq 0.5 \text{ m/s} \end{cases} \quad (5.13)$$

Cost Function

For the *Landing* simulation, the primary objective guiding the selection of the Cost Function was to ensure a safe and controlled pitch-down maneuver without compromising the structural integrity of the vehicle. This critical requirement necessitated that the optimizer guide the vehicle to a very low angular velocity. Specifically, the target angular velocity was set to $q_0 = -0.5^\circ/\text{s}$. To achieve this goal, a Cost

Function similar to that used in previous phases was established, as shown in the following equation:

$$J_L = \int_0^T |q(t) - q_0| dt \quad (5.14)$$

In this function, the integration of the difference between the actual angular velocity (q) and the target value (q_0) over the total time of the maneuver (T) serves as the objective for the optimizer. This formulation forces the system to minimize the deviation from the desired slow pitch rate throughout the landing phase and ensures a gentle and controlled final descent, which is paramount for the safety and success of the landing. It is important to note that, unlike in the *Descent* simulation, the objective parameter q_0 is now fixed for every optimization attempt.

5.4 First Results

The optimization performed with PSO and SNOPT yielded satisfactory results, achieving full compliance with the imposed constraints and maintaining a contained cost function in both simulation scenarios. The results are presented below, including the discretizations of the control surfaces.

Descent

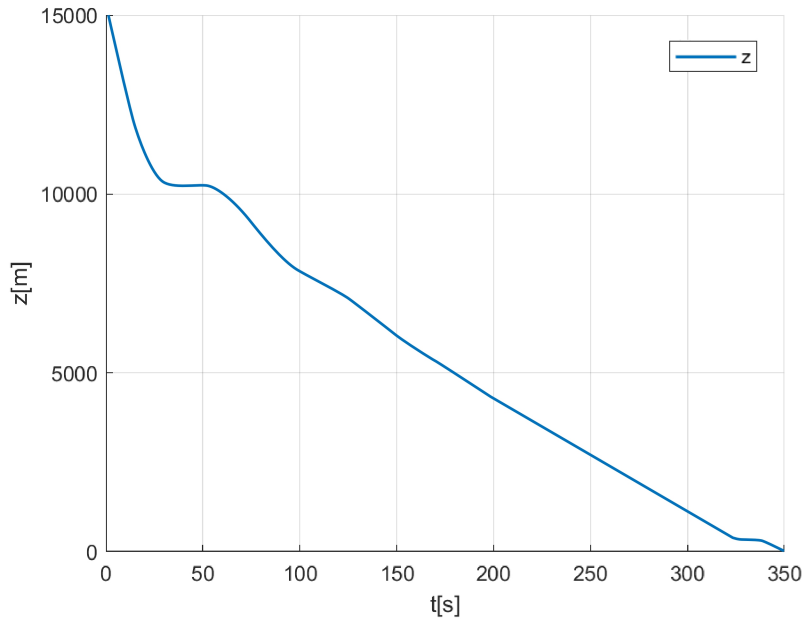


Figure 5.5: $z(t)$

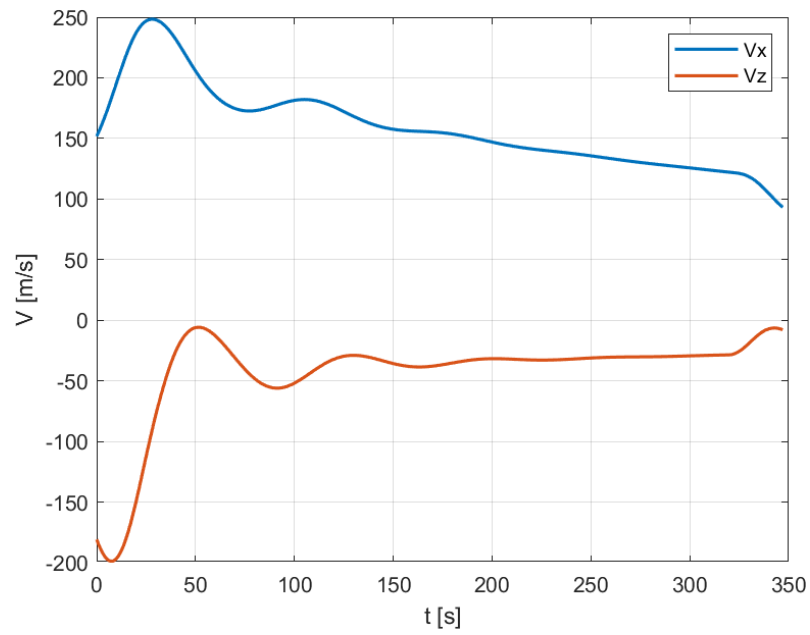


Figure 5.6: $v_x(t)$ e $v_z(t)$

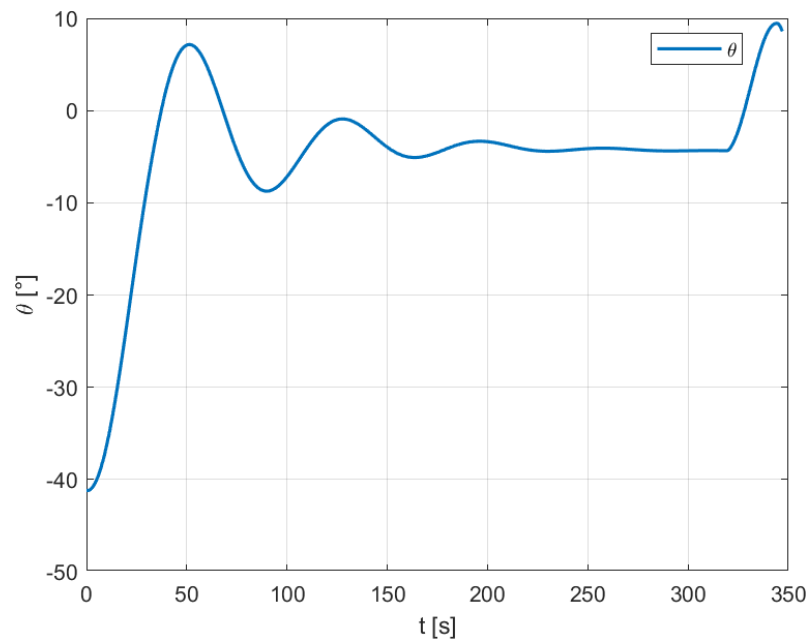


Figure 5.7: $\theta(t)$

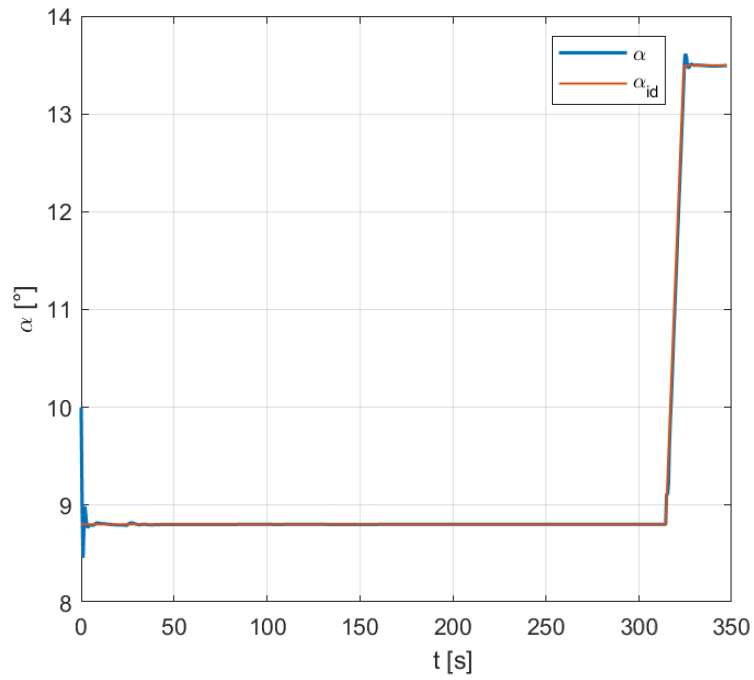


Figure 5.8: $\alpha_{id}(t)$ vs $\alpha(t)$

It's worth zooming in on the graphic of $\alpha(t)$ to understand the action of the *PID* controller:

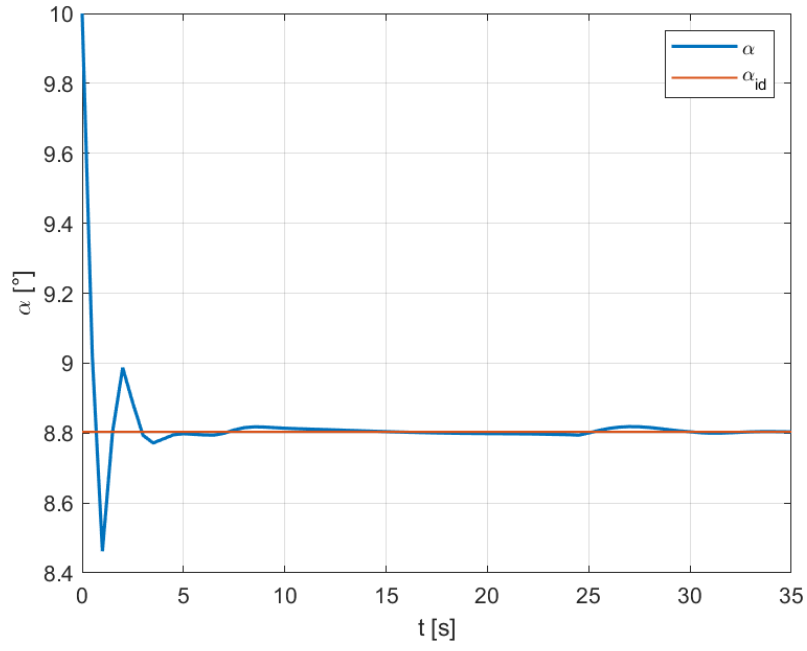


Figure 5.9: $\alpha_{id}(t)$ vs $\alpha(t)$, first seconds

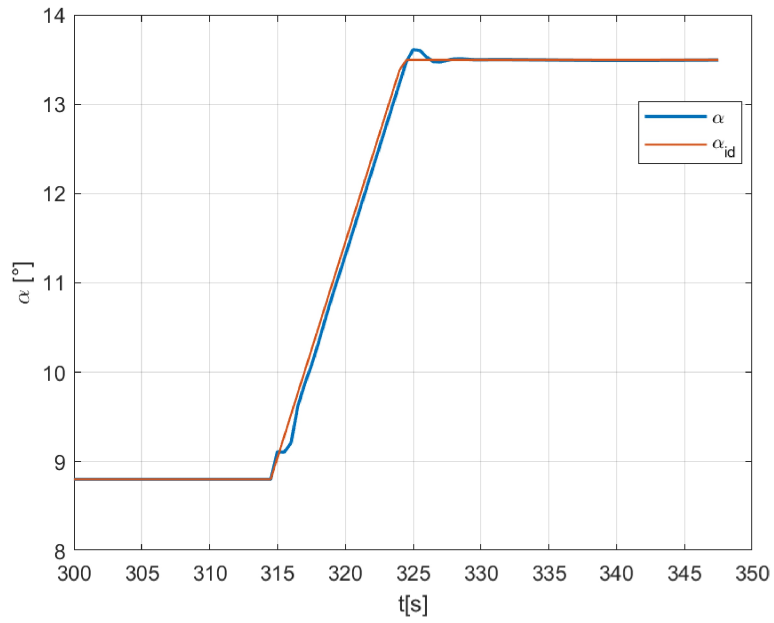


Figure 5.10: $\alpha_{id}(t)$ vs $\alpha(t)$, flare maneuver and last seconds

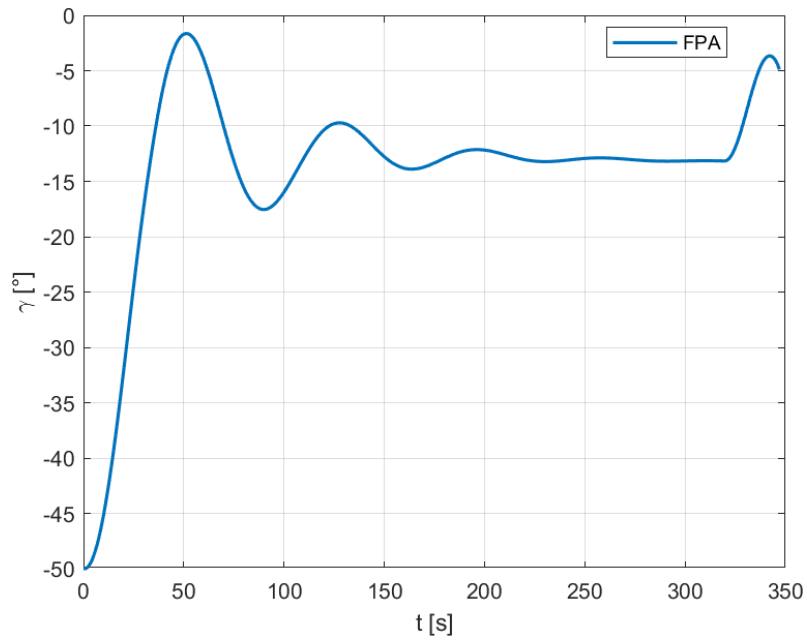


Figure 5.11: $FPA(t)$

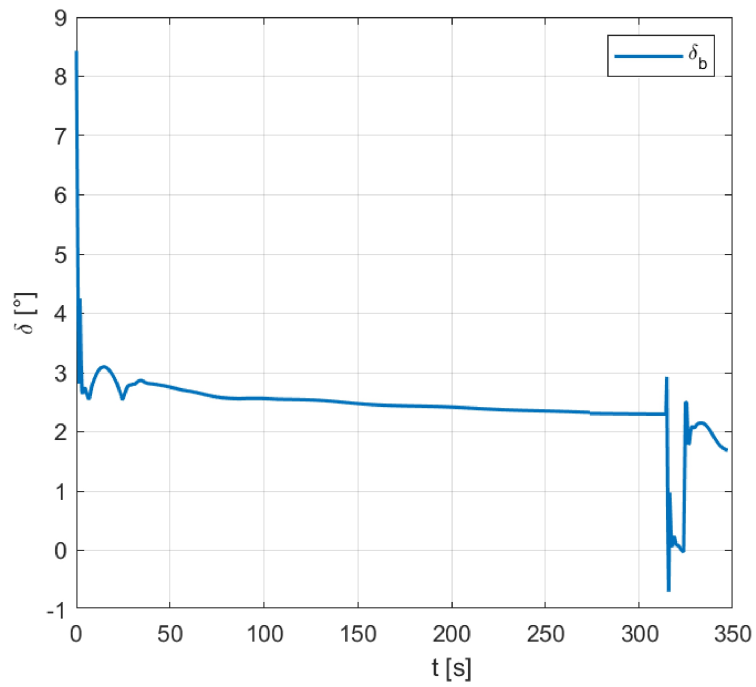
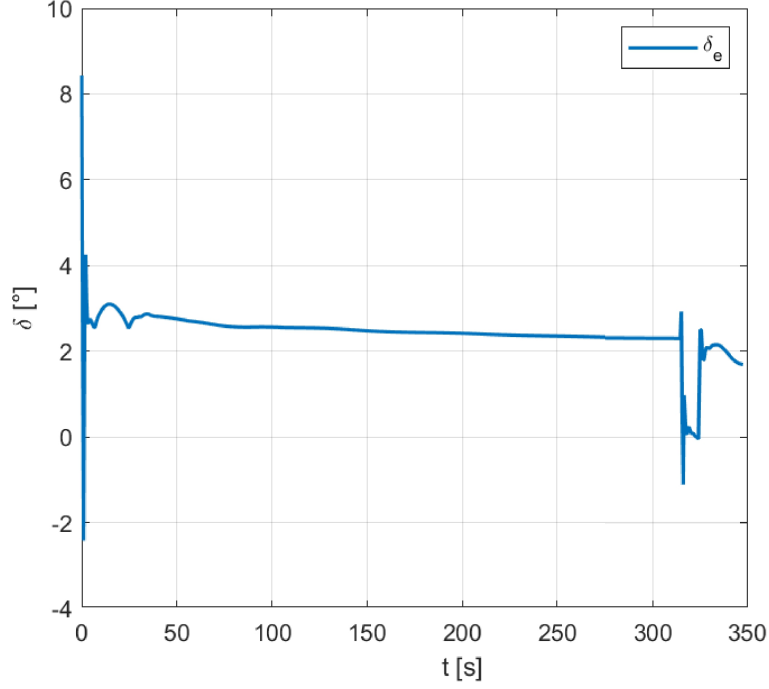


Figure 5.12: $\delta_b(t)$

Figure 5.13: $\delta_e(t)$

During the descent phase, the vehicle converges, after few instabilities, to the desired flight path angle FPA_0 (not the -20° settled in the guess but a slightly different value found by the optimizer during the process) and maintains it with stability. At approximately 600 meters altitude, the flare maneuver is initiated, beginning with an initial pitch-up and stabilizing to a final value of α until touchdown is achieved. The evolution of the main characteristic quantities follows this common pattern: an initial adjustment phase to reach stable conditions, maintained until the onset of the flare maneuver. The pitch angle θ remains stable and subsequently increases up to approximately 9° at touchdown, while the angle of attack α exhibits a similar trend. The vertical speed is stable throughout most of the trajectory and then gradually approaches zero toward the end of the simulation. Regarding the control variables, δ_b and δ_e remain nearly constant throughout the flight, thereby providing a stabilizing negative moment. In the last 30 seconds of simulation, the deflection rapidly decrease. This behavior subsequently induces the pitching-up moment required for the landing phase.

The optimized simulation for most of the variables over time (like θ and FPA) exhibits a characteristic behavior that can be interpreted in light of the **Turnpike property**. Initially, the trajectory shows oscillations as the system rapidly adjusts to the boundary conditions at the starting point. After this transient phase, the value stabilizes around a nearly constant value for the majority of the trajectory,

reflecting the system's tendency to remain close to a steady-state solution that minimizes the cost function. Toward the end of the simulation, additional oscillations are observed as the trajectory adapts to perform the flare maneuver and satisfy the terminal boundary conditions. This pattern is consistent with the turnpike phenomenon, whereby the optimal trajectory spends most of the horizon near a quasi-stationary state, with deviations primarily localized at the beginning and end of the time interval to enforce the required constraints.

Landing

The optimization for *Landing* phase and braking on the runway manages to perform a successful arrest of the vehicle while remaining in length constraints. It also manages to have a slow pitch down of the nose landing gear.

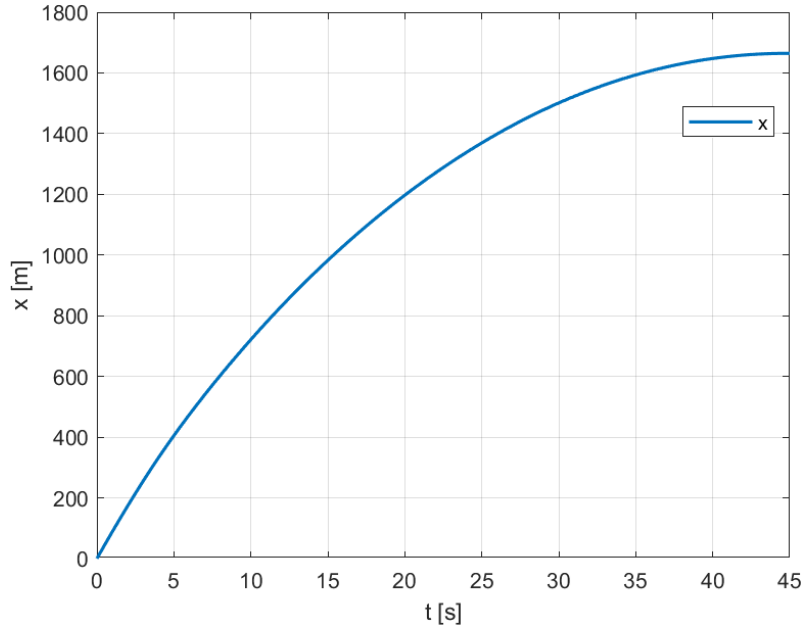


Figure 5.14: $x(t)$

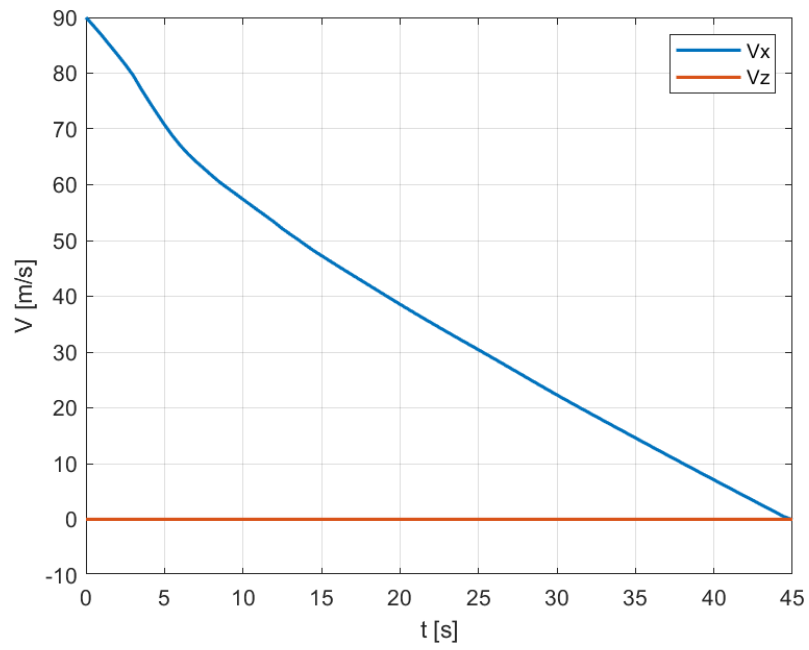


Figure 5.15: $v_x(t)$

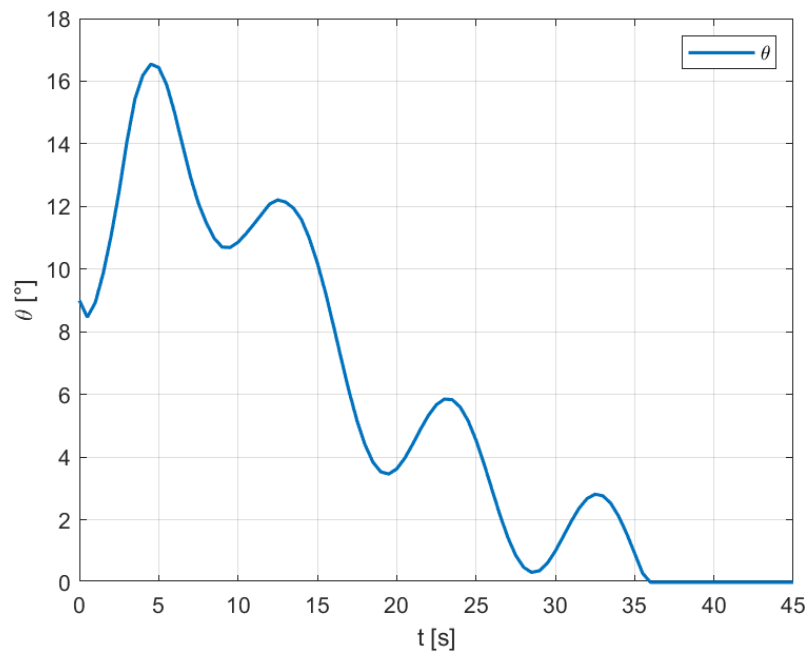


Figure 5.16: $\theta(t)$

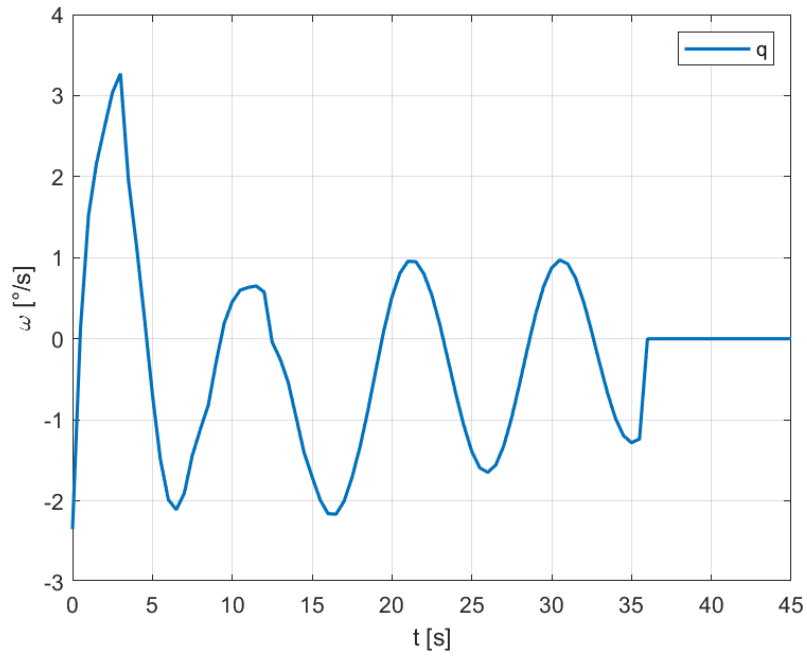


Figure 5.17: $q(t)$

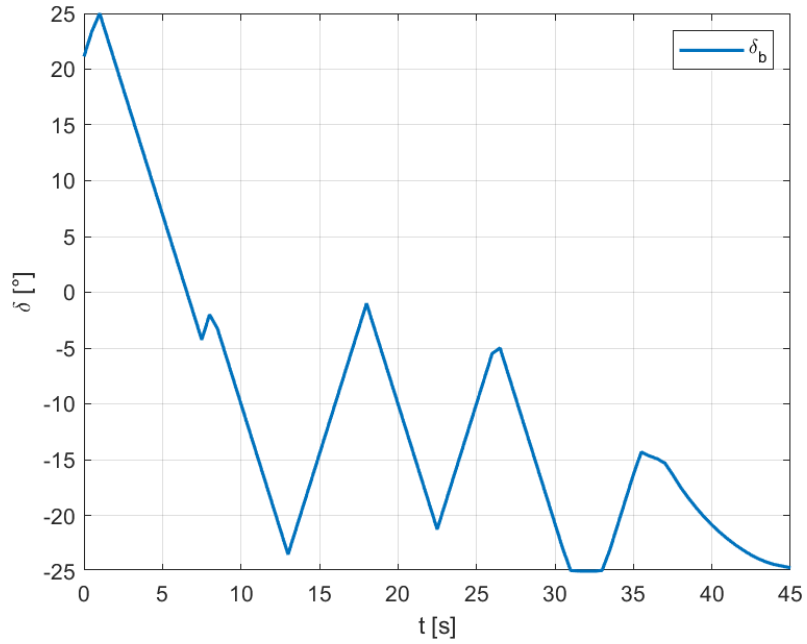
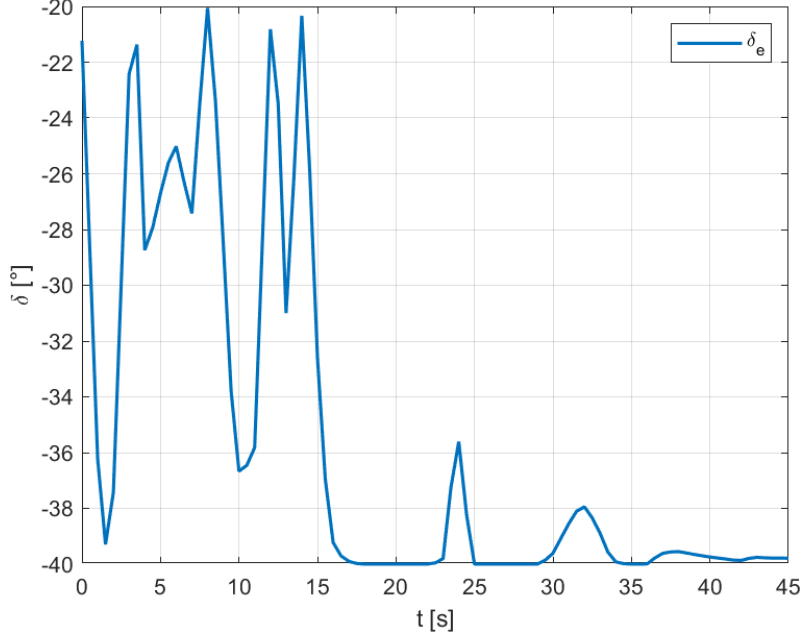


Figure 5.18: $\delta_b(t)$

Figure 5.19: $\delta_e(t)$

The optimal solution aims to stabilize the nose pitch-down rotation induced by the transposition of the linear vertical velocity at touchdown into angular velocity. In fact, both δ_b and δ_e assume negative values for the majority of the time, thereby generating a positive compensating moment. The pitch rate q remains very small in magnitude, not exceeding $-5^\circ/\text{s}$.

Although the results comply with all imposed constraints, it is noteworthy that the pitch-down control exhibits a certain degree of overshooting. The control inputs display an oscillatory behavior, alternately exceeding and then correcting in an attempt to track a desired pitch-rate value. This trend is reflected in the time histories of both the pitch angle θ , which oscillates rather than performing a constant decrease, and the angular velocity q . Such behavior may be attributed to the highly computational model employed to simulate the roll-out phase: more than one hundred variables are involved, and no dedicated controller—such as the PID used during the descent phase—is implemented. Consequently, the optimizer struggles to identify a strictly smooth and steady solution. This outcome provides an interesting point of comparison between the different optimization approaches, emphasizing how the constraints are nevertheless satisfied, albeit with higher computational times and the occurrence of overshooting phenomena.

6. Overall Requirements Definition and Future Development

To achieve the objective of this thesis, namely the validation of requirements for a horizontal re-entry vehicle, it is necessary to investigate which alternative configurations are capable of performing a successful landing in addition to the one just optimized. The underlying idea of performing this discretization is to obtain a feasibility map of the landing capability over a wide range of combinations of vehicle characteristics, by interpolating through the discretized values. This map can then be employed to identify the admissible characteristics of the vehicle according to specific requirements—for instance, determining the minimum achievable wing surface area or the minimum lift-to-drag ratio (L/D) for a fixed mass.

To this end, a series of optimizations must be carried out by varying the main characteristics of the vehicle—particularly aerodynamic properties and aerodynamic coefficients—as well as intrinsic parameters. The constraints are also varied in order to be as non-conservative as possible, while still remaining within the logistical limits of the territory.

The series of optimizations was structured as follows. First, the parameters to be varied were selected, as outlined below:

- m mass: 5 values between 10000 kg and 100000 kg have been chosen.
- S wing surface: varies from 100 m^2 to 300 m^2 in 3 values.
- L/D ratio: to perform aerodynamic performance variation, the value of L has been multiplied for $X_{L/D}$, a varying factor from 1 to 2. By doing this the L/D_{max} parameter varies from a value of 4 to 8 depending on the simulations.

As mentioned in the previous chapter, the variables dependent on these parameters are also parametrically scaled as functions of them. For instance, the mean aerodynamic chord (MAC) is parametrically scaled as a function of the wing surface area S , while the moment of inertia I_Y is parametrically scaled as a function of the vehicle mass.

These are the total variation parameters:

m [kg]	S [m ²]	$X_{L/D}$
10000	100	1
32500	200	1.5
55000	300	2
77500	-	-
100000	-	-

Table 6.1: Discretization of analysis parameters

The choice of these discretizations represents a compromise between ensuring a sufficiently broad range of parameter variations and avoiding excessive computational cost, which would otherwise result in prohibitively long solution times. To mitigate this issue and reduce computation time, the **Matlab** parallelization tool **parfor** is employed.

The **Matlab** code for requirements validation operates as follows:

1. A cell array ($C = \{5 \times 3 \times 3\}$) is generated, containing all possible combinations of the previously defined parameters. In the first **parfor** loop, each core processes one parameters combination, corresponding to a single element of C , and give it as input for **Simulink**.
2. The PSO+SNOPT optimizer attempts to converge to a solution for the two *Descent* simulations with the selected parameters, satisfying the imposed constraints while minimizing the cost function.
3. If the 2 simulation is successful, the 2 guess solutions (α_1 , α_2 , T_1 , T_2 , FPA_0 and K_p , K_i , K_d) and cost function + constraints values are stored in the vectors **bestU** and **bestF**. Conversely, if the simulation fails or does not respects constraints, the entries of these vectors are assigned the value **-inf**. Once all parameter combinations have been processed, the first **parfor** loop terminates.
4. A second **parfor** loop is then executed, performing optimization for the *Landing* simulations. This step is restricted to the parameter combinations that did not lead to failure during the *Descent* phase, i.e., only for the elements of C associated with maximum tolerance in **bestF**. The outcomes of the *Landing* simulations are similarly collected in the corresponding result vectors.

6.1 Results

The following figure illustrates the feasibility grid related to the *Guidance* optimization process. The axes represent the discretizations of the main parameters (m , S ,

and L/D). It can be observed that, in most cases, the guidance system successfully satisfies the imposed constraints and enables the vehicle to perform a complete landing after executing a flare maneuver, which reduces the velocity and aligns the vehicle in an appropriate landing attitude. Moreover, the value of the cost function remains sufficiently low across all feasible points, indicating that the vehicle is capable of reaching a steady flight-path angle within a relatively short time and maintaining it throughout the glide phase.

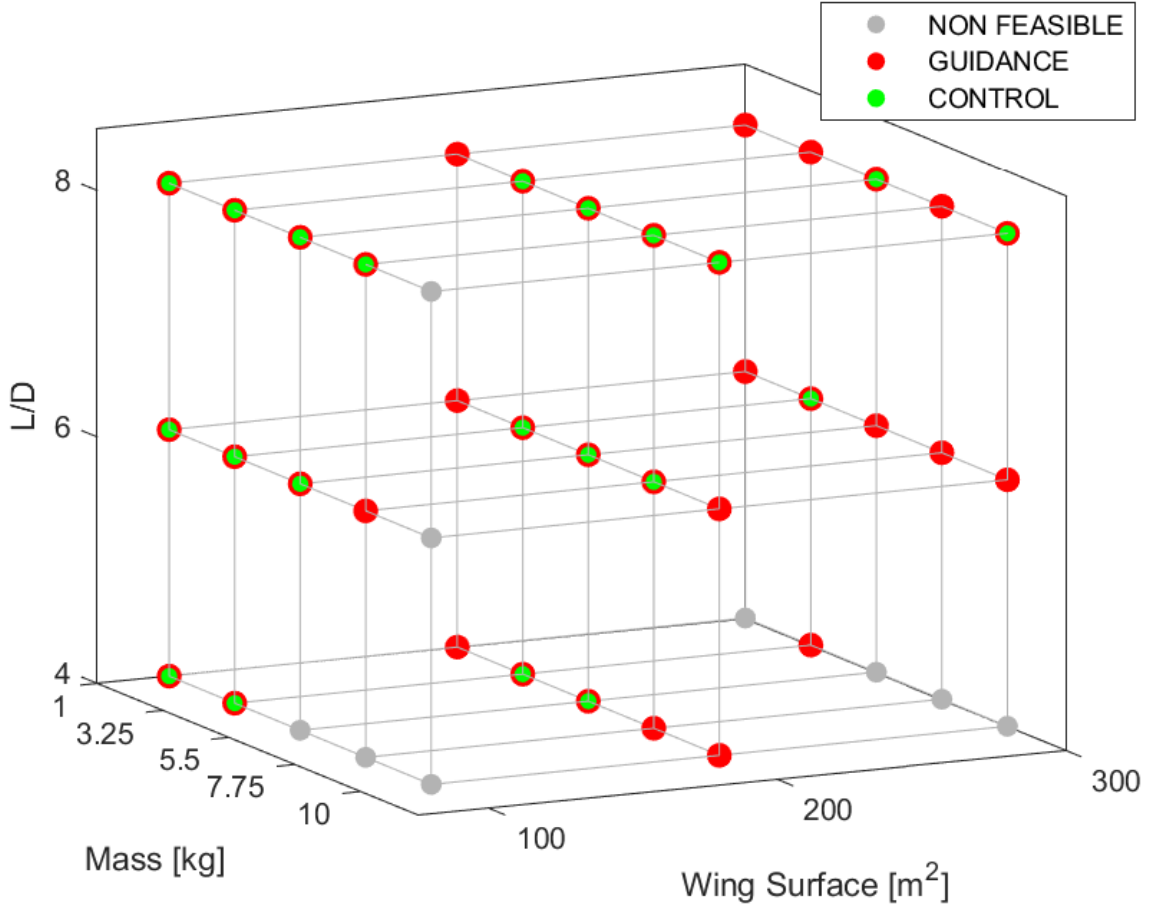


Figure 6.1: Final Results

The points highlighted in red correspond to feasible configurations, in which all constraints are satisfied within a tolerance of 10^{-4} . Conversely, some points are identified as infeasible, as under those specific conditions the vehicle is unable to complete a successful landing. This infeasibility may be attributed to an excessive mass relative to the wing surface area, or to a wing area or lift-to-drag ratio (L/D) that is too low to allow a proper flare maneuver. It is also interesting to note that an increase in wing surface area does not necessarily guarantee the success of the opti-

mization. This occurs because a larger wing surface leads not only to an increase in lift, but also to a proportional increase in drag. Therefore, for the same L/D ratio, a vehicle with a larger wing may experience greater aerodynamic resistance, making it more difficult to perform the flare maneuver effectively.

The plot is further enriched with the results of the *Control descent* (colored in green) which verifies whether the vehicle is effectively able to follow the guidance commands and complete both the maneuver and the landing. As can be observed, not all of the red points are reachable by the controlled vehicle, and this may be caused by several factors. The first and most significant factor is that the defined control surfaces may not generate a sufficient control moment to effectively follow the guidance trajectory. A second possible cause lies in the input of the PID control parameter: the controller tracks the evolution of the angle of attack, α , both in time and altitude, compensating for possible delays or anticipations in reaching flare altitude.

The results corresponding to the *landing phase* are also reported. These simulations were performed only for the configurations that exhibited acceptable feasibility during the descent phase, namely the green points in the feasibility grid. All of these simulations resulted in terms of feasibility and constraints satisfaction, successfully completing a smooth nose pitch-down maneuver and a deceleration within the considered runway length. Therefore, the landing results are not highlighted with a distinct color, but are instead included within the green points already shown in the figure.

Overall, the results confirm that the proposed guidance and control framework allows the vehicle to perform a complete and controlled landing within the considered parameter space, while clearly identifying the limits of feasibility associated with mass, aerodynamic efficiency, and control effectiveness.

6.2 Future Developments and Reflections on Research

The work presented in this thesis has laid the foundations for a systematic definition of the requirements for the horizontal landing of a re-entry vehicle. Nevertheless, several potential extensions can be envisaged. Future research could expand the modeling framework, refine the optimization methodology, and strengthen the coupling between aerodynamic, guidance, and control aspects. The following subsections outline several promising directions for advancement, together with broader reflections on the research methodology.

Degrees of Freedom of the Simulator

A natural extension of the present work would involve expanding the simulation environment from a three-degree-of-freedom (3-DOF) longitudinal model to a full six-degree-of-freedom (6-DOF) dynamic representation. This enhancement would enable the study of lateral-directional effects—such as sideslip, roll dynamics, and yaw control—which play a crucial role in the final alignment and crosswind compensation phases of runway approach.

The inclusion of these dynamics would inevitably increase both the mathematical and computational complexity of the problem. However, such complexity would be offset by the gain in physical realism and predictive fidelity, allowing for a more accurate assessment of control surface effectiveness, actuator authority, and aerodynamic coupling during landing. Moreover, a higher-fidelity simulator would enable the simultaneous optimization of longitudinal and lateral trajectories, contributing to a unified description of vehicle behavior during the complete terminal phase.

Aerodynamic Databases

The current implementation relies primarily on aerodynamic data derived from the Space Shuttle configuration. A valuable extension would consist of testing the proposed methodology against other aerodynamic databases, corresponding to different wing planforms, aspect ratios, or lifting-body geometries. For instance, data derived from slender delta wings, double-delta configurations, or blended-wing bodies could provide insights into how aerodynamic efficiency and control sensitivity influence landing feasibility.

The inclusion of diverse aerodynamic models would also facilitate the development of scaling laws applicable to a broader class of re-entry vehicles, ultimately supporting preliminary design activities. Future efforts could further consider the integration of surrogate aerodynamic databases generated through reduced-order modeling or machine-learning interpolation of CFD results, thus ensuring continuity between conceptual design and high-fidelity aerodynamic analysis.

Alternative Optimization Strategies

Another significant improvement could concern the optimization architecture itself. In the current framework, the descent and landing optimizations are performed sequentially through independent black-box modules. Although this approach simplifies implementation, it introduces discontinuities between the output of the descent phase and the initial conditions of the landing phase.

A more advanced methodology would merge the two stages into a single, coupled optimization problem, in which both black boxes are included. This unified formulation would ensure smoother transitions between flight regimes, better coupling of constraints, and potentially shorter computation times, since the optimizer would

handle the entire trajectory in a single run. Exploring other hybrid optimization techniques – such as *IPOPT* (Interior Point Optimizer) for gradient-based algorithms and or surrogate for stochastic algorithms — could further improve convergence robustness and facilitate the exploration of complex, multimodal solution spaces.

Advanced and Multi-Objective Control Laws

While the present study adopted a PID-based control architecture for its simplicity and robustness, future work could investigate more sophisticated control schemes. Linear Quadratic Regulators (LQR), Model Predictive Controllers (MPC), or adaptive and gain-scheduled control systems could be implemented to provide higher precision and robustness under varying aerodynamic conditions.

Furthermore, rather than limiting the control objective to pitch attitude, the controller could be designed to simultaneously regulate additional parameters such as flight path angle, velocity, or vertical acceleration. This multi-parameter control approach would allow the system to maintain trajectory adherence even under disturbances or modeling uncertainties, ultimately leading to smoother and more stable landing profiles.

Refinement of Requirement Mapping and Interpolation Techniques

The current mapping of aerodynamic and performance requirements could be refined by increasing the density of sampling points and adopting interpolation techniques to ensure smoother transitions between discrete results. By applying multidimensional interpolation or regression methods, the requirement maps could evolve from discrete feasibility domains to continuous response surfaces, thereby enabling a more intuitive and precise identification of optimal configurations.

Such an approach would be particularly beneficial for early design optimization, where the sensitivity of performance metrics to aerodynamic parameters plays a critical role in system-level trade-off analysis.

Convergence Enhancement and Sensitivity to Initial Guesses

A final avenue of improvement concerns the convergence behavior of the optimization process. As demonstrated in the experimental phase, the quality of the initial guess substantially influences the solver’s ability to converge toward feasible solutions. Future studies could undertake a systematic analysis of the solution landscape, focusing on the characterization of local minima and the development of heuristics for generating informed initial conditions.

Techniques such as continuation methods, warm-starting strategies, or surrogate-based pre-optimization could significantly improve convergence reliability and reduce computational time. This line of research would also contribute to a deeper

understanding of the underlying dynamics of the problem, shedding light on the sensitivity of optimal solutions to both model parameters and initial conditions.

Broader Reflections on the Research

Beyond the specific methodological extensions, the work presented herein highlights the importance of integrating multidisciplinary approaches in the design and analysis of reusable re-entry vehicles. The combination of flight dynamics, control theory, and numerical optimization proved to be a powerful framework for investigating complex aerospace problems characterized by strong nonlinearities and coupled constraints. Future developments should continue to emphasize this interdisciplinary perspective, promoting the creation of flexible simulation tools that can adapt to different mission profiles and vehicle architectures.

In conclusion, while the present study has achieved its primary goal of defining baseline requirements for horizontal landing, it also reveals a wide range of open challenges. Addressing them will not only enhance the technical maturity of the proposed methodology but also contribute to the broader advancement of reusable space transportation systems, bridging the gap between theoretical modeling and operational implementation.

7. Conclusions

This thesis has presented a comprehensive methodology for defining the aerodynamic and control requirements necessary to enable the horizontal landing of a reusable re-entry vehicle. Starting from a theoretical framework rooted in flight dynamics, guidance and control theory, and numerical optimization, the research developed an integrated approach capable of simulating and optimizing the final descent and landing phases with a high degree of fidelity and computational efficiency.

The work addressed the landing problem as a multi-phase optimization task, decomposing it into distinct yet interconnected stages — Descent Guidance, Descent Control, and Landing. This modular structure allowed for the independent assessment of each phase while ensuring global consistency in the vehicle’s dynamic response. The adopted optimization framework followed a hybrid strategy that combined a global and a local optimization stage. Specifically, a Stochastic Algorithm (PSO) was first employed to perform a global exploration of the solution space and to identify promising regions within the design domain, thereby reducing the likelihood of convergence to suboptimal local minima. Subsequently, a deterministic Sequential Quadratic Programming (SQP) method, implemented through the SNOPT solver, was applied to refine the solution locally and achieve rapid convergence to the optimal trajectory. This two-step approach effectively balanced exploration and exploitation, improving robustness and computational efficiency. The use of MATLAB and Simulink enabled a flexible simulation environment that integrated the physical modeling of forces and moments, the implementation of control laws, and the parallel execution of computational tasks through the Parallel Computing Toolbox.

The results obtained from the optimization process provided valuable insights into the feasibility of horizontal landing maneuvers for lifting-body and winged re-entry configurations. In particular, the analyses identified the key parameters— lift-to-drag ratio (L/D), surface area (S) and mass (m)—that delineate the boundary between feasible and non-feasible landing conditions. These findings allow for the establishment of minimum performance thresholds that future reusable vehicles must meet to achieve controlled runway landings without the aid of parafoil or retro-propulsion systems. The proposed framework thus contributes to bridging the gap between theoretical modeling and early design assessment, offering a practical tool

for preliminary mission analysis and vehicle design.

Beyond the immediate results, this research highlights the importance of coupling dynamic simulation with hybrid optimization techniques in the design of next-generation re-entry systems. The integration of realistic aerodynamic models, advanced control strategies, and parallel computational methods provides a scalable foundation for more complex analyses, including six-degree-of-freedom (6-DOF) dynamics, atmospheric uncertainties, and actuator nonlinearities. Moreover, the methodology can be extended to evaluate crosswind landings, variable-mass effects, or adaptive guidance laws for autonomous systems.

In conclusion, the thesis advances the current understanding of horizontal landing feasibility for reusable vehicles by offering both a methodological contribution and a quantitative characterization of critical requirements. While the presented model operates under simplifying assumptions, it establishes a solid basis for future investigations aimed at refining landing guidance algorithms, validating results with high-fidelity simulations, and ultimately supporting the development of safer, more efficient, and fully autonomous re-entry systems. The work thereby contributes to the long-term vision of sustainable and cost-effective access to space.

Bibliography

- [1] *Shuttle Entry Guidance*, Lyndon B. Johnson Space Center, NASA Technical Memorandum 78476, Houston, Texas, February 1979
- [2] Frank J. Regan and Satya M. Anandakrishnan, *Dynamics of Atmospheric Re-Entry*, AIAA Education Series, American Institute of Aeronautics and Astronautics, Washington DC, 1993, ISBN 978-1563470687
- [3] Mykel J. Kochenderfer, Tim A. Wheeler, and Kyle H. Wray, *Algorithms for Decision Making*, MIT Press, Cambridge, Massachusetts, 2022, ISBN 978-0262047012, <https://algorithmsbook.com/>
- [4] Jan Roskam, *Airplane Flight Dynamics and Automatic Flight Controls*, Volume 1, DARcorporation, 1995, ISBN 978-1884885550
- [5] Jan Roskam, *Airplane Flight Dynamics and Automatic Flight Controls*, Volume 2, DARcorporation, 1998, ISBN 978-1884885567
- [6] Leonid Bussler, Sven Stappert, Jascha Wilken, Martin Sippel, Ingrid Dietlein, *Analysis Of VTVL And VTHL Reusable Launch Vehicle Configurations*, DLR Institute of Space Systems, SART TN-009/2019, Bremen, Germany, 2019
- [7] ResearchGate figure, *Demonstration of the body and wind frames on the aircraft*, accessed 2023, https://www.researchgate.net/figure/Demonstration-of-the-body-and-wind-frames-on-the-aircraft_fig4_368832072
- [8] MathWorks Documentation, *Spatial Representation: Coordinate Systems and Conventions*, accessed 2023, <https://it.mathworks.com/help/fusion/gs/spatial-representation-coordinate-systems-and-conventions.html>
- [9] ResearchGate figure, *Aerodynamic lift and drag force on blade*, accessed 2023, https://www.researchgate.net/figure/Figure-2-Aerodynamic-lift-and-drag-force-on-blade_fig2_339040715
- [10] NASA Langley Research Center, *The Aeronautics of the Space Shuttle: Control Surfaces*, accessed 2023, <https://www.nasa.gov/centers-and-facilities/langley/the-aeronautics-of-the-space-shuttle/>

- [11] A. Miguel San Martin, Steven W. Lee, Edward C. Wong, *The Development of the MSL Guidance, Navigation, and Control System for Entry, Descent, and Landing*, Journal of Spacecraft and Rockets, Vol. 51, No. 4, pp. 1062–1078, 2014, DOI: 10.2514/1.A32660
- [12] Defense Systems Information Analysis Center (DSIAC), *Guidance, Navigation, and Control (GNC) Systems Overview*, Technical Report AD1182620, October 2022, <https://dsiac.dtic.mil/wp-content/uploads/2022/10/AD1182620.pdf>
- [13] Mykel J. Kochenderfer and Tim A. Wheeler, *Algorithms for Optimization*, 2nd edition, MIT Press, Cambridge, Massachusetts, London, England, 2022, ISBN 978-0262545983
- [14] Marios Papachristou, *Optimization Algorithms*, online project documentation, accessed 2023, <https://mpalaourg.me/project/optimization-algorithms/>
- [15] PID Controller scheme, <https://product-help.schneider-electric.com/Machine%20Expert/V1.1/en/Util/topics/pid.htm>
- [16] A. Del Oliveira, *Guidance And Control System Design For Reusable Launch Vehicle Descent And Precise Landing*, PhD Thesis, Politecnico di Milano, Department of Aerospace Science and Technology, 2015
- [17] A. Tewari, *Atmospheric and Space Flight Dynamics*, Birkhäuser, Boston, 2007, ISBN 978-0817644390
- [18] Frank J. Regan and Satya M. Anandakrishnan, *Dynamics of Atmospheric Re-Entry*, AIAA Education Series, 1993, ISBN 978-1563470687
- [19] Dr.-Ing. John Nassour, *Transformation Matrices*, Technische Universität Chemnitz, Lecture Notes, 2017, <https://www.tu-chemnitz.de/informatik/KI/edu/robotik/ws2017/trans.mat.pdf>
- [20] NASA, *Aerodynamic Design Data Book Orbiter Vehicle STS-1*, Volume 1, SD72-SH-0060, Rockwell International, 1975
- [21] E. Mooij and I. Barkana, *Stability Analysis of an Adaptive Guidance and Control System applied to a Winged Re-entry Vehicle*, Acta Astronautica, Vol. 59, Issue 1-5, pp. 37–57, 2006, DOI: 10.1016/j.actaastro.2006.02.002
- [22] Francisco Miguel Camacho Camara, *Development and Design of the Terminal Area Energy Management Guidance for a Reusable Launch Vehicle*, March 2003, Delft University of Technology

- [23] Mark E. Campbell, Jin-woo Lee, Eelco Scholte, David Rathbun, *Simulation and Flight Test of Autonomous Aircraft Estimation, Planning, and Control Algorithms*, November–December 2007, Cornell University, Ithaca, New York 14853, DOI: 10.2514/1.29719
- [24] , National Instruments, *The PID Controller & Theory Explained*, <https://www.ni.com/en/shop/labview/pid-theory-explained.html>, Updated 2025-03-07. Accessed 2025-10-10
- [25] Robert H. Daugherty and Thomas J. Yager, *Texture Modification of the Shuttle Landing Facility Runway at Kennedy Space Center*, NASA Technical Memorandum 3626, Langley Research Center, Hampton, Virginia, 1997
- [26] Roberto Galvez, Stephen Gaylor, Charles Young, Nancy Patrick, Dexter Johnson, Jose Ruiz, *The Space Shuttle and Its Operations*, NASA History Series, Chapter 3, 2023, <https://www.nasa.gov/wp-content/uploads/2023/04/wings-ch3a-pgs53-73.pdf>
- [27] Renton B. Carsley, *Space Shuttle Wheels And Brakes*, Rockwell International Corporation, Downey, California, 1975