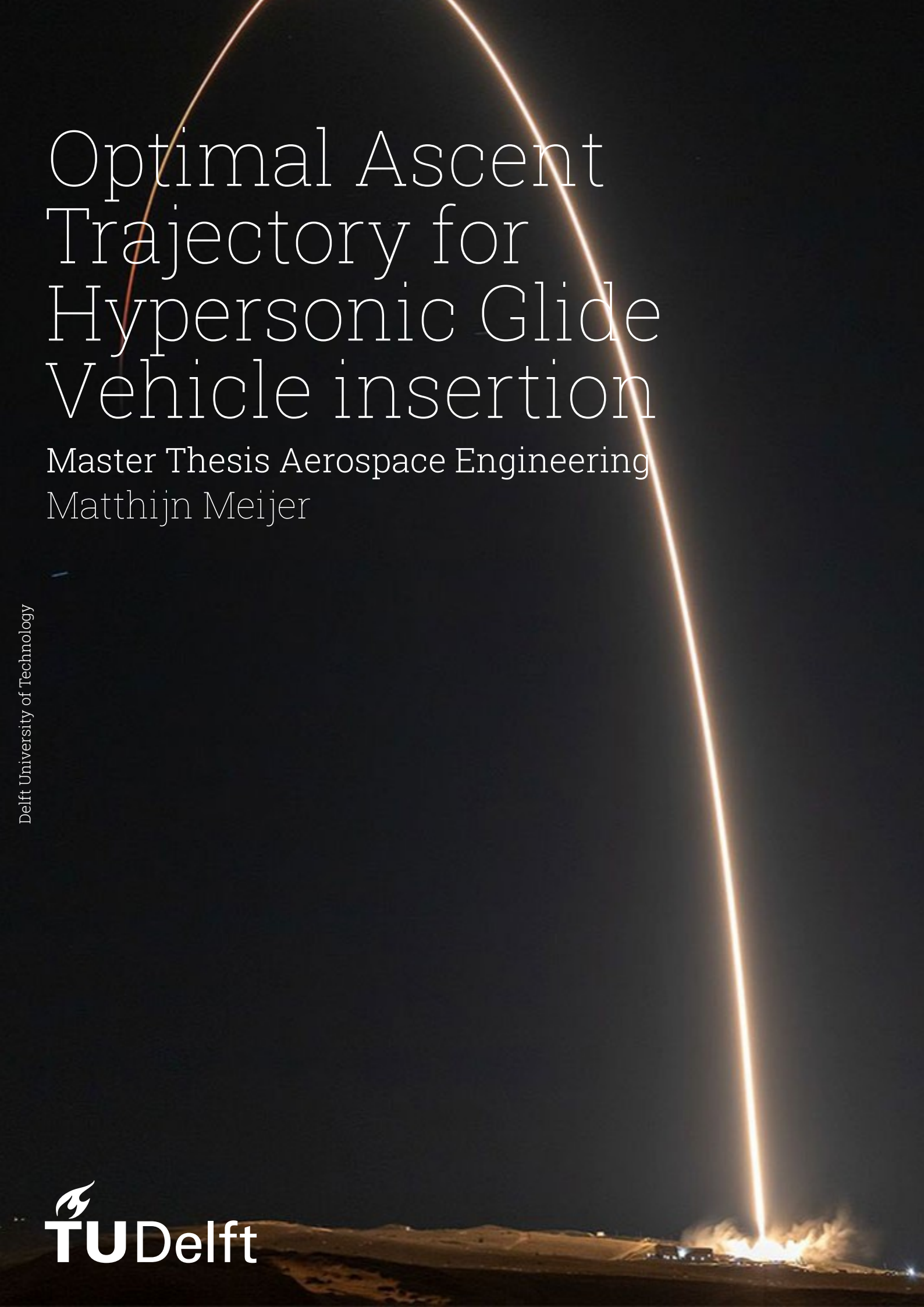
Optimal Ascent Trajectory for Hypersonic Glide Vehicle insertion

Master Thesis Aerospace Engineering

Matthijn Meijer

Delft University of Technology



TU Delft

Optimal Ascent Trajectory for Hypersonic Glide Vehicle insertion

by

Matthijn Meijer

at the Delft University of Technology.

Project duration:	February, 2025 – November, 2025	
Supervisor:	Botchu Jyoti	TU Delft Aerospace Engineering
Co-supervisor:	Mark Verveld	Royal Netherlands Aerospace Centre
Thesis committee:	Jian Gou	TU Delft Aerospace Engineering
	Marc Naeije	TU Delft Aerospace Engineering

Cover: A SpaceX Falcon 9 rocket carrying the NROL-126 mission and 20 Starlink satellite launched from Vandenberg Space Force Station in California on Saturday, Nov. 30, 2024 (Image credit: SpaceX)

Preface

After seven unforgettable years at Delft University of Technology, my time as a student comes to an end with this thesis. Throughout these years, I have had the opportunity to grow both personally and professionally, and this final work represents the culmination of everything I have learned along the way.

I would first like to express my sincere gratitude to Mark Verveld, my supervisor at NLR, for his invaluable guidance and insight throughout this project. His deep understanding of the subject and his ability to challenge ideas in a constructive way have greatly shaped my thinking and the direction of this research.

I would also like to thank Botchu Jyoti, my supervisor at TU Delft, for her support, encouragement, and critical feedback. Her perspective and attention to detail helped me refine my ideas and the writing process of this report.

Working on this thesis has been very rewarding. I found the topic of optimal ascent guidance for a hypersonic glide vehicle both fascinating and highly relevant. It represents a very current and important challenge within aerospace research. The problem's complexity and its connection to real-world applications made this work particularly engaging to explore.

Looking back, I am proud of the experiences, challenges, and growth that the years in Delft have brought me. This thesis marks the end of an incredible chapter and the beginning of a new one.

Matthijn Meijer
Amsterdam, November 2025

Abstract

This thesis presents the development of an open-loop optimization framework for computing the optimal ascent trajectory of a multi-stage booster, with specific application to the insertion of a Hypersonic Glide Vehicle (HGV) into its glide phase. The study addresses the challenge of guiding a launch vehicle through atmospheric and exo-atmospheric regimes while satisfying physical and terminal constraints. The HGV insertion case provides a relevant example, given the growing interest in boost-glide systems for both defense and research applications.

The optimization problem is formulated using the Pontryagin Maximum Principle (PMP), enabling an indirect method that solves the coupled state–costate dynamics with strict boundary conditions. The ascent is modeled as a 3-DoF point-mass in Cartesian coordinates, taking into account aerodynamic forces, Earth's rotation, and path-and-control constraints. This study presents a suitable constraint set and problem definition for the optimization problem that improves numerical convergence. A multiple-shooting approach is used to ensure convergence for the nonlinear dynamics, while a homotopy continuation strategy gradually incorporates aerodynamic and path constraints to improve numerical robustness. Furthermore, the developed trajectory optimization model allows for the implementation of arbitrary boosters that include staging and throttle management.

Results demonstrate smooth, dynamically consistent ascent trajectories that meet target insertion conditions required for an efficient HGV glide. In addition, this report introduces various evaluation techniques for the computed trajectories, together with verification methods that ensure their physical accuracy. Overall, the work delivers a comprehensive and extensible framework for optimal ascent guidance under realistic aerodynamic and operational constraints, with direct applicability to modern boost-glide mission design.

Contents

List of Figures

List of Tables

Nomenclature

List of Abbreviations

Abbreviation	Definition
BVP	Boundary Value Problem
DCM	Direction Cosine Matrix
DoF	Degrees of Freedom
ECEF	Earth-Centered Earth-Fixed
ECI	Earth-Centered Inertial
EOM	Equations of Motion
FDM	Finite Difference Method
GA	Genetic Algorithm
GEP	Generalized Equatorial Plane
GGI	Generalized geocentric inertial
GNC	Guidance, Navigation, and Control
HBG	Hypersonic Boost-glide vehicle
HGV	Hypersonic Glide Vehicle
KKT	Karush-Kuhn-Tucker
LGR	Legendre-Gauss-Radau
NED	North-East-Down
NLP	Nonlinear programming
NLR	Nederlands Lucht- en Ruimtevaartlaboratorium
OCP	Optimal Control Problem
PMP	Pontryagin's Maximum Principle
SQP	Sequential Quadratic Programming
STT	Skid-to-turn
TPBVP	Two-point Boundary Value Problem
VAFB	Vandenberg Space Force Base

Latin Symbols

Symbol	Meaning	Unit
A	Drag force	N
A_e	Nozzle exit area	m^2
a	Semi major axis	m
C_A	Axial force coefficient	-
C_N	Normal force coefficient	-
$C(u,t)$	Control constraint	-
e	Eccentricity	-
E	Eccentric anomaly	rad
$F_{g,f}$	Gravitational force vector	N
g	Gravitational acceleration	m/s^2
H	Hamiltonian	-
H_0	Hamiltonian independent on thrust	-
H_{scale}	Local scale height	m
h	Altitude	m

Symbol	Meaning	Unit
h_a	Specific angular momentum	m^2/s
I_b	Body axis orientation	-
I_b^*	Optimal body axis orientation	-
I_y	Unit vector body y-axis	-
I_N	Unit vector body normal axis	-
i	Inclination	rad
J	Objective function	-
m	Mass	kg
m_c	Dry mass	kg
m_p	Propellant mass	kg
M	Mach number	-
M_a	Mean anomaly	rad
n	Mean motion angular rate	rad/s
n_g	Load factor	-
N	Lift force	N
p	semi-latus rectum	m
$p_{ambient}$	Ambient pressure	Pa
$\boldsymbol{p}$	Costate vector	-
$\boldsymbol{p}_V$	Velocity costate vector	-
$\boldsymbol{p}_r$	Position costate vector	-
q	Dynamic pressure	Pa
q_c	Heatflux	W/m^2
Q_α	Maximum product $q \cdot \alpha$	Pa ·rad
R	Gas constant	$J{\cdot}kg^{-1}{\cdot}K^{-1}$
R_E	Earth Radius	m
R_N	Nose Radius	m
r	Distance	m
$S(x,u,t)$	Path constraint	-
S_{ref}	Frontal surface area	m^2
T	Thrust vector	N
T_K	Absolute temperature	K
t	time	s
V	Velocity vector	m/s
V_C	Circular velocity	m/s
V_r	Earth relative velocity vector	m/s
Y	Shooting nodes array	-

Greek Symbols

Symbol	Meaning	Unit
α	Angle of attack	rad
β	Sideslip angle	rad
ϵ	Specific orbital energy	m^2/s^2
η	Throttle variable	-
γ	Flightpath angle	rad
γ_{heat}	Specific heat ratio	-
λ	Homotopy parameter	-
μ	Gravitational parameter	m^3/s^2
ν	Constant multiplier vector	-
ω	Argument of periapsis	rad
ω	Rotational rate vector	rad/s
ω_E	Earth rotational speed	rad/s

Symbol	Meaning	Unit
ω_s	Schuler frequency	1/s
Ω	Right ascension of the ascending node	rad
$\phi(x(t_f), t_f, x(t_0), t_0)$	Terminal cost	-
Φ	Angle between V_r and $\boldsymbol{p}_V$	rad
$\Phi(x(t_0), t_0)$	Terminal condition	-
$\Psi(x(t_0), t_0)$	Initial condition	-
ρ	Atmospheric density	kg/m^3
ρ_0	Atmospheric density at sea level	kg/m^3
τ	Dimensionless time	-
θ	True anomaly	rad

Subscripts

Symbol	Meaning
0	Initial value
f	Final value
N	Number of shooting points
min	Minimum value
max	Maximum value
vac	Vacuum phase component
burn	Burn phase component
coast	Coast phase component
af	Transformation from aerodynamic frame to ECEF frame

1

Introduction

Near-space boosters play a crucial role in delivering payloads to specific position-velocity combinations, supporting a wide range of applications in both civilian and defense sectors. These boosters operate in the upper atmosphere and lower space regions, enabling efficient high-speed transport and interception missions. One prominent example is the boosted Hypersonic Glide Vehicle (HGV), which is initially launched by a rocket booster to reach a certain altitude and speed. After booster burnout, the HGV glides and skips through the atmosphere at hypersonic speeds of more than Mach 5. An example of a near-space booster that inserts a hypersonic glide vehicle is the boost phase of DARPA's HTV-2, seen below.

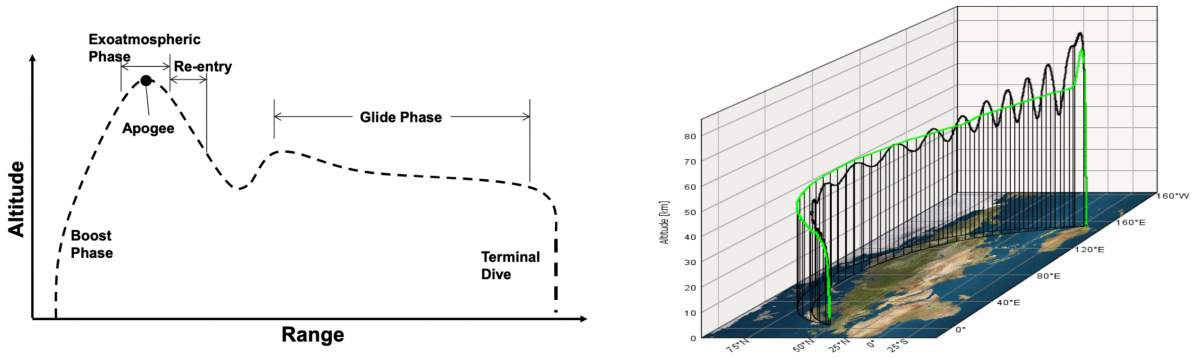
Figure 1.1: HTV-2 insertion schematic [1].

Near-space boosters have a diverse range of applications. In the civilian sector, hypersonic vehicles can serve as a new form of intercontinental passenger and cargo transport. This type of cargo vehicle can enable fast and precise delivery, boosting transportation efficiency. Meanwhile, a hypersonic passenger vehicle can significantly reduce travel time [2]. In defense and space exploration, near-space boosters offer capabilities such as intercepting fast-moving targets, including asteroids from outer space. Their ability to place payloads at precise locations with specific velocity conditions makes them valuable for a wide range of missions. Throughout this paper, the boosted HGV will serve as the primary example to analyze the performance, trajectory optimization, and terminal conditions of these boosters.

To ensure the successful operation of most autonomous flight vehicles, a well-designed guidance, navigation, and control (GNC) sub-system is essential. Given the model uncertainties and complex physical effects encountered at high Mach numbers, the methods used must be capable of ensuring mission success while compensating for any inaccuracies in the model assumptions [3].

For every application and type of vehicle, the ascent trajectory will look different. However, in general, the full trajectory of launch vehicles consists of similar phases. These are illustrated in figure 1.2a for

the specific case of an HGV. This study focuses on the trajectory from launch through booster ascent to the point at which the hypersonic-glide vehicle is inserted into its glide back toward Earth.



(a) HGV total flight scenario.[4]

(b) Three dimensional boosted HGV Path [5].

Figure 1.2: Boosted HGV trajectory example.

During this ascent trajectory, it is important to consider the necessary terminal conditions of the trajectory to ensure the success of the rest of the mission. For instance, in the case of the HGV, the ascent phase is followed by the glide phase, as can be seen in figure 1.2a. Therefore, the booster's ascent trajectory must satisfy specific terminal conditions to deliver its payload properly. These conditions, in turn, directly influence the dynamics of the earlier phases, making them essential for accurately modeling the entire trajectory. This approach will be further elaborated later.

1.1. Background

Assignment Description NLR

This thesis research is conducted at the Royal Netherlands Aerospace Centre (NLR), whose goal is to develop the optimal open-loop ascent trajectory of a booster, with possible closed-loop applications in mind, using an indirect optimization method. So, a primary requirement is that the solution is found analytically. This is crucial because, an analytical solution facilitates easy future integration into other simulation environments, avoiding the computational burden of direct numerical optimization and providing greater insight for sensitivity studies. Furthermore, the model should be a 3-DoF model, preferably with a rotating Earth. Finally, the rocket model must be arbitrary to enable optimization for various launch vehicles.

Goal of Research

The goal of this research is to find the optimal ascent trajectory of the boost phase of a hypersonic glide vehicle using an indirect optimization method. As mentioned before, the research will use an arbitrary booster model to allow for the implementation and optimization of different booster types. The research will also verify that the computed optimal ascent trajectory produces accurate results by verifying the physical accuracy and comparing it to other direct optimization models. Additionally, the research will evaluate the obtained results with a sensitivity analysis.

Document outline

The report is structured into 10 chapters. Chapter 2 presents the ascent optimization problem of this study. Also, the studied literature and the state-of-the-art in the field of boost phase trajectory optimization will be given. Further, it highlights the specific case study for the HGV. This chapter will present the novelty in the field of ascent optimization for an HGV, which will form the basis of the developed research questions. Following this novelty, Chapter 3 defines the research objectives, requirements, and questions. Chapter 4 describes the stakeholder analysis of NLR. This is to ensure that this study satisfies their requirements and expectations. Chapters 5, 6, and 7 discuss the research methodology, including modeling approaches and optimization methods used. The methodology used here is also based on a methodology trade-off that has been made during the literature study phase of this research. This trade-off can be found in the Appendix. Chapter 8 presents the results of the made optimization

model by comparing different cases with each other. Furthermore, chapter 9 presents a sensitivity analysis and a verification of all the obtained results. Lastly, chapter 10 gives the final conclusion and recommendations for future research.

2

The Ascent Optimization Problem

The optimization of the ascent guidance of near-space boosters is a very important part of mission design. Near-space boosters are key to transporting payloads to exact position-velocity targets, serving various purposes in both the civilian and defense sectors. The trajectory of the boost phase of near-space boosters is significantly different in comparison to the ascent trajectory of other launch vehicles. First of all, these boosted ascent phases follow a sub-orbital trajectory. In contrast, when a launch vehicle inserts a satellite into orbit, a part of the booster must reach and maintain orbital velocity around Earth. These two trajectories differ in key parameters like final altitude, velocity, and flightpath angle.

The goal of this chapter is to present the state-of-the-art in the field of trajectory optimization of near-space boosters, upon which the novelty of this work and the corresponding research gap are identified. In the next chapter, the research requirements and questions will be elaborated based on this novelty.

2.1. State-of-the-art

This chapter will explore how the boost phase for near-space boosters is modeled in existing literature (like boosted HGVs and Ground-Based Midcourse Defense vehicles). This includes the modeling of the rocket and its environment. Furthermore, the boost phase of an HGV is examined as a specific case study because it aligns well with the application and is currently a widely researched topic in the field. Chapter 2.2, discusses the different flight conditions the HGV goes through, because of its influence on the final conditions of the boosted ascent.

Additionally, the chapter will examine the methods used to optimize these trajectories, highlighting different approaches found in research. Finally, these findings will be analyzed and reflected upon at the end of the chapter, providing a foundation for the methodology used in this study.

2.1.1. Modeling of Launch Vehicle

Before a trajectory can be optimized, a detailed model of the launch vehicle's dynamics must first be established. This involves defining the equations of motion (EOM) that describe the system's behavior, which depend on the chosen reference frame and the required level of detail for accurately capturing the motion. Additionally, the booster itself must be characterized, specifying the type of booster used, its thrust capabilities, and the extent of thrust vectoring, which directly impacts the rocket's controllability. Finally, external forces acting on the vehicle must be considered, particularly aerodynamic forces that induce drag and lift on the vehicle. This plays a crucial role in shaping the vehicle's flight path.

Equations of Motion

There are various methods and reference frames used to describe the dynamics of a launch vehicle. This section presents the two most commonly used approaches in literature for modeling the motion of near-space booster trajectories. The dynamics of the hypersonic glider and its launch vehicle are often modeled as a point mass while still accounting for angular orientation, such as the angle of attack.

The first dynamical model that is prominent in literature is described by Yu Li (et. al) [6], Kai An (et. al)

[7], and Si-Yuan (et. al) [8]. Here, the launch vehicle is modeled to fly in a vertical plane over a spherical, non-rotating Earth. The motion is described using a spherical coordinate system. The following figure 2.1 gives a schematic of this dynamical model.

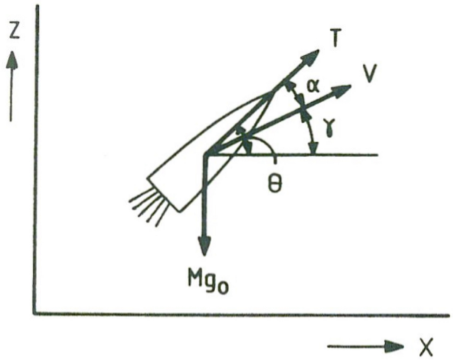


Figure 2.1: Schematic of vertical motion of launch vehicle.[9]

Here, T is the thrust, α is the angle of attack, and γ is the flightpath angle. Further, m is the mass, g is the acceleration due to gravity, and V represents the velocity vector.

Earth's gravity can also be chosen to be modeled as a sphere. Then only one variable is necessary to describe the gravitational acceleration, which is its position vector [8]. Li Yu (et. al.) presents more dynamical models like this, applied to boosted HGV trajectories [10] [11]. In these descriptions, the point mass can move in three-dimensional spaces with the equations of motion presented in a Cartesian framework.

However, another option is to also express these equations in spherical coordinates [12]. In this case, the three-dimensional point mass also has a heading angle (ψ), latitude (ϕ), and longitude (λ) that describe the state of the system [12]. Here, it is possible to take into account the influence of Earth's J_2 perturbations. Then, there are two different terms that describe the Earth's gravitational acceleration considering its oblateness (g_r and g_ϕ). These can be described using the following relations [12]. In these equations, R_e is Earth's radius.

$$g_r = \frac{\mu_E}{r^2}\left[1 - \frac{3}{2}J_2\left(\frac{R_E}{r}\right)^2(3\sin^2\phi - 1)\right] \tag{2.1}$$

$$g_\phi = -\frac{3\mu_E}{r^2}J_2\left(\frac{R_E}{r}\right)^2\sin\phi\cos\phi \tag{2.2}$$

Furthermore, the Earth's rotation is also implemented in these equations of motion. It is, of course, possible to use a model where the rotation of the earth is taken into account, but not its oblateness. This can be done to simplify the dynamics model of the booster.

Control of the launch vehicle

The controllability of a launch vehicle is determined by factors such as thrust and angle of attack, which appear in the equations of motion. In the literature, various scholars define their booster models, specifying the thrust capabilities and control conditions during the booster's ascent. These definitions directly influence the launch conditions and the extent to which the rocket can be maneuvered. The papers mentioned in this section describe models used in the literature to characterize near-space boosters for vehicles like boosted HGVs and ground-based midcourse boosters.

Yu Li et al. model the launch vehicle as a Minuteman III, a three-stage solid rocket booster [6]. Each stage ignites sequentially, cutting off and separating to shed inert mass. The thrust, mass flow rate,

and specific impulse for each motor are assumed to be constant. Li applies the same Minuteman III booster in other research focused on HGV ascent trajectory optimization [10] [11].

Si-Yuan Chen et al. also consider a solid-propelled booster with two stages [8]. To simplify the structure, this booster lacks a thrust termination system, meaning the fuel must be entirely consumed before burnout.

Kai An et al. define the control variable for a boosted HGV trajectory as the pitch angle, with no thrust variation [7]. This limitation aligns with the characteristics of a solid rocket booster, where thrust cannot be actively throttled.

J.R. Gottlieb et al. take a different approach by optimizing both the ascent trajectory and the booster of the HGV [13]. Their model also employs a solid rocket booster, reinforcing the prevalence of such systems in trajectory optimization research.

Both the work presented by S. Ann and Y. Jung control the flight path angle of a ground-based midcourse booster by only altering the thrust direction and not the throttle [14] [15]. Thus, indicating that the research does not cover the ability of thrust variation.

Furthermore, the paper by W. Chen et al. and N. Li et. al. optimize the trajectory of a booster designed to intercept hypersonic targets [16] [17]. These booster models use a solid propellant thruster. Therefore, the only control variables are the angle of attack and engine activation. Also, the paper by J. Lukacs et. al., which uses a ground-based midcourse vehicle, uses a booster with a solid propellant [18]. A ground-based midcourse vehicle is a rocket launched from the ground that is designed to intercept a target during the midcourse phase of its trajectory.

However, in the paper by G. Wu et al., throttle is one of the control variables [19]. In the paper, the ascent trajectory of an HGV is optimized to minimize fuel usage. Therefore, this does allow for the implementation of liquid propellant thrusters. However, the vehicle starts midcourse, and the used optimization approach makes it not able to be implemented for the use case of the research of this thesis. This will be discussed later in this chapter.

The literature demonstrates that extensive research has been conducted on the use of solid rocket boosters for the boost phase of an HGV. As a result, the primary control variable in these studies is the angle of attack of the launch vehicle and the duration of the coasting phase. Since solid rocket boosters do not allow for thrust modulation, the thrust level cannot be used to steer the vehicle in a desired direction. Instead, the controllability in the reviewed papers comes primarily from thrust vectoring, which influences the angle of attack and, consequently, the overall state and trajectory of the booster.

Aerodynamic forces

To model the ascent of a rocket accurately, it is essential to account for the aerodynamic forces acting on the launch vehicle. In literature, different levels of detail can be found to approximate these forces.

At the most basic level, aerodynamic lift and drag are expressed through constant coefficients that are combined with dynamic pressure and a reference area [7]. A higher level of fidelity can be achieved by incorporating altitude-dependent air density models, ranging from simple exponential approximations to more detailed standard atmosphere models such as the 1976 U.S. Standard Atmosphere [20, 21].

In more advanced treatments, the aerodynamic coefficients themselves are not considered constant. Some works assume that lift and drag coefficients depend only on the angle of attack [22], while others extend this by making them a function of both angle of attack and Mach number [6]. This allows for more realistic modeling of the aerodynamic behavior of launch vehicles during ascent.

The paper by J.R. Gottlieb uses Missile DATCOM to make Aerodynamic calculations [13]. Missile DATCOM is an aerodynamic prediction tool designed to estimate the aerodynamic characteristics of flight vehicles over a wide range of speeds and atmospheric conditions. The need for such a tool was recognized in the late 1970s by the Tri-Service Committee on Missiles and Projectile Aerodynamics, leading to the development of a structured method for aerodynamic analysis. The program was developed to provide engineers with reliable estimates of forces, moments, and stability derivatives based

on vehicle geometry and flight conditions [23]. Gottlieb used the tool to generate tables of aerodynamic coefficients, which are used for the trajectory modeling [13].

2.2. Case study: Insertion of the HGV

As mentioned earlier, in particular the boost phase of an HGV is examined. For the case of an HGV, the goal of its ascent is to reach the right altitude and speed for an optimal glide. In contrast, for orbital insertions, the focus is on achieving the necessary velocity and height for a stable orbit. A hypersonic boost-glide vehicle re-enters along a trajectory with a high lift-to-drag ratio, meaning its maximum downrange is not solely determined by the shutdown parameters. Instead, optimizing both the boost and glide phases together is essential for maximizing the missile's range [6]. Further, the re-entry phase of an HGV is different compared to more conventional re-entry trajectories. An HGV has smaller velocities compared to re-entry vehicles that have been in orbit, like the space shuttle. The latter vehicle has the goal of getting rid of lots of energy by banking. However, an HGV has the goal of maintaining and conserving its energy with the goal of increasing its range, for instance.

For a Hypersonic Glide Vehicle (HGV) to successfully complete its mission, it must enter the atmosphere under precise conditions. This requires an optimal insertion point where the HGV transitions from the boost phase to its skip-glide descent. The terminal conditions of the boost phase, defined by its height, velocity, and flight path angle, directly determine whether the vehicle can initiate its intended maneuver. Once separated from the booster, the HGV initially follows a ballistic re-entry, as it is released outside the atmosphere. However, once the atmosphere gets denser, thermal or mechanical failure should be avoided by staying within the defined re-entry corridor. Deviating from this path can lead to excessive heating or structural stress, making the vehicle uncontrollable. While re-entry constraints are less critical in the exoatmospheric phase, a precise insertion is crucial. If the vehicle enters with an incorrect velocity or flight path angle, it may fail to execute its maneuver upon atmospheric entry, significantly impacting its performance and survivability.

2.2.1. Heat flux and mechanical load constraint

The HGV has a maximum heat load that it is able to withstand during re-entry. This maximum heat load therefore determines one of the constraints of the re-entry corridor of the vehicle's descent. The Chapman cold-wall stagnation heat flux equation can be used to make an estimation of the heat flux on the HGV during re-entry. This equation is given by [24]:

$$q_c = \frac{c_1}{\sqrt{R_N}} \left(\frac{\rho}{\rho_0}\right)^{1/2} \left(\frac{V}{V_c}\right)^{c_2} = c^* \sqrt{\rho V^{c_2}} \tag{2.3}$$

Here, q_c is an estimation of the heat flux experienced by a spacecraft during re-entry. It considers factors such as the nose radius (R_N), atmospheric density (ρ), sea-level density (ρ_0), spacecraft velocity (V), and circular velocity at re-entry (V_c). Further, c_1 is the heat flux constant and c_2 the heat flux exponent. The cold-wall method provides a more conservative estimation of the heat flux compared to a hot-wall model, making it suitable for initial design considerations and ensuring a safe re-entry [24]. When the maximum allowable heat flux is known, this equation can be rewritten in order to obtain the corresponding air density. This equation can be seen below [20]:

$$\rho_{q,c} = \rho_0 \left(R_N^n \frac{q_{c,max}}{c^*} \left(\frac{V_c}{V}\right)^m \right)^{\frac{1}{1-n}} \tag{2.4}$$

In this equation, n will have different values based on the type of boundary layer of the airflow. Furthermore, the cosponsoring altitude can be determined following the computed air density. Considering a range of velocities, we can generate a height-velocity (h-V) constraint line of the re-entry corridor [20].

Furthermore, the re-entry vehicle also has a maximum mechanical load that it is able to withstand. The equations presented below allow for calculating the forces acting on a vehicle due to drag and lift [20].

$$n_g = \frac{\sqrt{A^2 + N^2}}{m g_0} \tag{2.5}$$

Here, the load factor (n_g) is calculated based on the drag force (A), lift force (N), vehicle mass (m), and gravitational acceleration at sea level (g_0). This load factor represents the total aerodynamic force

experienced by the vehicle, normalized by its weight. The equation can be rewritten to the following equation.

$$n_{g,max} = \frac{1}{2}\rho_g V^2 S_{ref} \frac{\sqrt{C_A^2 + C_N^2}}{mg_0} \tag{2.6}$$

This maximum load factor is a critical constraint for safe re-entry, as exceeding it can lead to structural damage.

By rearranging this equation when the maximum allowable load factor $n_{g,max}$ is known, it is possible to solve for the corresponding air density (ρ_g):

$$\rho_g = 2n_{g,max} \frac{mg_0}{V^2 S_{ref} \sqrt{C_N^2 + C_A^2}} \tag{2.7}$$

This equation allows us to determine the air density that the vehicle can safely encounter for a given velocity and maximum load factor. Furthermore, since air density is related to altitude, this calculation effectively translates the maximum load factor constraint into an altitude constraint. Just like for the stagnation heat flux, a h-V line can be determined for the re-entry corridor for the maximum allowable mechanical load [20].

The following figure 2.2, shows an overview of the loads dependent on the altitude of the re-entry vehicle. It can be seen that during re-entry, usually there is first a peak of the thermal load and then then mechanical load [20]. Next to these constraints, potentially also a maximum temperature limit can be considered for the HGV.

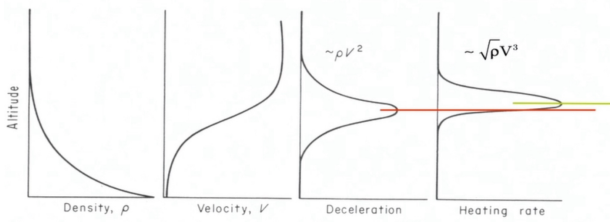


Figure 2.2: Re-entry loads.[20]

The presented equations above will not need to be modeled for the HGV during the ascent trajectory of this study. However, it shows all the factors the boost phase needs to consider in order to facilitate a proper insertion of the HGV.

2.2.2. Soft constraint

If a re-entry vehicle makes a gliding entry, it will glide slowly down. Therefore, the vehicle will have a small flight path angle. In this case, the vehicle will fly in an equilibrium glide condition. So, the Hypersonic Glide Vehicle is in a state were it descends at a nearly constant flight path angle, maintaining a balance between lift and weight. The flight path angle directly affects this condition, as changing this parameter will effect the amount of lift on the vehicle [20]. When the HGV enters the atmosphere at high speeds it decelerates due to aerodynamic drag. However, as explained, at the same time, the vehicle also generates lift to pitch up and leave the atmosphere [25]. So, the equilibrium glide condition is the theoretical ceiling under which no skipping occurs and therefore defines what is called the soft constraint. However, in some re-entry scenarios, exceeding this constraint may be intentional to initiate a skipping maneuver, allowing the HGV to momentarily leave the dense atmosphere before continuing its glide.

2.2.3. Re-entrty corridor

All the different constraints mentioned form the re-entry corridor. The HGV needs to stay within this corridor during its descent flight. The proper insertion of the HGV plays a crucial role in this process. The boost phase of the HGV needs to have the right terminal conditions in order to deliver the HGV

into its re-entry corridor. Figure 2.3, shows the different path constraints that the re-entry vehicle shall not cross. This creates the re-entry corridor that the vehicle needs to stay within.

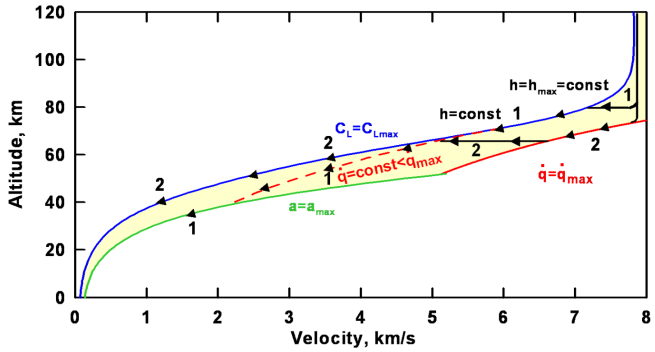


Figure 2.3: Altitude-Velocity re-entry corridor. [25]

In literature, this re-entry corridor is also frequently presented in an alternative way. An energy-drag or velocity-drag diagram shows the same corridor but presented by different parameters.

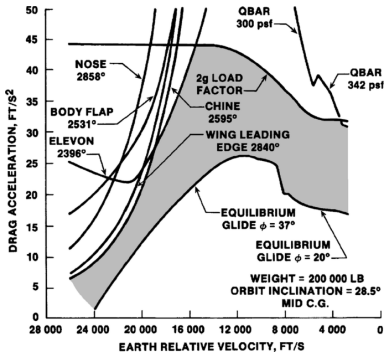


Figure 2.4: Drag-velocity curve. [26]

2.2.4. Final conditions of the HGV boost phase

The paper by J. Bradt et. al. [27] determines the descent trajectory of a hypersonic glider. In the work it can be seen that the HGV first makes a ballistic flight before it initiates its glide trajectory. After the ballistic flight, the vehicle will begin its skip or glide trajectory. The optimal insertion point for this phase varies depending on the type of Hypersonic Glide Vehicle, as different applications and vehicle designs require distinct descent trajectories and re-entry corridors. However, inserting the HGV into its ballistic re-entry trajectory under the right conditions, starting from an exoatmospheric environment, is a crucial step in the success of the entire mission. Therefore, dependent on the mission, the boost phase of the HGV will have a certain final condition for its velocity, altitude and flight path angle, so the HGV can optimally start on its descent.

2.3. Guidance methods for near-space booster trajectories

The guidance of a rocket refers to the system and algorithms that determine its trajectory to ensure it reaches its intended target or orbit. Ascent guidance is very important for a launch vehicle, since it provides the essential commands to steer the vehicle to its desired target [28]. Within the guidance of launch vehicle's a distinction can be made between closed-loop guidance and open-loop guidance.

Open-loop guidance:
In open-loop guidance, the system follows a predetermined trajectory without real-time corrections, as there is no feedback mechanism. The control action is independent of the system's output, meaning that once the launch occurs, no adjustments are made to the trajectory.

While guidance methods for the exoatmospheric phase are efficient and reliable, relying on open-loop guidance during the atmospheric flight phase has contributed to costly launch delays. This happens when the actual wind profile deviates significantly from the average profile used to determine the attitude control program [29].

Closed-loop guidance:
In closed-loop guidance, the control input depends on the system's output, meaning it continuously monitors the current state, such as position and velocity, and adjusts control inputs accordingly. This approach is essential for maintaining an optimal trajectory in the presence of dispersions [12]. By feeding the process output back to the input, the control action dynamically responds to real-time flight conditions to ensure the final objective is reached. Using feedback, the system continuously refines the trajectory, improving accuracy and robustness.

When an open-loop guidance optimization problem relies entirely on numerical optimization algorithms, the problem can become computationally intensive to solve. Integrating these algorithms into a closed-loop guidance system with a fast enough update rate to ensure robustness is quite challenging [30].

As a compromise, a common methodology is to operate in 'open-loop' mode during the high dynamic pressure portion of flight and then, based on a pre-determined time or event, switch to a closed-loop vacuum guidance scheme which operates on the premise that aerodynamic forces can be neglected.

2.3.1. Optimization methods for the ascent trajectory used in literature

The trajectory of a launch vehicle can be optimized in order to minimize or maximize a certain objective. There are many different methods to solve such an optimal control problem and compute the optimal trajectory. During this optimization, boundary conditions also need to be taken into account. This includes the initial conditions, terminal conditions, and path constraints that the trajectory needs to satisfy. These initial and terminal conditions can be defined based on the desired trajectory of the launch vehicle. Furthermore, how the path constraints can be determined will be elaborated further in chapter 5.

There are two different approaches to solving an optimization problem. This is achieved by either using a direct or indirect method to compute an optimum. Indirect and direct optimization methods are two fundamental approaches to solving optimal control problems, each with distinct characteristics and advantages. Indirect methods solve the necessary optimality conditions derived from Pontryagin's Maximum Principle (PMP), which states that an optimal control must maximize the Hamiltonian function. This function is a combination of the system dynamics and costates that represents the total cost of the system. In contrast, direct methods discretize the control problem and solve it numerically as a nonlinear optimization problem.

As stated previously, indirect methods are based on principles from the calculus of variations and Pontryagin's Maximum Principle. The Hamiltonian function helps to derive the necessary conditions for optimality. Once these conditions are established, the optimal control solution can be obtained either analytically or numerically, often using Newton's method. Newton's method is an iterative technique that finds a solution by repeatedly using the function's gradient to move closer to the point where it equals zero. To solve the resulting boundary value problems, also techniques like the multiple shooting method are commonly applied. Indirect methods typically yield highly accurate solutions but require solving complex differential equations and boundary conditions [31]. More details about the previously described indirect method, as well as the numerical techniques used to obtain a solution from it, are further elaborated in chapter 6. In contrast, direct methods transform the optimal control problem into a nonlinear programming problem (NLP), which involves finding the best solution to a set of nonlinear equations and constraints. Instead of deriving necessary conditions explicitly, the control and state variables are discretized and optimized numerically. A key advantage of this approach is that there is no need to manually derive the necessary conditions for optimality. However, direct methods often provide less precise solutions than indirect methods. Additionally, these methods are computationally less

effective and may struggle with multiple local minima, sometimes converging to a suboptimal solution rather than the true global solution [31].

In the papers discussed in Section 2.1.1, various methods are used to compute the optimal trajectory for a boost phase. All the studies reviewed apply direct optimization methods to determine the ascent trajectory.

Yu Li et al. optimize the ascent trajectory of the boost phase to maximize the downrange distance [6]. They use a direct shooting method in combination with Sequential Quadratic Programming (SQP) to solve the optimization problem. In two additional papers, Yu Li et al. again use an SQP solver to optimize the boosted flight phase of an HGV, this time focusing on maximizing either crossrange or downrange [10] [11].

J.R. Gottlieb et al. perform a coupled optimization of both design parameters of the booster carrying the HGV and the trajectory of the booster itself. They apply a direct optimization approach using a Genetic Algorithm [13]. However, they note that this method has a long convergence time and, because it relies on a stochastic process of population evaluation, it does not guarantee repeatability. This means that each optimization may produce slightly different solutions, making sensitivity analysis more difficult.

Kai An et al. aim to minimize the terminal time of the boost phase, meaning the time required to reach a specified terminal height. They use a numerical Legendre-Gauss-Radau (LGR) orthogonal collocation method, which, like the other approaches, falls under direct optimization [7].

Si-Yuan Chen et al. seek to maximize the velocity at the solid rocket shut-off point by implementing a burn-coast-burn strategy for the boost phase. Their research employs a pseudospectral method to compute the reference trajectory. This is another direct optimization approach, showing that indirect methods are not commonly used for this phase of flight [8]. Further, also W. Chen et. al. use a pseudospectral method to optimize their trajectory. Only in this case for a booster capable of intercepting hypersonic targets [16]. The same goes for N. Li et. al., who utilize the pseudospectral methods, but compare it to other direct optimization methods [17].

Furthermore, S. Ann and J. Jung minimize the flight time to a certain terminal state. This minimization is done with an indirect optimization method. However, this trajectory starts midcourse and does not take aerodynamic forces into account [14] [15]. Therefore, this approach lacks the applicability to a trajectory in an endoatmospheric environment starting from Earth's surface.

The paper by G. Wu et al. uses the particle swarm method for the optimization of the ascent trajectory [19]. This is a direct optimization method that is very effective at finding the global optimum of a problem. However, the high computational load would make it ineffective for implementation in a robust closed-loop guidance system. Furthermore, this method makes sensitivity analysis of the obtained result very challenging.

Overall, the reviewed studies consistently apply direct optimization methods for computing the optimal ascent trajectory of an HGV. Furthermore, when an analytical method was used, it was applied inappropriately and for an exoatmospheric environment, making it inadequate for the context of this thesis research.

2.4. Conclusion

In the reviewed literature, there is a notable absence of indirect optimization approaches and throttle control strategies for the open-loop guidance of the boost phase of a hypersonic glide vehicle (HGV).

When throttle control is considered, the optimization method employed is typically a direct method like the particle swarm optimization. While effective in some cases, this approach is generally computationally intensive and does not leverage the analytical structure offered by indirect methods.

Indirect optimization techniques have only been used twice in the reviewed studies. However, both instances apply to midcourse guidance starting in an exoatmospheric environment. As such, these methods are not suited for modeling ground-based ascent trajectories that include atmospheric effects.

Furthermore, in all studies focusing on boosted HGV ascent, the vehicles rely on solid rocket boosters, and the trajectory optimization is carried out using direct methods. This reveals a specific gap in the

literature: the absence of indirect optimization for throttleable, ground-launched HGVs with atmospheric flight. Addressing this gap represents a key novelty of this work and will be further discussed in the next section.

2.5. Takeaways

Underneath, a table can be found that summarizes all the essential information found in the most important papers of this chapter. The content of this table will be further elaborated in this section. Additionally, a more detailed version of this table, including the advantages and disadvantages of the approaches used by the scholars in their papers, can be found in Appendix A.

Table 2.1: Summary of research on ascent trajectory optimization and booster models.

Author	Research Objective	Used Optimization Method	Booster Model	Source
Yu Li et al	Optimizes the ascent trajectory of the boost phase to maximize the downrange distance.	Direct shooting method in combination with Sequential Quadratic Programming (SQP).	The Launch Vehicle is a Minuteman III, a three-stage solid rocket booster.	[6]
Yu Li et al	Optimizes the ascent trajectory of the boost phase to maximize the crossrange distance.	A SQP solver to optimize the boosted flight phase.	The Launch Vehicle is a Minuteman III, a three-stage solid rocket booster.	[10]
Yu Li et al	Optimizes the ascent trajectory of the boost phase to maximize the downrange distance.	A SQP solver to optimize the boosted flight phase.	The Launch Vehicle is a Minuteman III, a three-stage solid rocket booster.	[11]
Si-Yuan Chen et al.	The goal is to maximize the velocity at the solid rocket shut-off point.	The research uses a pseudospectral method to compute the reference trajectory. This is another direct optimization approach.	Solid-propelled booster with two stages.	[8]
Kai An et al.	The goal is to minimize the terminal time of the boost phase, meaning the time required to reach a specified terminal height.	A numerical Legendre-Gauss-Radau (LGR) orthogonal collocation method is used.	The booster is defined to have no thrust variation.	[7]
J.R. Gottlieb et al.	optimize both the ascent trajectory and the booster of the HGV.	A direct optimization approach using a Genetic Algorithm is applied.	The model employs a solid rocket booster.	[13]
S. Ann	Starting midcourse in an exoatmospheric environment, and minimizes a certain time to reach a terminal state.	An indirect optimization method is used to compute the reference guidance.	The booster is defined to have no thrust variation.	[14]
Y. Jung	Starting midcourse in an exoatmospheric environment and minimizing a certain time to reach a terminal state.	An indirect optimization method is used to compute the reference guidance.	The booster is defined to have no thrust variation.	[15]
G. Wu	Optimizes the fuel usage of the ascent trajectory of an HGV	A particle swarm method is used for the trajectory optimization.	The booster is defined to have thrust variation and a controllable angle of attack.	[19]
W. Chen	Optimizes the trajectory to intercept hypersonic targets.	Optimization is based on the direct pseudospectral method.	The rocket is a three-stage solid booster.	[16]
N. Li	Optimizes the trajectory to intercept hypersonic targets, using a combination of performance indicators like maximum terminal velocity, fuel consumption, and lateral range.	Optimization compares the pseudospectral method and the hp-ARPM method.	Uses angle of attack, bank angle, and engine activation as control variables.	[17]

J. Lukacs	Optimizes a combination of minimal time until reaching a terminal condition, maximizing terminal velocity, and path length of the trajectory of a ground-based Midcourse vehicle.	Optimization uses a direct-method-based approach that converts the optimization problem into a nonlinear programming problem.	The used rocket model has a fixed thrust profile.	[18]

Limitations of the shown research

All the papers focus on optimizing the boost-phase trajectory of near-space boosters using a solid rocket booster. Consequently, the angle of attack and coasting time are the only available control variables in the ascent trajectory optimization. This inherently limits the optimization process to solid rocket boosters. In contrast, if liquid propellant boosters were used, thrust could be controlled, introducing another control variable. However, none of the reviewed studies address the optimization of a ground-based boost trajectory with a liquid-propellant booster.

Furthermore, all studies apply a direct numerical optimization method to determine the optimal boost-phase trajectory. As highlighted by J.R. Gottlieb et al. [13], these numerical approaches often result in long computation times. Additionally, because they rely on numerical solvers, the repeatability and precision of the final solution cannot always be guaranteed. This lack of consistency makes it difficult to perform sensitivity analyses, as the solutions may vary across different optimization runs.

Another major limitation of numerical optimization methods is their computational effectiveness, particularly when applied in a closed-loop guidance system. Although direct methods, such as the particle swarm optimization techniques used in the paper by G. Wu, are effective at finding a global optimum, they come with a high computational cost [19]. As noted by Fl. Richard et al. [30], numerical methods are computationally intensive, making them challenging to implement in robust closed-loop guidance strategies, especially in an endoatmospheric environment where aerodynamic forces significantly impact the booster. In contrast, if the open-loop ascent trajectory were analytically solved, it could be easily integrated into a closed-loop system, ensuring robustness. Therefore, the models and numerical methods used in the reviewed literature are less suitable for real-time closed-loop applications in endoatmospheric scenarios.

Novelty

A key advantage of analytical solutions is their ability to facilitate rapid sensitivity analysis. Since an analytical approach provides mathematical expressions that describe how outputs depend on input parameters. If there are explicit equations for the ascent trajectory that directly show the dependence on aspects like thrust or aerodynamic forces, it is possible to instantly calculate sensitivities using mathematical differentiation. In contrast, numerical methods sometimes lack explicit equations and instead require running multiple full-scale simulations with slightly altered parameters (e.g., varying thrust) to determine sensitivity. Alternatively, when sensitivities are possible to be computed directly, as in the pseudospectral method, they may vary at each iteration and collocation point. Therefore, these approaches are computationally inefficient, requiring significant processing time and power.

The numerical methods used in previous studies are computationally inefficient and suffer from difficulties in ensuring convergence, particularly in endoatmospheric ascent. By employing an indirect method, where computations can be made analytically to solve the optimization problem, computation time can be reduced from minutes to seconds. This efficiency not only makes the optimization process faster but also enables real-time implementation in a closed-loop guidance system or the ability to be implemented in a more detailed simulation environment. Developing a method that supports closed-loop applications and the implementation in a wider range of simulation environments also aligns with NLR's research objectives, making this approach particularly relevant. Further discussion of this objective is provided in Chapter 4.

Furthermore, by introducing thrust as a control variable, different booster types can be incorporated into the optimization process. When thrust is a controllable parameter, the optimization is no longer limited to solid rocket boosters, but it can also be applied to liquid-propellant boosters. This expands the range of booster models that can be optimized, making the approach more versatile for different

mission requirements and vehicle configurations. This flexibility allows for the evaluation of more launch vehicle designs, helping in both the possibilities of design decisions and guidance strategies.

Key considerations

In conducting this research, several important challenges must be taken into account. First, the nonlinear nature of aerodynamic forces introduces significant complexity in the indirect optimization process, as it limits the ability to obtain a fully analytical solution. This requires the incorporation of numerical techniques or hybrid optimization approaches. Additionally, the success of indirect methods strongly depends on the availability of accurate initial guesses. Poor initial guesses may lead to numerical divergence or convergence to suboptimal solutions, making the development of a reliable initialization strategy essential. Finally, formulating the correct set of optimality conditions and analytically expressing the full mathematical problem requires careful attention, as a poor formulation highly influences the overall convergence and feasibility of the optimization problem. These considerations are discussed in more detail throughout in Appendix B. Here, different approaches for applying the indirect optimization method to a rocket ascent trajectory are analyzed and traded off to determine which best aligns with the research objectives and requirements, that are discussed in chapters 3 and 4.

3

Research Objective

Based on the findings from the literature review in Chapter 2, this chapter further defines the research objective. The following sections outline the study's goals and the specific research requirements that will be addressed.

3.1. Research Objective

The objective of this research is to compute the optimal ascent trajectory of the boost phase of a hypersonic glide vehicle (HGV) using an indirect optimization method.

3.2. Research Requirements

First of all, the research approach must allow the implementation and optimization of the trajectory of different booster types. This gives the following requirement that the research needs to comply with:

- *"The booster dynamics shall be modeled using an arbitrary booster model, allowing for the implementation and optimization of different booster types."*

Second, to ensure the validity and applicability of the computed optimal ascent trajectory, it is necessary to verify that the research produces accurate results. This can be achieved by comparing the optimized trajectory to other optimization methods and verifying the physical accuracy. This gives the second requirement:

- *"The optimal trajectory shall be verified, to ensure the research produces accurate results."*

Additionally, understanding the sensitivity of the indirect optimization method is crucial. Evaluating how initial inputs influence the outputs is important for assessing their practical use in ascent trajectory planning. This gives the last requirement of the research:

- *"The sensitivity of the optimization method shall be evaluated."*

3.3. Research Questions

3.3.1. Main Research Question

How can an indirect optimization method be applied to compute the optimal ascent trajectory of an HGV boost phase while accommodating different booster models with variable thrust?

3.3.2. Subquestions

1. *How can the dynamic model of the booster be verified to ensure the model is physically accurate?*
2. *What methods can be used for the evaluation of the optimization problem?*
3. *How effective is the implemented indirect optimization method for ascent trajectory planning in terms of sensitivity and its ability to find an optimal solution?*

By addressing these research questions, the research aims to fulfill the research objective and requirements. Ultimately, this helps contribute to the understanding of optimal ascent trajectory planning for the boost phase of near-space boost vehicles.

4

Stakeholder Analysis

Stakeholder analysis is an essential step in understanding the interests, expectations, and influence of key parties involved in a project. In this study, the primary stakeholders include the customer, the end user, and the funder. For this research, all these roles are fulfilled by the Royal Netherlands Aerospace Centre (NLR). As both the funder and the primary beneficiary of the study, NLR plays a central role in defining the objectives and requirements of the research.

4.1. NLR's Expectations and Requirements

NLR has specific expectations regarding the outcome of this study. The primary goal is to develop an open-loop model that computes the optimal ascent guidance of a booster, given certain terminal conditions. This optimization has to be done using an indirect optimization method. Furthermore, the application of the boost phase of an HGV works as a good novel case study for this problem. The model must be designed to facilitate further advancements in closed-loop applications, aligning with NLR's broader research objectives.

One key reason for developing this model is that an analytically solved open-loop guidance solution is easier to integrate into a closed-loop system compared to a directly solved model. As mentioned in chapter 2, the computational cost of direct optimization methods makes it challenging to implement into a robust closed-loop model or other more detailed simulation environments, while also incorporating aerodynamic forces. Since the simulation has to be applicable for a wide variety of applications, the Ascent Trajectory has to be able go through the entire atmosphere.

Additionally, NLR requires the model to be applicable to arbitrary rocket configurations. This requires the inclusion of variable thrust capabilities within the booster model, allowing for the analysis of a greater variety of existing and future booster concepts. The potential threats and capabilities of HGV technologies can therefore be studied and better understood from a defense perspective.

By meeting these requirements, the research will contribute to NLR's current efforts in studying the capabilities of hypersonic glide vehicle technologies. This broader research goal therefore matches with the research gap found in chapter 2.

4.2. Priorities of features

In chapter 2, the different aspects that are necessary or could possibly be used in the optimization of the trajectory were explained. These features correspond to the requirements that NLR desires for the research. Features refer to the available options that can be implemented within the dynamic model, vehicle control scheme, and the targeted performance criteria of the optimization. In this, the realization of some features is more important than others. Therefore, based on NLR's preference, these features are prioritized below in table 4.1.

Table 4.1: Priorities of possible features.

No.	Feature
1	Aerodynamic forces
2	Constraints
3	Rotating Earth
4	Angle of Attack control
5	Staging
5	Closed-loop
6	Sensitivity analysis
6	Variable thrust
7	Computational efficiency

The table reflects that it is first important to create a booster model that is physically accurate. Therefore, a sufficient implementation of aerodynamic forces, constraints, and a realistic dynamic model is most important. Following this, the table presents basic vehicle control options, such as angle of attack and staging. Here too, the importance of closed-loop applicability emerges, showing the need for the model to be suitable for broader simulation use in the future. The final three features in the table, 'sensitivity analysis', 'variable thrust', and 'computational efficiency', refer to the ability to perform efficient sensitivity analyses, implement various booster models, and enable broader simulation capabilities.

It's important to note that, according to the previously described novelty and the requirements mentioned above, the HGV is explicitly the highlighted case. For an HGV, it's generally desirable that the boost phase maximizes its range so that the total trajectory of the HGV is maximized. Therefore, the optimization goal will primarily be maximizing the booster's range. However, sometimes it's also desirable for the HGV not to maximize its range, but to fly to specific terminal conditions. Therefore, it's important that the developed optimization model has the ability to both maximize the range and fly to a defined terminal condition. The optimization model will be based on the previously mentioned research objective and requirements, and the above-mentioned stakeholder expectations of NLR.

5

Flight Dynamics

To answer the main research question: ”How can an indirect optimization method be applied to compute the optimal ascent trajectory of an HGV boost phase while accommodating different booster models with variable thrust”, a working dynamical and optimization model must be established. This will be implemented in MATLAB.

The Methodology trade-off table, shown in Figure B.1 in the appendix serves as a useful reference for selecting an initial strategy and methodology to guide this research. The first step is to model the rocket dynamics. Following the insights from the trade-off table, the papers by Yu Li et al. [10] [11], provide a strong foundation for this part of the work. This chapter will discuss the specific details of how the rocket dynamics are modeled.

The discussed literature in chapter 2 presents different methods on how to model the dynamics of the launch vehicle. Based on the trade-off and research requirements, it is decided that a Cartesian coordinate system is best applicable to this research, as will also be further elaborated in chapter 6.

5.1. Reference Frames

In order to express the motion of the launch vehicle, it is important to have a good understanding of the different reference frames that can be used for this. The five different frames that are used in this research to describe the orientation and motion of the launch vehicle are the Body frame, Aerodynamic frame, Earth-Centered Earth-Fixed Frame, the Vehicle-carried frame, and the perifocal frame.

5.1.1. Body frame

The body-fixed reference frame has its origin located at the center of mass of the launch vehicle [32]. Since the booster in this study is modeled as a point mass, the body frame is positioned at the location of the point mass. A schematic of the body-fixed frame is shown below. The frame is indicated by the red axes and labeled with the letter “p”.

As shown in figure 5.1, the x-axis of the body frame is aligned with the forward direction of the vehicle. In this research, the x-axis also defines the thrust direction. This makes the body-fixed reference frame a convenient choice for describing the thrust direction of the launch vehicle.

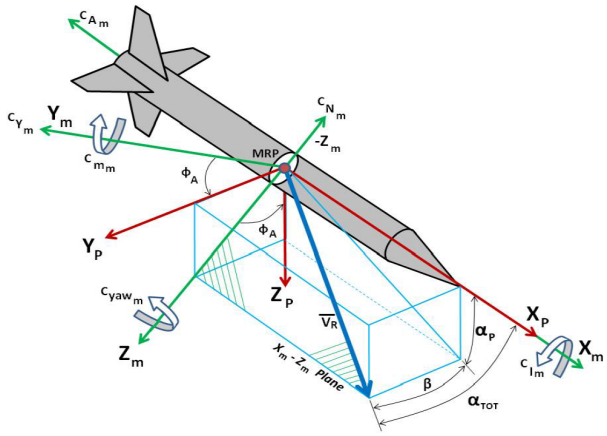


Figure 5.1: Body-fixed frame [32].

5.1.2. Aerodynamic frame

Then there is the aerodynamic reference frame. The aerodynamic reference frame is aligned with the velocity vector of the vehicle. As shown in Figure 5.1, the aerodynamic frame points in the same direction as the blue velocity vector. Specifically, the x-axis of the aerodynamic frame is aligned with the velocity vector. This means that, when expressing the velocity in this frame, it only has an x-component. The angle of attack (α) and sideslip angle (β) define the orientation of the aerodynamic frame relative to the body-fixed frame. These angles are also used as control variables for the rocket's guidance, making them essential for describing the thrust orientation in the aerodynamic frame.

5.1.3. Earth-Centered Earth-Fixed Frame

The Earth-Centered, Earth-Fixed (ECEF) frame represents positions in Cartesian coordinates (X, Y, Z), relative to the center of the Earth. The positive X-axis points towards the point where the equator intersects the prime meridian at 0° latitude and 0° longitude. A schematic of this frame is shown below [33].

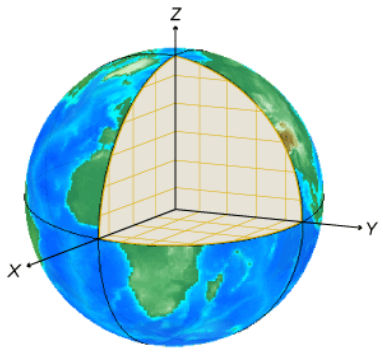


Figure 5.2: ECEF reference frame [33].

The ECEF frame is useful for describing the position and orientation of the spacecraft relative to the Earth's surface. It accounts for Earth's rotation. If Earth's rotation is not considered, this frame is referred to as the Earth-Centered Inertial (ECI) frame.

5.1.4. Vehicle-Carried Frame

The vehicle-carried reference frame, like the body-fixed frame, is located at the point mass representing the vehicle's position. However, it has a different orientation. This frame is also known as the North-East-Down (NED) frame. It uses the Cartesian coordinates (x;North, y;East, z;Down) to represent position relative to a local origin. Typically, the NED frame's local origin is positioned above the Earth's surface [33].

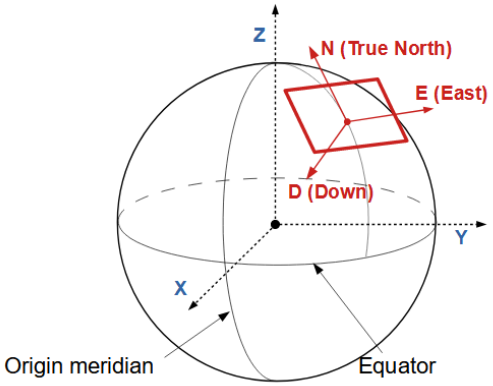


Figure 5.3: NED reference frame [34].

The NED frame is useful for describing the gravity vector acting on the vehicle. This is because gravity points toward the center of the Earth, which is also the center of the ECEF frame. As a result, the gravity vector aligns with the "down" direction of the NED frame.

5.1.5. Perifocal coordinate system

Lastly, there is the perifocal frame. The perifocal frame is a coordinate system used for describing an orbit. It is a Cartesian frame fixed in space and centered at the focus of the orbit, with its xy-plane coinciding with the orbital plane.

An orbital frame is a coordinate system defined by the geometry of an orbit using Keplerian elements. The position of the spacecraft is given by the radius vector r, which points from the center of the Earth to the satellite. The inclination i describes the tilt of the orbital plane with respect to the equatorial plane, while the right ascension of the ascending node Ω defines the angle from a fixed inertial reference direction to the line where the orbital plane intersects the equator [35]. Together with other orbital elements such as the argument of periapsis ω and the true anomaly ν (also sometimes denoted with θ), these parameters uniquely describe the orientation and shape of the orbit, as seen in the next figure [36].

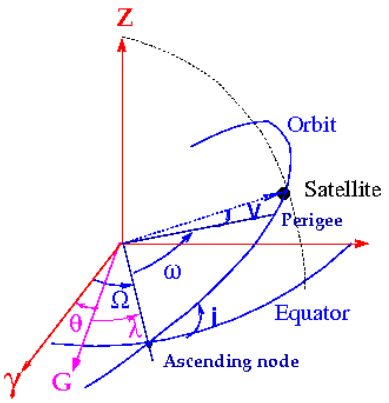


Figure 5.4: Orbital frame [36].

This framework is not only useful for satellites in closed orbits but also for sub-orbital trajectories. Sub-orbital trajectories intersect Earth's surface and therefore have a periapsis that is smaller than Earth's radius. However, the Keplerian elements still provide an efficient way to describe the flight path and its direction, making them very practical for analyzing and comparing different launch and re-entry paths.

The x-axis of the perifocal frame, denoted $\hat{p}$, points from the focus through periapsis, while the y-axis, $\hat{q}$, lies in the orbital plane at 90° true anomaly to $\hat{p}$. The z-axis, $\hat{w}$, is normal to the orbital plane in the direction of the orbital angular momentum vector $\hat{\mathbf{h}}$ [37]. Underneath, a schematic can be seen of this frame.

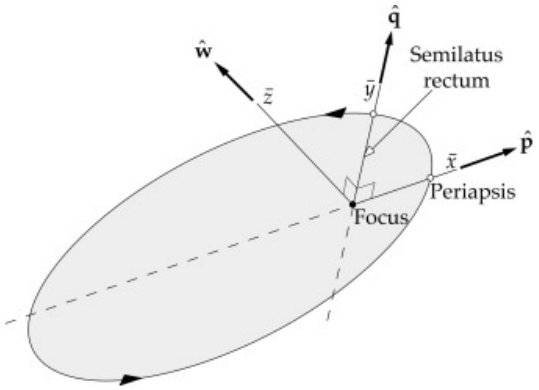


Figure 5.5: Perifocal frame [37].

Together, $\hat{p}$, $\hat{q}$, and $\hat{w}$ form the perifocal frame, which can be used for expressing orbital motion and transforming it to an inertial system such as ECI [37].

5.2. Modeling the Dynamics

To model the physical behavior of the booster, its motion is described using a 3 degrees of freedom (3 DoF) approach. This means the booster can translate freely along the three Cartesian axes (x, y, z), describing its position in space. In addition to its position, the orientation of the booster also plays an important role in defining its motion.

5.2.1. 3-DoF Equations of Motion

The ECI frame describes the dynamics of the vehicle in Cartesian coordinates. Yu Li et al. [10] demonstrated that this is an effective approach for modeling the dynamics of a booster. Another option is to model the dynamics using the ECEF frame. However, using the ECI frame to describe the equations of motion allows the derivation of the rocket's optimal control in an elegant manner, as will be shown in Chapter 6. Therefore, this research also adopts the Cartesian coordinates in the ECI frame to describe the motion of the vehicle. The next section explains how the different forces acting on the booster are modeled.

Gravity

One of the biggest forces acting on the launch vehicle is gravity. The gravitational force depends on the distance from the Earth and is a function of the gravitational parameter $\mu = 3.986 \times 10^{14}$ m^3/s^2 [38]. The direction of the gravitational force is straightforward to model: in the ECI frame, it points opposite to the unit vector of the vehicle's position, which is the same as the "down" direction in the vehicle-carried frame.

The gravitational force on the launch vehicle in the inertial frame is expressed as [39]:

$$\mathbf{F}_{g,f} = \frac{-\mu \cdot m}{|r|^3} \cdot r \tag{5.1}$$

Thrust

As explained earlier, the thrust vector points in the same direction as the body. Therefore, by knowing the orientation of the body in the ECI frame, the thrust direction can also be described in this frame. If the body's orientation is described by the unit vector I_b, the thrust force in the ECI frame is given by:

$$\mathbf{T} = T_{mag} \cdot I_b \tag{5.2}$$

Here T_{mag} represents the magnitude of the thrust force.

Aerodynamic relations

The aerodynamic forces acting on the vehicle depend on the dynamic pressure and air density. How the air density is modeled will be discussed later in this chapter. The magnitudes of the aerodynamic forces can be expressed as:

$$A = \frac{1}{2}\rho V_r^2 S_{\text{ref}} C_A(\text{Mach}, \alpha) \tag{5.3}$$

$$N = \frac{1}{2}\rho V_r^2 S_{\text{ref}} C_N(\text{Mach}, \alpha) \tag{5.4}$$

Here, V_r is the Earth's relative velocity. This can be calculated like [40]:

$$V_r = V - \omega_E \times r \tag{5.5}$$

In this equation, ω_E represents Earth's rotational speed, which is $7.2921159 \times 10^{-5} rad/s$ [41]. It is important to take Earth's rotation into account when considering aerodynamic forces. Earth's atmosphere rotates with the Earth due to gravity and friction with the surface. Therefore, this affects the magnitude of the rocket's aerodynamic forces.

The aerodynamic forces on the booster depend on both the angle of attack and the Mach number. The angle of attack dependence introduces a higher level of complexity than using constant lift and drag coefficients. However, it does more accurately explain the aerodynamic forces on the rocket.

The aerodynamic coefficients are defined as:

$$C_N(\alpha, M) = C_{N\alpha}(M) \cdot \alpha \tag{5.6}$$

$$C_A(\alpha, M) = C_{A_\alpha}(M) + C_N(\alpha, M) \cdot \sin\alpha \tag{5.7}$$

The Mach number dependence is also implemented and will be discussed in section 5.4. The values of $C_N(\alpha, M)$ and $C_{N_0}(M)$ will be obtained from a dataset, which will then be curve-fitted to derive an analytical relation for the aerodynamic coefficients [42]. This analytical form is important for formulating the indirect optimization method. As will be seen in Chapter 6, the co-state equations are highly dependent on the analytical expressions of the aerodynamic forces and their coefficients. Since the rocket in this study is assumed to be axisymmetric, a simple aerodynamic relation depending only on Mach number and angle of attack, as given above, is sufficient.

Launch Vehicle Mass

The mass of the launch vehicle has a significant impact on its motion. The total mass can be divided into several components: the structural mass, payload mass, and propellant mass. These components are combined into the total launch vehicle mass at $t = 0$, denoted as m_0. After all propellant has been consumed and the rocket enters its coasting phase, the mass becomes:

$$m(t_f) = m_0 - m_p(t_0)$$

where m_p is the propellant mass.

The time evolution of the mass also depends on the rocket's mass flow rate, which is described by equation 5.28. The mass of the vehicle over time is given by the following relation [9]:

$$\begin{cases} m(t) = m_0 - \int \dot{m}\, dt, & \text{while } m_p(t) > 0 \\ m(t) = m_0 - m_p(t_0), & \text{when } m_p(t) = 0 \end{cases} \tag{5.8}$$

Here, $m_p(t_0)$ represents the full propellant mass available at the start of the flight. This relation shows that the rocket's mass decreases as propellant is consumed, and remains constant once the propellant is depleted. Furthermore, between burn phases, the rocket jettisons the dead mass of the previous stage, discarding significant portions of structural mass during flight. So, the start of each burn phase will have a new initial mass.

Lastly, when the rocket engine throttles, the mass flow rate increases or decreases, directly affecting the rate of propellant depletion and thus the rocket's mass over time. This will be further discussed in Chapter 6.6.

5.2.2. Dynamical Model

The full equations of motion for the rocket are obtained by summing all the forces acting on the vehicle and dividing by its mass to compute the acceleration. Integrating this acceleration yields the velocity and, subsequently, the position of the booster. The equations of motion are described as follows [43]:

$$\dot{\mathbf{r}} = \mathbf{V} \tag{5.9}$$

$$\dot{\mathbf{V}} = \frac{1}{m(t)} \cdot (\mathbf{F}_{g,f} + \mathbf{A} + \mathbf{T} + \mathbf{N}) \tag{5.10}$$

$$\dot{m} = -\eta \frac{T}{g_0 I_{sp}} \tag{5.11}$$

5.2.3. Modeling the Body Orientation

In a 3-DoF model, the equations of motion describe the translation of the vehicle's center of gravity, without accounting for its attitude. The direction of the booster's velocity is characterized by the angle of attack, which is the angle between the velocity vector and the longitudinal axis of the rocket in the body frame, and the flight path angle, which is the angle between the velocity vector and the local horizontal in the inertial frame. This chapter presents two methods that have been investigated during the research in order to model the orientation of the rocket. Furthermore, it will be explained which of these methods are applied in the different parts of the research.

Skid-to-turn

First of all, there is the skid-to-turn method. Here, the goal is to steer the vehicle by controlling its attitude relative to the incoming flow, using aerodynamic incidence angles. With the skid-to-turn method, the rocket's dynamics do not rely on rolling its lifting surfaces to generate lateral forces that initiate a turn. In STT mode, the roll angle is typically held constant or left uncontrolled. This principle is important because lateral accelerations in the aerodynamic frame directly cause the velocity vector to change its direction. The rate at which this velocity vector rotates with respect to the inertial frame can be expressed as a rotational rate vector ω_{af}, defined by:

$$\omega_{af} = \frac{1}{V} \begin{bmatrix} 0 & -\dot{V}_z & \dot{V}_y \end{bmatrix}^T \tag{5.12}$$

Here V is the magnitude of the velocity vector in the aerodynamic frame, and $\dot{V}_y$ and $\dot{V}_z$ are the lateral acceleration components. This equation is derived from the derivative of the velocity vector. In other words, the change in direction of the velocity vector is directly proportional to the applied lateral acceleration.

In this approach, the quaternions are applied to describe the booster's attitude using four Euler parameters. A quaternion is a hypercomplex number that describes the orientation of an object in a reference frame by defining both a rotation axis and the rotation around this axis. Four parameters are needed: three for the axis definition and one for the rotation itself [20]. As shown below, the imaginary part contains the axis parameters and the real part corresponds to the rotation [44]:

$$\mathbf{q} = q_0 + q_1 \hat{i} + q_2 \hat{j} + q_3 \hat{k} \tag{5.13}$$

$$\mathbf{q} = [q_0, q_1, q_2, q_3] \tag{5.14}$$

This representation is closely related to the axis-angle formulation of 3D rotations. By Euler's rotation theorem, any rotation can be described by a unit vector and an angle, which directly map to the four quaternion components. Figure 5.6 illustrates this concept [45].

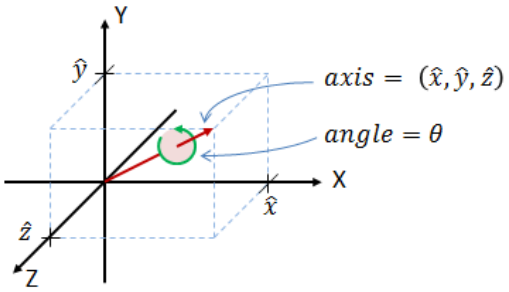


Figure 5.6: Quaternions schematic description [45]

Given axis–angle parameters $(\theta, \hat{x}, \hat{y}, \hat{z})$, the quaternion is defined as [45]:

$$\mathbf{q} = (q_0, q_1, q_2, q_3) \tag{5.15}$$

$$q_0 = \cos\left(\frac{\theta}{2}\right), \quad q_1 = \hat{x}\sin\left(\frac{\theta}{2}\right), \quad q_2 = \hat{y}\sin\left(\frac{\theta}{2}\right), \quad q_3 = \hat{z}\sin\left(\frac{\theta}{2}\right) \tag{5.16}$$

Here, q_0 depends only on the rotation angle, while (q_1, q_2, q_3) represent the axis scaled by a common factor. The norm of a rotation quaternion is always unity [45].

A key advantage of quaternions is that they avoid gimbal lock, which occurs when two gimbal axes align, reducing the system's degrees of freedom and preventing smooth rotation adjustments [46]. For this reason, quaternions are commonly used in aerospace dynamics. They are also computationally efficient, reducing the reliance on trigonometric functions.

In this case, the quaternions are used to represent the orientation of the velocity vector relative to the inertial frame during a skid-to-turn maneuver. This is achieved by combining the aerodynamic rotational rate with the Euler parameter kinematic differential equation [47]:

$$\dot{q}_{af} = \frac{1}{2} \cdot B(q_{af}) \cdot \omega_{af} \tag{5.17}$$

where

$$B = \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \tag{5.18}$$

$$\begin{aligned} q_{af} &= \int_{q_0} \dot{q}_{af}\, dt \\ &= [q_0, q_1, q_2, q_3] \end{aligned} \tag{5.19}$$

Integrating this relation yields the updated quaternion, which describes the orientation of the aerodynamic frame with respect to the ECEF frame (or the ECI frame if Earth's rotation is neglected). The corresponding direction cosine matrix (DCM) is expressed in terms of the Euler parameters as [47]:

$$DCM = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{5.20}$$

Thus, once the body acceleration, angle of attack, and sideslip are known, quaternion kinematics provide the orientation of both velocity and body frames with respect to an inertial frame.

Finally, combining quaternions for orientation with Cartesian coordinates for position yields a complete and efficient vehicle state description. This dynamical model follows the block scheme in Figure 5.7.

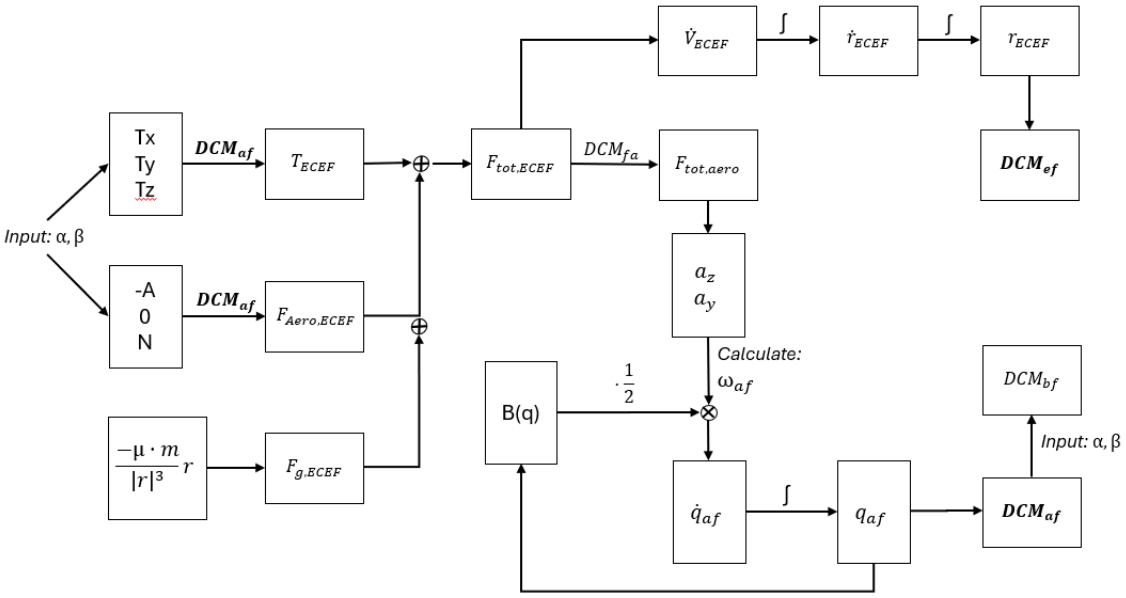


Figure 5.7: Overview of the modeled dynamical system

In this scheme, the vehicle's dynamics are driven by inputs such as angle of attack and sideslip. These inputs determine the transformation between aerodynamic, body, and vehicle-carried frames and the ECEF or ECI frame through the DCMs.

Zero-sideslip formulation

The next method that is used to define the body orientation is based on the "fly into the wind" maneuver. The vehicle body axis frame $\{1_b, 1_y, 1_z\}$ defines the rocket's orientation in flight. In this method, the frame is defined to enforce zero sideslip, meaning the vehicle always flies such that its symmetric plane lies in the plane formed by the body axis 1_b and the Earth-relative velocity vector V_r. This principle is called the *fly into the wind* maneuver, which ensures the rocket rolls about its longitudinal axis 1_b to cancel sideslip when crosswinds are present. Effectively, the rocket orients itself to align with the relative airflow, nullifying the sideslip angle $\beta = 0$ [40].

The unit vectors defining the frame are constructed as follows. The body x-axis is aligned with 1_b. The body y-axis is defined as

$$1_y = \frac{1_{V_r} \times 1_b}{\|1_{V_r} \times 1_b\|}, \tag{5.21}$$

where $1_{V_r} = V_r/\|V_r\|$ is the unit vector of the Earth relative velocity [40]. The body z-axis completes the right-handed system as

$$1_z = 1_b \times 1_y. \tag{5.22}$$

The body-normal unit vector 1_n is then given by

$$1_n = 1_b \times \frac{(1_b \times V_r)}{\|1_b \times V_r\|}, \quad (\alpha > 0) \tag{5.23}$$

with α the angle of attack. For $\alpha < 0$, to avoid a sudden 180° flip, 1_n is redefined as

$$1_n = 1_b \times \frac{(V_r \times 1_b)}{\|1_b \times V_r\|}. \tag{5.24}$$

A more robust definition of 1_y, valid for both $\alpha > 0$ and $\alpha < 0$, is

$$1_y = \frac{1_{V_r} \times 1_b}{\sin \alpha}, \tag{5.25}$$

which avoids discontinuities when α changes sign, since 1_{V_r} and 1_b may cross over.

The paper by P. Lu et. al. gives an expression for the aerodynamic vectors working on the booster [43]. These are based on the magnitude of the aerodynamic forces given in 5.3 and 5.4.

$$\mathbf{A} = \text{-}A \cdot \mathbf{1}_b, \quad \mathbf{N} = N \cdot \mathbf{1}_n \tag{5.26}$$

Here, **A** and **N** represent the aerodynamic axial and normal forces, respectively. The vector $\mathbf{1}_n$ describes the body's normal axis.

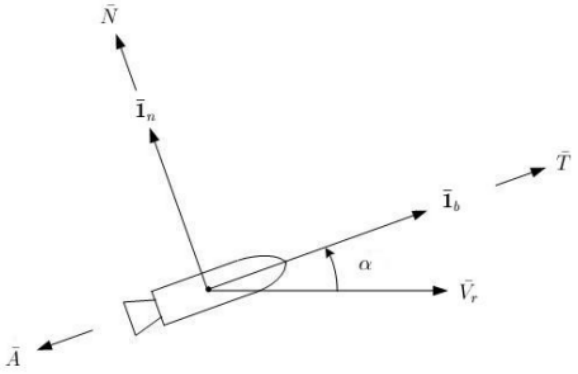


Figure 5.8: Booster body frame with relative velocity. [40]

Using the zero sideslip formulation in combination with defining the aerodynamic relations in this way is particularly helpful for the indirect optimization method. Once the thrust direction and the magnitude of the Earth's relative velocity are known, so is the complete body orientation I_b and the aerodynamic forces. It enables a clean analytical formulation of the optimality condition, using the primer vector theory, which will be further discussed in Chapter 6. This elegant way of obtaining the optimal body axis rotation simplifies the optimization problem. Therefore, this method is adopted in the final optimization model.

In summary, the fly into the wind principle ensures that the vehicle adjusts its roll such that the sideslip angle remains zero. The entire body orientation is then fully defined by 1_b, 1_y, and 1_z, guaranteeing a body frame aligned with the velocity vector, eliminating sideslip. The only downside is that the rocket may perform more excessive roll maneuvers compared to a scenario where the sideslip is not set to zero [40].

The STT approach could potentially also be applied within the indirect optimization framework. However, incorporating quaternion kinematics and rotation matrices makes it considerably more complex to derive the optimality conditions and corresponding control laws compared to the fly into the wind principle. Moreover, the indirect optimization of ascent trajectories using the fly into the wind approach is well established in the literature, whereas significantly fewer studies address the STT method. As a result, more robust boundary and optimality conditions are readily available for implementation in this study. As will be further explained in chapter 6.5, such well-defined conditions are crucial for ensuring convergence of the optimization model. Nevertheless, the skid-to-turn approach is not disregarded

fully in this study. For verification purposes, a Monte Carlo model is developed that generates multiple ascent trajectories based on randomized angle of attack and sideslip inputs. In this context, the STT model is applied because it provides a straightforward way to introduce random control variations. This application will be further discussed in chapter 9.

5.3. Control Variables

The optimization problem consists of five different control variables: the body axis rotation (two-dimensional), engine throttle, the coasting period, and the engine burn time.

The first control variable is the body axis rotation, which defines the orientation of the thrust vector with respect to the inertial frame. Through the equations of motion, this directly influences the direction of the acceleration of the rocket. When the ”fly into the wind” maneuver is used, the body axis rotation is defined by its roll angle and the angle of attack. The body axis rotation, therefore, also influences the lift and drag of the rocket. This is because of the dependency of the angle of attack on the aerodynamic coefficients.

The second control variable is the coasting period, which represents the duration between successive propulsion phases. During this period, the rocket follows a purely ballistic trajectory without thrust. Adjusting the length of the coast has huge effects on the shape and terminal conditions of the trajectory.

Variable thrust can be realized by the throttling of the engine. This is possible for boosters with liquid propellant. Varying the thrust will add another control variable and therefore enhance the control capabilities of the rocket. In the equations of motion, it can be seen that the thrust level has a direct effect on the acceleration of the launch vehicle.

When the thrust is a control variable, the engine throttle can be described with variable η. With this new variable, the magnitude of the dimensionless thrust can be calculated with [43]:

$$T = \eta \cdot \frac{[T_{vac} + \Delta T(r)]}{m(t) \cdot g_0} \tag{5.27}$$

Here, $\Delta T(r) \leq 0$ is the thrust loss inside the atmosphere as a function of altitude through the dependence on ambient pressure. The magnitude of the full vacuum thrust is described with T_{vac}. The thrust level is therefore bounded by $0 \leq T \leq T_{vac}$. Furthermore, $m(t)$ is the mass of the booster at a time t. Lastly, as seen before, g_0 is the gravitational acceleration on the surface of the Earth. In a similar way, the mass flow rate of the booster can be described with the following relation [43]:

$$\dot{m} = -\eta \frac{T_{vac}}{g_0 \cdot I_{sp}(p_{ambient})} \tag{5.28}$$

In this equation, $I_{sp}(p_{ambient})$ is the specific impulse of the engine of the booster. Generally, it is a function of ambient pressure.

Furthermore, when the engine is able to throttle, it is also able to cut off the engine early if this is deemed more efficient. This means the engines will stop producing thrust before all propellant is burned. This adds the last control variable, namely the burn time of the stage.

5.4. Environmental Models

In this study, the aerodynamic data are represented by analytical expressions rather than directly relying on discrete lookup tables. This choice is necessary because the indirect optimization method, based on the Pontryagin Maximum Principle, requires analytical expressions in the equations of motion to evaluate the optimality conditions. Also, the derived costates have terms based on these environmental models. These require analytical descriptions, as will be discussed in chapter 6. Further, when the dynamics are expressed as continuous functions, the optimizer can compute gradients more efficiently and reliably. This improves the convergence behavior of the solver, since the gradient information can be obtained directly from smooth functional forms instead of through the discontinuous steps associated with table look-ups. The studied literature on indirect optimization methods did not clarify the methods used to determine the analytical expressions for the different environmental and aerodynamic parameters. Therefore, this section will explain how these can be obtained.

The downside of this approach is that the aerodynamic data must be approximated, and thus the exact fidelity of the original dataset is not fully preserved. However, the effect of these approximations on the overall trajectory solution is minimal. For the optimization objective considered in this work, the level of accuracy provided by the analytical fits is sufficiently detailed and does not compromise the validity of the results greatly.

Equations 5.6 and 5.7 show the aerodynamic relations used in this study. As can be seen in these equations, C_{N_α} and C_{A_α} are dependent on the Mach number of the rocket. The Mach number is the ratio between the velocity and the speed of sound of an object [48]:

$$M = \frac{V}{a} \tag{5.29}$$

However, the speed of sound is dependent on the temperature of the medium. The speed of sound can be computed with the following equation [48]:

$$a = \sqrt{\gamma_{heat} \cdot R \cdot T_K} \tag{5.30}$$

Here, γ_{heat} is the ratio of specific heats, R is the gas constant, and T_K is the absolute temperature. Now, in order to compute the Mach number, the temperature needs to be determined. The temperature is dependent on the altitude of the rocket.

In this work, the temperature–altitude relation is derived from the U.S. Standard Atmosphere 1976 dataset [21]. Tabulated values of temperature as a function of altitude up to 100 km are taken from here. The tabulated data are first stored as altitude–temperature pairs. To enable the use of this relation in the indirect optimization method, where analytical expressions are required, the dataset is transformed into continuous functional forms.

Two approaches are compared. The first uses a cubic spline interpolation, which provides a smooth curve that passes exactly through the tabulated data points. This ensures high fidelity but results in a piecewise-defined function that is less convenient for analytical derivations.

The second approach is a polynomial fit of degree 11, applied to the entire altitude range (0–100 km). This produces a single analytical expression for temperature as a function of altitude. This gives a full analytical expression. The fitting accuracy is evaluated by comparing the polynomial results with the tabulated values, showing that the maximum deviation remains small. Although the polynomial fit introduces an approximation, the error is minimal compared to the scale of the problem, and the smooth analytical form significantly improves numerical convergence of the optimizer compared to using discrete table look-ups. Furthermore, it forms a great solution for the necessity to form analytical expressions to be used in the costate definitions. Underneath a figure can be seen, showing both the 11-degree polynomial compared to the cubic spline interpolation.

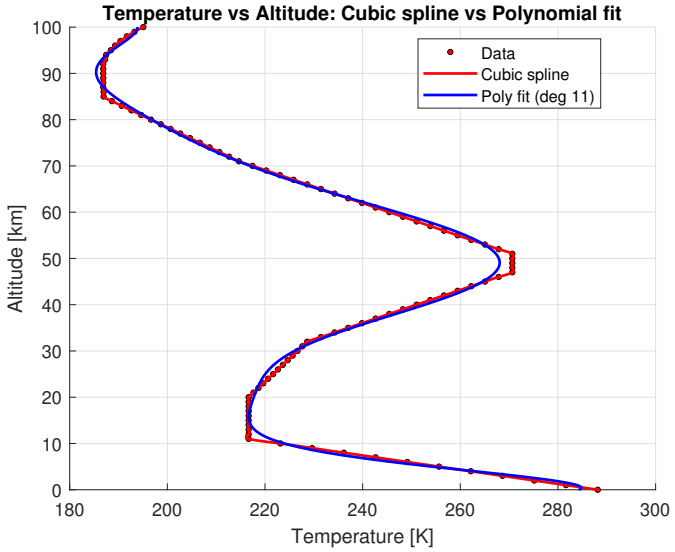


Figure 5.9: Analytical Pressure-Altitude formulation.

This figure clearly shows that the polynomial does not perfectly match the U.S. Atmosphere 1976 data, but that it is able to make a pretty close approximation. When the temperature is known following the altitude, the local atmospheric scale height can also be computed [20].

$$H_{scale} = \frac{R \cdot T_K}{g_0} \tag{5.31}$$

With this parameter and the altitude of the rocket, the air density in this study is determined with the following relation [20].

$$\frac{\rho}{\rho_0} = e^{-\frac{h}{H_{scale}}} \tag{5.32}$$

Furthermore, since the Mach number can now be determined, it can be used in the computation of the aerodynamic forces. For this study, the following table is used to explain the C_{N_α} and C_{A_α} coefficients of the rocket in relation to the Mach number.

Table 5.1: Aerodynamic coefficients as a function of Mach number.

Mach	0	0.65	0.69	0.74	0.75	0.85	0.89	0.90	0.95	1.00	1.04	1.14	1.19
C_{A_α}	0.650	0.650	0.650	0.672	0.678	0.734	0.756	0.761	0.789	0.817	0.839	0.895	0.923
C_{N_α} [1/deg]	0.240	0.240	0.253	0.267	0.270	0.291	0.296	0.296	0.299	0.300	0.300	0.299	0.297

1.20	1.25	1.40	1.50	1.60	1.80	2.00	2.20	2.50	3.00	4.00	5.00
0.929	0.919	0.892	0.873	0.855	0.818	0.781	0.744	0.689	0.597	0.597	0.597
0.297	0.294	0.283	0.273	0.263	0.243	0.225	0.210	0.192	0.174	0.150	0.135

In this work, the coefficients $C_{N\alpha}$ and C_{A_α} are modeled as a function of Mach number. The next step is to form an analytical expression for this tabular data. The aerodynamic data show a characteristic trend in which the coefficient remains nearly constant at low Mach numbers, exhibits a distinct peak in the transonic regime, and then asymptotically approaches a constant value at higher Mach numbers. A polynomial fit was initially attempted, but this method was unable to reproduce the correct physical

behavior. In particular, a polynomial cannot capture both the local transonic bump and the asymptotic flattening of the coefficient at higher Mach numbers, and instead introduces oscillations.

The coefficients $C_{N\alpha}$ and C_{A_α} are modeled in MATLAB with a function that combines a high-Mach asymptote, a smooth low-to-high Mach transition, and a transonic peak. A least-squares objective is then defined as the difference between the model and the data, and the parameters are estimated through nonlinear fitting using MATLAB's `fminsearch` algorithm. The resulting function can be evaluated at any Mach number and provides a smooth analytical expression for $C_{N\alpha}(M)$.

Compared to the polynomial approximation, the fitted parametric model achieves smaller errors across the full Mach range, guarantees smoothness of the function, and preserves the correct asymptotic behavior. The resulting analytical expressions for $C_{N\alpha}(M)$ and C_{A_α} are therefore well-suited for use within the indirect optimization method. Underneath, both the plots can be seen for the analytical expressions formed for the $C_{N\alpha}(M)$ and C_{A_α} coefficients. Here, it can be seen that the polynomial fit is not sufficient to use for the approximation.

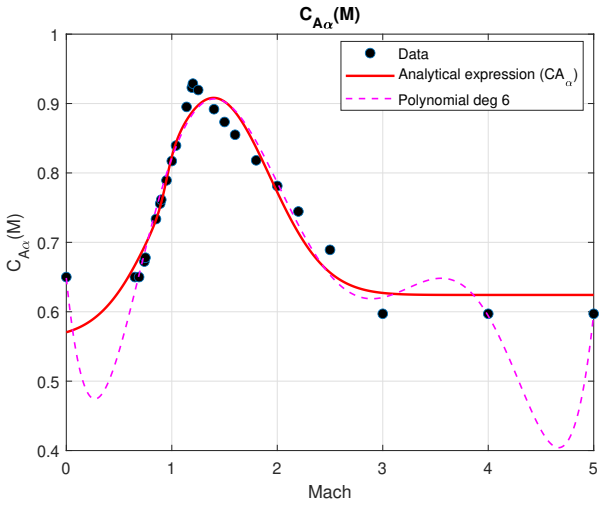


Figure 5.10: Analytical Mach-Drag formulation.

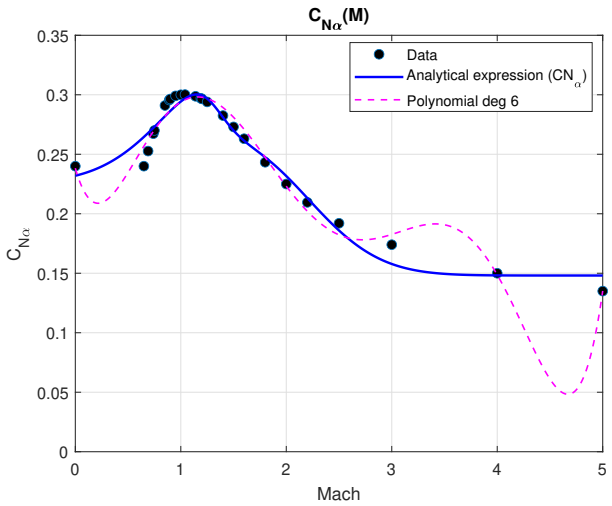


Figure 5.11: Analytical Mach-Lift formulation.

Following the fminsearch, the analytical expressions are extruded and implemented in the aerodynamics of the MATLAB model.

Lastly, the ambient pressure as a function of altitude is modeled using data from the U.S. Standard Atmosphere 1976 [21]. This is required because the thrust produced by a rocket engine depends not only on the chamber pressure and nozzle geometry but also on the surrounding ambient pressure. At lower altitudes, where ambient pressure is high, the effective thrust is reduced. Conversely, at higher altitudes, the ambient pressure decreases and the thrust approaches the ideal vacuum value. In the present model, the thrust correction is included through the term [49]:

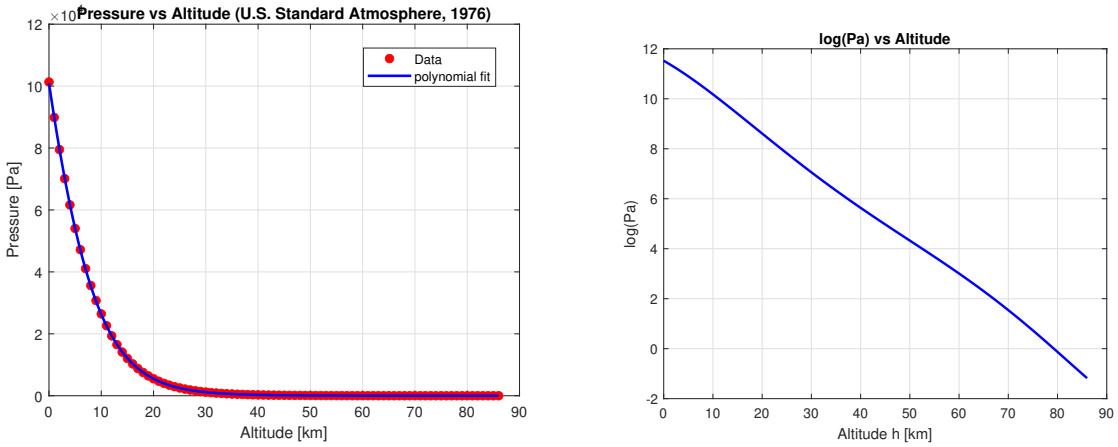
$$\Delta T = A_e \cdot p_{\text{ambient}},$$

Here, A_e is the nozzle exit area. The effective thrust then becomes

$$T = T_{\text{vac}} \cdot \eta - \Delta T,$$

with T_{vac} the vacuum thrust and η the throttle. By using the U.S. Standard Atmosphere 1976 for the ambient pressure-altitude relation, this correction can be applied across the full ascent trajectory.

A polynomial fit of the logarithm of pressure versus altitude is constructed, resulting in an analytical expression that can be directly evaluated within the equations of motion. The figure below shows the result of this polynomial fit. Here, the red dots present the data from the U.S. Standard Atmosphere 1976 [21] and the blue line the fitting function.



(a) Pressure-altitude profile.

(b) Logarithmic pressure-altitude relation.

Figure 5.12: Analytical pressure–altitude formulation.

The derived analytical expressions of the different functions explained in this section can be found in Appendix D.

5.5. Constraints

Throughout the trajectory, the rocket is bounded by certain path and control constraints. These define the maximum control actuation that the booster is able to deliver, and within what bounds the ascent trajectory needs to be.

Path constraints

Similarly to how the HGV has a maximum allowable mechanical load during its re-entry trajectory, as discussed in chapter 2.2, there are also similar path constraints for the ascent trajectory. First, there is the overload constraint or bending load constraint. This aims to assure flight safety, since it is important to restrict the normal overload [50]. If the load on the launch vehicle is too high, the structure of the body of the launch vehicle can mechanically fail. The normal and axial overloads can be calculated as

follows [51]:

$$\begin{cases} N_m = \frac{|T\sin(\alpha)+N|}{m\cdot g}, \\ N_x = \frac{T-A}{m\cdot g}, \end{cases} \tag{5.33}$$

Here, T is the thrust produced by the booster itself, N the lift force, and α the angle of attack of the booster. Further, A denotes the axial aerodynamics force (or the drag force). Both the normal overload (N_m) and the axial overload (N_x) should not exceed the maximum allowable overload.

Second, there is the maximum dynamic pressure that the launch vehicle is able to withstand. This constraint aims to ensure the stability of the vehicle structure [50]. Therefore, the dynamic pressure on the rocket has to remain below its maximum allowable dynamic pressure. This constraint can be defined with the following equation [7]:

$$q = \frac{1}{2}\rho V^2 \leq q_{max} \tag{5.34}$$

Furthermore, during the boost phase of a launch vehicle, generally the heat flux does not exceed its maximum limitation [7]. Therefore, this heat flux constraint is not considered as a path constraint for the ascent trajectory.

Control constraints

Lastly, the booster is subjected to certain control constraints. This is because the control actuation of the booster is limited by the physical limitations of its thrust vectoring capabilities. So, a maximum angle of attack and angle of attack rate are defined. These maximum angles are defined in the following relations:

$$\begin{cases} \alpha_{\min} \leq \alpha \leq \alpha_{\max} \\ \dot{\alpha}_{\min} \leq \dot{\alpha} \leq \dot{\alpha}_{\max} \end{cases} \tag{5.35}$$

6

Guidance and Optimization

As outlined in the previous chapters, the indirect optimization method is selected for the optimization of the ascent trajectory. The Methodology trade-off of figure B.1, shows that particularly the approach described in the work of P. Lu et al. [43] [42], is relevant for this study and aligns well with the research objectives.

As can be read in the appendix, the indirect optimization method was chosen over direct approaches due to its higher computational efficiency and superior capability for sensitivity analysis. The method allows efficient evaluation of how parameter variations affect the optimal solution, while avoiding the heavy computational load and discretizations required by direct methods. Following the requirements stated in chapters 3 and 4, the indirect method is a more effective choice for the research objectives in this study.

This chapter will first dive deeper into how the indirect optimization method exactly works. Afterwards, it will explain how this method is applied to the ascent optimization problem of this study. It is divided into two parts: the atmospheric part and the vacuum part of the trajectory. Finally, the terminal conditions of the optimization problem are defined. So, the goal of this chapter is to explain how the optimization of the ascent trajectory is determined.

6.1. The Indirect Optimization Method

In this section, an overview of the indirect method for solving optimal control problems (OCPs) is provided. Specifically, optimal control problems of dynamic systems that are time-dependent. OCPs focus on determining the control input that optimizes a given performance measure while adhering to specific boundary constraints, as well as state and control limitations. The Pontryagin's Maximum Principle (PMP) is a method that uses calculus of variations to solve such an OCP. Let x and u represent the state and control vectors, respectively. The OCP can be formulated as follows [52]:

$$\text{Minimize: } J = \phi(x(t_f), t_f, x(t_0), t_0) + \int_{t_0}^{t_f} L(x, u, t)dt, \tag{6.1}$$

$$\text{Subject to: } \dot{x} = f(x, u, t), \tag{6.2}$$

$$\Psi(x(t_0), t_0) = 0, \tag{6.3}$$

$$\Phi(x(t_f), t_f) = 0, \tag{6.4}$$

Here J represents the cost functional, ϕ denotes the terminal cost, and $L(x, u, t)$ is the running cost, with t_0 typically set to zero. Equation 6.2 describes the system's equations of motion (EOMs). Indirect methods must also satisfy the boundary constraints Ψ and Φ, as given in equations 6.3 and 6.4, respectively. Additionally, constraints on states and control inputs along the trajectory are considered later. Applying this principle to Eq. 6.1 results in a set of conditions that, when met, identify a potential optimal solution. These conditions, known as the First-Order Optimality Conditions, combine with the existing constraints to define the Hamiltonian Boundary-Value Problem [53]. The Hamiltonian function is given by [52]:

$$H = L(x, u, t) + \boldsymbol{p}^T(t) f(x, u, t), \tag{6.5}$$

where $\mathbf{p}^T(t)$ represents the co-state vector, which has the same dimension as x. The associated co-state dynamics are defined as:

$$\dot{\boldsymbol{p}} = -\left[\frac{\partial H}{\partial x}\right]^T. \tag{6.6}$$

For cases where the control is unbounded, the optimal control law u^* can be derived using [52]:

$$\frac{\partial H}{\partial u} = 0. \tag{6.7}$$

The co-state boundary conditions at the initial and final times are given by:

$$\boldsymbol{p}(t_0) = \nu_0^T \frac{\Psi}{\partial x(t_0)}, \tag{6.8}$$

$$\boldsymbol{p}(t_f) = \frac{\phi}{\partial x(t_f)} + \nu_f^T \frac{\Phi}{\partial x(t_f)}, \tag{6.9}$$

where ν_0 and ν_f are constant Lagrange multipliers used to enforce these boundary conditions. In problems with a free final time, the following additional condition must be met [52]:

$$\left(H + \frac{\phi}{\partial t} + \nu_f^T \frac{\Phi}{\partial t}\right)_{t=t_f} = 0. \tag{6.10}$$

These equations define the necessary conditions for optimality, forming a well-defined two-point boundary value problem (TPBVP) that can be solved using numerical methods such as MATLAB's `bvp4c` solver [54].

Furthermore, it can happen that the control inputs appear linearly in the Hamiltonian; this is known as bang-bang control. This is a control strategy often observed in certain optimal control problems. In such cases, the optimal control typically switches between two extreme values: an upper and a lower bound. Instead of gradually varying, the control ”jumps” directly from one bound to the other, resulting in a discontinuous control profile. A situation where the control takes on a value between these bounds is known as a singular control arc. However, in optimal control, singular arcs are generally undesirable, as they complicate the solution process. Therefore, the optimal control is often designed to assume only the extreme values, leading to the characteristic bang-bang behavior [55].

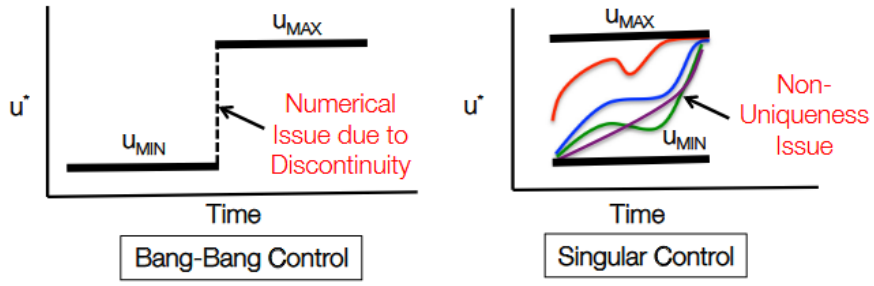


Figure 6.1: Schematic of bang-bang control [52].

Of course, it is also possible that the system is bounded by path constraints or that the input of the control is bounded. Control constraints are typically expressed as inequalities of the form:

$$C(u,t) \leq 0. \tag{6.11}$$

To enforce these constraints, the control law is determined by setting $C = 0$ at the constraint boundary. In the standard approach, the Hamiltonian with the control constraints leads to the modified expression [52]:

$$H = L + \boldsymbol{p}^T f + \mu_c^T C, \tag{6.12}$$

Here μ represents the multipliers associated with the control constraints. These additional terms modify the equations of motion for the co-states along constrained arcs [52]. In the optimization model of this study, these control constraints mean, for instance, a limitation on the magnitude of the angle of attack of the rocket. All the physical control constraints on the rocket were elaborated in chapter 5.5. Next to this, path constraints also impose restrictions on the system states and controls throughout the trajectory. They are given by [52]:

$$S(x,u,t) \leq 0. \tag{6.13}$$

The presence of these constraints complicates the boundary-value problem (BVP) by introducing corner conditions in the co-states and splitting the trajectory into multiple segments. To determine the control history along constrained arcs, the time derivatives of the path constraints are computed to obtain an explicit dependence on the control variable. If this occurs at the q^{th} derivative, the Hamiltonian is augmented accordingly [52]:

$$H = L + \boldsymbol{p}^T f + \mu_s^T S^q. \tag{6.14}$$

An example of finding the control history along constraints arcs can be seen with the path constraint on the dynamic pressure on the booster, which will be discussed in section 6.6. Here, the control variable is the thrust level, which is adapted in order not to exceed the path constraint. The formulation from equation 6.14 ensures that both control and path constraints are satisfied while optimizing the system's trajectory according to PMP. By incorporating multipliers for these constraints, the method accounts for discontinuities in the co-states and maintains feasibility across constrained arcs.

In summary, in the context of the rocket ascent problem, the abstract constraints correspond to physical limitations of the vehicle and its environment. The control constraint $C(u,t)$ represents bounds on control inputs such as the magnitude of the angle of attack or thrust level, while the path constraint $S(x,u,t)$ captures flight envelope limits like dynamic pressure q_{dyn}, or constraining the angle of attack multiplied with the dynamic pressure to prevent structural failure. Furthermore, the actual implementation of these control and path constraints into the optimization model will be elaborated in section 6.3.2.

6.2. Optimization Objective

In this section, the optimization problem of this study will be defined, following the theory presented in the previous section. The approach by P. Lu offers an effective framework for applying this method to compute the optimal control inputs for the booster's ascent trajectory [43]. Another paper by P. Lu adopts the same approach and provides a more detailed and comprehensive elaboration of this same method [40]. Therefore, this approach provides a solid foundation for this research focused on finding the optimal solution. The first and most crucial objective is to determine the optimal control input that guides the direction of the launch vehicle. This section will further elaborate on how this optimization is realized.

In chapters 3 and 4, it is explained what the final optimization should solve. The ascent trajectory is optimized in alignment with the HGV's mission objective. There are many possible objectives that an HGV can have. However, in the scope of this study, two of the most applicable scenarios will be focused on.

As presented in chapter 4, often the goal of an HGV is to maximize its range. To maximize the overall mission range, the booster delivering the HGV also has to be optimized to maximize its own range. So, the range maximization of the rocket is the first objective. However, the objective of an HGV is not always to maximize range. In such cases, the booster must deliver the HGV to specific final conditions that best enable its glide trajectory towards a target. The aim is therefore to reach these conditions as quickly as possible, minimizing the overall flight time. Therefore, the optimization program created in this study will aim to solve both of these situations.

It is also necessary for the HGV to be delivered with a negative flight path angle. This angle, however, must not be too steep; if it is excessively negative, the vehicle will leave its designated corridor and risk burning up. Therefore, the negative terminal flight path angle causes the rocket to end its flight in a descent trajectory. For this trajectory that ends in a descent, a maximum range inherently corresponds to a maximum terminal velocity. So, no greater range can be achieved if no higher feasible terminal velocity is possible under the same final conditions. The verification of this proposition will be shown later in Chapter 9.

How this objective can be optimized is elaborated in the next sections of this chapter. Here, the optimization is split into the atmospheric and vacuum parts of the trajectory.

6.3. Optimality of the Atmospheric Ascent

The atmospheric ascent is the first part of the trajectory that contains the burn of the first stage. This is therefore a fixed final time problem, since the propellant burn time is known. Only when the rocket throttles does the burn time change. However, as will be explained in chapter 6.6, the throttle profile is determined outside the optimization loop discussed in this section. Therefore, even in cases where throttling occurs, the total burn time remains predefined, and the problem can still be treated as a fixed-final-time formulation.

The trajectory of the first stage is an endo-atmospheric flight. For the optimization problem, first a performance index needs to be chosen that has to be minimized. As previously discussed, the objective is to maximize the range and thus the terminal velocity of the trajectory. As can be seen in the verification chapter of this report, maximization of the orbital energy at the terminal point also leads to maximizing the terminal velocity. Therefore, the performance index that needs to be minimized is [43]:

$$J = \phi(\mathbf{r}_f, \mathbf{V}_f, t_f) = \frac{1}{r_f} - V_f^2 \tag{6.15}$$

For this research, first the body axis orientation is optimized, while the thrust level is not varied. Therefore, the mass $(m(t))$ is treated as a prescribed function of time, since the mass flow rate is constant. Therefore, mass is not treated as a state variable and does not appear in the Hamiltonian function.

The Hamiltonian function is defined following the Equations of Motion and the objective function presented in chapter 5. The Hamiltonian function looks as follows[43]:

$$H = \boldsymbol{p}_r^T \boldsymbol{V} + \boldsymbol{p}_V^T \left[\boldsymbol{g} + \frac{(T-A)}{m} \mathbf{1}_b + \frac{N}{m} \mathbf{1}_n \right] + \mu_s^T \boldsymbol{S}(\boldsymbol{r}, \boldsymbol{V}, t) + \mu \left(\mathbf{1}_b^T \mathbf{1}_b - 1 \right) \tag{6.16}$$

The objective function is based on a specific terminal cost, picked such that the range of the trajectory is maximized. Therefore, the Hamiltonian is solely based on the state and costate equations, since only a running cost is part of the Hamiltonian, which in this optimization is absent. This Hamiltonian function will be used to find the optimal control I_b^* to have maximum terminal velocity. In the Hamiltonian function, "$\mu\left(\boldsymbol{I}_b^T \boldsymbol{I}_b - 1\right)$" is a constraint function, which basically tells that I_b is a unit vector. So, I_b needs to be normalized in order to satisfy:

$$\|\boldsymbol{I}_b\| \equiv 1 \tag{6.17}$$

Furthermore, "$\mu_s^T \cdot \boldsymbol{S}(\boldsymbol{r}, \boldsymbol{V}, t)$" is the part of the function that presents the path constraints. This part of the Hamiltonian function will be elaborated on later in this chapter.

Lastly, the co-states of the system are presented by "$\boldsymbol{p}_r$" and "$\boldsymbol{p}_v$". These are the co-states of the position and velocity, respectively.

6.3.1. Co-state Equations

Next, it is essential to determine the co-state dynamics of the system. The costates have no actual physical meaning. However, they provide important boundary conditions and influence the formulation of the Hamiltonian equation. The co-state dynamics can be derived using equation 6.6.

For clarity, a simplified representation of the co-states is provided below, normalized by mass and neglecting the aerodynamic forces acting on the booster [56]:

$$\dot{\boldsymbol{p}}_r = -\frac{\partial H}{\partial \boldsymbol{r}} = \frac{1}{\|\boldsymbol{r}\|^3}\boldsymbol{p}_V - \frac{3\boldsymbol{p}_V^T \boldsymbol{r}}{\|\boldsymbol{r}\|^5}\boldsymbol{r} \tag{6.18}$$

$$\dot{\boldsymbol{p}}_V = -\frac{\partial H}{\partial \boldsymbol{V}} = -\boldsymbol{p}_r \tag{6.19}$$

However, in the optimization problem of this research, aerodynamic forces cannot be neglected. Therefore, the co-state dynamics are more complex within the scope of this work. By incorporating aerodynamic forces and applying the previously discussed environmental models, which depend on the position of the booster, the derivative of the Hamiltonian with respect to the state is given by the following expression [40]:

$$\begin{aligned}
\dot{\boldsymbol{p}}_r \;=\; & \frac{1}{r^3}\, p_V \;-\; \left[\frac{3\, a_{pvb}}{r^4} + a_{pvb}\left(T_r - A_{pr} + \frac{1}{2V_r}\, C_\rho\, V_s^2\, C_{A,Mach}\, \frac{\partial V_s}{\partial r}\right)\right. \\
& \left. + a_{pvn}\left(N_{pr} - \frac{1}{2V_r}\, C_\rho\, V_s^2\, C_{N,Mach}\, \frac{\partial V_s}{\partial r}\right)\right] \frac{\mathbf{r}}{r} \\
& + C_\rho\, \boldsymbol{\omega}_E \times \left\{ a_{pvb}\left[\left(C_A + \frac{1}{2V_r}\, V_s C_{A,Mach}\right)\mathbf{V}_r + \frac{1}{2}\, C_{A\alpha}\, V_r^2\, \frac{\partial \alpha}{\partial \mathbf{V}}\right]\right. \\
& \left. - a_{pvn}\left[\left(C_N + \frac{1}{2V_r}\, V_s C_{N,Mach}\right)\mathbf{V}_r + \frac{1}{2}\, C_{N\alpha}\, V_r^2\, \frac{\partial \alpha}{\partial \mathbf{V}}\right]\right\},
\end{aligned} \tag{6.20}$$

$$\begin{aligned}
\dot{\boldsymbol{p}}_V \;=\; & -\, p_r + C_\rho\left[a_{pvb}\left(C_A + \frac{1}{2V_r}\, V_s C_{A,Mach}\right) - a_{pvn}\left(C_N + \frac{1}{2V_r}\, V_s C_{N,Mach}\right)\right. \\
& \left. + \frac{1}{2}(a_{pvb} C_{A\alpha} - a_{pvn} C_{N\alpha})\, \frac{\cos\alpha}{\sin\alpha}\, \mathbf{V}_r\right] \;-\; \frac{C_\rho\, V_r}{2\sin\alpha}(a_{pvb} C_{A\alpha} - a_{pvn} C_{N\alpha})\; \mathbf{1}_b.
\end{aligned} \tag{6.21}$$

The definitions of the parameters used here are consistent with those introduced earlier in this report. However, there are a couple of new parameters, which are defined in the following way [40].

$$\rho_r = \frac{\partial \rho}{\partial r}, \quad T_r = \frac{\partial T}{\partial r}, \quad C_\rho = \frac{\rho_0 S_{ref} R_0}{m(t)},$$

and the following expressions are introduced:

$$A_{pr} = \frac{V_r^2 S_{ref} R_0 \rho_0 C_A \rho_r}{2m(t)}, \quad N_{pr} = \frac{V_r^2 S_{ref} R_0 \rho_0 C_N \rho_r}{2m(t)},$$

$$C_{A\alpha} = \frac{\partial C_A}{\partial \alpha}, \quad C_{N\alpha} = \frac{\partial C_N}{\partial \alpha},$$

$$C_{AMach} = \frac{\partial C_A}{\partial \text{Mach}}, \quad C_{NMach} = \frac{\partial C_N}{\partial \text{Mach}},$$

$$a_{pvb} = \boldsymbol{p}_V^T \mathbf{1}_b = \boldsymbol{p}_V \cos(\Phi - \alpha), \quad a_{pvn} = \boldsymbol{p}_V^T \mathbf{1}_n = \boldsymbol{p}_V \sin(\Phi - \alpha).$$

From the velocity relation

$$V_r = V - \boldsymbol{\omega}_E \times r,$$

The variation of the angle of attack with respect to velocity can be written as

$$\frac{\partial \alpha}{\partial \mathbf{V}} = \frac{1}{V_r \sin \alpha} \left(\cos \alpha \, \mathbf{1}_v - \mathbf{1}_b \right).$$

For this reason, it is useful to have algebraic relations, as defined in chapter 5.4. This way, the aerodynamic and Mach-dependent relations of the environmental models, together with their derivatives, can be used in the indirect optimization method. Some of the relations presented above are included in Appendix D, as their full expressions are too lengthy to list here. Furthermore, algebraic relations support convergence of the problem by providing the optimizer with more accurate gradient information.

6.3.2. Constraint formulation in the Indirect Method

Following chapter 5, the control input of the booster is limited and the booster is limited to path constraints that it can not exceed in order to have a feasible trajectory.

Control and Path constraints

The constraints influence the Hamiltonian function, as can be seen in equation 6.12. The constrained optimal control input of the angle of attack looks as follows:

$$\alpha^* \in [-\alpha_{max}, \alpha_{max}] \tag{6.22}$$

The angle of attack does not show bang-bang control behavior, since it does not appear linear in the Hamiltonian function. Therefore, the Hamiltonian function is defined as below:

$$H = L + \mathbf{p}^T f + \mu_1(-\alpha - \alpha_{max}) + \mu_2(\alpha - \alpha_{max}) \tag{6.23}$$

These multipliers (μ) can be either active or inactive ($\mu = 0$), depending on the angle of attack.

$$\mu_1 = \begin{cases} 0, & \alpha \geq \alpha_{max} \\ \frac{\partial H}{\partial \alpha}, & \alpha = \alpha_{max} \end{cases} \tag{6.24}$$

$$\mu_2 = \begin{cases} 0, & \alpha \geq \alpha_{max} \\ -\frac{\partial H}{\partial \alpha}, & \alpha = \alpha_{max} \end{cases} \tag{6.25}$$

These relations basically make the Hamiltonian function go to zero when the angle of attack is at its maximum. Therefore, automatically limiting the angle of attack can get an even higher magnitude.

Next, there are the path constraints. In literature, one of the most prominent path constraints for launch vehicles is the product of dynamic pressure and the angle of attack. Like defined in the research of P. Lu as [40]:

$$|q\alpha| \leq Q_\alpha \tag{6.26}$$

This constraint is based on the dynamic pressure constraint that is defined in equation 5.34. So, this Q_α path constraint is implemented in the current model of this research. In equations 6.14 it can be

seen how a path constraint influences the Hamiltonian function. The constraint function can be defined in the following way.

$$S_1 = q \cdot \alpha - Q_\alpha \leq 0 \tag{6.27}$$

The control I_b appears in this path constraint through the angle of attack (α). So, when the constraint is active, $S_1 = 0$. In this case the control of the angle of attack can be found by:

$$\alpha = \frac{Q_\alpha}{q} \tag{6.28}$$

Here, "q" is the dynamic pressure of the rocket at a certain time (t).

As will be explained later, the dynamic pressure constraint is also implemented [40]:

$$S_2 = q - q_{max} \leq 0 \tag{6.29}$$

Using the engine throttle to enforce this constraint is the most efficient strategy [40]. However, there is a separate section dedicated to the implementation of this constraint.

6.3.3. Primer Vector theory

The following relations present a possible way to define the state of the ascent of a booster [57].

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{r}} \\ \dot{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{g}(\mathbf{r}) + \Gamma \mathbf{u} \end{bmatrix} \tag{6.30}$$

Here, Γ is the thrust per unit mass. In this case, where the state is defined in such a way that the body axis orientation (I_b) only enters the Hamiltonian as a part of the derivative of the velocity. The optimal body axis orientation (I_b) is the one that maximizes the Hamiltonian, within the set of admissible controls. This can therefore be defined as [57]:

$$I_b^* = \frac{\boldsymbol{p}_v}{|\boldsymbol{p}_v|} \tag{6.31}$$

Therefore, the optimal body axis rotation can be determined by computing the unit vector of the velocity costate vector. This unit vector is called the primer vector. It presents a very elegant way to compute the full body orientation, together with the equations 5.21 and 5.24.

When aerodynamic forces are incorporated into the equations of motion, the optimal orientation of the body axis can still be determined using the primer vector. However, it will be slightly different due to a correction needed for the aerodynamic forces. The optimal control problem can be found by solving equation 6.7. This gives the optimality condition for this research:

$$\frac{\partial H}{\partial \mathbf{1}_b} = 0 \tag{6.32}$$

This ensures that the Hamiltonian is maximized with respect to the body-axis direction. Following the work of P. Lu et. al [43], it can be concluded that the optimal thrust vector lies in the plane spanned by the primer vector $\boldsymbol{p}_v$ and the Earth relative velocity vector $\boldsymbol{V}_r$. This is illustrated schematically below in figure 6.2.

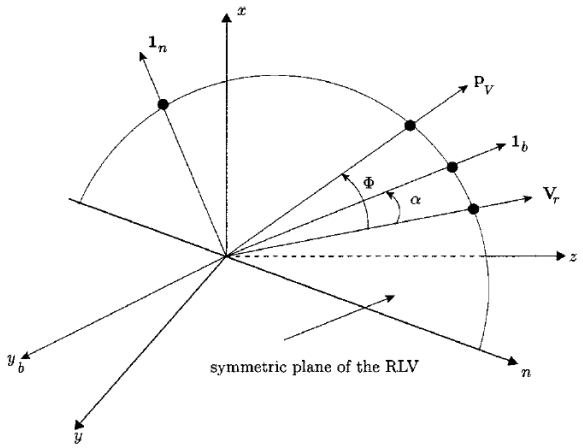


Figure 6.2: Body axes and primer vector[43].

After much differentiation and algebraic operation, the optimal body-axis direction can thus be expressed as a linear combination of these two vectors [43]:

$$\mathbf{1}_b^* = c_1 \boldsymbol{p}_v + c_2 \boldsymbol{V}_r \tag{6.33}$$

where c_1 and c_2 are scalar functions of the state and co-state variables. As atmospheric density decreases during flight, the influence of drag diminishes, leading to $c_2 \to 0$, and the optimal body-axis vector aligns solely with the primer vector $\boldsymbol{p}_v$. Therefore, in this case, the optimal body axis is computed just like in equation 6.31.

Furthermore, since the zero sideslip method (”Fly into the wind” maneuver) is used for modeling the body orientation, the symmetric plane of the body axis is aligned with the Earth’s relative velocity vector. Therefore, when aerodynamic forces are active on the rocket, the problem reduces to a one-dimensional search within the plane formed by $\boldsymbol{p}_v$ and $\boldsymbol{V}_r$. This highly simplifies the problem. In summary, the Hamiltonian is maximized when I_b lies in the plane between the primer vector and the relative velocity vector. So, the calculation of the primer vector from the velocity costate, combined with the determination of the optimal angle of attack, directly provides the full body axis rotation.

The work of P. Lu shows how exactly this optimal angle of attack can be found, following $\partial H / \partial \alpha$. This leads to the following optimality condition [43]:

$$\tan(\Phi - \alpha)(T - A + N_\alpha) - (A_\alpha + N) = 0 \tag{6.34}$$

Here Φ is the angle between the vectors p_v and V_r. Further, A_α and N_α are the derivatives of the aerodynamic forces with respect to α. This equation must be solved numerically for the angle of attack α. This is done with the Newton-Raphson method, seen in the algorithm of the pseudocode below. This formulation now explicitly includes aerodynamic forces, making the solution applicable to the more realistic case of a launch vehicle flying through the atmosphere. Once ”α” is known, the optimal body-axis orientation is computed directly from the following analytical relation [43]:

$$\mathbf{1}_b^* = \left(\frac{\sin \alpha}{\sin \Phi} \right) \mathbf{1}_{p_V} + \left(\frac{\sin(\Phi - \alpha)}{\sin \Phi} \right) \mathbf{1}_{V_r} \tag{6.35}$$

This approach, shown by P. Lu [43], provides a computationally efficient method for determining the optimal body orientation. It reduces the problem to computing the unit vector of the velocity costate and solving a nonlinear equation for ”α”. This avoids the need to compute the full derivative of the Hamiltonian with respect to the control variable, as described in equation 6.7. Therefore, this gave the rationale for defining the Equation of Motion and force equations, as presented in chapter 5.

The pseudocode underneath gives the algorithm that is implemented in MATLAB, which is used to compute the optimal body axis rotation. The full code implementation of the computation algorithm for the angle of attack can be found in Appendix C.

Algorithm 1 Computation of α and I_b pseudocode

procedure Calculate_Alpha($\mathbf{p}_V$, $\mathbf{V}$, T, ρ, q_{dyn}, p, α_{prev})
  $\cos\Phi \leftarrow \dfrac{p_V \cdot V}{\|p_V\|\,\|V\|}$; $\cos\Phi \leftarrow \mathsf{clip}(\cos\Phi, -1, 1)$
  $\Phi \leftarrow \arccos(\cos\Phi)$
  $V_{\mathsf{mag}} \leftarrow \|V\|$
  Set $\alpha_{\max}$, $\alpha_{step\_\max}$, and Q_α from defined parameters
  $\alpha_\star \leftarrow 0$ ▷ Newton iterate for optimal α
  for $i = 1$ **to** 10 **do**
    Compute $C_\ell, C_d, A, A_\alpha, A'_\alpha, N, N_\alpha$ from $(V, \rho, \alpha_\star)$
    $F \leftarrow \tan(\Phi - \alpha_\star)\left(T - A + N_\alpha\right) - \left(A_\alpha + N\right)$
    $dF \leftarrow \dfrac{\partial F}{\partial \alpha}$ ▷ analytic derivative
    $\alpha_\star \leftarrow \alpha_\star - \dfrac{F}{dF + \epsilon}$
  end for
  $\alpha_\star \leftarrow \mathsf{clip}(\alpha_\star, -\alpha_{\max}, \alpha_{\max})$ ▷ AoA constraint
  if $\left|q_{\mathsf{dyn}} \cdot \alpha_\star\right| \le Q_\alpha$ **then** ▷ AoA & Dynamic pressure constraint
    $\alpha_{\mathsf{constrained}} \leftarrow \alpha_\star$
  else
    $\alpha_q \leftarrow Q_\alpha / q_{\mathsf{dyn}}$
    $\alpha_{\mathsf{constrained}} \leftarrow \mathsf{sign}(\alpha_\star) \cdot \min(\alpha_q, \alpha_{\max})$
  end if
  $\Delta \leftarrow \alpha_{\mathsf{constrained}} - \alpha_{\mathsf{prev}}$ ▷ AoA rate constraint
  if $\Delta > \alpha_{\mathsf{step\_max}}$ **then**
    $\Delta \leftarrow \alpha_{\mathsf{step\_max}}$
  else if $\Delta < -\alpha_{\mathsf{step\_max}}$ **then**
    $\Delta \leftarrow -\alpha_{\mathsf{step\_max}}$
  end if
  $\alpha \leftarrow \alpha_{\mathsf{prev}} + \Delta$
  return (α, Φ)
end procedure

Require: Inputs: V_r, p_V
$a_1 \leftarrow \dfrac{\sin(\alpha)}{\Phi}$
$b_1 \leftarrow \dfrac{\cos(\alpha) - \cos(\Phi)\,\cos(\Phi - \alpha)}{\Phi^2}$
$I_b \leftarrow a_1 \cdot \dfrac{p_V}{\|p_V\|} + b_1 \cdot \dfrac{V_r}{\|V_r\|}$

As shown in the algorithm, the derivatives of the aerodynamic forces with respect to the angle of attack are required. To obtain their analytical expressions, the derivatives of the aerodynamic coefficients with respect to the angle of attack are also needed. Their expressions are provided in Appendix D.

The algorithm computes the angle of attack using ten iterative loops of the Newton–Raphson method. It was found that the algorithm typically converges to a solution within about five iterations. However, to ensure numerical robustness and guarantee convergence, ten iterations were selected. This way, no numerical instabilities or divergence were observed, even for rapidly changing control profiles. Increasing the number of Newton–Raphson iterations significantly affects the overall computation time of the optimal atmospheric trajectory solution, as the above algorithm is executed at every time step within each new iteration of the atmospheric solution process. The solution process of the entire atmospheric

trajectory is further described in Chapter 7.2. Since no changes in the trajectory were observed when the number of iterations was increased (to for example fifteen), ten iterations were deemed sufficient.

It can be seen that in this algorithm, the control and path constraints on the booster are also included with $\alpha_{\max}$, $\alpha_{step\_\max}$, and Q_α. Here, α_{prev} is the angle of attack of the previous time step in the trajectory. This way, the angle of attack rate constraints can be enforced. All the constraints are enforced outside of the Newton-Raphson loop. Therefore, enforcing the constraints does not influence the convergence or numerical stability of the computation of the angle of attack.

6.3.4. Takeaway of Atmospheric Ascent Optimality

In summary, this section established the full formulation of the atmospheric ascent optimization problem. The optimization objective was defined to maximize the terminal velocity, from which the corresponding Hamiltonian and co-state dynamics were derived. The necessary control and path constraints were identified and formulated, ensuring that the vehicle's physical limits are respected throughout the trajectory. Furthermore, the methodology for determining the optimal body-axis orientation was developed for both vacuum and atmospheric flight conditions, accounting for aerodynamic effects in the latter. Finally, the computational algorithm used to determine the optimal angle of attack and body-axis orientation was presented, providing the foundation for the atmospheric ascent solution discussed in the next chapter.

6.4. Analytical Vacuum Optimal Ascent Solution

Since only a fraction of the trajectory takes place within the endo-atmospheric environment, an optimal trajectory must also be determined for the remaining exo-atmospheric vacuum phase. Because the rocket consists of multiple stages, the solution must account for both a burn phase of the remaining stage as well as the coasting phases. As discussed previously, the latter implies that the coasting duration acts as a control variable in the optimal guidance problem.

Following the requirements, the booster model shall allow for throttling. A liquid-propellant booster can cut off its engine before the propellant is fully depleted. This introduces an additional control variable: the burn time of the second stage. This section will therefore address the optimization of the guidance for the remaining stage, the optimal coasting time, and the optimal burn time. How these sections of the optimized trajectory are tied together in one total trajectory is explained in the next chapter 7.

When coast phases and multiple burns are introduced, the optimal control problem becomes significantly more challenging and highly sensitive. This increased sensitivity arises mainly from two factors: long coast arcs amplify even small deviations in the initial conditions at the start of the coast, and the final stage typically provides much lower thrust acceleration, making it difficult to perform substantial trajectory corrections [40]. Therefore, for the vacuum ascent, the optimization problem needs to be defined such that it facilitates faster and more reliable convergence to a solution. This is achieved by choosing a method that provides a closed-form solution to the greatest extent possible.

6.4.1. Burn arc solution

For the burn arc solution, the Analytical Vacuum Optimal Ascent Solution method is used, presented by P. Lu et. al [43]. The main building blocks of this method are: the linear gravity approximation, closed-form solutions of the costate dynamics, and closed-form solutions of the state equation using quadrature formulas. At the start of each guidance cycle, let r_0 denote the dimensional position vector. The gravitational acceleration is then approximated as a linear function of r [43]:

$$g = -\left(\frac{\mu_E}{r_0^3}\right) r = -\omega_s^2 r \tag{6.36}$$

where μ_E is Earth's gravitational parameter. Furthermore, ω_s is called the Schuler frequency. Introducing $g_0 = \mu_E / r_0^2$ as the gravitational acceleration at r_0, we normalize distance by r_0, time by $\sqrt{r_0/g_0}$, and velocity by $\sqrt{r_0 g_0}$. For vacuum flight ($A = N = 0$), the nondimensional equations of motion are [43]:

$$\dot{r} = V \tag{6.37}$$

$$\dot{V} = -r + T(\tau) I_b \tag{6.38}$$

Using nondimensional equations of motion improves convergence because it scales the variables to comparable magnitudes, reducing numerical ill-conditioning in the optimizer. This prevents certain states from dominating the solution and allows the optimization algorithm to explore the search space more efficiently. Here, $T(\tau)$ is the normalized thrust, $T(\tau) = T_{\text{vac}}/(m(\tau)g_0)$ with τ the dimensionless time.

The Hamiltonian is given by equation 6.16, only the Aerodynamic forces in this equation can be removed. P. Lu et. al. present a closed-form solution of the costates derived from the non-dimensional equations of motion and the Hamiltonian. This close form solution is [43]:

$$\begin{bmatrix} p_V(\tau) \\ -p_r(\tau) \end{bmatrix} = \begin{bmatrix} \cos\tau I_3 & \sin\tau I_3 \\ -\sin\tau I_3 & \cos\tau I_3 \end{bmatrix} \begin{bmatrix} p_{V0} \\ -p_{r0} \end{bmatrix} \equiv \Omega(\tau) \begin{bmatrix} p_{V0} \\ -p_{r0} \end{bmatrix} \tag{6.39}$$

Here p_{V0} and p_{r0} are the initial costates. Then, quadrature thrust integrals are used to obtain a closed-form solution of the state [43].

$$I_c(\tau) = \int_0^\tau I_{p_V}(\zeta) \cos\zeta T(\zeta) d\zeta, \quad I_s(\tau) = \int_0^\tau I_{p_V}(\zeta) \sin\zeta T(\zeta) d\zeta \tag{6.40}$$

The thrust integrals $I_c(\tau)$ and $I_s(\tau)$ represent the accumulated effect of the time-varying thrust, projected along cosine and sine weightings of the primer direction. They capture how the thrust history influences the state, allowing equation 6.41 to express $r(\tau)$ and $V(\tau)$ in a closed form solution [43].

$$\begin{bmatrix} r(\tau) \\ V(\tau) \end{bmatrix} = \Omega(\tau) \begin{bmatrix} r_0 \\ V_0 \end{bmatrix} + \Gamma(\tau) \begin{bmatrix} I_c(\tau) \\ I_s(\tau) \end{bmatrix} \tag{6.41}$$

with

$$\Gamma(\tau) = \begin{bmatrix} \sin\tau I_3 & -\cos\tau I_3 \\ \cos\tau I_3 & \sin\tau I_3 \end{bmatrix} \tag{6.42}$$

The thrust integrals I_c and I_s cannot be expressed in a simple closed form, so they must be evaluated numerically, meaning the integral is approximated by summing function values at selected points rather than solving it exactly. A common approach is Milne's rule. Milne's rule approximates the integral of a function by replacing it with a polynomial that passes through a set of equally spaced sample points and then integrating that polynomial exactly. The result is a weighted sum of the function values at those points, ensuring high-order accuracy for smooth functions. The weights are obtained by integrating the basis functions used to approximate the curve between the data points, and therefore depend only on the locations of those points, not on the function values [58]. This makes the integrals practical to compute and insert back into the state solution. Here, τ_{togo} is the dimensionless time remaining until the end of the engine burn. The parameter $\delta = \tau_{\text{togo}}/4$ is simply a subdivision of this interval into four equal parts, which is required by Milne's rule to evaluate the integral using five evenly spaced points. Milne's rule gives [43]:

$$I_i(\tau_{\text{togo}}) = \frac{\tau_{\text{togo}}}{90} \left[7I_i(0) + 32I_i(\delta) + 12I_i(2\delta) + 32I_i(3\delta) + 7I_i(4\delta)\right], \quad i = c, s \tag{6.43}$$

Given the above method, the final state and costate have become functions of p_{v0} and p_{r0}. Therefore, the terminal conditions that need to be satisfied are functions of the unknowns p_{v0}, p_{r0}, and τ_{togo}. This

makes the optimization problem a root-finding problem. Furthermore, how the terminal conditions are defined will be explained later in this chapter.

Underneath a pseudo code can be found on how exactly the above closed form solutions are calculated, given a certain input for p_{v0}, p_{r0}, and τ_{togo}.

Algorithm 2 Second Stage Vacuum Propagation with Milne Quadrature

Require: Inputs: p_{v0}, p_{r0}, τ_{final}, r_0, v_0, m_0, $\dot{m}$, g_0, T_{vac}

$I \leftarrow I_3$ ▷ identity matrix
$X_0 \leftarrow [r_0; v_0]$
$h \leftarrow \tau_{\text{final}}/4$
$\tau_\cdot \leftarrow \text{linspace}(0, \tau_{\text{final}}, 5)$

for $k = 1$ to 5 **do**
  $t \leftarrow \tau(k)$
  $\Omega(t) \leftarrow \begin{bmatrix} \cos t\, I & \sin t\, I \\ -\sin t\, I & \cos t\, I \end{bmatrix}$
  $\Lambda(t) \leftarrow \Omega(t)[p_{v0}; -p_{r0}]$
  $p_V(t) \leftarrow \Lambda(t)_{1:3}$;
  $pR(t) \leftarrow -\Lambda(t)_{4:6}$
  $\hat{p}(t) \leftarrow p_V(t)/\|p_V(t)\|$
  $m(t) \leftarrow m_0 - \dot{m}t$;
  $T(t) \leftarrow T_{\text{vac}}/(m(t)g_0)$
  $I_c^{\cdot}(:,k) \leftarrow \hat{p}(t)\cos(t)T(t)$
  $I_s^{\cdot}(:,k) \leftarrow \hat{p}(t)\sin(t)T(t)$
end for

$I_c \leftarrow \frac{2h}{45}\left(7I_c^{\cdot}(:,1) + 32I_c^{\cdot}(:,2) + 12I_c^{\cdot}(:,3) + 32I_c^{\cdot}(:,4) + 7I_c^{\cdot}(:,5)\right)$
$I_s \leftarrow \frac{2h}{45}\left(7I_s^{\cdot}(:,1) + 32I_s^{\cdot}(:,2) + 12I_s^{\cdot}(:,3) + 32I_s^{\cdot}(:,4) + 7I_s^{\cdot}(:,5)\right)$

$t \leftarrow \tau_{\text{final}}$
$\Omega(t) \leftarrow \begin{bmatrix} \cos t\, I & \sin t\, I \\ -\sin t\, I & \cos t\, I \end{bmatrix}$
$\Gamma(t) \leftarrow \begin{bmatrix} \sin t\, I & -\cos t\, I \\ \cos t\, I & \sin t\, I \end{bmatrix}$
$X(t) \leftarrow \Omega(t)X_0 + \Gamma(t)[I_c; I_s]$

This pseudocode is based on the MATLAB code of the state and co-state propagation during the burn arc of the analytical vacuum solution. This code can be found in Appendix C.

In summary, the methodology described above provides a formulation that enables the derivation of a closed-form solution for the state and costate evolution during the burn arc of the vacuum phase. This closed-form solution is achieved through the use of non-dimensional equations of motion, the linear gravity approximation, and the inclusion of thrust integrals.

6.4.2. Coast Arc solution

Then the coast trajectory needs to be optimized. This means particularly that the coasting duration needs to be optimized. This section will first explain how the optimal coasting time is determined, and then how the state and costate are propagated during the coast.

In order to optimize the coast and burn period of a booster, a switching function is needed. Optimizing the coasting time of a booster helps to decide when to apply thrust or remain in a coasting phase. In the case of P. Lu, this decision between maximum thrust and zero thrust depends on the switching function [42]. So, the switching function determines the switching between burn and coast arcs. The Hamiltonian of the optimization problem can be rewritten in order to separate the parts of the Hamiltonian that

are dependent and independent of the thrust. The Hamiltonian then looks as follows [59]:

$$H = H_0 + T \cdot S \tag{6.44}$$

Here, H_0 is the part of the Hamiltonian that is independent of the thrust. For a free final time problem, the Hamiltonian is identically zero along the optimal trajectory [60]:

$$H(t) = 0, \quad t_0 \leq t \leq t_f$$

Further, the switching function S is the partial derivative of the Hamiltonian with respect to thrust. This function S determines when to use the booster's thrust and when to coast. During the coasting phase, the thrust of the booster is zero ($T = 0$). The Hamiltonian is maximized following the relation [60]:

$$T = \begin{cases} T_{\max} & S > 0 \\ 0 & S < 0 \end{cases} \tag{6.45}$$

The thrust switches on or off at certain switching points, t_i. The switching function at each switching point is zero:

$$S(t_i) = 0, \quad i = 1, 2, ..., n$$

Here, n is the number of free switching points. So, at each switching point, $S = 0$ and therefore $H_0 = 0$. Determining these switching times will lead to solving the optimal coasting period [60]. These conditions ensure that the transition between powered flight and coasting is determined optimally, with the switching function acting as an indicator for when thrust should start or stop.

As an application, consider a launch vehicle with multiple stages. After the first stage has burned to completion, the second stage does not necessarily need to ignite immediately. Instead, it may be optimal to insert a coast period during which the vehicle follows a ballistic trajectory before starting the second stage burn. The duration of this coast arc is not fixed in advance but is determined as part of the optimization problem.

The necessary condition for finding the ignition time t_1 of the second stage reduces simply to [60]:

$$H_0(t_1) = 0 \tag{6.46}$$

This condition provides a direct way to determine the start of the next burn without requiring integration of additional co-states. In this way, the optimal coasting period can be identified, which improves the overall performance of the ascent trajectory by avoiding inefficient or premature ignition of the next stage.

During a coast arc, the state of the booster is propagated using Keplerian motion, while the costates are propagated in the same manner as the vacuum burn arc solution explained previously. As seen in the requirements for the hypersonic glide vehicle (HGV), it's important that it is inserted with a negative flight path angle to ensure proper entry into the atmosphere. This makes the inclusion of a final coast phase necessary, which is a new aspect introduced in this research, compared to the studied literature. As a result, the optimization problem formulated here explicitly includes a final coast, in contrast to the analytical framework of P. Lu et al. [40], where such a terminal coast is not present. Therefore, when a two-stage rocket is used, the trajectory consists of two burn arcs and two coast arcs.

Coast arc propagation

The coast arcs are propagated by using the initial position and velocity together with the coasting time, which is determined by the optimization model. The coast arc is propagated using Keplerian motion. In trajectory optimization, it is significantly more efficient to use the closed-form Keplerian solution instead of directly integrating the equations of motion. Numerical integration of the Newtonian dynamics is computationally expensive in contrast with a Keplerian formulation. The Keplerian formulation offers a closed-form solution for coast arcs, providing an exact relation between the initial state $(\mathbf{r}_0, \mathbf{v}_0)$ and the future state without the need for numerical integration.

In the optimization, each coast arc is defined by the state at its start, $\mathbf{r}_0,\ \mathbf{v}_0$, and the coast duration Δt. This is an input from the optimizer, as will be seen in chapter 7. From these, the specific angular momentum $\mathbf{h}_a = \mathbf{r}_0 \times \mathbf{v}_0$ (with $h_a = \|\mathbf{h}_a\|$) is computed, which remains constant throughout the coast [35]. The eccentricity vector follows as [35]:

$$\mathbf{e} \;=\; \frac{\mathbf{v}_0 \times \mathbf{h}_a}{\mu} \;-\; \frac{\mathbf{r}_0}{\|\mathbf{r}_0\|}, \qquad e = \|\mathbf{e}\|, \tag{6.47}$$

where μ is the gravitational parameter. Using vis–viva, the semi-major axis is [35]:

$$a \;=\; \left(\frac{2}{\|\mathbf{r}_0\|} - \frac{\|\mathbf{v}_0\|^2}{\mu}\right)^{-1} \tag{6.48}$$

To propagate the coast, it is important to first determine the initial true anomaly θ_0. Using $p = h_a^2/\mu$ and $r_0 = \|\mathbf{r}_0\|$. Here, p is the semi-latus rectum. This can be computed with the following equation.

$$\theta_0 \;=\; \arccos\!\left(\frac{(h_a^2/\mu) - r_0}{e\, r_0}\right), \tag{6.49}$$

Next, the true anomalies from the second point up to the final point of the coast need to be determined. The true anomaly of the first point is known, and the coast duration is given as an input by the optimizer. To compute the true anomalies, first the mean anomaly and the eccentric anomaly need to be computed.

The eccentric anomaly E is an auxiliary angle used in elliptical orbits, defined geometrically in the orbital ellipse to simplify the relation between time and position. The mean anomaly M_a is a measure of time along the orbit, increasing linearly with time from perigee [35]. The eccentric anomaly can be visualized in the figure below.

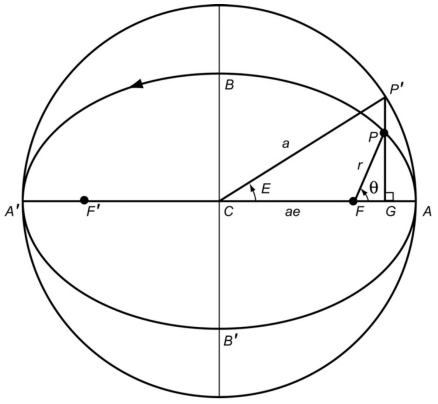


Figure 6.3: Eccentric Anomaly in orbital frame [35]

Here e is the orbital eccentricity. Furthermore, the mean anomaly can be computed with the following relation [35].

$$M_a \;=\; M_{a,0} + n \cdot (t_N - t_0), \quad n = \sqrt{\frac{\mu}{a^3}} \tag{6.50}$$

So, n gives the mean motion of the angular rate at which the satellite moves along its orbit, which is based on the orbital period. This is then used to compute the mean anomaly at a certain time.

The eccentric anomaly can then be computed with the following equation [35]:

$$E^{(k+1)} \;=\; M_a(t) + e \cdot \sin E^{(k)} \tag{6.51}$$

This is an iterative process. First, an initial guess for $E^{(k)}$ needs to be given. In this study, an initial guess of π is provided. Following the equation, a new value for the eccentric anomaly is computed. This loop is continued until the eccentric anomaly is converged to a certain value, and $\left|E^{(k+1)} - E^{(k)}\right| < \varepsilon$.

Once the eccentric anomaly is determined, the true anomaly can be computed. This is done with the following equation.

$$\theta \;=\; 2\arctan\!\Big(\sqrt{\tfrac{1+e}{1-e}}\tan\tfrac{E}{2}\Big). \tag{6.52}$$

Next, with $r = \dfrac{p}{1+e\cos\theta}$ the position and velocity in the PQW (perifocal) frame are [37]:

$$\mathbf{r}_{\text{pqw}} = \begin{bmatrix} r\cos\theta \\ r\sin\theta \\ 0 \end{bmatrix}, \qquad v_r = \frac{\mu}{h_a} e\sin\theta, \quad v_\theta = \frac{\mu}{h_a}(1+e\cos\theta), \quad \mathbf{v}_{\text{pqw}} = \begin{bmatrix} v_r\cos\theta - v_\theta\sin\theta \\ v_r\sin\theta + v_\theta\cos\theta \\ 0 \end{bmatrix}. \tag{6.53}$$

Finally, the state in the perifocal frame needs to be transformed to the ECI frame. This can be done with the following rotation matrix [37]:

$$R_{\text{PQW}\to I} = \begin{bmatrix} \cos\Omega\,\cos\omega - \sin\Omega\,\sin\omega\,\cos i & -\cos\Omega\,\sin\omega - \sin\Omega\,\cos\omega\,\cos i & \sin\Omega\,\sin i \\ \sin\Omega\,\cos\omega + \cos\Omega\,\sin\omega\,\cos i & -\sin\Omega\,\sin\omega + \cos\Omega\,\cos\omega\,\cos i & -\cos\Omega\,\sin i \\ \sin\omega\,\sin i & \cos\omega\,\sin i & \cos i \end{bmatrix} \tag{6.54}$$

In this matrix, Ω is the right ascension of the ascending node, i the inclination, and ω the argument of perigee. This matrix is equivalent to the rotation matrix $R_{\text{PQW}\to I} = [\,\hat{\mathbf{e}} \;\; \hat{\mathbf{p}} \;\; \hat{\mathbf{h}_a}\,]$. Here, the eccentricity direction, $\hat{\mathbf{e}} = \mathbf{e}/e$, is the same direction as the PQW x-axis, as can be seen in figure 5.5. Then, the angular momentum direction, $\hat{\mathbf{h}_a} = \mathbf{h}_a/h_a$, is the PQW z-axis. Lastly, the cross product between these vectors gives the last axis, $\hat{\mathbf{p}} = \hat{\mathbf{h}_a} \times \hat{\mathbf{e}}$.

The pseudocode of the iterative process of finding the eccentric anomaly and computing the position and velocity in the perifocal frame can be seen below.

Algorithm 3 True–anomaly and state computation along the coast

Require: time grid $\{t_i\}_{i=0}^{N}$, $M_{a,0}$, n, e, h_a, μ
for $i = 0$ to N **do**
  $M_{a,i} \leftarrow M_{a,0} + n\,(t_i - t_0)$
  $E \leftarrow \pi$ ▷ initial guess
  repeat
    $E_{\text{new}} \leftarrow M_{a,i} + e\sin E$
    $\delta \leftarrow |E_{\text{new}} - E|$
    $E \leftarrow E_{\text{new}}$
  until $\delta < \varepsilon$ **or** iterations exceed max
  $\theta_i \leftarrow 2\arctan\!\Big(\sqrt{\tfrac{1+e}{1-e}}\tan\tfrac{E}{2}\Big)$
  $r_i \leftarrow \dfrac{h_a^2/\mu}{1+e\cos\theta_i}$
  $v_r \leftarrow \dfrac{\mu}{h_a}\,e\sin\theta_i; \quad v_\theta \leftarrow \dfrac{\mu}{h_a}\,(1+e\cos\theta_i)$
  $\mathbf{r}_{\text{pqw},i} \leftarrow [\,r_i\cos\theta_i,\; r_i\sin\theta_i,\; 0\,]^{\mathsf{T}}$
  $\mathbf{v}_{\text{pqw},i} \leftarrow [\,v_r\cos\theta_i - v_\theta\sin\theta_i,\; v_r\sin\theta_i + v_\theta\cos\theta_i,\; 0\,]^{\mathsf{T}}$
end for

Here, N is the number of points in the coast. This pseudocode gives a representation of how, in MATLAB, the closed-form solution is computed for the state during the coast, given the initial conditions and the coast duration t_f. The computed state and costate at the end of the coast need to satisfy $H_0(t_f) = 0$ in order to have an optimal coast. How the full state and co-state are propagated can be found in Appendix C, here the MATLAB code can be examined.

6.4.3. Takeaway of Vacuum Ascent Optimality

This section developed the analytical guidance solution for the vacuum phase. For the burn arc, the objective was defined, and the state and costate equations were derived using a linear gravity approximation and non-dimensional equations of motion. By using the thrust integrals presented in equation 6.43, a closed-form solution can be obtained. For the coast arc, the method to determine the optimal coasting time was explained, using the part of the Hamiltonian that is independent of thrust, as shown in equation 6.46. Furthermore, the state and costate propagation was described using Keplerian motion. The algorithms for the propagation of the state and costate across both the burn and coast arcs were also presented. The main goal of this section was to demonstrate that, with the presented methods, closed-form solutions can be achieved for both the burn and coast arcs. How these two parts are combined into one complete trajectory is explained in Chapter 7.

6.5. Boundary Conditions

Logically, the trajectory is bounded by initial and final conditions. First of all, the initial position of the booster is located at the launch site where the trajectory simulation starts. More specifically, the simulation is assumed to start at an altitude of 100 meters above the launch site. The rocket first flies straight up to account for tower clearance. The velocity and mass at the end of this short vertical segment are then used as the initial conditions for the rest of the ascent and guidance. This means that the initial conditions of the state are fixed. Consequently, the initial conditions of the co-state are free and therefore yet undetermined [61].

At the end of the trajectory, the booster must meet the prescribed terminal conditions on its state. These constraints ensure a correct insertion of the HGV into its re-entry corridor, as explained in chapter 2.2. Consequently, the boost trajectory is designed and optimized subject to these fixed terminal conditions. As described in equation 6.9, the trajectory also has specified transversality conditions. These follow from the defined terminal conditions and the objective function. Furthermore, when there are $k \leq 6$ scalar terminal conditions, this means there are $6 - k$ transversality conditions [62].

At the end of the trajectory, the HGV needs to be inserted. This means that it needs to have a certain specified altitude and flightpath angle. Furthermore, the HGV also needs to be inserted into a certain desired direction. Lastly, depending on the objective, the HGV needs to be inserted with either a maximum velocity (to ensure maximum range) or a specified velocity.

6.5.1. Reduced Transversality Conditions Approach

Defining the terminal and transversality conditions properly is a very important task in solving the optimization problem. This definition can have a huge influence on the outcome and convergence of the optimizer. The approach by B. Pan [62] provides a robust method for this. The paper addresses the long-standing difficulty of defining the transversality conditions in indirect optimal guidance. More specifically, in defining the reduced transversality conditions (RTCs). The reduced transversality conditions are the transversality conditions (as defined in equation 6.9), but then with the unknown constant multiplier υ removed. This unknown constant multiplier can be removed by algebraically redefining the transversality conditions and thereby eliminating them from the equation. Removing the unknown constant multiplier simplifies the optimization problem, since there are fewer variables that need to be solved.

The key insight in B. Pan's approach is that, for launch insertion and orbital-transfer problems, the terminal constraints almost always fix a subset of the six classical orbital elements $(a, e, i, \Omega, \omega, \nu)$ and leave the others free. Writing the terminal requirements in orbital elements cleanly separates constrained and free variables. This is something that is messy or absent when defining the terminal requirements using Cartesian coordinates. Pan defines the RTCs systematically by determining which final orbital elements are free. Every free orbital element yields one RTC. RTCs are analogous to the usual transversality conditions for the same problem when the terminal constraints are expressed with orbital elements rather than Cartesian states [62].

During this study, it was found that defining the terminal requirements in Cartesian coordinates creates the risk of constraint redundancy. This means that removing a constraint would not change the solution of the optimization. Having unnecessary constraints also greatly affects the convergence of the

optimization problem. Pan's orbital-element method avoids this by providing a pre-compiled, independent set of RTCs, one per free element. This reduces the problem complexity and improves numerical robustness and speed by preventing unnecessary or ill-defined constraints. It was found that when the approach by B. Pan [62] was adopted in the optimization problem, it greatly improved the convergence and speed of the optimization compared to using Cartesian coordinates.

6.5.2. Definition of terminal requirements

As explained before, the trajectory needs to meet specified terminal conditions. One of these conditions is that it needs to fly in a certain desired direction. Because the terminal conditions are expressed in orbital elements, the terminal direction of the trajectory is specified by the inclination i, i.e., the angle between the final orbital plane and the Earth's equatorial plane. In addition to the orbital-plane direction, a terminal constraint on the following parameters needs to be imposed: (i) the altitude, expressed via the position magnitude $\|\mathbf{r}_f\|$; (ii) the terminal flight–path angle γ_f; and (iii) the terminal speed $\|\mathbf{V}_f\|$.

Previously, the optimization problem and objective have been defined, resulting in the Hamiltonian given in equation 6.16. As mentioned before, for the endo-atmospheric initial stage boost, the Hamiltonian is the same for different terminal cost objectives, since only the running cost is included in the Hamiltonian definition. Therefore, with the first endo-atmospheric burn arc, the optimal law for body-axis rotation is similar if the problem is posed to maximize terminal orbital energy with a fixed final time or to minimize flight time with a free final time [40]. However, this is not the case for the terminal transversality conditions, since the terminal cost objective is present in the transversality condition (equation 6.9). Therefore, the defined terminal requirements will influence the burn arc and coast arc of the analytical vacuum optimal ascent solution.

Two possible approaches to defining the terminal requirements of the analytical vacuum optimal ascent solution are investigated in this study. The first approach is by defining the terminal requirements, such that the objective is to minimize the flight time. In this case, the terminal velocity can still be maximized (in order to maximize the range) by sequentially adapting the terminal desired velocity until no higher feasible velocity is possible anymore. This approach is similar to the one by P. Lu [40], where sequentially parameters are adjusted until the desired orbital energy is met. The second approach is by defining the requirements such that the objective is to maximize the orbital energy. Both approaches yield the same result, as can be seen in the verification in chapter 9. However, they use other methods to arrive at these results. The defined terminal conditions and reduced transversality conditions of both approaches can be found below.

To start, the first approach has four terminal conditions: the terminal flightpath angle γ, the terminal inclination i, the terminal magnitude of the position vector $|r|$, and the terminal magnitude of the velocity vector $|v|$. The terminal conditions are [62]:

$$\mathbf{r}_f^\top \mathbf{r}_f - r_f^{*2} \tag{6.55}$$

$$\mathbf{V}_f^\top \mathbf{V}_f - V_f^{*2} \tag{6.56}$$

$$\mathbf{1}_z^\top (\mathbf{r}_f \times \mathbf{V}_f) - \|\mathbf{r}_f \times \mathbf{V}_f\| \cos i^* \tag{6.57}$$

$$\mathbf{r}_f^\top \mathbf{V}_f - r_f^* V_f^* \sin \gamma_f^* \tag{6.58}$$

This gives four scalar constraints that prescribe the a, i, e, and θ [62]. This leaves the ω and Ω free. Based on these free orbital parameters, the RTCs are determined. The last 2 scalar constraints of the reduced transversality conditions are [62]:

$$\Big(\mathbf{p}_{r_f} \times \mathbf{r}_f + \mathbf{p}_{V_f} \times \mathbf{V}_f\Big)^\top \mathbf{h}_{a,f} \tag{6.59}$$

$$\left(\mathbf{p}_{r_f} \times \mathbf{r}_f + \mathbf{p}_{V_f} \times \mathbf{V}_f\right)^\top \mathbf{1}_z \tag{6.60}$$

The second approach has three terminal conditions: the terminal flightpath angle γ, the terminal inclination i, and the terminal magnitude of the position vector $|r|$. Furthermore, the terminal cost of this approach is to maximize orbital energy. The given terminal conditions and transversality conditions are [40]:

$$\frac{1}{2}\,\mathbf{r}_f^T\mathbf{r}_f \;-\; \frac{1}{2}\,r_f^{*2} = 0 \tag{6.61}$$

$$\mathbf{1}_N^T(\mathbf{r}_f \times \mathbf{V}_f) \;-\; \|\mathbf{r}_f \times \mathbf{V}_f\| \cos i^* = 0 \tag{6.62}$$

$$\mathbf{r}_f^T\mathbf{V}_f \;-\; r_f\, V_f \sin\gamma_f^* = 0 \tag{6.63}$$

$$\left(\mathbf{V}_f^T\mathbf{p}_{\mathbf{r}_f}\right) r_f^2 \;-\; \left(\mathbf{r}_f^T\mathbf{p}_{\mathbf{V}_f}\right) V_f^2 \;+\; \left(\mathbf{r}_f^T\mathbf{V}_f\right)\left(V_f^2 - \mathbf{r}_f^T\mathbf{p}_{\mathbf{r}_f}\right) = 0 \tag{6.64}$$

$$\mathbf{V}_f^T\mathbf{p}_{\mathbf{V}_f} \;-\; V_f^2 = 0 \tag{6.65}$$

$$\left(\mathbf{h}_{a,f}^T\mathbf{p}_{\mathbf{r}_f}\right)\left[\mathbf{h}_{a,f}^T(\mathbf{r}_f \times \mathbf{1}_N)\right] \;+\; \left(\mathbf{h}_{a,f}^T\mathbf{p}_{\mathbf{V}_f}\right)\left[\mathbf{h}_{a,f}^T(\mathbf{V}_f \times \mathbf{1}_N)\right] = 0 \tag{6.66}$$

Ultimately, approach one is chosen for the final optimization program developed in MATLAB. Following the requirements, it is necessary for the ascent trajectory to maximize its range. However, on some occasions, it might be desired to let the booster fly to a certain specified terminal state. In this case, the first approach can be used, since it allows for defining a specified terminal speed. Then it is not necessary to sequentially find the maximum possible velocity. With this approach, one optimization code package is able to solve both simulation scenarios.

6.5.3. Terminal Hamiltonian

Lastly, the optimization problem of the analytical vacuum optimal ascent solution is a free final time problem. As described in section 6.4, the coasting time and burn time can have variable length. Of course, in the case of range maximization, it is expected that the stages burn all their propellant. The terminal condition for the Hamiltonian (equation 6.10) is used to solve the optimal burn and coasting times. For the objective function described previously, this yields the following relation that makes the final constraint $H(\tau_F) > 0$ [40]. This condition is numerically difficult to enforce because it can have a great mismatch in numerical scaling. For minimum–time problems, the costate dynamics are homogeneous, so the problem is not influenced by any positive scaling of the costate:

$$\mathbf{p}(\tau) \;\mapsto\; \kappa\,\mathbf{p}(\tau), \qquad \kappa > 0,$$

which leaves the optimal control and the state trajectory unchanged. Consequently, once the terminal constraints and reduced transversality conditions are satisfied, one can always rescale $\mathbf{p}$ so that the endpoint Hamiltonian takes its required value [40]. This makes the condition of equation 6.10 unnecessary and can harm the convergence of the optimizer. A robust replacement is to fix the scale of the costate by a normalization constraint [43]:

$$\|\mathbf{p}(\tau_f)\| = 1,$$

Therefore, enforcing that the magnitude of the costate vector is constant.

6.6. Engine Throttle

It has been shown by A. Calise that intermediate thrust levels are not feasible both within the atmosphere and in a vacuum for virtually every circumstance that might be encountered in any practical application [63]. In the context of rocket ascent optimization, these intermediate thrust levels correspond to throttle settings between maximum and minimum thrust. Calise proved that it does not exist for any physically valid trajectory, since it does not satisfy the first-order necessary conditions for optimality following the Hamiltonian. For this conclusion, it is assumed that the vehicle always flies at a

positive angle of attack and that the only control variables are the throttle and the body axis orientation [63].

As a result, the optimal thrust profile is of bang-bang type, meaning the thrust switches between its maximum and minimum values [63]. In practical rocket systems, however, true bang-bang control is not implemented because of hardware and flight constraints. Engine throttling cannot change instantaneously without causing structural loads or instability, and many engines have limited throttle range or restart capability. Once the main engines are shut down, restarting them is generally not desirable. Therefore, the available propellant is burned completely during the initial burn phase, after which the vehicle follows a coast phase whose duration is determined as part of the optimization [60].

Because of this reason, the throttle is thus mainly used as a constraint-enforcing variable, to keep the dynamic pressure below a prescribed limit,

$$q_{\text{dyn}} \leq q_{\text{max}}.$$

In the dynamic pressure constraint $S = q - q_{\max} \leq 0$, the body-axis direction I_b influences how q changes over time. By expressing $q = \frac{1}{2}\rho(r)V_r^2$, a first-order approximation can be used to predict $q(t+\delta)$ and derive a simple closed-loop throttle law that limits how quickly q increases. This method is well-suited for onboard guidance since it only requires local information and can be tuned to avoid throttle oscillations while keeping $q \leq q_{\max}$.

The engine throttle $\eta(t)$ can be treated as a prescribed time-varying command primarily used to enforce the dynamic pressure constraint during the endo-atmospheric phase. In this case, the throttle is determined externally and therefore not treated as a control variable within the optimization. Outside the atmosphere, the dynamic pressure is negligible, and the throttle scheduling for q enforcement is not required.

When deciding how much throttle to use at the current time t, the goal is to ensure that the dynamic pressure q does not exceed its maximum allowable value $q_{\max}$ in the next small time step δ. By using the following relation, the throttle can be computed such that the dynamic pressure q does not exceed its maximum allowable value [43].

$$\eta(t) \leq \frac{q_{\max} - q(t) - b_q\delta}{a_q\delta} \;\triangleq\; \eta_q$$

where a_q and b_q are coefficients that describe how q changes with time and throttle.

The actual throttle command is then determined according to [43]:

$$\eta = \begin{cases} \eta_{\text{prb}}, & \text{if } \eta_q > \eta_{\text{prb}} \\ \eta_q, & \text{if } \eta_{\min} \leq \eta_q \leq \eta_{\text{prb}} \\ \eta_{\min}, & \text{if } \eta_q < \eta_{\min} \end{cases}$$

where η_{prb} is usually the maximum allowable throttle prescribed and $\eta_{\min}$ is the minimum allowable throttle of the boost phase. For this study η_{prb} is set at 1 and $\eta_{\min}$ at 0. In practice, if the dynamic pressure is well below its limit, the throttle remains at the nominal value η_{prb}. When the pressure approaches $q_{\max}$, the throttle is reduced to η_q. If the pressure limit still cannot be satisfied within the allowed throttle range, the engine operates at its minimum setting $\eta_{\min}$.

The coefficients in the above equation are derived in the following way:

$$a_q = \rho\left(V_r \cdot \left(\tfrac{T}{m} I_b\right)\right),$$

$$b_q = \frac{1}{2r}\,\rho_r\, V_r^2\left(r \cdot V\right) \;+\; \rho\left(V_r \cdot (A + B + C + D)\right),$$

Here, the building blocks (A, B, C, D) are defined the like:

$$A = -\frac{A_{\text{drag}}}{m} I_b,$$
$$B = -\frac{1}{r^3} r,$$
$$C = \frac{N_{\text{drag}}}{m} I_n,$$
$$D = -\boldsymbol{\omega}_E \times V,$$

where A_{drag} and N_{drag} are the aerodynamic froces, m is the mass, I_b and I_n are the body and normal unit vectors, r is the position vector and $\boldsymbol{\omega}_E$ the Earth rotation vector. Furthermore, the "jitter" parameter δ also needed to be tuned in this study. When different booster models are used, it is possible that this parameter needs to be tuned again. However, for the booster model used in this study, the jitter parameter was found using trial and error. This booster model will be presented in chapter 8. A value between $0.8 - 1.2$ gave good results. A higher δ damps the throttle too extensively, while a lower value produces very extreme and oscillating jitter behavior. The extreme oscillating, bang–bang like, throttle can, in principle, be more efficient. However, rapid or extreme throttle oscillations are generally impossible and undesirable for real engines due to hardware limits and structural concerns.

The throttle thus serves mainly as an atmospheric constraint enforcement mechanism; it is active when q approaches $q_{\max}$ and is unnecessary outside the atmosphere where dynamic pressure is negligible. However, the bang-bang characteristic of the optimal thrust profile also allows for an early engine cut-off when this is beneficial for the mission objective, such as in cases where the goal is not to maximize the range. This is implemented in the optimization model, as previously discussed.

Because the first stage is modeled as a fixed-final-time burn, reducing the throttle lowers the mass flow. For the same scheduled burn duration, this means less propellant is consumed when the engine is throttled, which is not desired because the stage should deplete its propellant. To avoid leaving propellant unused, the MATLAB model produced in this study applies a time correction whenever the throttle is reduced. The burn time is lengthened according to the reduced mass flow so that the total propellant consumed matches the amount that would have been expelled at nominal throttle. In this way, the first stage always burns its available propellant to depletion even when throttling is commanded.

7

Numerical Solution Process

Since answering the main research question requires finding an optimal ascent trajectory using an indirect method, a key focus of this research is developing a method capable of determining an optimal guidance solution for the rocket. Analytical solutions to the PMP indirect method are only feasible for highly simplified problems that allow for closed-form expressions. Simulations showed that incorporating aerodynamic effects significantly increases the system's complexity, making a full analytical solution unfeasible. As identified in the ranked priorities of Chapter 4, modeling aerodynamic forces is essential for an accurate ascent trajectory simulation. Therefore, to provide a meaningful answer to the research question, this chapter provides the numerical process of finding an optimal solution. As explained in the methodology trade-off in the appendix, a hybrid analytical-numerical approach is used. This hybrid method is based on the information provided in chapter 6, and the numerical methods explained in this chapter.

One of the biggest challenges in this study is making sure the developed MATLAB optimization code is able to converge to a solution. This chapter will provide an approach to how this is realized. The optimization consists of three parts. First, there is the vacuum trajectory. Secondly, there is the endoamtospheric optimization. And last, there is the process of tying these two optimizations together. Therefore, this chapter is divided into three sections that explain these aspects of the optimization.

7.1. Computing the vacuum solution

In order to solve the entire optimization problem, first a vacuum solution needs to be found. This trajectory consists of two burn and coast arcs. Chapter 6 explained how these arcs are propagated. As explained in the previous chapter, the initial condition for the state is set, and the goal is to fly to a specific final condition. Section 6.5 defines the terminal and transversality conditions that the trajectory must satisfy to optimally reach a specified inclination, flight-path angle, altitude, and velocity. The final velocity is then adjusted sequentially to find the maximum range. In this optimization, the optimal coasting times and burn times need to be found. The main goal is to maximize the range of the booster. Here, it can be expected that all propellant is depleted.

Therefore, as a basic sanity check, the total propellant burn time should be consistent with the stage flight time. If the target point does not require the maximum feasible velocity, some propellant may remain. So the second–stage burn time is treated as a control variable, and the cutoff time is chosen by the optimizer. It should be noted that in this specific case, the first stage is assumed to burn to depletion and is therefore not a control variable. As will be explained in section 7.3, the transition from the atmospheric to the vacuum model is triggered once a predefined altitude is reached, beyond which aerodynamic forces can be reasonably neglected. This transition occurs during the coast phase between the two stages. Therefore, the second stage of the trajectory is assumed to fly in a vacuum. If the first–stage burn time were free, the optimizer might select an unrealistically short first–stage burn to switch too quickly to the vacuum phase. To prevent this modeling bias, the first stage is fixed to burn to depletion, and only the second–stage burn time is treated as a control variable.

The trajectory consists of four different segments. Each of these segments is propagated from an initial state and costate given by the optimizer. The goal is to satisfy the terminal and transversality conditions, together with the optimality conditions for the coast time and the final time, so that the solution is optimal. However, next to this the optimizer needs to produce one continuous trajectory. Therefore, the endpoint of one segment of the trajectory needs to be equal to the starting point of the next segment. To ensure this, continuity conditions are used. These continuity conditions look as follows:

$$\mathbf{y}^{-}_{\mathrm{coast}_1} - \mathbf{y}^{+}_{\mathrm{coast}_1} = 0,$$
$$\mathbf{y}^{-}_{\mathrm{stage}_2} - \mathbf{y}^{+}_{\mathrm{stage}_2} = 0,$$
$$\mathbf{y}^{-}_{\mathrm{coast}_2} - \mathbf{y}^{+}_{\mathrm{coast}_2} = 0.$$

Adding these additional requirements ties the whole optimization together into one continuous solution. The following figure schematically describes these continuity conditions.

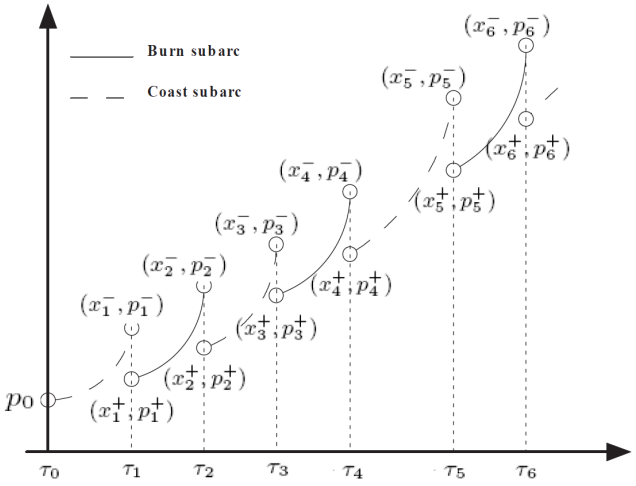


Figure 7.1: Multiple Shooting Formulation for N-burn Problem [59].

Now, the optimization problem consists of six unknowns for the initial conditions of the costate values p_V and p_R, two unknowns for the coasting times, one unknown for the final burn time, and 12 unknowns for each of the three nodes from which the trajectory segments are propagated. This gives a total of 45 unknowns. Then, there are four terminal conditions, two transversality conditions, two conditions for the optimal coast duration (equation 6.46), one condition for the terminal Hamiltonian (equation 6.5.3), and 12 continuity conditions for each of the nodes. This makes the dimensions of the zero-finding problem 45.

This approach is similar to the one by P. Lu et. al [40]. However, that trajectory consists of two burn arcs and only one coast arc. The optimization method in this study has the possibility of adding a final cost. This increases the number of control variables and the dimensions of the zero-finding problem.

In order to optimize the trajectory and solve all the unknown variables while also satisfying the required conditions, the `fmincon` optimizer is used in MATLAB. This optimizer is used because of the possibility of defining both an objective function and a constraint function. Compared to other fast solvers such as `fsolve`, `patternsearch`, and `lsqnonlin`, the `fmincon` solver is better suited for this problem because it can handle both equality and inequality constraints directly. In contrast, `fsolve` and `lsqnonlin` are designed only for unconstrained root-finding problems. This gives `fmincon` a great advantage since the optimization problem can be described more effectively. The `patternsearch` method, although robust for discontinuous problems, is derivative-free and therefore significantly slower and less efficient for problems with many control variables, like the 45 unknowns in the problem of this study. Similarly,

`lsqnonlin` minimizes the sum of squared residuals and is less flexible for problems with complex nonlinear constraints. In contrast, `fmincon` is the most effective choice for solving the constrained optimal trajectory problem compared to the previously mentioned optimizers.

In this study, the continuity constraints and the four terminal conditions are enforced as constraint functions. The objective function is the residuals of the transversality conditions (seen in equations 6.59 and 6.60), the two optimal–coast–time conditions, and the terminal Hamiltonian condition. This setup gave the best convergence and the most accurate results. The `fmincon` solver was configured with tight convergence tolerances. The Function Tolerance is set at $\tau_{fun} = 10^{-8}$, which makes the optimization stop when the change in the objective function becomes smaller than the tolerance. The Optimality Tolerance is given a value of $\tau_{opt} = 10^{-9}$. This makes the optimization stop when the gradient of the optimality falls below the specified threshold. Further, the ConstraintTolerance is $\tau_{con} = 10^{-5}$. This defines the maximum allowed residual for all constraints. For the launch scenario discussed in Chapter 8.3, the residual array of the zero-finding problem converged to 3.93×10^{-7}. All terminal and continuity constraints were satisfied with errors smaller than 10^{-5}. These results show that the optimizer reached a stable and consistent solution.

The optimization starts by providing the optimizer with the 45 initial guesses for the unknowns. The given initial guess for the burn time of the second stage is its total maximum burn time. For the coast durations, a guess of 50 seconds is given. The multiple shooting optimization model showed not to be sensitive to the initial guess of the burn time, coast duration, and the initial costates. Last, an initial guess needs to be given for the 3 nodes. Unlike the other initialization choices, the guesses for the shooting nodes are crucial for convergence. For this initialization, the method of Lu et al. [40] cannot be used. This formulation employs a different number of trajectory arcs and targets orbital insertion of the final stage. This mismatch in phase structure and terminal objective makes their guess incompatible with the present problem. The following section will explain how these initial nodes are guessed.

Shooting nodes initial guess

The nodes are guessed by approximating the path as a ballistic trajectory, starting from an impulsive shot. From the starting location, the initial position $\mathbf{r}_0$ is known. The initial speed is approximated as the impulsive shot equal to the sum of the stage ΔV's from the rocket equation,

$$\Delta V_1 = I_{\mathrm{sp},1}\, g_0 \ln\frac{m_{0,1}}{m_{0,1} - m_{p,1}}, \qquad \Delta V_2 = I_{\mathrm{sp},2}\, g_0 \ln\frac{m_{0,1}}{m_{0,1} - m_{p,2}}, \qquad v_0 \approx \Delta V_1 + \Delta V_2.$$

With this v_0 at $r_0 = \|\mathbf{r}_0\|$, the specific orbital energy and semi-major axis follow from [35]:

$$\mathcal{E} \;=\; \tfrac{1}{2}v_0^2 - \frac{\mu}{r_0}, \qquad a \;=\; -\frac{\mu}{2\mathcal{E}}.$$

All points are assumed to lie in the target orbital plane, which is the final required inclination. From the terminal conditions, the final radius r_f and the terminal flight-path angle γ_f are known. Since the orbital energy is constant, the terminal speed can be computed.

$$v_f \;=\; \sqrt{2\mathcal{E} + \frac{2\mu}{r_f}}\,.$$

The specific angular momentum of the trajectory is then [35]:

$$h_a \;=\; r_f\, v_f \cos\gamma_f,$$

Furthermore, the semi-latus rectum and eccentricity are computed in the following way.

$$p \;=\; \frac{h_a^2}{\mu}, \qquad e \;=\; \sqrt{1 - \frac{p}{a}}.$$

And the apogee and perigee radii follow from:

$$r_a \;=\; \frac{p}{1-e}, \qquad r_p \;=\; \frac{p}{1+e}.$$

The third and last shooting node is the start of the final coast. For the initial guess, this node is assumed to be located at the apogee of the trajectory. Here, true anomaly $\theta_3 = \pi$, radius $r_3 = r_a$. The speed at the apogee follows from the vis-viva equation [35]:

$$v_3 \;=\; \sqrt{\frac{2\mu}{r_3} - \frac{\mu}{a}}.$$

Then the guess for the first two nodes needs to be determined. The true anomaly of the initial location is computed with the following relation [35].

$$\cos\theta_0 \;=\; \frac{p/r_0 - 1}{e}$$

Nodes 1 and 2 are placed one-third and two-thirds of the way in true anomaly between θ_0 to π:

$$\theta_1 \;=\; \theta_0 + \tfrac{1}{3}(\pi - \theta_0), \qquad \theta_2 \;=\; \theta_0 + \tfrac{2}{3}(\pi - \theta_0).$$

At each θ_i ($i = 1, 2$), the radius and speed are

$$r_i \;=\; \frac{p}{1 + e\cos\theta_i}, \qquad v_i \;=\; \sqrt{\frac{2\mu}{r_i} - \frac{\mu}{a}}.$$

The corresponding position and velocity in the perifocal frame and ECI frame are constructed with the relations that are elaborated in chapter 6.4.2.

Then the initial guesses for the costates need to be made for every shooting node. The velocity costates are set equal to the unit vectors of the velocity at each of the nodes.

$$\mathbf{p}_{V,i} \;=\; \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} \quad (i = 1, 2, 3),$$

Lastly, the position costates are initialized as small constants, like:

$$\mathbf{p}_{r,i} \;=\; 10^{-3} \cdot [1,\, 1,\, 1]^T.$$

This gives three nodes $(\mathbf{r}_i, \mathbf{v}_i, \mathbf{p}_{V,i}, \mathbf{p}_{R,i})$ that lie on the same conic defined by $\mathcal{E}$, a, p, and e, with Node 3 at apogee and Nodes 1–2 evenly spaced in true anomaly between the start geometry and apogee.

7.2. Computing the atmospheric ascent solution

The goal of the atmospheric part of the trajectory is to optimally fly to a certain condition from where the remaining vacuum part of the trajectory is optimized. This condition consists of specified values of the costate. How this point is determined will be explained later in the chapter. In this study, three different approaches are investigated to compute a solution of the atmospheric trajectory.

First of all, for all these different approaches the equations of motion (equations 5.9 and 5.10) are first made non-dimensional. This improves convergence of the optimization because it scales the variables to comparable magnitudes. The non-dimensional equations of motion are [64]:

$$\begin{cases} r' = V, \\ V' = -\dfrac{1}{r^3}\, r \;+\; (T - A)\, \mathbf{1}_b \;+\; N\, \mathbf{1}_n. \end{cases}$$

So the variables are nondimensionalized to improve conditioning. Distances are scaled by the Earth radius r_e: $\tilde{\mathbf{r}} = \mathbf{r}/r_e$. Velocities are made non-dimensional using the characteristic speed $v_* = \sqrt{r_e g_0}$: $\tilde{\mathbf{V}} = \mathbf{V}/v_*$. Time is scaled by $t_* = \sqrt{r_e/g_0}$: $\tilde{t} = t/t_*$. Forces are scaled by the instantaneous weight $m(t) \cdot g_0$: $\tilde{\mathbf{F}} = \mathbf{F}/(m(t)g_0)$ [64]. The mass flow rate is nondimensionalized with the same time scale: $\tilde{\dot{m}} = \dot{m} \cdot \sqrt{r_e/g_0}$ [40].

The problem is posed as a boundary-value problem: the initial state is known and the terminal requirements define the boundary conditions. The initial costates are unknown and are treated as optimization variables. For a trial set of costates, the trajectory is propagated and, at each time step, the optimal body-axis direction is computed using the primer-vector theory (see Section 6.3.3). The optimization then needs to find the initial costates so the terminal conditions are satisfied. How this is done will be discussed, following the three investigated approaches.

Approach 1

For the first approach of finding the optimal initial costate values, the developed MATLAB model used a collocation method. At times, this produced singular Jacobian matrices, which made the problem unsolvable. However, this issue did not occur with the shooting method. The shooting method is implemented with the `fmincon` solver. The `fmincon` solver allows constraint functions, which are used to enforce the boundary conditions. This makes it a very effective solver to use for the optimization problem. The decision variables are the initial costates. For each guess, the optimal body-axis direction is computed (via the primer-vector logic), and the state is propagated by integrating the equations of motion with an `ODE` solver. The optimizer repeats this process until the terminal conditions are met without violating the constraints. A drawback is that the method is very sensitive to the initial guess supplied to `fmincon`, which can make convergence to a feasible solution difficult.

To address the convergence issue, a homotopy approach is applied. In this approach, a simpler problem that is easy to solve is gradually transformed into the full problem of interest. For example, a vacuum solution is obtained first; then atmospheric effects are introduced step by step. This strategy is useful for problems with aerodynamics, where strong nonlinearities make the optimal solution difficult to find.

A homotopy parameter $\lambda \in [0, 1]$ is introduced to transition from the simplified problem ($\lambda = 0$) to the full problem ($\lambda = 1$) by adding complexity in small increments [65]. In this procedure, the solution obtained at the homotopy step k is used as the initial guess for step $k + 1$. In this study, four homotopy runs are performed in sequence: (1) gradually introduce aerodynamics, (2) add Earth rotation, (3) add control constraints, and (4) add path constraints. For this approach, the aerodynamic homotopy was implemented using ten equally sized increments, while the remaining homotopy loops each used five increments. This staged procedure initially improved convergence for the atmospheric case; however, as additional homotopy stages were included, the method again encountered difficulties converging to a feasible solution.

Approach 2

A second solution strategy adopts a multiple shooting formulation inspired by Lu et al. [40]. Multiple shooting breaks the trajectory into short sections, so errors from a rough initial guess don't grow over long times. The solver can adjust each section separately, which reduces sensitivity and makes the whole problem easier to solve. The time interval $[t_0, t_f]$ is split into N sub-intervals of equal length $\Delta t = (t_f - t_0)/N$ with nodes $t_k = t_0 + k\,\Delta t$, $k = 0, \ldots, N$. At each node an unknown vector $y_k \in \mathbb{R}^{2n}$ is introduced that stacks state and costate, $y_k = \left(x_k^T\ p_k^T\right)^T$. Treating $\{y_k\}_{k=0}^{N}$ as independent variables breaks the long propagation into short segments [40].

The differential equations $y' = f(t, y)$ are approximated over each subinterval by a second–order central difference at the mid–point $t_{k-1/2} = t_{k-1} + \Delta t/2$ [66]:

$$\frac{1}{\Delta t}\left(y_k - y_{k-1}\right) \;=\; f\left(t_{k-1/2},\ \frac{y_k + y_{k-1}}{2}\right).$$

Rearranging defines the per–segment residual

$$E_k(y_k, y_{k-1}) \;\triangleq\; y_k - y_{k-1} \;-\; \Delta t \cdot\, f\left(t_{k-1/2},\ \frac{y_k + y_{k-1}}{2}\right) \;=\; 0, \qquad k = 1, \ldots, M.$$

For instance, with the given equations of motion and the finite difference formulation, the residuals of the state will be computed like:

$$E_{k,c}^{(r)} = \left(r_{k+1,c} - r_{k-1,c}\right) - 2\Delta t \cdot\, v_{k,c} \tag{7.1}$$

$$E_{k,c}^{(V)} = \left(r_{k+1,c} - 2r_{k,c} + r_{k-1,c}\right) - \Delta t^2\left[-\,q_k\, r_{k,c} + \left(p_k - A_{\mathsf{acc},k}\right) I_{b,k,c} + N_{\mathsf{acc},k}\, I_{n,k,c}\right] \tag{7.2}$$

Here Δt is the node spacing, $r_{k,c}$ and $v_{k,c}$ are the position and velocity components at node k, q_k is the gravity factor ($q_k = 1/\|r_k\|^3$ in the nondimensional model), p_k the (dimensionless) thrust magnitude, $A_{\text{acc},k}$ and $N_{\text{acc},k}$ the axial and normal aerodynamic accelerations, and $I_{b,k,c}$, $I_{n,k,c}$ the c-components of the body-axis and normal-direction unit vectors at node k. The finite difference method (FDM) helps to further simplify the equations of motion for the optimizer.

Together with the boundary conditions at the ends, $B_0(y_0) = 0$ and $B_N(y_N) = 0$, all constraints are collected into a single nonlinear system.

$$E(Y) = 0, \qquad Y \;=\; \left(y_0^T \; y_1^T \; \cdots \; y_N^T\right)^T \in \mathbb{R}^{2n(N+1)}.$$

The goal is to drive the residual array $E(Y)$ to zero. This then enforces both the dynamics via the finite difference relations and the boundary conditions.

First, the algebraic system $E(Y) = 0$ was solved with a modified Newton iteration. The control variables are all the $2n(N+1)$ points in the Y array. At iteration j, the search direction Δx_j solves the linearized system [67]:

$$\Delta x_j \;=\; -\left[\frac{\partial E(Y_{j-1})}{\partial Y}\right]^{-1} \cdot E(Y_{j-1}).$$

A line–search parameter $\sigma_j \in (0, 1]$ is then chosen to guarantee a decrease of the residual norm, and the unknowns are updated as [67]:

$$Y_j \;=\; Y_{j-1} + \sigma_j \cdot \Delta x_j.$$

The Jacobian is computed by numerical differencing: each decision variable is perturbed slightly to observe the change in the residual vector $E(Y)$. For variable Y_j, the (i, j) entry of the Jacobian is approximated by [68]:

$$\left[\frac{\partial E(Y_{j-1})}{\partial Y}\right] \;\approx\; \frac{E_i\big(Y + \varepsilon\, e_j\big) \;-\; E_i\big(Y - \varepsilon\, e_j\big)}{2\,\varepsilon}$$

For the optimization, an initial guess is also necessary. The initial guess is here a reference trajectory. A good initial guess is very important in order to make this approach a nice continuous trajectory. The reference trajectory chosen is the first stage trajectory of the initially produced vacuum solution. From this solution, $N + 1$ evenly spaced points of the state and costate are taken. This serves as a robust initial guess.

For vacuum trajectories, the multiple–shooting scheme converged much more robustly than the first approach. However, once aerodynamic effects were included, the modified Newton iterations became too slow to be a feasible approach for the optimization. The strong nonlinearities and repeated Jacobian evaluations greatly increased the computation time. Therefore, instead of the developed Modified Newton method, the residual matrix is solved using the `fmincon` optimizer.

Using `fmincon` together with the reference trajectory and homotopy loops consistently produced converged solutions to the optimization problem. For this approach, the same homotopy pattern and number of increments were used as in the first approach. This worked better than the previous approaches. However, when the trajectory was checked for physical accuracy, a mismatch appeared between the change in momentum (mass times velocity) and the time–integral of the applied forces. The cause was small but nonzero entries in the residual vector $E(Y)$. Even with tight optimality and constraint tolerances, `fmincon` can accept a solution while some residuals have not fully gone to zero. It can be seen in equation 7.2, that when the residual is not zero, a state is picked that does not exactly match the equations of motion. These remaining errors accumulate along the trajectory and lead to an impulse mismatch. Increasing the number of increments in the homotopy loop helped mitigate this issue, although it did not completely resolve it.

Approach 3

The last approach has been investigated. Like the second method, it uses a multiple–shooting setup. The decision vector is again a stacked array Y that contains the state and costates at all shooting points. The key difference is how the dynamics are enforced: instead of forming finite–difference residuals, the method propagates the state and costate forward from each shooting point over a short segment of

length t/N, where t is the first–stage flight time and N is the number of segments. This makes the propagation similar to the first approach, but only over short, fixed time windows.

Because each segment is short and many iterations are needed in `fmincon`, a simple Euler integrator is used instead of a full `ODE` solver. This is sufficient for the small step size and is more efficient for repeated evaluations. The step update uses the costate dynamics given in 6.20, 6.21, and the non-dimensional state equations described in chapter 7.2 to compute the next values from the current ones. The algorithm below shows how this is done using the Euler integration.

Algorithm 4 Single Euler time step update

Require: current state $(\mathbf{r}, \mathbf{v}, \mathbf{p}_V, \mathbf{p}_R, m)$, control α, step Δt, parameters

$(\dot{\mathbf{r}}, \dot{\mathbf{v}}, \dot{\mathbf{p}}_V, \dot{\mathbf{p}}_R, \dot{m})$

$\mathbf{r}_{\text{next}} \leftarrow \mathbf{r} + \Delta t \cdot \dot{\mathbf{r}}$ ▷ $\dot{\mathbf{r}} = \mathbf{v}$

$\mathbf{v}_{\text{next}} \leftarrow \mathbf{v} + \Delta t \cdot \dot{\mathbf{v}}$

$\mathbf{p}_{V,\text{next}} \leftarrow \mathbf{p}_V + \Delta t \cdot \dot{\mathbf{p}}_V$

$\mathbf{p}_{r,\text{next}} \leftarrow \mathbf{p}_R + \Delta t \cdot \dot{\mathbf{p}}_R$

$m_{\text{next}} \leftarrow m + \Delta t \cdot \dot{m}$ ▷ $\dot{m} < 0$ during burn

return $(\mathbf{r}_{\text{next}}, \mathbf{v}_{\text{next}}, \mathbf{p}_{V,\text{next}}, \mathbf{p}_{R,\text{next}}, m_{\text{next}})$

Since the state is directly propagated with the equations of motion, this method is less prone to the impulse balance errors observed in the second approach. As shown earlier, the first approach, which also propagated the dynamics, did not suffer from this problem. In these equations of motion, for each time step the optimal body axis rotation is computed following the methodology presented in chapter 6.3.3.

Furthermore, this approach again applies a homotopy strategy. It starts from a reference trajectory that is the first stage path obtained from the initial vacuum solution. From this reference, N nodes are selected, and from each node the state and costate are propagated over one short segment. In the first homotopy loop, the computed trajectory matches the vacuum reference. The homotopy level is then increased step by step, and after each step the previous solution is used as the initial guess for the next optimization. The goal remains to ensure that the terminal point of the first stage trajectory satisfies the specified end conditions. However, compared to the previous approaches, this method required only a single homotopy loop for the aerodynamic effects, using ten incremental steps to achieve a stable and converged solution. This reduction in homotopy steps significantly decreased the overall computation time of the optimization.

In the multiple–shooting method, each segment is propagated independently. To make a single continuous trajectory, the end of one segment must equal the start of the next. This is enforced by continuity constraints. For two consecutive segments this gives

$$\mathbf{y}_k^+ - \mathbf{y}_{k+1}^- = 0, \qquad \mathbf{y}_{k+1}^+ - \mathbf{y}_{k+2}^- = 0,$$

where $\mathbf{y}_k^+$ is the state–costate at the end of segment k and $\mathbf{y}_k^-$ is the state–costate at the start of segment k. In the optimization, these continuity residuals need to be driven to zero. Now, the optimization problem consists of 6 unknowns for the initial p_{V0} and p_{R0}, and $12 \cdot (N)$ unknowns for the node placements. Then, there are 6 constraints for the terminal p_{VN} and p_{RN}, and $12 \cdot (N)$ continuity conditions that need to be satisfied.

In the optimization problem, it was found that treating the continuity conditions as the constraint functions and the terminal conditions as the objective function produced the most accurate results. Here, the optimization proceeded to the next homotopy loop once the constraint tolerance of 10^{-12} and the optimality tolerance of 10^{-7} were satisfied. Typically, each homotopy loop required about 3 to 5 iterations of `fmincon`, corresponding to approximately 1,060 to 1,590 function evaluations. As a result, each homotopy step was computed efficiently, leading to a fast overall computation of the atmospheric solution.

In the reviewed literature, no example was found of solving the atmospheric ascent optimization with `fmincon` coupled to a multiple–shooting scheme that propagates from shooting points and enforces

continuity at their junctions. This approach performed best among all methods tested. It converges reliably and produces accurate results. The solution meets the specified terminal costate conditions and yields a physically feasible trajectory. The physical feasibility is confirmed in Chapter 9.

7.3. Combining the Atmospheric and Vacuum algorithm

The last part of the optimization is combining the atmospheric solution and the vacuum solution together.

The first part of the trajectory is, of course, the first stage. To solve the first stage trajectory, an initial guess is needed before it can be solved. This initial guess is generated by computing the entire vacuum trajectory. From the entire vacuum trajectory, only the points are taken from the first stage. This first stage vacuum solution serves as a great initial guess that leads to convergence of the atmospheric solution. As explained before, the terminal conditions of the first stage atmospheric solution are the six initial costate values of the first coast.

After the previously described homotopy process, the atmospheric burn arc is solved. However, due to the influence of aerodynamic forces on the booster, its altitude and velocity at the terminal point are lower than in the initial vacuum-based estimate. Therefore, from this terminal point of the burn arc, a new vacuum solution needs to be generated. Apart from some changes, this optimization works in the same way as the computation of the initial vacuum solution discussed in chapter 7.1. However, the initial conditions are now set at the end of the first stage, and there is one burn arc less. Therefore, the dimension of the zero finding problem is now 33, since the shooting node and the 12 continuity conditions of the end of the first stage are no longer present.

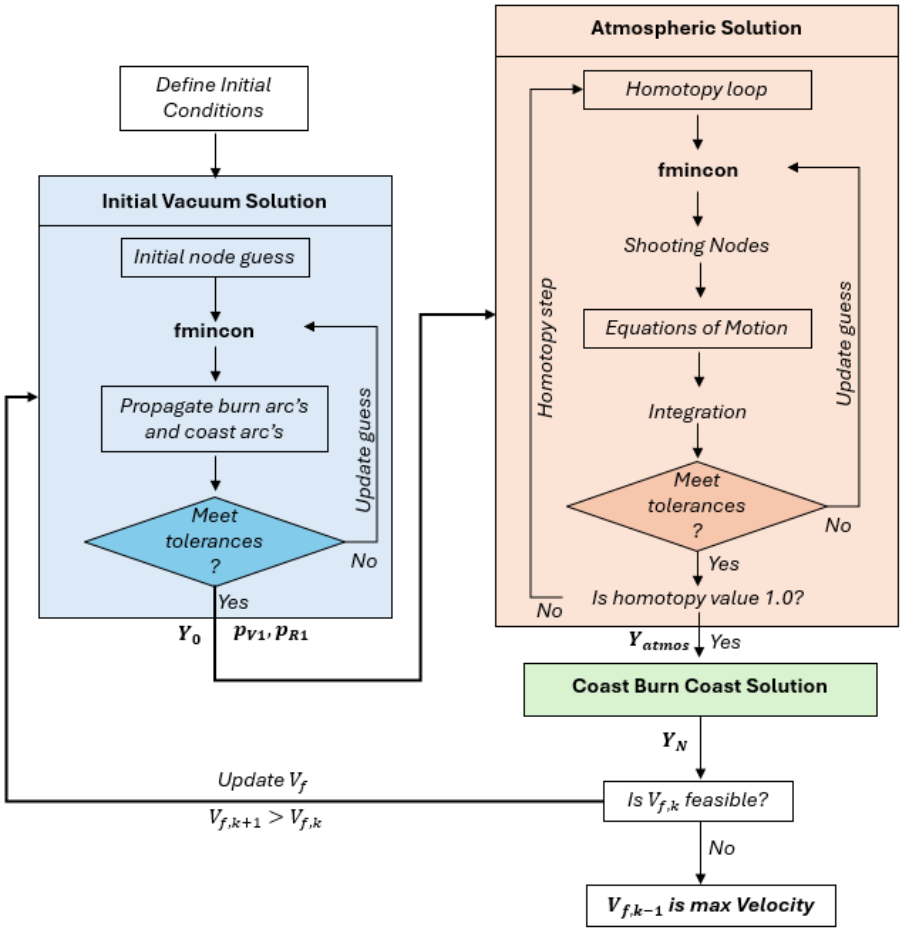


Figure 7.2: Schematic overview of the code functions

As mentioned previously, the maximum velocity is found by sequentially adapting the final velocity. If a certain terminal velocity satisfies the set terminal conditions and gives a feasible solution, a for loop can be used to iteratively increase the final velocity until no feasible solution is possible anymore. Then the maximum velocity and the maximum range have been found. In figure 7.2, a schematic can be found. This schematic explains how the different code files work together in order to find the optimal trajectory.

A check needs to be performed to ensure that the complete atmospheric and vacuum solutions form a continuous trajectory. The continuity of the state variables is straightforward to achieve, as the initial boundary conditions of the vacuum solution are directly taken from the final state output of the atmospheric solution. The costate at the end of the first stage vacuum solution serves as the terminal objective function of the atmospheric optimization. If, after the full optimization of both the atmospheric and vacuum solutions, a slight offset remains between the costates at the interface between the two solutions, a new atmospheric optimization can be performed with an adjusted objective for the terminal costates (p_{VN} and p_{RN}). In the case presented in chapter 8, this procedure is performed twice, similar to the iterative cycle described by P. Lu [40]. It should be noted, however, that a mismatch in costates does not affect the physical feasibility of the overall trajectory. Ensuring continuity in the state variables is the primary requirement for a feasible solution.

Not all launch vehicles have a first stage that reaches altitudes where aerodynamic forces can be neglected. Especially for launch vehicles that make a sub-orbital flight. For example, the booster used to launch the HTV-2 hypersonic glide vehicle is the Minotaur rocket [69]. This rocket has a relatively short first stage that is likely to end while aerodynamic forces are still significant. When this is the case, in the vacuum solution this portion of the coast is propagated using the standard equations of motion up to an altitude at which aerodynamic forces can reasonably be neglected. In this work, an altitude of 60 km is chosen as the point beyond which aerodynamic forces are neglected, since above this altitude the atmospheric density is sufficiently low to not have a major influence on the trajectory. So far, with the selected terminal conditions used in the simulations presented in chapter 8, corresponding to common HGV insertion conditions at around 100 km altitude, there have been no cases in which the coast phase is too short and the second burn still occurs within the atmosphere. However, if such scenarios are to be considered in future work, adaptations to the model are required. A potential approach would be to introduce two extra homotopy loops, one for the first coast phase and one for the second burn, to gradually incorporate aerodynamic effects throughout those segments.

The drawback of this current approach, with a clear intersection between the atmospheric and vacuum solution, is that it may introduce a sharp transition in the optimization between the atmospheric and vacuum segments of the coast. As a potential improvement, the entire first coast could instead be solved using the homotopy approach applied to the atmospheric trajectory. This refinement will be considered in the recommendations for future research section.

For the `fmincon` optimizations in the different code files, the sequential quadratic programming (SQP) algorithm has been selected. This algorithm has proven effective in finding optimized trajectories and, ultimately, in maximizing the terminal velocity. In the future, the SQP algorithm of `fmincon` can be converted to C++, which allows an even faster computation time for the optimization. It should be noted, however, that if the rocket model is modified, the parameters of the `fmincon` optimizer may need to be tuned. In particular, certain launch scenarios, especially those involving high inclinations, can cause difficulties with convergence.

8

Results

This chapter presents the results of the optimization model developed in MATLAB, as described in the previous chapter. First, the constant parameters used in the simulation are summarized. Then, the optimized trajectory and the key flight characteristics are presented.

Three different scenarios are compared. The first is a maximum-range trajectory launched from Vandenberg in a retrograde direction. This is chosen because it is the only available reference trajectory for a hypersonic glide vehicle boost phase found during this study. The second is a maximum-range trajectory launched from Kourou in a prograde direction, representing a more conventional launch scenario. Finally, a third scenario considers a Kourou launch with a trajectory that does not maximize range, to highlight how the optimizer adjusts the trajectory in order to dissipate energy compared to the second scenario.

8.1. Constant parameters

First, a booster model is selected. Hypersonic glide vehicles are conventionally launched using intercontinental ballistic missiles, such as the Minuteman or the Minotaur rocket. However, detailed specifications of these rockets are largely not publicly available. The Minotaur IV rocket was used for the launch of the HTV-2 during DARPA's flight tests in 2010 and 2011. The boost phase of this trajectory is analyzed in detail by Wright [70], where information on the booster model is also provided. In this study, a simplified booster model is used. Due to time constraints, the MATLAB framework currently supports only two-stage booster configurations, although the optimization method itself can be extended to three or more stages. Since the Minotaur uses three stages, the present analysis considers two of them to construct the booster model.

Table 8.1: Rocket parameters used for the booster model [70]

Parameter	Stage 1	Stage 2
Thrust T [N]	2,224,908	289,199
Specific impulse I_{sp} [s]	282	300
Propellant mass m_p [kg]	45,370	7,070
Dry mass m_c [kg]	3,620	640
Burn time t_{burn} [s]	56.4	72
Mass flow $\dot{m}$ [kg/s]	804.8	98.2

Section 8.2 further discusses the differences in stage velocities and total range between the booster configurations of the reference case and the current study.

The engine used in the Minotaur rocket is the SR-118. For the exit diameter of the first stage a nozzle diameter of 1600 mm is taken. Furthermore, the payload mass is chosen to be 1000 kg [70]. This is the mass of the HGV.

The aerodynamic characteristics of the booster depend on the reference area, which is chosen to be $S_{ref} = 2.2\,\mathrm{m}^2$. Since the diameter of the Minotaur rocket is not specified in the literature examined for this study, the surface area is estimated based on the diameter of the Minuteman III rocket [71]. As the Minuteman III is the predecessor of the Minotaur, it provides a suitable reference for this parameter. Furthermore, the aerodynamic coefficients of the lift and the drag forces on the booster are defined by the fit curves and polynomials, defined in chapter 5.4.

Then, the control and path constraint parameters need to be defined. These can be found in the following table.

Table 8.2: Constraint parameters for the ascent trajectory

Parameter	Value	Unit	Source
Maximum angle of attack $\alpha_{\max}$	25	deg	[13]
Maximum rate of change of α	10	deg/s	[13]
$Q\alpha$ limit	1.4364×10^5	Pa·deg	[40]
Maximum dynamic pressure $q_{\max}$	5.0×10^5	Pa	[72]

It should be noted that following the requirements in chapter 4, the booster model is arbitrary. It can be quickly adapted in order to generate new trajectories with other rockets.

8.2. Maximum Range Vandenberg Launch scenario

For the first case, a maximum-range trajectory is computed with a launch from Vandenberg Air Force Base (VAFB) in California. Although retrograde launches are less conventional for standard missions, this choice follows the analysis by Wright [70] of the HTV-2 test flight, which also used a retrograde trajectory from VAFB. The launch site is located at a latitude of $34.75°$ and a longitude of $-120.52°$.

The booster must satisfy the following terminal conditions in order to properly inject the hypersonic glide vehicle (HGV) into its glide phase:

Table 8.3: Terminal trajectory conditions for the VAFB launch

Parameter	Value	Unit
Altitude	100	km
Flight path angle	-5	deg
Inclination	145	deg

Here, the inclination of $145°$ is chosen based on the approximate flight direction of the HTV-2 launch as reported by Wright [70].

Simulating the maximum range trajectory gives the following results:

Table 8.4: Trajectory results for the VAFB launch

Quantity	Value	Unit
Velocity	6322	m/s
Range	1140.6	km

The range is calculated by determining the angle between the launch location and the final position on Earth's surface, and then converting this angle into a distance using a spherical Earth assumption.

The following ground track trajectory plot confirms that the direction of the trajectory in this study is similar to the direction of the reference trajectory.

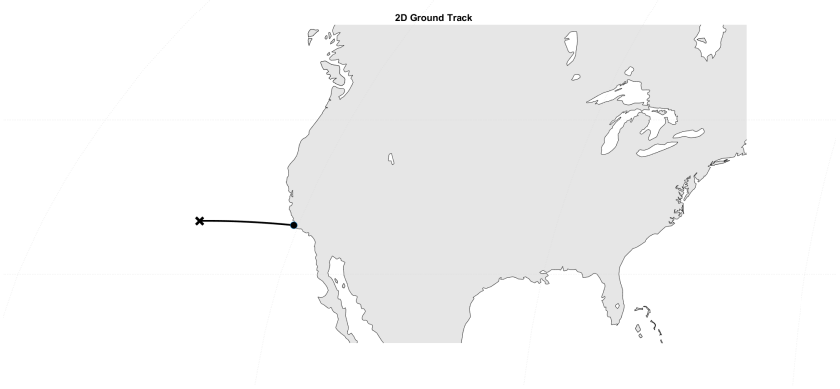


Figure 8.1: 2D Ground Track - VAFB Launch.

In contrast, the three-stage Minotaur rocket trajectory analyzed by Wright showed a range of about 1700 km and a terminal velocity of 7.16 km/s [70]. The larger range can be explained by the fact that the additional stage provides more delta-V to the booster.

The theoretical delta-V of each stage is compared between the two cases. This value is calculated using the rocket equation shown below, where the subscript i denotes the stage number [49].

$$\Delta V_i = I_{\mathrm{sp},i} \cdot g_0 \cdot \ln \frac{m_{0,i}}{m_{0,i} - m_{p,i}}, \tag{8.1}$$

Table 8.5: Comparison of stage ΔV between the reference case and this study.

	Reference Configuration	**Two stage configuration**
ΔV_1 [km/s]	2.1	4.2
ΔV_2 [km/s]	3.4	*N.A.*
ΔV_3 [km/s]	4.9	4.9
ΔV_T [km/s]	10.4	9.1

Since the table presents the maximum theoretical velocity for each stage, the specific impulse used in these calculations is taken under vacuum conditions. In the actual reference trajectory, Wright showed the additional stage produced an actual ΔV_2 of approximately 1.6 km/s after the first stage burnout [70]. Furthermore, as seen in the table, the additional theoretical delta-V of the three-stage configuration is 1.3 km/s.

This means that, in total, the three-stage Minotaur trajectory achieves more velocity compared to the two-stage launch considered in this study. Underneath in figure 8.2, a plot can be seen of the magnitude of the velocity of the booster during the simulated trajectory.

Here it can be seen that in practice, the actual difference in final velocity with the reference case is closer to 0.84 km/s. Figure 8.2 shows the first stage ends at a velocity of about 2.9 km/s. In contrast, the first stage in the reference case reaches a burnout velocity of only 1.4 km/s, which is significantly lower [70].

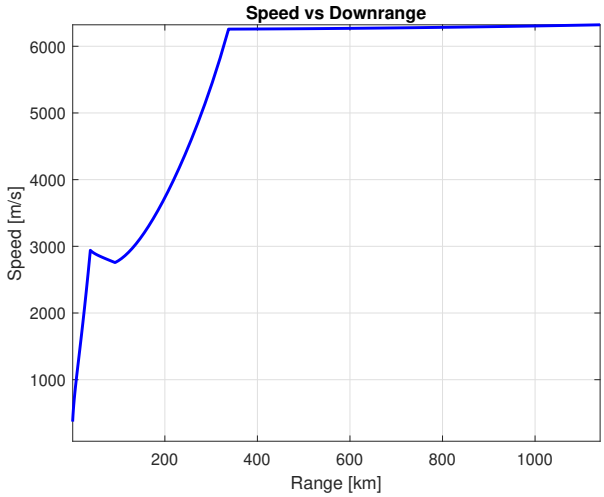


Figure 8.2: Speed Range plot - VAFB Launch.

First of all, this difference in first stage burnout velocity is caused by the fact that the first stage in the reference case must accelerate the additional mass of the extra stage of the three-stage configuration. As shown in table 8.5, this results in a significant loss of velocity. Consequently, in the simulated case of this study, the lighter overall rocket mass enables the first stage to generate a higher ΔV.

Furthermore, in the reference trajectory, the first two stages are ignited almost back-to-back with virtually no coasting in between. As a result, the rocket follows a largely vertical path during the early part of the ascent, which increases gravity losses. As stated in the analyses of Wright, the first stage ends at a burnout angle of about $87°$, and the second stage at about $10°$ [70]. The reason for the steep ascent of the first stage in the reference trajectory is the requirement to ensure that the drop zone of the empty casing remains close to the launch site. As a result, the vehicle must initially fly almost vertically. In this study, however, the designated drop zones of the stages have not been taken into account. Therefore, the first stage in the simulated case of this study ends at a flight path angle of $41.5°$, and is therefore able to add more horizontal velocity. This also reduces the difference in final velocity between the two cases, compared to the additional delta-V of the extra booster.

Additionally, the reference trajectory accounts for performance losses resulting from the lower specific impulse at lower altitudes. This effect has not yet been incorporated into the current model and is therefore recommended as an important addition for future research. This also led to the higher final velocity of the first stage in this study. In summary, the three factors discussed above together explain the difference in the first stage velocity magnitude.

The reverse occurs when the last stages of both cases are compared. The third stage of the reference trajectory is the same booster as the second stage in the simulated trajectory of this study. In the reference case, this booster provides almost 4 km/s of ΔV [70], whereas in the simulated case it delivers only about 3.5 km/s. The additional stage in the reference case enables a more optimal flight-path angle for the ignition of the final stage, as the second stage burn already ends near the target altitude. In comparison, the final stage of the case of this study needs to put relatively more energy into increasing altitude rather than increasing horizontal velocity.

Furthermore, the altitude range plot of the two-stage trajectory looks as follows.

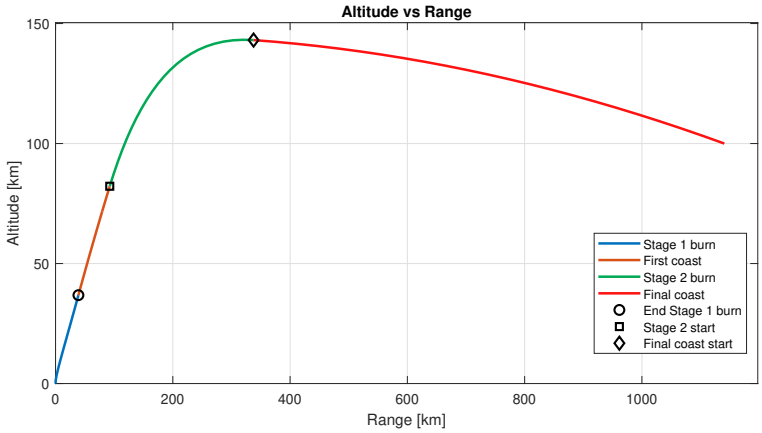


Figure 8.3: VAFB launch Altitude-range plot.

The different colors of the line segments and the markers indicate the various parts of the trajectory. From these plots, several similarities with the reference launch trajectory can be observed. This reference trajectory, denoted as Trajectory B, is provided in Appendix E. First, the part of the trajectory that covers the largest range is the final coast phase. This is also the case in the reference trajectory.

According to Wright, in the reference case the third stage flies almost horizontally for a large portion of the trajectory, minimizing gravity losses. This means that the final stage flies near the apogee of the trajectory. The same behavior can be observed in the optimized trajectory of this study. The last stage ignites shortly before apogee and completes its burn right after the apogee. At this point, the flight path angle is close to zero, allowing the booster to maximize its horizontal velocity and thereby extend its range.

This is also confirmed by the study of L. Gronlund et. al. [73]. This study states that the range of a suborbital missile is greatest when the final boost occurs near the apogee of the trajectory. At this point, the vehicle's vertical velocity is nearly zero, so all of the thrust contributes directly to horizontal velocity instead of fighting gravity. If the final boost were applied earlier, a portion of the energy would be wasted on climbing higher, leading to larger gravity losses. By burning near the apogee, these losses are minimized and the vehicle gains the maximum possible downrange distance. In essence, this timing ensures that the available energy is used most efficiently to extend the range rather than to increase altitude. Figure 8.3 follows the same motion. After the initial boost phase, the rocket coasts upward until reaching the upper part of its trajectory, where the final booster is ignited while the vehicle is flying almost horizontally.

8.3. Maximum Range Kourou Launch scenario

The second launch scenario is the launch from Europe's spaceport in Kourou. The launch site is located at a latitude of $5.16°$ and a longitude of $-52.65°$. This launch is eastward in a prograde direction. The specified terminal conditions for the launch vehicle are given below. These terminal values are typical for delivering an HGV in its glide corridor.

Table 8.6: Terminal trajectory conditions for the Kourou maximum range launch

Parameter	Value	Unit
Altitude	100	km
Flight path angle	-3	deg
Inclination	10	deg

Simulating the maximum range trajectory gives the following results:

Table 8.7: Trajectory results for the Kourou maximum range launch

Quantity	Value	Unit
Velocity	7827.3	m/s
Range	2067.7	km

Furthermore, a three-dimensional representation of the trajectory over the globe is shown in the figure underneath.

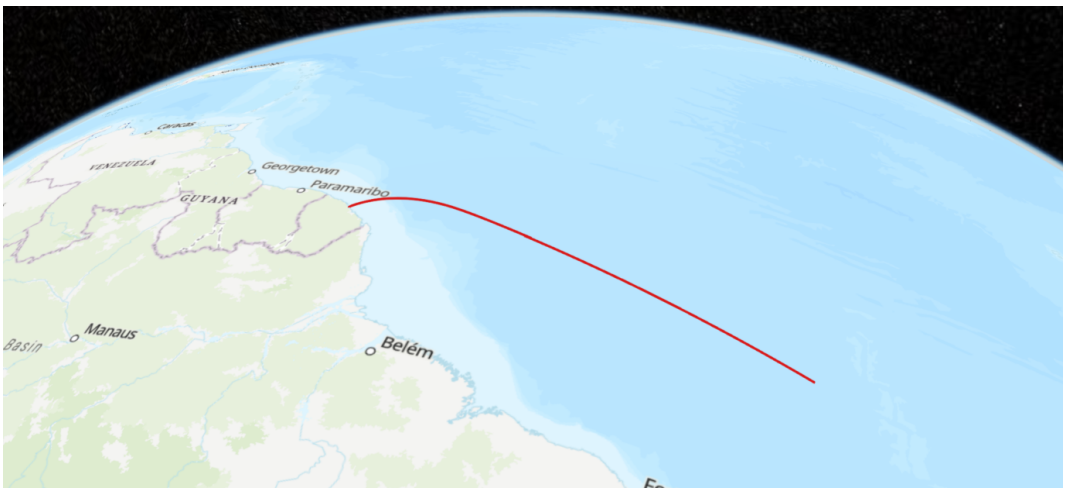


Figure 8.4: 3D Globe Trajectory - Kourou Launch.

From the results, it can be seen that the final velocity and range are far higher than the previous launch scenario. This has everything to do with the launch latitude and direction. As mentioned earlier, the Earth rotates around its own axis, and both the launch direction and latitude determine how this rotation influences the rocket's motion. Launching eastward provides a centrifugal boost, since the rocket's initial inertial velocity already includes the eastward surface speed of the Earth. The benefit is strongest near the equator, where the rotational surface speed is highest. The launch from Kourou, being close to the equator, therefore gains a larger benefit from the velocity component in the eastward direction than the westward launch from Vandenberg. Overall, this leads to a higher efficiency and greater final velocity and range.

Figure 8.5 confirms this. It can be seen that the velocity at the end of the first burn is 3.3 km/s. This is 0.4 km/s higher compared to the launch scenario at Vandenberg.

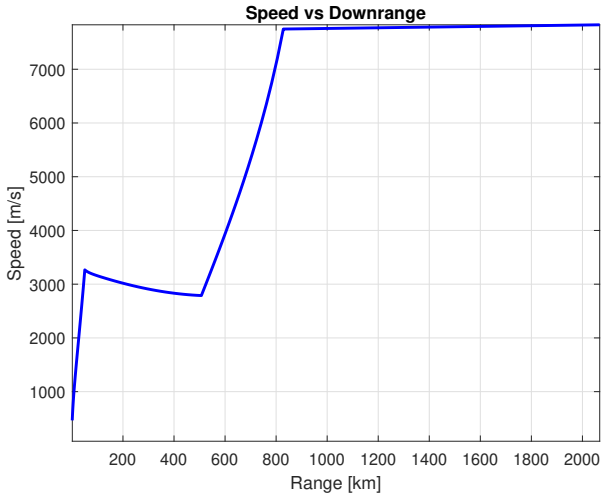
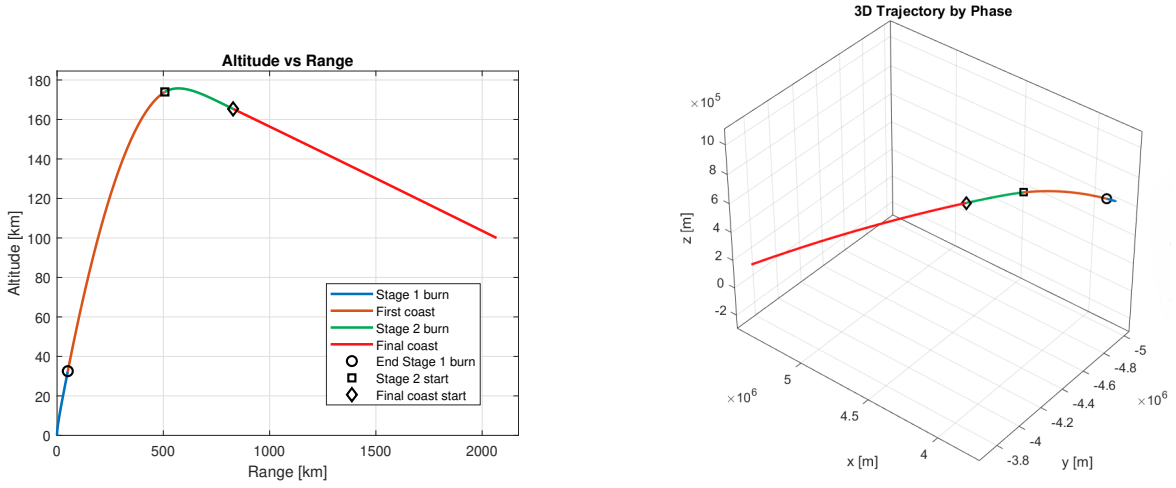


Figure 8.5: Speed profile Kourou maximum range scenario.

The same effect is also shown in the next figure. Here, the altitude at the end of the first burn is 32.5 km, which is 2.4 km higher compared to the launch scenario at Vandenberg.



(a) Range-Altitude plot.

(b) 3D Trajectory.

Figure 8.6: Kourou launch trajectory progression.

Compared to the Vandenberg launch scenario, it can also be seen that an increase in the terminal velocity also comes with an increase in the apogee altitude of the trajectory. A higher terminal velocity indicates that the final coast phase has greater orbital energy. This directly corresponds to a larger semi-major axis, and therefore a higher apogee altitude of the trajectory. Furthermore, because the final flight path angle is shallower than in the previous Vandenberg launch scenario, the final coast phase of the trajectory is more horizontal. Consequently, the final boost can be ignited closer to the apogee, allowing the vehicle to gain more horizontal velocity and thus achieve a greater range. The influence of the flight path angle on the maximum range will be further discussed in the sensitivity analysis in the next chapter.

The first optimized coast phase lasts $165.8\,\mathrm{s}$, after which the rocket reaches a nearly vertical orientation and the final stage is ignited. The final burn lasts $162.7\,\mathrm{s}$ and brings the rocket to its specified terminal conditions. Despite the shorter burn duration, this phase covers a significantly larger downrange dis-

tance. It should be noted that although the final coast segment may appear linear in the trajectory plot, this is not the case. The booster performs a ballistic flight described by Keplerian motion. However, because the velocity in the final coast is close to the circular orbital velocity at the corresponding altitude, the flight-path angle continues to change, but only gradually compared to the earlier flight phases.

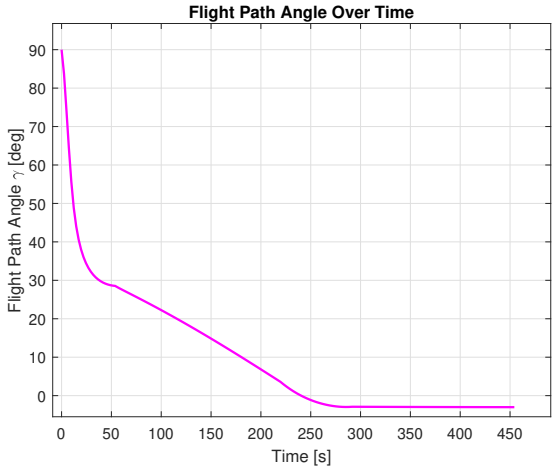


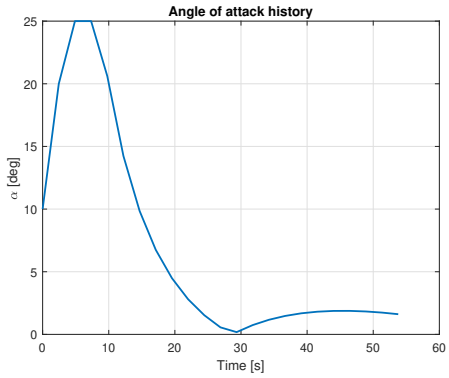
Figure 8.7: Flight path angle profile for the Kourou maximum range scenario.

Figure 8.7 shows the flightpath angle profile of the Earth relative velocity. Here it can be seen that the flight path angle γ is close to zero when the final stage is ignited, confirming the findings of L. Gronlund [73]. The final stage starts around $t = 220s$ at $\gamma = 3.5°$, and is burned out around $t = 292s$ at $\gamma = -2.9°$. This behavior is also consistent with the study of L. Gronlund et. al. [73], that the final stage is ignited around the apogee of the trajectory. Furthermore, in the flight data of the suborbital trajectory of the sounding rocket *Scout*, a similar strategy can be found, which also showed ignition of its final stage close to the apogee [74].

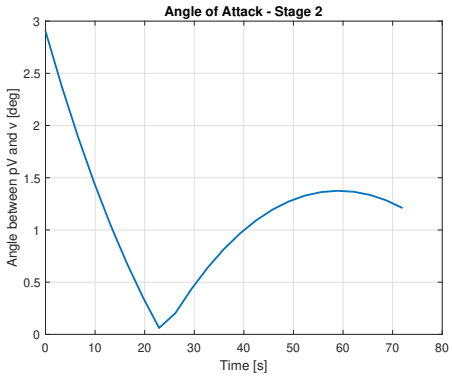
Furthermore, a very similar flight path angle profile is shown by F. Morgado et. al. [75] for an optimized two-stage trajectory. During lift-off, the rocket initially climbs vertically before gradually pitching over through a gravity turn. Throughout the coast phase, the flight path angle remains slightly positive to maintain ascent, with a brief horizontal turn before coasting to increase velocity and minimize gravity losses. This closely matches the behavior observed in the Kourou launch scenario, confirming that the trajectory follows a comparable optimal guidance strategy.

8.3.1. constraints check

Lastly, the actual steering control and constraints of the booster have been examined. Below, the angle of attack profiles for both stages are shown.



(a) Angle of Attack Stage 1.



(b) Angle of Attack Stage 2.

Figure 8.8: Angle of Attack profiles for both stages of the Kourou launch.

From these profiles, it can be seen that during the first stage, the control constraint becomes active at the $25°$ constraint limit. Furthermore, as shown in Figure 8.8a, the angle of attack is constrained by the rate limit during the first two time steps. Consequently, it cannot change by more than ten degrees per time step. This shows that the vehicle reaches its maximum steering capability during this phase and that both the angle of attack constraint and the angle of attack rate constraint are saturated.

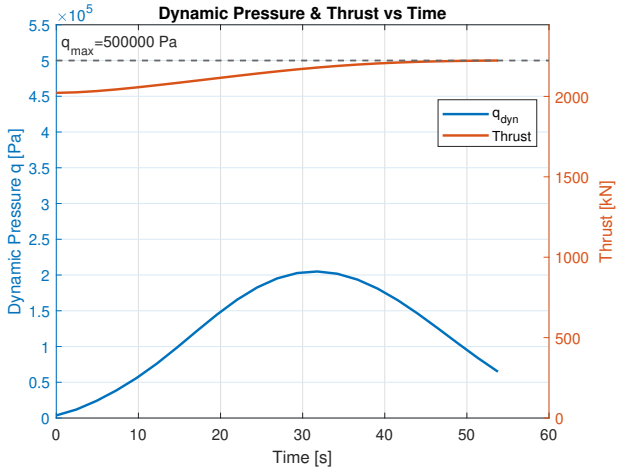


Figure 8.9: Thrust level and dynamic pressure - Kourou Launch.

In Figure 8.9, the evolution of the dynamic pressure and thrust level can be observed. A clear peak in dynamic pressure occurs around $t = 32\,\mathrm{s}$, after which it decreases rapidly. This behavior results from the balance between the increasing velocity and the decreasing air density, which respectively raise and lower the dynamic pressure during ascent. The dynamic pressure constraint is not fully matched, and the same observation holds for the $q\alpha$ constraint. If the booster would increase its horizontal velocity on a lower altitude, or if strong crosswinds were present, the aerodynamic forces and dynamic pressure acting on the booster would increase and approach the constraint limits. If the dynamic pressure constraint were to be exceeded, the MATLAB model would reduce the thrust level. Similarly, if the product of the angle of attack and dynamic pressure exceeded its limit, the optimizer would decrease the angle of attack, as shown in Algorithm 1. In the upcoming sensitivity analysis, this dynamic pressure limit will be adjusted to evaluate its influence on the throttle behavior. The thrust level increases steadily throughout the first stage, approaching its maximum value toward the end of the burn.

8.4. Kourou launch scenario for a non-maximum range trajectory

The final scenario analyzed in this study uses the same launch location in Kourou and the same eastward launch direction. However, in this case, the trajectory is not optimized for maximum range or velocity. Instead, the objective is to reach specific terminal conditions. This approach is relevant because it is not always desirable for the booster to deliver the HGV with maximum speed or range. In some missions, an excessively high terminal velocity could exceed the desired entry conditions for the glide vehicle. Therefore, the following terminal conditions are specified with a lower terminal velocity.

Table 8.8: Terminal trajectory conditions for the Kourou non-maximum range launch

Parameter	Value	Unit
Altitude	105	km
Velocity	6000	m/s
Flight path angle	-3	deg
Inclination	10	deg

Ultimately, this resulted in a range of 1500.329 km, which is as expected, lower than that of the maximum

range trajectory. The range altitude plot of the trajectory looks the following.

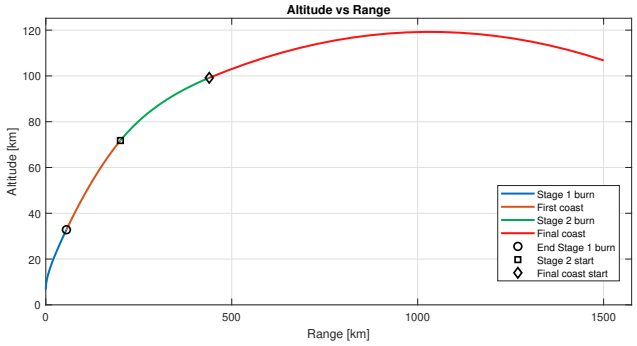


Figure 8.10: Range-Altitude plot Kourou non maximum range scenario.

From this figure, it is clear that the first coast ends at a lower altitude. Consequently, the final burn starts well before the apogee of the trajectory. Therefore, the second stage loses more energy due to gravitational losses. The same can be seen in the figure of the flight path angle.

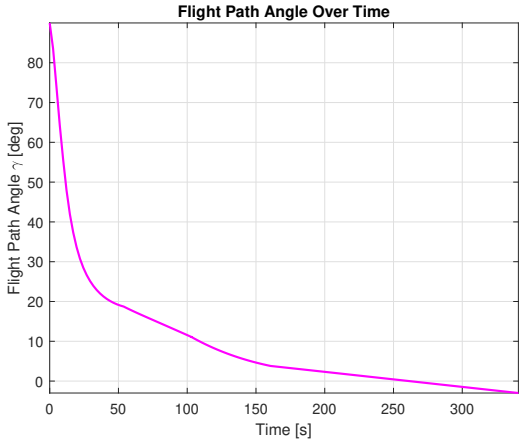


Figure 8.11: Flight path angle Kourou non maximum range scenario.

The final burn and coast of the trajectory have a flight path angle that is significantly higher compared to the case of the maximum range trajectory. Here, the final stage starts around $t = 101$ s at $\gamma = 11.5°$, and is burned out around $t = 159s$ at $\gamma = 3.7°$. The trajectory of this final stage is clearly more vertical compared to the maximum range case. Energy that could have been put into gaining horizontal velocity is now lost in gaining altitude.

Furthermore, the second stage is cut off early, with a burn time of 57.5 s instead of the 72 s required to deplete all propellant at full throttle. This means some propellant remains unused to satisfy the prescribed terminal conditions.

The work by D. Wright also presents a case where a three-stage Minotaur rocket performs an energy management maneuver to reduce its total energy and achieve a lower terminal velocity [70]. This energy-management reference trajectory, referred to as Trajectory A, is shown in Appendix E. In that study, the final stage is also ignited well before the apogee of the trajectory, similar to the results obtained in this work. Additionally, the reference trajectory includes a dog-leg maneuver to dissipate excess energy. In this maneuver, the direction of flight is altered near the end of the first stage, causing

part of the energy to be expended in rotating the plane of the trajectory, thereby reducing the overall kinetic energy.

Below, the ground track of the trajectory for this scenario is shown. It can be observed that the booster follows a smooth, continuous path without performing a dog-leg maneuver. This indicates that no additional energy is dissipated through lateral steering, in contrast to the reference case by D. Wright [70].

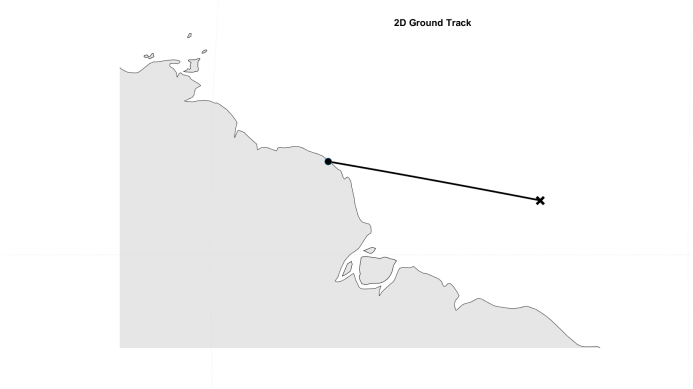


Figure 8.12: Ground track Kourou non maximum range scenario.

In this study, the engine of the second stage is ignited well before the apogee of the trajectory and cut off early to dissipate excess energy. In contrast to the reference case, no dog-leg maneuver is required. The reference trajectory by D.Wright [70] employs solid rocket boosters, which can not be shut down early, and due to the additional stage, a greater amount of energy needs to be dissipated to achieve the desired terminal conditions. Thus, this explains the need for the dog-leg maneuver. In principle, the method used in this study could potentially also generate a dog-leg maneuver, as no restrictions are imposed on the rocket to prevent such a trajectory. However, this capability should be verified in future work, for example by evaluating whether energy can be dissipated using a solid rocket booster.

8.5. Summary of the Result findings

With the presented theory, it has been shown that the method can successfully optimize a sub-orbital trajectory for HGV insertion. The algorithm optimizes body orientation, coast durations, and burn times, while incorporating a throttle schedule to enforce the dynamic-pressure constraint.

By inspecting the optimizer output, a clear systematic pattern appears in how the two stages are used. After the first burn the booster coasts toward apogee. Just before the apogee the final stage is ignited and burns at a near-zero flight-path angle to maximize horizontal velocity and therefore range. After burnout, the remaining part of the trajectory is the final coast until it meets the required final conditions. During this flight phase, most of the range is covered, compared to the other flight segments. Conversely, when the mission requires dissipating energy, to avoid an excessive terminal velocity or range, the final burn is started earlier and may be cut off before all propellant is used. The gravitational losses and an earlier burn termination are used to reduce the vehicle's final energy.

As mentioned in chapter 7.1, due to the way the optimization problem is defined and solved, for the latter case the assumption is made that all the propellant of the first stage is burned. Therefore, only the second stage has a varying burn time. Therefore, in the future the first stage burn time can be added to the control variables. This analysis can reveal how the optimizer reallocates thrust between stages when the objective is not range maximization. In other words, it may show whether the solver shifts more burned propellant to the upper stage to meet specific terminal conditions, and those insights can guide design or mission trade-offs.

The chosen optimization strategy reliably produces converged solutions, and the MATLAB `fmincon` solver delivers results far faster than the modified-Newton methods commonly reported in the literature.

The model does, however, become more sensitive due to the added final coast phase. This is because the final coast is the longest and the last phase that also offers no opportunity for corrective control. Small changes earlier in the trajectory can make meeting terminal conditions more challenging. This sensitivity is most noticeable for high-inclination launches and has been shown to potentially cause convergence issues. However, this is rarely problematic for prograde scenarios such as Kourou. The reduced convergence at higher inclinations is possibly caused by greater sensitivity to small errors and a smaller benefit from Earth's rotation. Since the target velocity vector deviates more from Earth's rotational direction, smaller steering or thrust errors can lead to larger deviations in the final inclination, making the solution space smaller and harder to converge. In addition, higher-inclination launches gain less benefit from Earth's rotational velocity, requiring more ΔV and possibly stronger steering to reach the set terminal conditions. In practice, a retuning of `fmincon` options, like trying alternative algorithms and tolerances, can help restore convergence for new launch cases. The tolerances described in chapter 7.2 are currently used to ensure that the trajectory is converged correctly. However, it is still essential to check the residuals of the optimality and transversality conditions to assess the validity of the obtained solution. The maximum number of iterations in the current model is set to 10,000, although the actual number required is typically much lower, usually less than half of this value. This high limit ensures that the solver does not terminate prematurely. In chapter 10, future work will be recommended to help further overcome the sensitivity issue.

Lastly, the results presented in this chapter demonstrate that the theory used in the optimization model can answer the main research question: "*How can an indirect optimization method be applied to compute the optimal ascent trajectory of an HGV boost phase while accommodating different booster models with variable thrust?*" The model is formulated using Pontryagin's Maximum Principle, and solved with a shooting method for the vacuum solution, and then augmented with aerodynamics via a homotopy continuation. This has been shown to accommodate different booster models with variable thrust and to successfully solve the optimization problem.

9

Sensitivity Analysis & Verification

It is important to verify whether the presented results are accurate. In line with two of the three sub-questions, it is necessary to evaluate the physical accuracy of the trajectory and assess the optimality of the obtained solution. Therefore, this chapter consists of two sections. The first shows a sensitivity analysis done on certain parameters used in the optimization. The second presents the verification steps performed to evaluate the optimal trajectory.

9.1. Sensitivity Analysis

The goal of this section is to help verify the results of the previous chapter by checking whether deviations in certain conditions lead to the expected changes in the outcome. For this analysis, the inclination, HGV mass, flight path angle, and launch latitude sensitivities are considered. These are especially important parameters, as they are free variables in the optimization problem that determine key aspects such as the launch direction, final trajectory, and insertion conditions of the HGV. Further, the inclination, HGV mass, and latitude serve as effective parameters to verify whether the optimization model produces physically consistent results, since their influence on the range can be logically reasoned.

In addition to confirming intuitive effects, this analysis also examines less obvious aspects, such as the influence of the propellant mass fraction of each stage and the effects of the dynamic pressure constraint and mass flow rate. This sensitivity analysis demonstrates that the optimization model could potentially also support mission planning or be used to optimize the booster design for specific objectives.

Since many different parameters are involved in the optimization, the six key parameters described above have been selected for this study. For each of these parameters, the sensitivity gradient has been evaluated locally around a base case, which is the maximum range trajectory from Kourou discussed in the previous chapter. All sensitivity analyses are therefore performed relative to this nominal situation.

As presented in the previous chapter, the maximum range Kourou launch scenario has the following outcome and terminal conditions.

Table 9.1: Summarized Kourou maximum range launch results

Parameter	Value	Unit
HGV Mass	1000	kg
Altitude	100	km
Flight path angle	-3	deg
Inclination	10	deg
Final Velocity	7827.3	m/s
Final Range	2067.7	km

9.1.1. Inclination Sensitivity

The first sensitivity that has been evaluated is the terminal inclination of the trajectory. This influences the direction in which the booster flies.

Table 9.2: Sensitivity of the inclination.

Inclination [°]	**Maximum Range**[km]
13	2063.283
7	2071.436

Given the base case, the average local sensitivity gradient is -1.192 km/°. So, this means that a smaller inclination will lead to an increase in range. This result is logical, since a smaller inclination means the rocket's trajectory is more aligned with Earth's eastward rotation. Therefore, the rocket profits more from this rotation, as its heading is more closely aligned with the Earth's eastward inertial velocity component. Ultimately, the rocket needs less ΔV to reach the set terminal conditions.

9.1.2. HGV Mass Sensitivity

The second sensitivity is the HGV mass. If the mass of the payload increases, this would naturally lead to a lower range. The following table shows the sensitivity analysis.

Table 9.3: Sensitivity of the payload mass.

HGV Mass [kg]	**Maximum Range**[km]
1010	1973.804
990	2153.272

This gives an average local sensitivity gradient of -8.97 km/kg. The direction of this gradient is expected, as an increase in mass requires more energy to accelerate and propel the rocket along its trajectory. Consequently, for the same thrust level, the vehicle achieves lower acceleration and overall performance.

9.1.3. Flightpath angle Sensitivity

The next parameter that is analyzed is the flightpath angle.

Table 9.4: Sensitivity of the flightpath angle.

Flightpath angle [°]	**Maximum Range**[km]
-2.9	2026.9
-3.1	2105.8

This gives an average local sensitivity gradient of -394.01 km/°. So, a slightly steeper and more negative flightpath angle would increase the range of the booster. This gradient clearly shows that the range is highly sensitive to the flightpath angle. Therefore, it is an important parameter to consider when planning the HGV mission. The insertion flight path angle greatly influences the vehicle's achievable range and overall trajectory.

However, the effect that this parameter has on the range of the trajectory is less intuitive compared to the previously analyzed parameters. The terminal flightpath angle has two main effects on the trajectory that also influence the range. A shallower terminal flightpath angle means that the velocity component of the trajectory puts more energy into covering horizontal distance. However, a shallower final flightpath angle also means a lower apogee and a shorter final coast arc. This means covering less distance. So, a steeper more negative flightpath angle has less horizontal speed but a longer final coast. Also, changing the final flightpath angle changes the ballistic shape of the trajectory. Therefore, it is not clear to tell right away what flightpath angle leads to the longest range. The figure underneath compares the two trajectories of the sensitivity analysis.

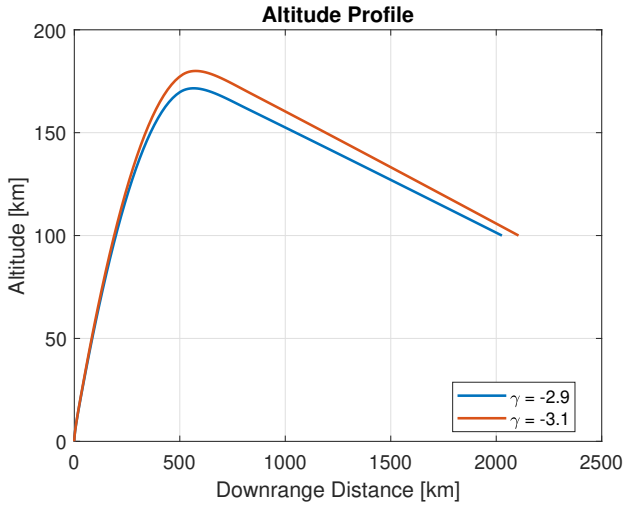


Figure 9.1: Comparison of flightpath angle sensitivity.

When the terminal flight path angle decreases from $3.1°$ to $2.9°$, the final velocity increases slightly from 7825 m/s to 7830 m/s. However, the final coast duration shortens from 166.9 s to 157.9 s, showing that a shallower angle leads to a shorter coast despite the gain in velocity. In this case locally, a steeper flightpath angle leads to a longer coast time and thus a larger range. Therefore, here the effect of a bigger apogee outweighs the loss of velocity.

9.1.4. Latitude Sensitivity

The fourth parameter examined is the launch latitude. As discussed earlier, a smaller latitude increases the influence of Earth's rotation, since the surface speed is higher when the booster is launched closer to the equator. The sensitivity analysis yielded the following result:

Table 9.5: Sensitivity of the latitude.

Latitude [°]	**Maximum Range** [km]
6	2033.7
4	2106.3

This corresponds to a local sensitivity gradient of $-36.9,$ km/°, indicating that a decrease in latitude increases the achieved range. This behavior is expected, as launches closer to the equator receive a larger eastward inertial velocity from Earth's rotation. At higher latitudes, this rotational advantage diminishes, leading to a shorter maximum range. It should be noted that latitude has the opposite effect for westward launches, since in this case the booster moves against Earth's rotation. As a result, the initial inertial velocity from Earth's rotation acts in the opposite direction. Therefore, the rocket requires more energy to reach the same terminal velocity as an eastward launch.

9.1.5. Mass fraction Sensitivity

The next sensitivity analysis concerns the propellant mass fraction of the first and second stages. As shown in Table 8.1, the booster model uses a first-stage propellant mass of $45{,}371$ kg and a second-stage propellant mass of $7{,}069$ kg. This results in a first-stage propellant mass fraction of 0.8652 relative to the total propellant mass (and $1 - 0.8652$ for the second stage). In this analysis, this ratio is slightly varied while assuming that the structural mass of both stages remains constant. Since the deviations are small, the influence of this assumption on the results is expected to be negligible. The sensitivity analysis resulted in the following table.

Table 9.6: Sensitivity of the mass fraction.

Mass fraction	**Maximum Range**[km]
0.8662	2022.634
0.8642	2109.102

This shows a local sensitivity gradient of 43.2 km gain in range for every 0.1 percent decrease in the mass fraction. The effect of varying the propellant mass fraction is not immediately intuitive. Therefore, this sensitivity analysis provides valuable insight into how adjusting the distribution of propellant between stages can help optimize the booster configuration to achieve a greater range. The table above indicates that the booster's range can be improved by reducing the propellant mass fraction of the first stage and allocating that mass to the second stage.

The observed increase in range for the configuration with a smaller first-stage propellant mass fraction can be explained by two main effects. First, shifting propellant to the second stage reduces the amount of thrust delivered in the dense, lower atmosphere. So, more of the thrusting occurs at higher altitudes where the air is thinner and drag losses are smaller. Second, the altered mass distribution increases the total available ΔV. Following the rocket equation, below the actual ΔV computations for the two cases are compared.

$$\begin{aligned} \text{Case } 0.8662: \quad & \Delta V_1 = 4.2813, \quad \Delta V_2 = 4.89602, \\ & \Delta V_{\text{total}} = 9.18 \text{ km/s} \\ \text{Case } 0.8642: \quad & \Delta V_1 = 4.2578, \quad \Delta V_2 = 4.93150, \\ & \Delta V_{\text{total}} = 9.19 \text{ km/s} \end{aligned}$$

Together, this additional ΔV combined with reduced drag losses at higher altitude produce the increase in downrange distance observed.

A similar conclusion is reported by Gottlieb et. al [13]. Gottlieb shows that shifting mass/thrust into the upper stage reduces drag impulse and improves downrange performance. He also adds to this that the path of the first stage is influenced by the second stage design characteristics. This finding confirms the result shown in this sensitivity analysis.

The sensitivity results also indicate a clear coupling between the first and second stages. Changes in the propellant mass or burn time of one stage directly affect the performance and separation conditions of the other, influencing parameters such as velocity, altitude, and structural loading at stage separation. The design effect of the mass fraction can also impact the structural loading, as a higher propellant mass in the first stage results in greater acceleration in the denser part of the atmosphere, leading to higher dynamic pressure. In addition, variations in the propellant mass fraction alter the initial conditions for the second stage, resulting in a different overall trajectory.

A sensitivity analysis like the one shown above could potentially be used to find the optimum mass fraction for the two-stage booster. However, due to time constraints this was not possible anymore and is recommended for future research.

9.1.6. Dynamic pressure constraint & mass flow rate Sensitivity

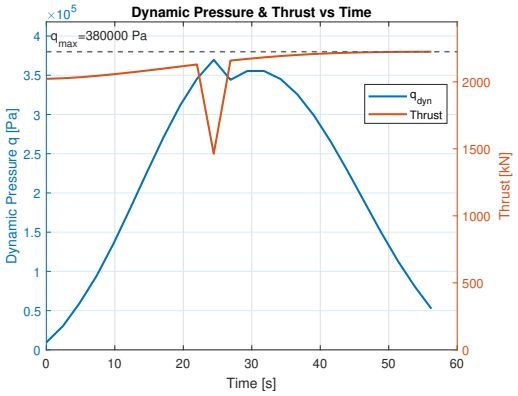
The final sensitivity analysis examines the dynamic pressure constraint. This parameter influences the throttling of the booster and therefore the thrust level and the mass flow rate of the engine. Since the dynamic pressure constraint is not violated in the Kourou launch case, the VAFB scenario is used for this sensitivity study because its ascent showed to encounter higher aerodynamic loading and is therefore closer to the constraint value. The dynamic pressure limit is lowered to 3.8×10^5 Pa and 3.5×10^5 Pa to evaluate how a tighter constraint affects throttle scheduling, thrust level, and propellant mass flow. VAFB is chosen specifically because its typical peak dynamic pressures are nearer these test values, making it the more relevant case to reveal the operational impact of the constraint change.

The change in the boundary condition gives the following results.

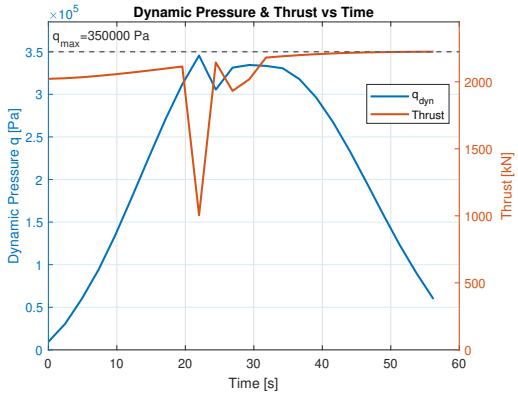
Table 9.7: Sensitivity of the mass fraction.

Dynamic pressure constraint $[Pa]$	**Maximum Range** [km]
3.8 $\times 10^5$	1250.4
3.5 $\times 10^5$	1486.8

The table shows that the range increases when the throttle is decreased. The local range gradient between these two points is 7.9 km per 1.0 [kPa] reduction in the constraint value. The following two figures compare the dynamic pressure and the thrust level of both cases.



(a) Dynamic pressure 3.8 $\times 10^5$.



(b) Dynamic pressure 3.5 $\times 10^5$.

Figure 9.2: Comparison of the throttle between the sensitivity cases.

Here, it is clear that when the dynamic pressure constraint is lower, the engine of the booster has to apply a lower thrust level.

When the dynamic pressure is $5e5 \times 10^5$[Pa], like in the base scenario in the previous chapter, the rocket has a range of 1140.6 km. Imposing a throttle to satisfy a lower q–limit reduces the propellant mass flow rate $\dot{m}$ and therefore lengthens the burn time. Therefore, a part of the boost is delivered at higher, less dense altitudes. This reduces drag losses and can improve downrange performance. This effect is visible in the comparison figure below. This shows the drag impulse between the two sensitivity cases.

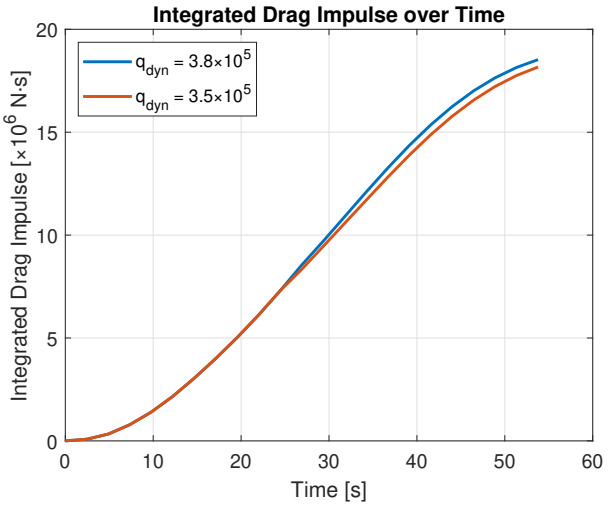


Figure 9.3: Drag impulse comparison.

It can be seen that when the dynamic pressure constraint is lower, the total impulse drag is also lower. This is because more of the thrust is applied at a higher altitude with less drag. The lower drag during the first stage leads to an increase in range at the end of the trajectory.

Since Calise demonstrated within the PMP framework that intermediate thrust levels are not optimal, as discussed in chapter 6.6 [63], the observed performance improvement suggests that introducing an intermediate coast phase during the first stage could be beneficial. However, as previously mentioned, such an approach is considered undesirable due to the physical limitations of booster engines [60].

9.2. Verification

Furthermore, the results obtained in chapter 8 also need to be verified. This section consists of three parts. First, the physical verification will be discussed. Secondly, the used optimality conditions will be verified. Last, the numerical verification will be presented.

9.2.1. Physical Verification

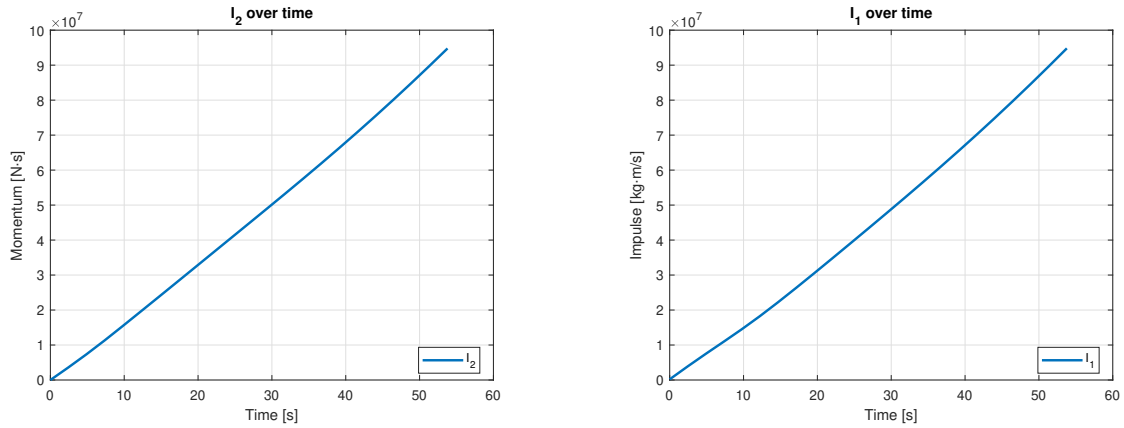
The physical verification is very important. This shows whether the obtained results are actually possible and whether the optimization model does not produce unrealistic or impossible results. Therefore, each of the phases in the trajectory is verified with energy and impulse balances. The case used for the verification is the maximum range trajectory that is launched from Kourou, described in chapter 8.3.

First stage verification

The next method of verifying the physical accuracy of the launch vehicle is based on the law of conservation of momentum. According to this principle, the total external forces acting on the rocket, integrated over time, must be equal to the change in momentum of the rocket and the expelled propellant. This means that the total impulse generated by the rocket's engines and external forces must match the overall change in momentum of the system, ensuring that no momentum is artificially created or lost during the simulation. This can be verified with the following relation [49]:

$$\int_0^t F_{\text{tot}} \cdot dt = \int_{v_0}^{v} dV \cdot m + v \int_{m_0}^{m} dm \tag{9.1}$$

Here, the left side of the equation gives the integrated resulting force on the rocket over time, and the right side gives the total change in momentum of the rocket and the propellant. The right side and left side of the equation are plotted and can be seen in the figure below.



(a) Change in rocket and propellant momentum.

(b) Total force integrated over time.

Figure 9.4: Conservation of Momentum verification.

It can clearly be seen that these results match, indicating that the rocket dynamics model in MATLAB and the optimization framework do not artificially create or lose momentum during the simulation.

Coast verification

Furthermore, the two coasts are also verified. This is done by using the energy balance. This can be described with the following equation:

$$\frac{1}{2} \cdot v(t_1)^2 \cdot m(t_1) + m(t_1) \cdot g \cdot h(t_1) = \frac{1}{2} \cdot v(t_2)^2 \cdot m(t_2) + m(t_2) \cdot g \cdot h(t_2) \tag{9.2}$$

However, since mass is constant along the coast, this can be reduced to:

$$\Delta h \cdot g = \frac{1}{2}(v(t_2)^2 - v(t_1)^2) \tag{9.3}$$

The first coast starts at a velocity of $v_1 = 3338.1$ m/s and ends with a velocity of $v_2 = 2849.8$ m/s. Following equation 9.3, this means that between these points the altitude can increase with $\Delta h = 154$ km. The altitude at the first point is $h_1 = 40$ km and ends with an altitude of $h_2 = 194$ km. Therefore, this means that the coast is physically possible.

The second coast starts at a velocity of $v_3 = 7748.4$ m/s and ends with a velocity of $v_4 = 7827.3$ m/s. Furthermore, the altitude at the first point is $h_3 = 165.3$ km and ends with an altitude of $h_4 = 100$ km. The energy at the start and at the end of the coast is both 3.164×10^7[J]. Therefore, this means that the coast is physically possible.

Second stage verification

Lastly, the verification of the final stage needs to be done. This verification is also done with the energy balance. During the final-stage burn, the energy balance differs from the coast because an active thrust force is present. Consequently, the change in the vehicle's kinetic and potential energy must equal the work performed by that thrust. So, the energy balance is based on the equations of motion, described with equations 6.37 and 6.38.

For the non-dimensional equations of motion, the total energy at a specific point can be computed as follows.

$$E(\tau) \;\equiv\; \tfrac{1}{2}\|\mathbf{V}(\tau)\|^2 + \tfrac{1}{2}\|\mathbf{r}(\tau)\|^2. \tag{9.4}$$

Note that here the non-dimensional equations of motion are used. This changes the magnitude of the values used, however, the energy balance should still be zero. Because the gravitational force in the non-dimensional equations of motion is given by $-r$, the associated potential energy follows from integrating this term with respect to position, yielding $U(r) = \frac{1}{2}\, r^2$.

The time derivative of this balance is:

$$\frac{dE}{d\tau} = \mathbf{V}\cdot\dot{\mathbf{V}} + \mathbf{r}\cdot\dot{\mathbf{r}} \tag{9.5}$$

$$= \mathbf{V}\cdot\big(-\mathbf{r} + T\,\mathbf{I}_b\big) + \mathbf{r}\cdot\mathbf{V} \tag{9.6}$$

$$= \mathbf{V}\cdot\big(T\,\mathbf{I}_b\big), \tag{9.7}$$

Ultimately, this gives the following energy balance.

$$E(\tau_f) - E(0) \;=\; \int_0^{\tau_f} \mathbf{V}(\tau)\cdot\big(T(\tau)\,\mathbf{I}_b(\tau)\big)\, d\tau \;. \tag{9.8}$$

The second stage gives the following residual of the energy balance.

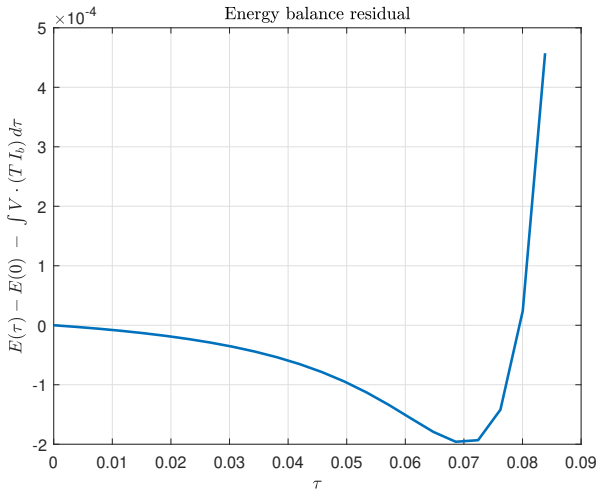


Figure 9.5: Energy balance final stage.

This shows that the Energy balance is practically zero along the trajectory of the second stage.

This means that following the different energy balances and impulse balance, all the segments of the trajectory are physically possible.

Maximum feasible ΔV check

The ideal rocket equation relates the change in velocity of a rocket (ΔV) to the effective exhaust velocity (v_e) and the ratio of the initial mass (m_0) to the final mass (m_f). It is expressed as [49]:

$$\Delta V = v_e \ln \frac{m_0}{m_f} \tag{9.9}$$

$$v_e = g_0 \cdot I_{sp} \tag{9.10}$$

This equation shows that the maximum achievable ΔV depends solely on the propellant mass and the exhaust velocity. For a given rocket, it is physically impossible to exceed the ΔV predicted by this equation. Since there are two stages, each producing a certain amount of ΔV, the following two computed values will be evaluated in the total speed figure.

$$\Delta V_1 = I_{\mathrm{sp},1}\, g_0 \ln\left(\frac{m_{0,1}}{m_{0,1} - m_{p,1}}\right) = 4.165\, km/s$$

$$\Delta V_2 = \Delta V_1 + I_{\mathrm{sp},2}\, g_0 \ln\left(\frac{m_{0,2}}{m_{0,2} - m_{p,2}}\right) = 9.077\, km/s$$

In figure 9.6 below, this maximum possible ΔV is highlighted with the dotted red and blue line. Here it can be seen that the actual velocity of the rocket remains below the maximum achievable ΔV. Based on this verification criterion, the optimization model does not produce velocities that are physically impossible.

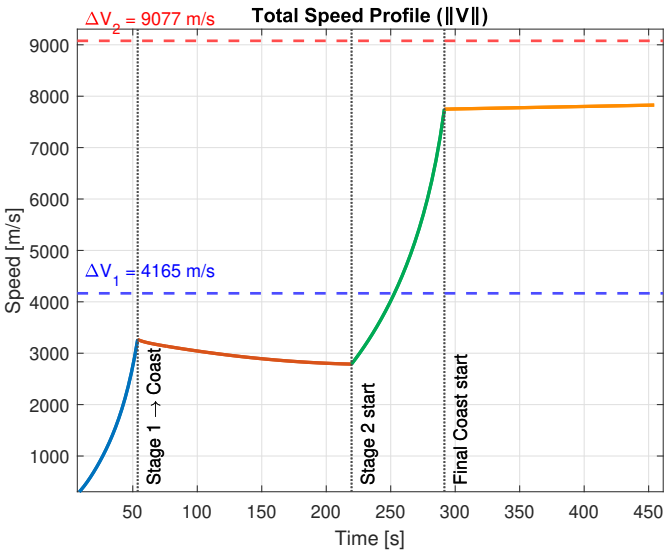


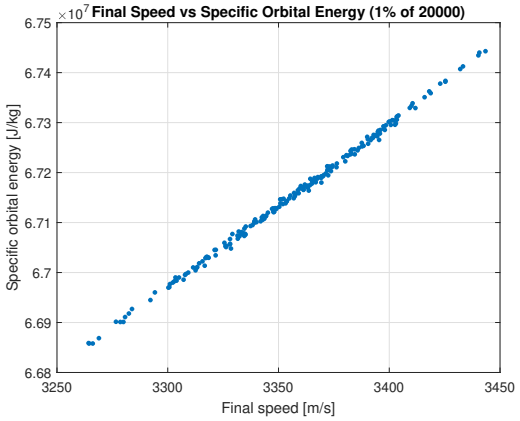
Figure 9.6: Maximum possible ΔV.

9.2.2. Optimality Verification

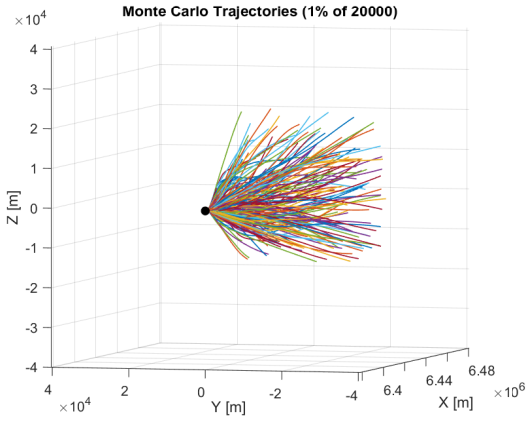
The optimization objective is to maximize the range of the booster. During the optimization process, some methods are used in order to define this optimization objective. These are described in chapter 6 and are based on the studied literature. However, as an additional confirmation, this section verifies several key assumptions used in the formulation of the optimization problem.

Maximum orbital energy

The first stage uses the premise that the maximum orbital energy also leads to the maximum velocity. This is verified with the use of a Monte Carlo simulation. Here, random control inputs are given for the angle of attack and the side slip angle. This follows the dynamics discussed in section 5.2.2. These trajectories are terminated after a certain amount of time. The final orbital energy and final velocity are plotted for all the terminal points of the different trajectories. This can be seen in the following figure.



(a) Orbital Energy - velocity plot.



(b) 3D view of the Monte Carlo trajectories.

Figure 9.7: Angle of attack and side slip angle Monte Carlo simulation

Here, the specific orbital energy is computed in the following way.

$$e = \frac{\mu}{|r_f|} + \frac{1}{2} \cdot |V_f|^2 \tag{9.11}$$

For the Monte Carlo study, 20,000 trajectories were generated. To preserve clarity in the figure, only a random 1% subset is displayed. The full dataset was used for all quantitative analyses, while the plotted subset is shown solely to improve the readability of the figures.

This shows there is a linear relation between the terminal orbital energy and the final velocity. Moreover, when varying key input parameters in the Monte Carlo simulation, such as booster characteristics or the HGV mass, the same linear relationship consistently emerges. Therefore, confirming that the assumption was correctly made.

Maximum velocity

The next verification examines the premise that, for trajectories ending in descent, a larger terminal velocity leads to a larger downrange distance. Physically, a higher terminal speed increases the trajectory's orbital energy and horizontal velocity, which generally produces a greater range.

Here again, a Monte Carlo simulation is made to verify this assumption. The simulation generates many ballistic trajectories by randomly sampling burnout speeds v_0 and flight path angles γ_0 at a fixed altitude. For each sample, the initial position and velocity vectors are used to compute the orbital elements, such as the semi-major axis a, eccentricity e, and true anomaly. Then the magnitude of the range and final velocities are compared. The terminal point is reached when the trajectory intersects with a certain set of constant altitudes. From this intersection, the range is determined using the angle between the burnout and intersect points. The terminal velocity V_f is obtained from the vis-viva equation, and the terminal flight path angle γ_f is computed from the orbital geometry. Only descending cases with $\gamma_f < 0$ are retained. In this way, the code outputs the relation between range, terminal velocity, and flight path angle for a large set of possible ballistic flights. Currently, the burnout point is 60km and the intersect point is 50km. Many more values were to be tested, however, the sampled case here already confirms the premise that $V_{\max}$ corresponds to $\mathrm{Range}_{\max}$ for trajectories that end in descent.

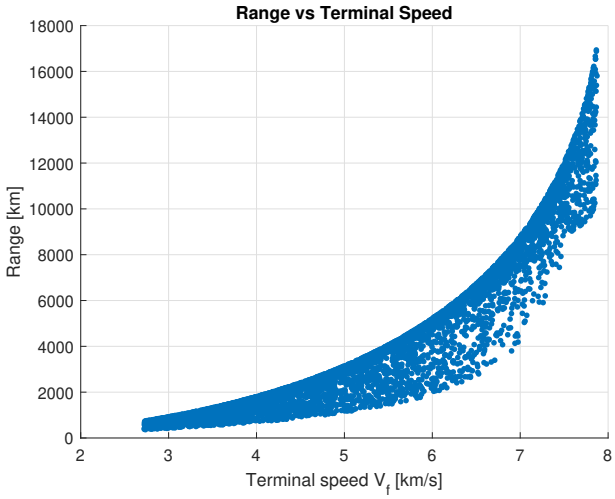


Figure 9.8: Maximum Range is Maximum Velocity.

This shows there is a clear relation between an increase in velocity and an increase in range. Thus, for the trajectories that end in descent, maximizing velocity also maximizes range. However, in the above figure the points have varying terminal flightpath angles. Therefore, it is clearer to analyze this verification result in a 3D plot with the terminal flightpath angle also on one of the axes. In the following plot, colors are also added for different values of the flightpath angle to make the figure more understandable.

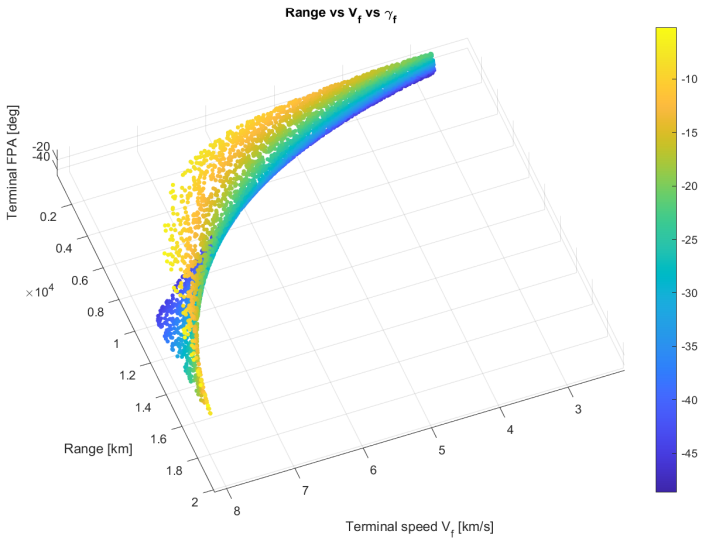


Figure 9.9: Range, Velocity, Flightpath angle Monte Carlo.

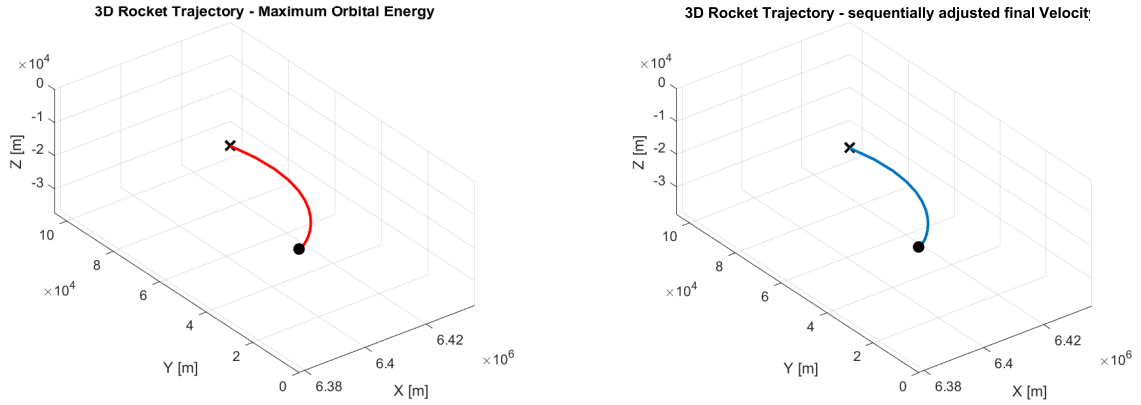
The flight-path angle in the 3D plot also confirms the statement made in the sensitivity analysis that a lower terminal flight-path angle does not automatically yield a larger range. It reflects a trade-off between a longer coast time (higher apogee) and increased horizontal speed. What effect dominates depends on the specific trajectory.

Furthermore, the result of this verification is in line with the statement by Gronlund *et al.* [73] that, for the suborbital trajectories considered, a maximum-range solution is similar to a minimum-energy solution, so the two objectives can be used interchangeably.

This verification shows that the maximum range is reached when the terminal velocity is maximized. Therefore, the assumption made for the vacuum solution optimization is correct.

Sequentially adapting the final velocity

The final verification compares two solution strategies: sequentially finding the maximum feasible terminal velocity and directly optimizing for maximum orbital energy. For the cases examined, both approaches produce equivalent trajectories, confirming the consistency of the numerical procedure described in Chapter 7. For different launch scenario's the two optimization methods are compared. In the figure 9.10 below, a single-stage scenario is highlighted.

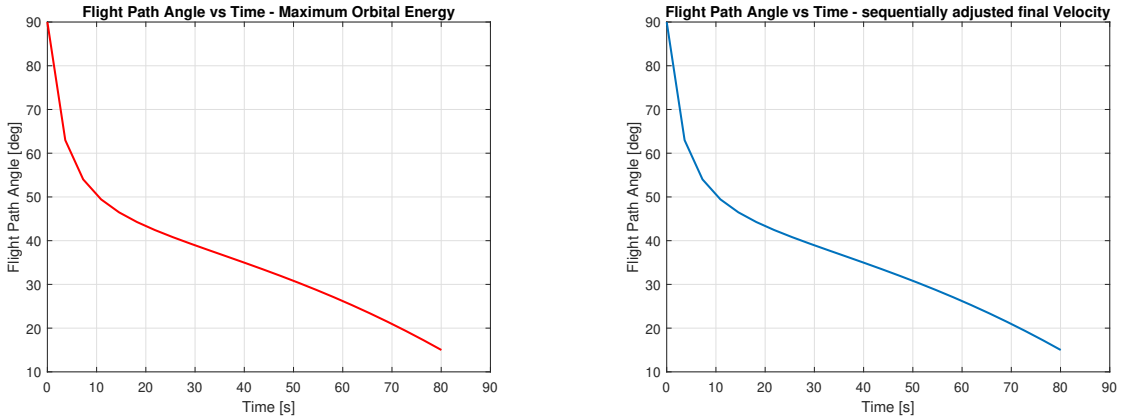


(a) Maximum Orbital Energy.

(b) Sequentially adapting the terminal velocity.

Figure 9.10: Verification of finding the maximum velocity

It can be seen that the trajectories result in the exact same results. In the next figure of the flightpath angle progression, it can be seen more clearly.



(a) Maximum Orbital Energy.

(b) Sequentially adapting the terminal velocity.

Figure 9.11: Flightpath angle comparison

This shows that both optimization strategies give the same result.

9.2.3. Numerical Verification

Lastly, two verifications are done of the numerical result that was found by the optimization model that was developed in this study. The goal of this verification is to try and use an alternative optimizer to see if it yields the same found solution as the optimizer of this study. The optimizer of this verification is therefore not required to be computationally fast. So, the two verification models used are run on the server of NLR. Here, 58 cores are available to run the optimization. As will be seen in the following sections, this additional computing power can be used to verify if the answer found is optimal.

Verification Vacuum Solution

The goal of this verification is to explore the full solution space of the vacuum problem described in Chapter 6 and to test whether the 'fmincon' solution is truly optimal or trapped in a local minimum. To do this, two global search methods were applied: simulated annealing and a genetic algorithm. These methods perform broad, population- or temperature-driven searches that are less prone to getting stuck in local minima than gradient-based optimizers. Both optimizers were executed on the NLR server using many cores to maximize the chance of finding a better solution if one exists.

Simulated annealing imitates the physical process of cooling a heated metal. The algorithm explores the design space by moving to nearby candidate solutions and occasionally accepts worse solutions so it can escape local minima. The willingness to accept worse solutions decreases over time according to a cooling schedule, so the search becomes more conservative as the algorithm progresses. A genetic algorithm maintains a population of candidate solutions that evolve over successive generations using selection, crossover, and mutation operators inspired by natural evolution. Promising solutions are preferentially selected to produce offspring, crossover recombines information between parents, and mutation introduces random variation to preserve diversity. Over many generations, the population tends to converge toward high-quality solutions.

Both methods were implemented in MATLAB, but the genetic algorithm was ultimately selected for the final optimization. The GA allows direct specification of nonlinear constraint functions, which avoids the need to define and tune penalty terms by trial and error. Developing the penalty function for the constraint functions, if done incorrectly, can quickly lead to the optimizer getting stuck in undesired local minima. This direct handling of constraints makes the GA approach more convenient and efficient for the current optimization problem.

The population size and maximum number of generations were deliberately set high to let the genetic algorithm explore the solution space as thoroughly as possible. Furthermore, the computing power of

the available cores of the NLR server allows for an increase in both population size and generations while still keeping the computation time moderate. This improves the chance of finding a high-quality solution.

The following figure shows the progress of the Genetic Algorithm.

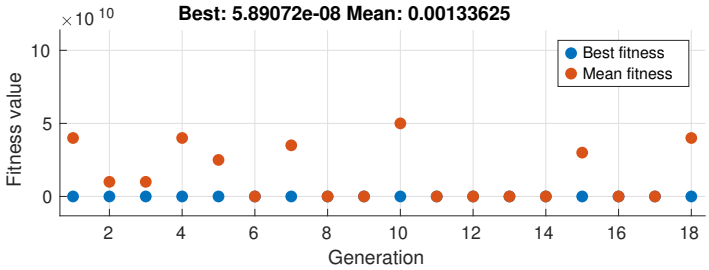


Figure 9.12: Progress Genetic Algorithm Optimizer.

The solution of the Kourou maximum range launch from chapter 8 was verified. The genetic algorithm returned the same vacuum solution as obtained with `fmincon`. This agreement increases confidence that the `fmincon` result is not a local minimum within the chosen formulation and bounds, since an independent global search reproduced the same solution.

Verification Atmospheric Solution

The atmospheric solution was verified using a Monte Carlo approach to remove dependence on any single initial guess. During development, it became apparent that the atmospheric optimizer frequently became trapped by the initial guesses, so a randomized search was applied to search the solution space more broadly.

In practice, entries of the `Y` array are randomly perturbed to generate alternative trajectories. These are the non-dimensional state and costate values at each shooting point, see Chapter 7. Because these perturbations are purely stochastic, the Monte Carlo runs are not biased by a particular seed trajectory and are far less likely to stall in the same local minimum repeatedly. The resulting trajectories are used to check whether lower objective values exist and to assess the robustness of the atmospheric optimization routine.

With this Monte Carlo approach, no better atmospheric solution was found. Unfortunately, due to time constraints no alternative method has been used to verify the atmospheric trajectory.

A complete comparison that jointly optimizes the atmospheric and vacuum phases was not performed, also due to time constraints. A full combined optimization remains future work and is recommended in the concluding recommendations for future research. For verification, a fully coupled verification would provide a more consistent evaluation of the transition between the atmospheric and vacuum phases and better capture how changes in one phase influence the other. This could lead to new insights into the trajectory optimality conclusions by confirming whether the separately verified atmospheric and vacuum phases together form a truly globally optimal trajectory.

9.3. Summary of the Sensitivity and Verification findings

This chapter demonstrates that the sensitivity analyses carried out follow the expected physical behavior and agree with examples from the literature. The sensitivity analysis not only confirms intuitive trends but also reveals less obvious couplings between design choices of the booster and missions planning choices of the HGV insertion. The optimization model is a useful tool for exploring booster design and HGV mission planning. Therefore, this also partly answers the last sub-question of this study. The indirect optimization method has proven to be quite effective for the use of a sensitivity analysis, because of its ability to quickly produce new trajectories and results. The non-stochastic nature of the indirect method makes it easy to repeat the same result. Furthermore, the fast computation time makes

it effective at iterating through different parameter values. However, while this local sensitivity approach provides valuable insight into how each parameter influences the trajectory near the nominal case, and how specific booster designs could be improved for the given ascent problem, it does not capture the full nonlinear behavior of the rocket ascent problem. The results cannot be directly generalized due to the high nonlinearity of the ascent optimization problem. A broader, global sensitivity analysis would be required to identify nonlinear interactions not captured in this study.

The verification steps confirm that the assumptions used in the optimization are consistent and that the computed trajectories are physically feasible. Therefore, the converged solutions of the optimization model produce trajectories that are physically possible. These verification steps simultaneously answer the first sub-question of this study. "*How can the dynamic model of the booster be verified to ensure the model is physically accurate?*". The used energy and impulse balance methods have proven to be effective at proving the physical accuracy of the model. Furthermore, the sensitivity analysis, different Monte Carlo simulations, and the evaluation of the made assumptions also showed to answer the second sub-question of the report: "*What methods can be used for the evaluation of the optimization problem?*".

The verification of the numerical solution demonstrates that the vacuum-phase result is not just a local minimum. Independent global searches reproduced the same solution, increasing confidence in its optimality within the chosen formulation. For future work, it is recommended to perform a full, coupled verification that jointly optimizes the atmospheric and vacuum phases, such a combined test would ensure optimality across the interface and capture any coupling effects between the two models. It is recommended to apply a completely different numerical approach from the one used in this study to ensure that the verification of the solution is fully independent. So, the numerical verification does help to verify the optimality of the found solution of the indirect optimization method, and thereby answer the last sub-question of this research: "*How effective is the implemented indirect optimization method for ascent trajectory planning in terms of sensitivity and its ability to find an optimal solution?*". Also, the literature evaluated in chapter 8 confirmed that found results show a trajectory that has the characteristics of a maximum range trajectory. Therefore, it helps to substantiate the numerical verification.

10

Conclusion

The objective of this study has been to compute the optimal ascent trajectory of the boost phase of a hypersonic glide vehicle using an indirect optimization method.

It has been shown that the developed optimization framework, which combines a vacuum-phase solution with aerodynamics introduced via a homotopy process, provides a robust numerical implementation of the problem formulated using Pontryagin's Maximum Principle. This hybrid approach keeps the structure of the indirect optimality conditions while allowing the solver to progressively introduce aerodynamic effects, yielding reliable and physically consistent trajectories.

The analytic structure of the indirect optimization comes with practical challenges. A principal drawback is its sensitivity to the initial guess. This issue was already found during the literature study. The solver must satisfy coupled state–costate boundary conditions and transversality relations. Combined with the increased sensitivity introduced by a long final coast, this can make convergence difficult. The optimization model uses a shooting and homotopy method in order to tackle this problem. A second downside is the relative complexity of the problem formulation itself. Deriving the costate equations, boundary conditions, and correct optimality conditions is complex compared to the problem formulation of other optimization methods. Mistakes in the mathematical setup easily produce infeasible or physically incorrect solutions. That complexity increases the implementation and debugging burden compared with many direct methods.

On the positive side, the indirect approach is very efficient once it converges. The method typically finds high-accuracy solutions far faster than many direct methods, which makes it attractive for use inside larger simulation environments. Another practical advantage is that path and terminal constraints are handled naturally through the optimality and transversality conditions. So, there is less need to design and tune ad hoc penalty functions to enforce feasibility.

The main research question of this study was:

- *How can an indirect optimization method be applied to compute the optimal ascent trajectory of an HGV boost phase while accommodating different booster models with variable thrust?*

The methodology presented in chapters 5, 6, and 7 thoroughly addressed this research question. These chapters elaborate on how the indirect method can be applied to compute the optimal ascent trajectory. Furthermore, chapter 8 gives the results following the presented methodology used in the MATLAB model. This clearly shows the model is able to compute an optimal ascent trajectory of an HGV boost phase while accommodating different booster models with variable thrust.

Combining coasting phases with variable-thrust capability in the indirect optimization presented here is novel for the literature of the HGV boost-phase optimization that has been found during this study. Furthermore, the inclusion of a final coast phase is also a new element of this study compared to the coast and angle of attack optimization literature discussed in Chapter 6.

Next, this work formulates a suitable constraint set and problem definition for the optimization problem that improves numerical convergence. Likewise, both the established terminal conditions and the formulation of complex aerodynamic models in an analytical manner, help the convergence of the numerical model. Also, a new strategy was required to compute the initial guess for the shooting points for the vacuum solution. The shooting points proposed in the referenced theory proved infeasible for the trajectories considered here, so they were replaced by a procedure adapted to the present problem formulation and boundary conditions.

Finally, the developed numerical model, based on a shooting method together with MATLAB's `fmincon`, provides substantially faster solution times than the modified-Newton approaches reported in the literature.

The result is that the optimization method produces trajectories that maximize range by coasting until close to apogee and then performing the final burn almost horizontally to maximize horizontal velocity. This behavior is consistent with the optimal-range strategy described in other work.

The three sub-questions in this research were:

1. *How can the dynamic model of the booster be verified to ensure the model is physically accurate?*
2. *What methods can be used for the evaluation of the optimization problem?*
3. *How effective is the implemented indirect optimization method for ascent trajectory planning in terms of sensitivity and its ability to find an optimal solution?*

These sub-questions have been addressed in chapter 9. This section has gone over the whole verification of the physical accuracy, the evaluation of the made assumptions, and the optimality of the found solution. This chapter also presented different ways to evaluate the made optimization and its results in, among other things, the sensitivity analysis. Because of this, the final three sub-questions, presented in chapter 3 of this study, have been answered.

Finally, the work presented in this study introduces a novel approach to optimizing the boost phase of hypersonic glide vehicles. Furthermore, due to the flexibility of the developed booster model and the ease with which terminal conditions can be adjusted, the optimization framework can also be readily applied to other near-space booster applications.

10.1. Recommendations for future research

Following the conclusions and the final remarks made in chapters 8 and 9, also some recommendations for future research are made in this section of the report.

10.1.1. Recommended Additions for Dynamical Model

Because this study uses an indirect optimization method, the amount of detailed environmental modeling that could be included was limited. At present, the model assumes a spherical Earth. An extension for future work is to include the oblateness term (the J_2 perturbation) in the equations of motion. The J_2 term can be incorporated into the indirect formulation as an additional perturbing acceleration and implemented using the polynomial or curve-fit gravity strategy introduced in Chapter 5.

Another physical effect not accounted for in the current optimization model is the variation of specific impulse with ambient pressure (altitude), which here has only been modeled indirectly through a prescribed thrust level. Nozzle performance changes with altitude (leading to over- or under-expansion of the exhaust) and alters engine efficiency. Including a model for this effect would improve the fidelity of the propulsion model. Implementing altitude-dependent specific impulse and nozzle flow correction in future work is recommended, as it can influence mass-flow, burn timing and ultimately the optimal staging and trajectory solutions.

10.1.2. Recommended Additions Numerical Optimization

As a recommendation for future research, an alternative approach for computing the initial guess of the shooting nodes for the analytical vacuum solution should be investigated. The present study was limited by time was not able to compare alternative methods. Yet the importance of a close initial guess for indirect optimization is well known. A more accurate initialization strategy would keep the solver close

to a feasible solution and substantially improve convergence reliability, especially for sensitive cases like with a long final coast. Furthermore, a tailor-made vacuum guess should be generated for the final coast-burn-coast solution, seen in figure 7.2. Currently, this model uses the same guess as the initial vacuum solution.

In addition, extending the homotopy procedure to also cover the entire first coast, instead of adding aerodynamic effects only to part of it, is recommended. A full coast homotopy that gradually introduces aerodynamic forces to the whole coast phase is expected to produce a smoother solution, which improves the convergence of the optimizer. Implementing these two improvements together should significantly increase the robustness of the indirect solver.

Lastly, the optimized atmospheric and vacuum solutions should be validated against alternative numerical methods to increase confidence in their optimality. Due to time constraints, this additional verification was not carried out here, but it is recommended for future work. One promising direction is reinforcement learning. It can serve both as an independent verifier and as a practical alternative to the indirect method used in this study. Reinforcement-learning approaches do not require derivation of costate equations or transversality conditions, and once trained they can produce optimal trajectories quickly. For these reasons, Reinforcement Learning is worth investigating as part of a future verification and method-comparison study.

10.2. Final Assessment of Compliance with Project Requirements

This project was carried out to meet the requirements set out by the stakeholder (NLR) in Chapter 4. The primary mandate was to develop an indirect (PMP-based) optimization model that computes optimal booster ascent guidance for prescribed terminal conditions, using the boost phase of an HGV as a representative case study. The work presented in this thesis satisfies these requirements as follows.

- **Arbitrary booster model:** The implementation uses a modular, arbitrary booster representation that can be replaced with minimal effort, meeting the stakeholder requirement for rapid substitution of vehicle models.
- **Physical verification:** Key physical effects required by the specification were modeled and validated. The model's physical accuracy was checked through verification steps, like energy balances, Monte Carlo perturbations, and alternative optimizer checks.
- **Staging and throttle:** Multi-stage boosters and variable thrust have been implemented and integrated into the optimization framework, fulfilling the stakeholder demand to include staging dynamics and throttle capability.
- **Mission objective:** The core objective: maximizing booster range while inserting an HGV under prescribed terminal constraint, was achieved and demonstrated in the numerical results.
- **Sensitivity analysis:** The computed trajectories were verified against independent checks, and a sensitivity analysis was performed. The findings have also been confirmed with examples found in the literature.
- **Prioritized features:** When reviewing the priorities of possible features, presented in chapter 3, it can be seen that all the desired features have been implemented in the optimization model.

In summary, the developed optimization framework meets the project requirements: it provides a validated, physically consistent, and flexible indirect optimization tool for booster ascent guidance and HGV insertion. Remaining recommendations for future work will further strengthen the applicability and robustness of the made model.

References

[1] Dan Hope. *DARPA readies hypersonic aircraft for mach 20 launch test*. Aug. 2011. url: https://www.space.com/12601-darpa-falcon-hypersonic-aircraft-launch-test-htv-2.html.

[2] Ding Yibo et al. “Review of control and guidance technology on hypersonic vehicle”. In: *Chinese Journal of Aeronautics* 35.7 (2022), pp. 1–18.

[3] Johannes Autenrieb et al. “Towards a Control-Centric Modelling and Simulation-Framework for Hypersonic Glide Vehicles”. In: *German Aeronautics and Space Congress (DLRK), Bremen, Germany*. 2021.

[4] Chegg. *Solved develop a conceptual design for a hypersonic boost*. Dec. 2023. url: https://www.chegg.com/homework-help/questions-and-answers/develop-conceptual-design-hypersonic-boost-glide-vehicle-uses-rocket-booster-accelerate-un-q125845884.

[5] Martin Sippel et al. “system investigations of the spaceliner concept in FAST20XX”. In: *17th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*. 2011, p. 2294.

[6] Yu Li, Naigang Cui, and Siyuan Rong. “Trajectory optimization for hypersonic boost-glide missile considering aeroheating”. In: *Aircraft Engineering and Aerospace Technology* 81.1 (2009), pp. 3–13.

[7] Kai An et al. “A framework of trajectory design and optimization for the hypersonic gliding vehicle”. In: *Aerospace science and technology* 106 (2020), p. 106110.

[8] Si-Yuan Chen and Qun-Li Xia. “A multiconstrained ascent guidance method for solid rocket-powered launch vehicles”. In: *International Journal of Aerospace Engineering* 2016.1 (2016), p. 11.

[9] M. Naeije and W. Simons. *Launch trajectories 1 - AE4870A Rocket Motion*. 2024.

[10] Yu Li and Nai-gang Cui. “Maximum crossrange for hypersonic boost-glide missile”. In: *2008 2nd International Symposium on Systems and Control in Aerospace and Astronautics*. IEEE. 2008, pp. 1–6.

[11] Yu Li and Nai-gang Cui. “Optimal attack trajectory for hypersonic boost-glide missile in maximum reachable domain”. In: *2009 International Conference on Mechatronics and Automation*. IEEE. 2009, pp. 4012–4017.

[12] Vijith Mukundan et al. “Trajectory optimization and guidance for solid motor propelled launch vehicle ascent”. In: *2019 American Control Conference (ACC)*. IEEE. 2019, pp. 3575–3580.

[13] J R Gottlieb et al. *Coupled Vehicle and Trajectory Design Optimization for Boost-Glide Hypersonic Systems*. Jan. 2024. url: https://arc.aiaa.org/doi/abs/10.2514/6.2024-0373.

[14] Sungjun Ann. “Exo-atmospheric Midcourse Guidance Law for Anti-ballistic Missile Using Response Surface Method”. In: *Seoul National University*. 2019.

[15] Yeontaek Jung. “Zero-Effort-Miss Midcourse Guidance Law Based on Predictor-Corrector for Solid-fuel Multi-stage Anti-ballistic Missile”. In: *Seoul National University*. 2019.

[16] Wenyu Chen et al. “Near space interceptor on-line correction research based on predictive intercept point”. In: *IOP Conference Series: Materials Science and Engineering*. Vol. 768. 4. IOP Publishing. 2020.

[17] Ning Li, Tao Zhang, and Wenyu Chen. “Trajectory optimization method of interceptor for high-speed gliding target via Hp-adaptive Radau pseudo-spectral method”. In: *IEEE Access* (2024).

[18] John A Lukacs IV and Oleg A Yakimenko. “Trajectory-shaping guidance for interception of ballistic missiles during the boost phase”. In: *Journal of Guidance, Control, and Dynamics* 31.5 (2008), pp. 1524–1531.

[19] Ge Wu et al. “Ascent trajectory optimization of hypersonic vehicle based on improved particle swarm algorithm”. In: *2015 Chinese Automation Congress (CAC)*. IEEE. 2015.

[20] Erwin Mooij. *Re-entry systems*. Springer Nature, 2024.

[21] National Aeronautics and Space Administration. *U.S. Standard Atmosphere 1976*. 1976.

[22] K Brown, E Harrold, and G Johnson. “Some new results on space shuttle atmospheric ascent optimization”. In: *Guidance, Control and Flight Mechanics Conference*. 1970, p. 978.

[23] W. Blake. *Missle DatCom User’s Manual - 2008 Revision*. 2008.

[24] E. Mooij. *Re-entry Guidance for Path-Constraint Tracking*. 2017. url: https://pure.tudelft.nl/ws/portalfiles/portal/20437837/AIAA_2017_1265.pdf.

[25] M. Tausche. *High Lift-over-Drag Earth Re-entry Executive Summary*. 2009. url: https://nebula.esa.int/sites/default/files/neb_study/995/C21735ExS.pdf.

[26] U. Walter. *Planetary Entry*. Springer, 2019.

[27] J. Bradt, M. Jessick, and J. Hardtla. *Optimal Guidance for Future Space Applications*. 1987.

[28] YU Qi, CHEN Wanchun, and YU Wenbin. “Approximate analytical solutions for three-dimensional ascent trajectory of a solid-fuel launch vehicle with time-varying mass flow rate”. In: *Chinese Journal of Aeronautics* 37.10 (2024), pp. 275–293.

[29] Anthony J Calise, Nahum Melamed, and Seungjae Lee. “Design and evaluation of a three-dimensional optimal ascent guidance algorithm”. In: *Journal of Guidance, Control, and Dynamics* 21.6 (1998), pp. 867–875.

[30] Fl Richard and B Christophe. “Open and closed loop guidance for an airbreathing winged launch vehicle”. In: *Acta Astronautica* 35.2-3 (1995), pp. 83–97.

[31] O. von Stryk and R. Zdenek Bulirsch. *Direct and Indirect Methods for Trajectory Optimization*. 1992. url: https://www.researchgate.net/profile/Oskar-Von-Stryk/publication/225872927_Direct_and_Indirect_Methods_for_Trajectory_Optimization/links/53da44480cf2631430c808f1/Direct-and-Indirect-Methods-for-Trajectory-Optimization.pdf.

[32] G.L. Watts and J.W. McCarter. *Missile Aerodynamics for Ascent and Re-entry*. Nov. 2012.

[33] Mathworks. *Comparison of 3-D Coordinate Systems*. url: https://nl.mathworks.com/help/map/choose-a-3-d-coordinate-system.html.

[34] SGB-systems. *Reference coordinate frames*. url: https://support.sbg-systems.com/sc/kb/latest/underlying-maths-conventions/reference-coordinate-frames.

[35] Kevin Cowan. *Fundamentals of Astrodynamics - AE4874*. 2023.

[36] *European Space Agency - Two-body Problem*. 2011. url: https://gssc.esa.int/navipedia/index.php/Two-body_Problem.

[37] Howard D Curtis. *Orbital mechanics for engineering students*. Butterworth-Heinemann, 2019.

[38] NASA Jet Propulsion Laboratory. *Astrodynamic Parameters*. url: https://ssd.jpl.nasa.gov/astro_par.html.

[39] Peter F. Gath and Anthony J. Calise. *Optimization of Launch Vehicle Ascent Trajectories with Path Constraints and Coast Arcs*. 2001.

[40] P. Lu. *ASSESSMENT OF ADAPTIVE GUIDANCE FOR RESPONSIVE LAUNCH VEHICLES AND SPACECRAFT*. Apr. 2009. url: https://apps.dtic.mil/sti/tr/pdf/ADA501767.pdf.

[41] *The World Book Encyclopedia*. Vol. 6. World Book Inc., 1984.

[42] Ping Lu, Lijun Zhang, and Hongsheng Sun. “Ascent guidance for responsive launch: a fixed-point approach”. In: *AIAA guidance, navigation, and control conference and exhibit*. 2005, p. 6453.

[43] Ping Lu, Hongsheng Sun, and Bruce Tsai. “Closed-loop endoatmospheric ascent guidance”. In: *Journal of Guidance, Control, and Dynamics* 26.2 (2003), pp. 283–294.

[44] J Sonderegger. “Vertical T tical Take-Off and Landing Contr e-Off and Landing Control via Dual-Quaternions and ol via Dual-Quaternions and Sliding Mode”. PhD thesis. Scholarly Commons, 2022.

[45] D Rose. *Rotation quaternions, and how to use them*. May 2015. url: https://danceswithcode.net/engineeringnotes/quaternions/quaternions.html.

[46] D Hoag. *Apollo Imu Gimbal Lock*. 1963. url: https://www.nasa.gov/history/alsj/e-1344.htm.

[47] Hanspeter Schaub and John L. Junkins. *Analytical Mechanics of Space Systems*. AIAA Education Series, 2003.

[48] NASA. *Speed of Sound*. url: https://www.grc.nasa.gov/www/k-12/BGP/sound.html.

[49] B.T.C. Zandbergen. *Thermal Rocket Propulsion (version 2.09)*. 2022.

[50] Jinlong Huang, Chunzhen Sun, and Wei Feng. "Ascent trajectory design method for air-breathing combined power hypersonic vehicle". In: *International Conference on Guidance, Navigation and Control*. Springer. 2022, pp. 3970–3980.

[51] M. Lin et al. *Multiconstrained ascent trajectory optimization using an improved particle swarm optimization method - lin - 2021 - International Journal of Aerospace Engineering - Wiley Online Library*. 2021. url: https://onlinelibrary.wiley.com/doi/10.1155/2021/6647440.

[52] Michael J Mall Grant and Ehsan Taheri. "Uniform Trigonometrization Method for Optimal Control Problems with Control Bounds and State Path Constraints". In: *Journal of Spacecraft and Rockets* (2020).

[53] G. Stolwijk. "A Hybrid Optimal Control Approach to Multi-Aircraft Formation Flying". Technical University of Delft, 2017.

[54] Kshitij Mall, Atharva Awasthi, and Daniel DeLaurentis. "Explaining Optimal Trajectories Using Indirect Methods and Explainable AI". In: *AIAA SCITECH 2024 Forum*. 2024, p. 2120.

[55] M. Soledad Aronna. *The Pontryagin maximum principle. Part II*. 2021.

[56] Yuan Li et al. *Optimal Control of Ascent Trajectory for Launch Vehicles: A Convex Approach*. Institute of Electrical and Electronics Engineers, 2017.

[57] B. Conway. *Spacecraft Trajectory Optimization*. Cambridge University Press, 2010.

[58] Willie Aboumrad. *Notes on Newton-Cotes quadrature - CME 108/MATH 114*. 2020. url: https://web.stanford.edu/class/math114/lecture_notes/nc_quad.pdf.

[59] Pan Binfeng and P. Lu. "Rapid optimization of multiburn rocket trajectories revisited". In: *AIAA Guidance, Navigation and Control Conference*. 2009.

[60] Greg Dukeman and Anthony Calise. "Enhancements to an atmospheric ascent guidance algorithm". In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 2003, p. 5638.

[61] D. Kirk. *Optimal Control Theory An Introduction*. Dover Publications Inc., 1998.

[62] Binfeng Pan et al. "Reduced transversality conditions in optimal space trajectories". In: *Journal of Guidance, Control, and Dynamics* 36.5 (2013), pp. 1289–1300.

[63] Anthony J Calise and Nico Brandt. "Generation of launch vehicle abort trajectories using a hybrid optimization method". In: *Journal of guidance, control, and dynamics* 27.6 (2004), pp. 929–929.

[64] Peter F Gath. *Improvements to a hybrid algorithm for rapid generation of 3-d optimal launch vehicle ascent trajectories*. diplom. de, 1999.

[65] L. Watson and R. Haftka. "Modern Homotopy Methods In Optimization". In: *Virginia Polytechnic Institute and State Unversity*. 1988.

[66] Josef Stoer et al. *Introduction to numerical analysis*. Vol. 1993. Springer, 1980.

[67] Peter Deuflhard. "A modified Newton method for the solution of ill-conditioned systems of nonlinear equations with application to multiple shooting". In: *Numerische Mathematik* 22.4 (1974), pp. 289–315.

[68] Kirk J Vanden and Paul D Orkwis. "Comparison of numerical and analytical Jacobians". In: *AIAA journal* 34.6 (1996), pp. 1125–1129.

[69] James M Acton. "Hypersonic boost-glide weapons". In: *Science & Global Security* 23.3 (2015), pp. 191–219.

[70] David Wright. “Research note to hypersonic boost-glide weapons by James M. Acton: analysis of the boost phase of the HTV-2 hypersonic glider tests”. In: *Science & Global Security* 23.3 (2015), pp. 220–229.

[71] Public Affairs Office Air Force Global Strike Command. Feb. 2019. url: https://www.af.mil/About-Us/Fact-Sheets/Display/Article/104466/lgm-30g-minuteman-iii/.

[72] Dongdong Yao et al. “Sliding Mode Formation Control for Multiple Hypersonic Glide Vehicles”. In: (2023).

[73] Lisbeth Gronlund and David C Wright. “Depressed trajectory SLBMs: A technical evaluation and arms control possibilities”. In: *Science & Global Security* 3.1-2 (1992), pp. 101–159.

[74] James A Nagy. *Flight performance data from the scout x-258 rocket motor*. Tech. rep. 1965.

[75] FMP Morgado, AC Marta, and PJS Gil. “Multistage rocket preliminary design and trajectory optimization using a multidisciplinary approach”. In: *Structural and Multidisciplinary Optimization* 65.7 (2022), p. 192.

[76] Nesrin Sarigul-Klijn et al. “Survey of planetary entry guidance algorithms”. In: *Progress in Aerospace Sciences* 68 (2014), pp. 64–74.

[77] Greg Dukeman. “Atmospheric ascent guidance for rocket-powered launch vehicles”. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 2002, p. 4559.

[78] Michael S Branicky, Vivek S Borkar, and Sanjoy K Mitter. “A unified framework for hybrid control: Model and optimal control theory”. In: *IEEE transactions on automatic control* 43.1 (1998), pp. 31–45.

[79] Martin SK Leung and Anthony J Calise. “Hybrid approach to near-optimal launch vehicle guidance”. In: *Journal of Guidance, Control, and Dynamics* 17.5 (1994), pp. 881–888.

[80] Christian Kirches. *Fast numerical methods for mixed-integer nonlinear model-predictive control*. Vieweg+Teubner Verlag / Springer Fachmedien Wiesbaden, Wiesbaden, 2011.

[81] H. Brunner. *Collocation methods for Volterra integral and related functional differential equations*. Cambridge University Press, 2004.

[82] S. Hartjes. “An Optimal Control Approach to Helicopter Noise and Emissions Abatement”. Technical University of Delft, 2015.

[83] Y Wang et al. *Optimal trajectory planning for trains – A pseudospectral method and a mixed integer linear programming approach*. 2013.

[84] Bailing Tian and Qun Zong. “Optimal guidance for reentry vehicles based on indirect Legendre pseudospectral method”. In: *Acta Astronautica* 68.7-8 (2011), pp. 1176–1184.

[85] GuoQiang Huang, YuPing Lu, and Ying Nan. “A survey of numerical algorithms for trajectory optimization of flight vehicles”. In: *Science China Technological Sciences* 55 (2012), pp. 2538–2560.

[86] B Goodman. *Sequential Quadratic Programming*. 2016. url: https://optimization.cbe.cornell.edu/index.php?title=Sequential_quadratic_programming.

[87] NASA. 2024. url: https://nssdc.gsfc.nasa.gov/planetary/factsheet/earthfact.html.

A

State-of-the-art Overview

Author	Research Objective	Used Optimization Method	Booster Model	Advantage	Disadvantage	Source
Yu Li et al	Optimizes the ascent trajectory of the boost phase to maximize the downrange distance.	Direct shooting method in combination with Sequential Quadratic Programming (SQP).	The Launch Vehicle is a Minuteman III, a three-stage solid rocket booster.	The used method showed to be effective in optimizing the performance of the HGV with nonlinear constraint functions.	No variable thrust possible in booster model. Therefore, only limited amount of booster models can be optimized. Furthermore, the direct optimization approach is difficult to perform sensitivity analysis on.	[6]
Yu Li et al	Optimizes the ascent trajectory of the boost phase to maximize the crossrange distance.	Direct shooting method in combination with Sequential Quadratic Programming (SQP).	The Launch Vehicle is a Minuteman III, a three-stage solid rocket booster.	The used method showed to be effective in optimizing the performance of the HGV with nonlinear constraint functions..	No variable thrust possible in booster model. Therefore, only limited amount of booster models can be optimized. Furthermore, the used optimization approach is difficult to perform sensitivity analysis on.	[10]

Yu Li et al	Optimizes the ascent trajectory of the boost phase to maximize the downrange distance.	Direct shooting method in combination with Sequential Quadratic Programming (SQP).	The Launch Vehicle is a Minuteman III, a three-stage solid rocket booster.	The used method showed to be effective in optimizing the performance of the HGV with nonlinear constraint functions.	No variable thrust possible in booster model. Therefore, only limited amount of booster models can be optimized. Furthermore, the used optimization approach is difficult to perform sensitivity analysis on.	[11]
Si-Yuan Chen et al.	The goal is to maximize the velocity at the solid rocket shutoff point.	The research uses a pseudospectral method to compute the reference trajectory. This is another direct optimization approach.	Solid-propelled booster with two stages.	The used optimization method showed to produce robust results in the presence of dispersion due to the closed-loop approach for the second stage of the booster.	No variable thrust possible in booster model. Therefore, only limited amount of booster models can be optimized. Furthermore, the closed-loop method is only used when aerodynamic forces are ignored. The numerical optimization approach could pose difficulties when implemented in an endo-atmospheric closed-loop system.	[8]
Kai An et al.	The goal is to minimize the terminal time of the boost phase, meaning the time required to reach a specified terminal height.	A numerical Legendre-Gauss-Radau (LGR) orthogonal collocation method is used.	The booster is defined to have no thrust variation.	More information about the trajectory can be obtained by influencing the density of collocation points.	No variable thrust possible in booster model. So, only limited amount of booster models can be optimized. Furthermore, the initial parameters of the state showed to influence the trajectory profile. The LGR orthogonal collocation method optimization approach is difficult for sensitivity analysis.	[7]
J.R. Gottlieb et al.	Optimizes both the ascent trajectory and the booster of the HGV.	A direct optimization approach using a Genetic Algorithm is applied.	The model employs a solid rocket booster.	The used method showed to be successful in the simultaneous vehicle and trajectory design optimization.	The numerical approach often resulted in long computation times. Therefore, the repeatability and precision of the final solution cannot be guaranteed. This makes sensitivity analysis and the implementation in more detailed simulation environments difficult.	[13]

S. Ann	Starting midcourse in exoatmospheric environment and minimizes a certain time to reach a terminal state.	An indirect optimization method is used to compute the reference guidance.	The booster is defined to have no thrust variation.	The used optimization method is effective to use for a robust reference trajectory.	It is not possible to implement a liquid propellant thruster. Furthermore, the indirect model used is not applied to a ground based ascent incorporating atmospheric effects.	[14]
Y. Jung	Starting midcourse in exoatmospheric environment and minimizes a certain time to reach a terminal state.	An indirect optimization method is used to compute the reference guidance.	The booster is defined to have no thrust variation.	The used optimization method is effective to use for a robust reference trajectory, which can be used for predictor-corrector guidance.	It is not possible to implement a liquid propellant thruster. Furthermore, the indirect model used is not applied to a ground based ascent incorporating atmospheric effects.	[15]
G. Wu	Optimizes the fuel usage of the ascent trajectory of a HGV	A particle swarm method is used for the trajectory optimization.	The booster is defined to have thrust variation and controllable angle of attack.	The particle swarm method has a big change of finding the global compared to other optimization methods.	Limited sensitivity analysis possible due to the stochastic nature of the optimization method. Also computation time is too slow for broader simulation implementations.	[19]
W. Chen	Optimizes the trajectory to intercept hypersonic targets.	Optimization is based of the direct pseudospectral method.	The rocket is a three stage solid booster.	The proposed algorithm showed to be effective for the proposed guidance algorithm.	No variable thrust possible in booster model. Therefore, only limited amount of booster models can be optimized. Furthermore, the numerical optimization approach could pose difficulties when implemented in an endo-atmospheric closed-loop system.	[16]

Table A.1: Extended summary of research on ascent trajectory optimization and booster models.

N. Li	Optimizes the trajectory to intercept hypersonic targets, using a combination of performance indicators like maximum terminal velocity, fuel consumption, and lateral range.	Optimization compares the pseudospectral method and hp-ARPM method.	Uses angle of attack, bank angle, and engine activation as control variable.	The paper shows a good comparison between the two direct optimization methods.	No variable thrust as control variable. Furthermore, the numerical optimization approach could pose difficulties for sensitivity analysis and closed-loop implementations.	[17]
J. Lukacs	Optimizes a combination of minimal time until reaching a terminal condition, maximize terminal velocity and path length of the trajectory of a ground-based Midcourse vehicle.	Optimization uses a direct-method-based approach that converts the optimization problem in a nonlinear programming problem.	The used rocket model has a fixed thrust profile.	Unlike gradient-based methods, which require accurate initial guesses, the used optimization method is not highly sensitive to initial conditions.	No variable thrust possible in booster model. Therefore, only limited amount of booster models can be optimized. Furthermore, the direct optimization approach is difficult to perform sensitivity analysis on.	[18]

Table A.2: Continued - Extended summary of research on ascent trajectory optimization and booster models.

B

Methodology Trade-off

This section presents methods that can be used to compute the optimal ascent trajectory. As described in the research objective, the goal is to determine which approach is the most effective in achieving an optimal solution, and to verify whether this solution is accurate. The methods discussed here can be used to determine this optimum solution and ultimately need to be evaluated based on their ability to find the global optimum while considering relevant constraints and computational efficiency.

B.1. Modeling approaches of the ascent trajectory

This section presents some reference frames and dynamic models that were not found in the literature presented in chapter 2. These models and reference frames provide potential other approaches to define the motion of the launch vehicle.

B.1.1. 3-DoF

To model the physical behavior of the booster, its motion is described using a 3 degrees of freedom (3 DoF) approach. This means the booster can translate freely along the three Cartesian axes (x, y, z), describing its position in space. In addition to its position, the orientation of the booster also plays an important role in defining its motion. The direction of the booster's velocity is characterized by the angle of attack and the flight path angle. The body frame is fixed to the rocket itself, with its axes aligned along the vehicle, the x-axis points along the rocket's nose. The Earth-centered inertial (ECI) frame, on the other hand, is a fixed, non-rotating frame centered on Earth, used to describe the trajectory relative to space. Further, it is also possible to use an Earth-centered Earth-fixed (ECEF) reference system, which is a non-inertial rotating reference system with the origin at the Earth center of mass.

Quaternions

A quaternion is a hyper complex number that can be used to describe the orientation of an object in a reference frame. How this works is further elaborated in chapter 5.2.3.

The reason for the use of this type of reference frame is to prevent singularities happening because of gimbal lock. Gimbal lock happens when the movement of a vehicle causes two of the three gimbal axes to become aligned. When this occurs, all three axes lie in the same plane, and the system loses one degree of freedom. This means that it can no longer rotate independently around a certain direction. Even though any vehicle orientation can still be represented by specific gimbal angles, the system is unable to smoothly adjust to certain changes in orientation once it reaches this locked state. This issue is one of the reasons why quaternions are often used instead of gimbals for tracking rotations, as they do not suffer from gimbal lock [46].

Generalized geocentric inertial coordinate system

This is a system where the coordinate of the vehicle is described by its heading and position on a sphere. In the GGI system, the motion of a rocket "M" is described by the generalized longitude λ, generalized latitude ϕ, and generalized heading angle ψ. As can be seen in the figure below.

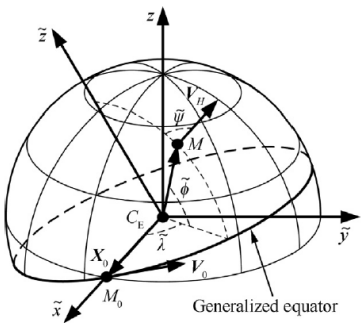


Figure B.1: Generalized geocentric inertial coordinate system.[28]

Here longitude λ describes the angle between the x-axis and the projection of vector X_m on Generalized Equatorial Plane (GEP). Here, X_m is the vector from C_E to M. Latitude ϕ, is the angle between X_m and the GEP. Lastly, the heading angle ψ, describes the heading of the spacecraft, by the angle of the local horizontal component V_H of the current velocity vector w.r.t. the local generalized meridian [28].

The equation of motion of the system are the following,in the work of Qi YU, 2024 [28]:

$$\frac{d\tilde{\lambda}}{dt} = \frac{V\cos\gamma\sin\tilde{\psi}}{R\cos\varphi} \tag{B.1}$$

$$\frac{d\tilde{\varphi}}{dt} = \frac{V\cos\gamma\cos\tilde{\psi}}{R} \tag{B.2}$$

$$\frac{dR}{dt} = V\sin\gamma \tag{B.3}$$

$$\frac{dV}{dt} = \frac{P}{m}\cos\alpha\cos\beta - g\sin\gamma \tag{B.4}$$

$$\frac{d\gamma}{dt} = \frac{P\sin\alpha}{mV} - \frac{g\cos\gamma}{V} + \frac{V\cos\gamma}{R} \tag{B.5}$$

$$\frac{d\tilde{\psi}}{dt} = \frac{P\cos\alpha\sin\beta}{mV\cos\gamma} + \frac{V\cos\gamma\sin\tilde{\psi}\tan\varphi}{R} \tag{B.6}$$

$$\frac{dm}{dt} = -q_m \tag{B.7}$$

Here, R is the distance between C_E and M. Velocity is V relative to the inertial space, m the mass of the system, γ the flight path angle, α is the angle of attack, β the side slip angle, P is the thrust, and g is the local acceleration of gravity. It is good to note that for these equations aerodynamic forces and Earth's rotation are neglected [28].

B.2. Indirect optimization approaches of the control variables

The advantage of analytical solutions is that it is simple to implement and the computation time is low. However, the downside is that the computed trajectory has possibly a lower fidelity and thus is less accurate compared to numerical formulations. The main benefit of analytical guidance is its straightforward approach and rapid computation, making it well-suited for the quick decision-making required in hypersonic entry. In contrast, numerical methods can be less reliable, as they may either fail to converge to a solution or require significantly more time to do so [76].

However, indirect optimization methods can also pose some difficulties. In the paper by Dukeman, the optimization problem is computed using calculus of variations by solving the states and co-states. However, no analytical solution can be found of his defined atmospheric state/co-state system. This is because of the high nonlinearity of the system. Dukeman therefore uses the Runge Kutta method with ten integration steps to obtain a solution for the guidance. Then, he applies the maximum principle to the Hamiltonian, and solves this numerically to find the optimal control of the system. Incorporating aerodynamic forces into the problem significantly increases its sensitivity and computational complexity. Unlike the vacuum case, where simple and well-defined formulas exist, the full optimization from liftoff to burnout becomes much harder to solve reliably in real-time [77].

Analytically addressed Optimal control problems are solved using indirect optimization methods, since this provides an analytical framework. The indirect method is usually based on the calculus of variations and Pontryagin's Maximum Principle, as mentioned in chapter 2. It establishes necessary conditions like adjoint equations and transversality conditions, finds optimal solutions by maximizing the Hamiltonian concerning the control, and transforms the original optimal control problem into a multi-point boundary value problem. It is important to note that a high-quality initial guesses of the adjoints are required, since the indirect methods can be sensitive the initial guess.

First a simplified scenario can be computed and then with hompotopy procedures the problem can be gradually made more complex. For instance, first a vacuum solution can be computed. Afterwards, with a homotopy procedure, atmospheric effects can be gradually introduced. The use of this homotopy, reduces the sensitivity of initial guesses of the co-states [77]. This will process will be further discussed later in the chapter.

B.2.1. Hybrid of analytical and numerical algorithm

There are also methods that solve optimal control problems with a hybrid of indirect and direct numerical methods. These methods could be feasible, as long as the computation time stays minimal. At the foundation of a Hybrid Optimal Control problem is a hybrid system, defined as a system that incorporates both continuous and discrete variables [78]. The use of a hybrid method, where a numerical optimization method is used in combination with an analytical method, has shown to be more suitable for real-time applications than a purely numerical approach [79]. A hybrid optimization method can be used when solving the state and co-state equations in Pontryagin's Maximum Principle becomes too complex or impractical. In such cases, numerical techniques are employed to approximate the solution, allowing for a more feasible optimization process while still having the advantages of PMP. Therefore, systems that are too complex to solve algebraically, such as those involving highly nonlinear aerodynamic force equations, can still be addressed using numerical methods.

There are many different numerical methods that can be used for a hybrid analytical numerical approach. Therefore, some important hybrid optimization methods that are used frequently in literature are discussed in this section.

Direct Shooting method

The direct shooting method is a technique used to solve problems by first making an initial guess for the starting conditions. It then simulates the system step by step, using numerical calculations to track the movement. If the outcome does not match the desired result, the starting conditions are adjusted, and the process is repeated until the correct solution is found. While this method is easy to implement, it works best for problems where high accuracy is not critical. This is because it is highly dependent on the initial guess. If the guess is far from the correct solution, the method may not converge, requiring adjustments or different starting points. Furthermore, it is possible that no solution is found due to singularities or numerical errors accumulating over integration. Highly nonlinear or unstable systems require an initial guess close to the optimal solution, which can be difficult to obtain [80].

Method of Collocation

Collocation is a method used to find an approximate solution to differential equations by dividing the time period into smaller intervals. The solution is represented using simple functions like polynomials within each interval. The unknown values in these functions are found by ensuring the solution is continuous at the points where the segments meet, and the time derivatives of the functions satisfy the differential equations at certain points within each segment [79]. In practice, the collocation method

is implemented by first selecting a grid that divides the problem's domain into sub-intervals. Within each sub-interval, a polynomial approximation is used to represent the solution. A set of collocation points is chosen, typically based on well-known numerical schemes such as Gauss, Radau, or Lobatto points. The differential equation is then enforced at these collocation points, generating a system of nonlinear algebraic equations. These equations are solved using numerical techniques such as Newton's method, direct shooting method or other iterative solvers. The method effectively transforms the continuous differential equation into a discrete numerical problem that can be efficiently handled by computers. By refining the grid or increasing the polynomial degree, accuracy can be improved, though this may also increase computational cost [81].

In [82], it is also noted that direct collocation enables the imposition of a complex set of path constraints. Furthermore, it is shown that Pseudospectral collocation method is the most effective and offers the greatest accuracy. The Legendre Pseudospectral Method is a collocation method, that combines elements of both indirect and direct numerical approaches, transforming the optimal control problem into a nonlinear programming problem for faster convergence. The user defines the equations of motion, constraints, and objectives based on the specific problem [76]. In the pseudospectral method, the state and control functions are approximated using Legendre polynomials based on interpolation at specific collocation points. This method is mainly well suited for approximating smooth functions, integrations and differentiations [83]. However, also this method still has the issue that a good initial guess greatly influences the rate of convergence and the probability of getting a good result [84].

Direct Multiple Shooting Method

A Variation of the presented methods above is the multiple shooting method. It is an enhanced shooting method, developed to address the stability and convergence issues that can arise, particularly in long or sensitive trajectory optimization problems. Instead of solving for the entire time interval from a single starting point, the method splits the differential equations into smaller intervals [85]. In each sub-interval, the shooting method is applied, treating each sub-interval as an independent boundary value problem. This approach is more stable than direct shooting because the problem is split into parts, reducing the accumulation of numerical errors. However, it comes with the trade-off of higher computational costs, and like the direct shooting method, a good initial guess is still essential for success [85]. In order to reduce the complexity of state and co-state a Multiple shooting method can be used. Specifically, using just one additional intermediate shooting point, along with the initial shooting point, can greatly decrease the sensitivity to the initial guesses for the co-states [77].

The Newton method can be used to iteratively solve a system of equations. Therefore, it is able to solve the problem of the collocation, and the multiple shooting method. Newton's Method iteratively refines a guess by adjusting it proportionally to the function's rate of change and inversely to its acceleration, ensuring convergence towards critical points by preventing overshooting and handling non-convex problems. The logic behind this approach is that a long, steep incline in a function will not be close to a critical point, so the improvement should be large. Further, a shallow incline that is rapidly expiring is likely to be near a critical point, so the improvement should be small [86].

Figure B.2, gives an example of how all these methods can be used to solve a hybrid optimization problem.

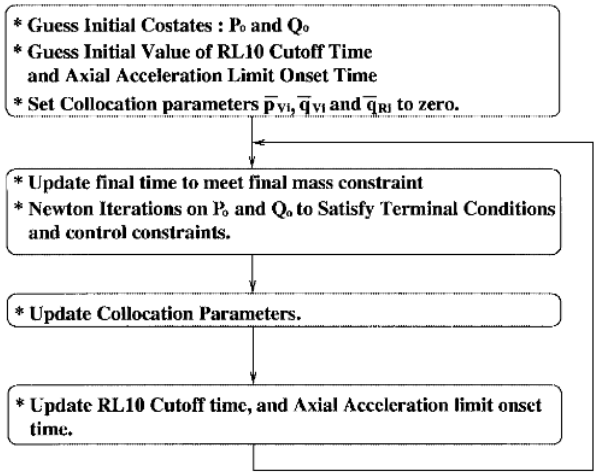


Figure B.2: Example of hybrid method computational procedure [29].

In Ref. [29], RL10 cutoff stands for the cutoff time of one of the boosters of the launch vehicle. It can be seen that A. J. Calise et. al. [29], used a collocation method in combination with a Newton method to iteratively solve the co-state equations that follow from the Hamiltonian defined in Eq. 6.5.

B.3. Control variables

Chapter 5.3 discussed the different control variables that can be used to shape the rockets trajectory. Different papers focus on optimizing differnt control variables.

For instance, the paper by P. Lu [42], optimizes the trajectory of a launch vehicle with three stages. Here an important focus is on optimizing the coast times. The burn times of the first two stages are fixed. However, the coast and burn times of the third stage need to be optimized. In the papers presented in this section [42] [59] [60], the mass flow rate is also part of the state equations of the dynamic model of the system. This means that also the throttle of the rocket is a control variable. Furthermore, in this presented example the state takes on the following form $\dot{x}(t) = [\dot{r}, \dot{V}, \dot{m}]$, and has the co-states $\mathbf{p}_r$, $\mathbf{p}_v$, $\mathbf{p}_m$ for the position, velocity and mass flow respectively.

B.4. Boundary Conditions

The optimization problem of the ascent trajectory of the boost phase is bounded by different kind of constraints. These constraints give information about the initial conditions and final conditions of the booster. Furthermore, it provides insights into the boundaries within which the booster's trajectory must remain. All these boundry conditions create what is called a boundary value problem. In previous sections in this chapter many different strategies are discussed to solve these kind of problems. Solving this optimal control problem with a shooting method combined with a homotopy procedure is a popular method to solve a boundary-value problem [77]. Later in this chapter, the homotopy procedure will be further elaborated. The rest of this section will go deeper into the initial conditions, final conditions and path constraints that the ascent trajectory of the booster is exposed to.

B.4.1. Initial Conditions

Logically, the trajectory of the booster will start with its initial conditions. The initial conditions of the booster state are fairly self-explanatory. Using a spherical non-rotation earth reference frame, the initial conditions of the booster can be described the following way:

$$\mathbf{x_0} = \begin{bmatrix} \mathbf{V_0} \\ \gamma_\mathbf{0} \\ \mathbf{h_0} \\ \mathbf{R_0} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \gamma_\mathbf{0} \\ \mathbf{0} \\ \mathbf{6378 \cdot 10^3 m} \end{bmatrix} \tag{B.8}$$

The initial conditions of the height and velocity are naturally zero. However, different ascent trajectories might require a different initial condition for the flight path angle (γ_0). Further, the initial condition for R_0 is equal to the radius of the earth which is $6.378km$ [87].

For the co-states the initial conditions are a little bit more complicated. When the initial conditions of the state are fixed it means that the initial conditions of the co-state are free and therefore yet undetermined [61]. This also follows from the equation for the initial condition of the co-state of Eq. 6.8. It is important to be aware of this when working with the Pontryagin's Maximum Principle, since it affects the process of solving the state and co-state equations.

B.4.2. Terminal Conditions

The boost phase also has certain terminal conditions. These are the final conditions that the state of the booster need to satisfy. For instance, in order to properly insert an HGV in its re-entry trajectory or to reach a certain target at a specified velocity. This means that the boost phase has certain fixed terminal conditions. Similarly like the initial conditions, when one of the parameters of the state has fixed terminal conditions, it means that the co-state of this parameter is free. Further, if the terminal condition of a certain state is free, it means that the terminal condition of this co-state is zero. This all follows from the transversality conditions presented by Eq. 6.9. From this equation it also follows that when it is desired to minimize or maximize a certain terminal value the state, the co-state of this parameter has a value of one at the terminal condition [61].

For the specific case of a Hypersonic Glide Vehicle, the terminal conditions of the booster are influenced by the initial conditions of the HGV's descent maneuver. Therefore, it must begin its maneuver from an optimal entry point, requiring certain initial conditions for a successful skip and glide descent. These conditions like, height, velocity, and flight path angle, are determined by the terminal state of the boost phase. Once separated from the booster, the HGV first undergoes a ballistic re-entry since it is released in exoatmospheric conditions. Like previously discussed in chapter 2.2, to ensure a controlled descent, the HGV must remain within its designated re-entry corridor. Otherwise, excessive heat and mechanical stress could lead to failure. While re-entry constraints, both soft and hard, have minimal impact during the exoatmospheric phase, proper insertion remains critical. If the vehicle enters with excessive velocity or an incorrect flight path angle, it may be unable to execute its intended maneuver upon reaching atmospheric conditions, risking thermal or structural failure. The limits of these constraints and the optimal conditions for the HGV to initiate its glide-skip trajectory will differ per mission and per vehicle. Dependent on this, the boost phase of the HGV will require certain terminal conditions for the height, velocity and flight path angle to facilitate the proper insertion of the HGV into its descent trajectory.

B.4.3. Constraints

Apart from the initial and final conditions being constraint, it can also be the case that the trajectory of the rocket is bounded by certain path and control constraints. These define the maximum control actuation that the booster is able to deliver and between what bounds the ascent trajectory needs to be. In chapter 5.5, these different constraints have been further elaborated.

B.5. Homotopy

Homotopy is a technique used for solving complex optimization problems. Here, a simpler, more easily solvable problem is gradually transformed into the complex problem of interest. For instance, first a vacuum solution can obtained, after which atmospheric effect can be introduced to compute the real solution. Homotopy procedures can be especially helpful when solving problems including atmospheric effects, because the non-linearity makes solving the optimal solution challenging.

An example of such a homotopy method is achieved by introducing a homotopy parameter λ, which transitions between a simplified version of the problem ($\lambda = 0$) and the full optimization problem ($\lambda = 1$). As λ increases in small increments, additional complexities are introduced [65]. For ascent optimization, this approach is beneficial when applying Pontryagin's Maximum Principle and solving co-state equations using the Shooting Method or Newton's Method. Homotopy can therefore be used to help achieve the final research objective, by starting to solve a simplified optimization problem and gradually transforming it into the desired optimization problem.

B.6. Methodology Trade-off

A trade-off has been made between the different papers that propose methods to solve the optimization problem. This includes the theory and methodology discussed in Chapters **??** and **??**. The trade-off is based on several features that are important for this research, which are summarized and prioritized in Table 4.1 in Chapter 4. In short, the comparison looks at which features each method supports and how well they fit with the overall goals of the research. These features reflect the different aspects that can be included in the optimized trajectory.

In the table the different features can be seen, that all are given a score for each of the studied papers. The worst score is presented with symbol ”–”, the second to worst score with ”-”, the second to best score wit ”+”, and the best possible score is given by ”++”. The meaning and scoring of all the different features will be further discussed after table B.1.

Table B.1: Trade-off table of papers on modeling approaches, control variables, and optimization techniques.

Paper	Guidance Method	Features								
		Dynamic modeling			Control variables			Optimization method		
		Rotating Earth	Aero-dynamic Com-plexity	Con-straints	Variable Thrust	Staging	Angle of Attack	Sensi-tivity Analy-sis	Closed-loop Possi-bility	Comp. Effi-ciency
Yu Li [6]	Four order Runge-Kutta intergration with SQP.	–	++	++	–	–	++	-	+	-
Yu Li [10]	Four order Runge-Kutta intergration with SQP.	++	++	++	–	–	++	-	+	-
Yu Li [11]	Four order Runge-Kutta intergration with SQP.	++	++	++	–	–	++	-	+	-
Si-Yuan Chen [8]	Pseudospectral method.	–	-	++	–	++	++	-	+	-
K. An [7]	LGR collocation method.	–	++	++	–	–	++	-	+	-
J. Gottlieb [13]	Genetic Algorithm.	–	++	++	–	++	++	–	–	–
S. Ann [14]	Indirect optimization method.	++	–	++	–	–	++	+	++	+
Y. Jung [15]	Indirect optimization method.	++	–	+	–	++	++	+	++	+
W. Chen [16]	Pseudospectral method.	–	++	++	–	–	++	-	+	-
N. Li [17]	Pseudospectral & Hp-arm method.	–	++	++	–	–	++	-	+	-
J. Lukacs [18]	Directmethod based approach.	–	–	+	–	–	++	-	+	-
G. Wu [19]	Particle Swarm method.	–	++	++	++	–	++	–	–	–
G. Dukeman [77]	Indirect optimization with shooting method.	++	+	++	–	–	++	+	++	+
G. Dukeman [60]	Indirect optimization with shooting method.	–	-	++	–	++	++	+	++	+
K. Mall [54]	Indirect optimization method, with a Giuseppe solver.	–	++	-	–	–	++	+	++	+
B. Pan [59]	Indirect optimization method, with multiple shooting method.	–	–	+	–	++	++	+	++	+
P. Lu [43]	Indirect optimization method, with forward difference.	++	++	++	++	–	++	+	++	+
P. Lu [42]	Indirect optimization method, with multiple shooting method.	–	++	-	–	++	++	+	++	+

A. Calise [29]	Hybrid analytical and numerical approach.	++	++	++	–	–	++	+	++	+
M. Leung [79]	Hybrid analytical and numerical approach.	–	++	+	–	–	++	+	++	+

First, there are the different features related to the dynamic modeling of the booster. This includes the level of detail used to represent the booster's environment. For example, regarding the inclusion of a "Rotating Earth," the highest score (++) is given when Earth's rotation is included in the model, and the lowest score (–) is assigned when it is neglected. Next, the feature "Aerodynamic complexity" refers to how detailed the modeling of aerodynamic forces is. The lowest score is given when no aerodynamic forces are included. A slightly better score is assigned when aerodynamic coefficients are assumed to be constant. A higher score is given when the coefficients depend on parameters like Mach number and velocity. The best score is awarded when aerodynamics are modeled as a function of both Mach number and angle of attack. Finally, the treatment of constraints on the booster is considered. The lowest score is given when no constraints are applied to the ascent. A slightly better score is assigned when only initial and terminal conditions are defined. A higher score is given when either path or control constraints are included. The highest score is reserved for cases where both path and control constraints are present in the model.

Then there are the different control variable features: variable thrust, staging, and angle of attack. In the table, the lowest score is given when a specific type of control variable is not used or optimized in the paper. Conversely, the highest score is given when the control variable is both used and optimized in the study. It should be noted that variable thrust inherently allows for the possibility of coasting, since the thrust control variable can be reduced to zero, meaning the booster is coasting. Staging, as a control variable, also enables coasting, but with the added feature that a stage of the booster is separated during this phase to reduce the overall mass of the vehicle.

For the feature of sensitivity analysis, the lowest score is given to direct optimization methods that typically have a high computational load. These methods often lack direct mathematical expressions to describe how changes in input affect the output, making sensitivity analysis difficult. The next lowest score is assigned to direct methods that do allow for some mathematical expression of sensitivity, such as pseudospectral methods, which use KKT conditions. However, the sensitivity can vary between time steps or collocation points, making the analysis less consistent. A higher score is given to indirect methods that are solved numerically. These methods have explicit expressions for sensitivities, though they still need to be evaluated using numerical techniques. An advantage here is that, unlike in pseudospectral methods, the sensitivities do not vary across different points in the solution. The highest score is given to indirect methods that are solved fully analytically. In these cases, sensitivities can be determined directly by analytically evaluating the explicit expressions, offering consistent and immediate results. However, such fully analytical solutions are not found in the literature presented in the table.

For the feature of sensitivity analysis, the worst score has been given to direct optimization methods that generally have a high computational load. For these method usually there are no direct mathematical expressions that can describe the sensitivity of certain input on output values. The second to worst score is given to direct methods that do have a possibility of mathmetically expressing the sensitivty, like pseudospectral methods (through KKT-conditions), however, can have a different sensitivities at different iterations step. Like with the pseudospectral method the sensitivity can differ per collocation point. The second to best score is given to indirect methods that are solved with numerical methods. Here, explicit expression are present of the sensitivities, however, also needs to be solved using numerical methods. The upside is, with this method there are no collocation points that have different sensitivities. The best score is given to indirect methods that are solved full analytically. Here, sensitivities are also instantly found through analtucally calculating the present explicit sensitivity expressions. The latter is only not present in the literature in the table.

When considering closed-loop possibilities and computational efficiency, comparing specific papers and methods becomes challenging due to the variety of factors influencing computational time and load. These factors include available computing power and the complexity of the dynamics involved.

Rather than comparing individual methods directly, it is more effective to evaluate how each method inherently addresses the optimization problem. In general, it is more computationally efficient to define and solve a problem analytically than to search for an optimal solution at every integration step. As a result, nonlinear programming (NLP) problems are typically less efficient than indirect methods. Indirect methods leverage the optimality conditions directly, solving the analytically derived system for the necessary conditions for optimality, which reduces the problem to a root-finding issue. In contrast, NLP methods approximate the optimal conditions at each time step, which adds computational complexity. However, indirect methods are not the only viable option for more complex closed-loop simulations. Depending on the specific problem, particularly if there are many state and co-state equations to solve, numerical methods can also lead to longer computation times as well. Despite this, the potential of indirect methods to solve problems as analytically as possible makes them inherently promising for computational efficiency. In the evaluation table, the highest score is reserved for guidance laws that approach the optimal trajectory fully analytically through a closed-form solution. However, none of the methods presented in the reviewed literature achieve a completely analytical approach. Therefore, the next highest score is given to solutions that are partially analytical, so indirect methods combined with numerical solving techniques. On the other end of the scale, the lowest score is assigned to direct methods with high computational demands, such as genetic algorithms or particle swarm optimization. In between these extremes are direct methods that iteratively converge to an optimal solution but are not as computationally intensive as the aforementioned techniques. Furthermore, due to the higher computational load of some direct optimization methods, they are difficult to implement in closed-loop simulation environments. As a result, these methods receive the lowest score for the feature ”Closed-loop possibilities”. A better score is given to direct methods that have been shown in literature to be suitable for closed-loop guidance. The highest score is awarded to indirect methods, thanks to their generally higher computational efficiency, which makes them more suitable for implementation in closed-loop guidance without significantly affecting the overall computation time of the simulation.

B.6.1. Conclusion of methodology

From table B.1 it can be seen that there is no clear presented methodology used in the studied literature that is able to perfectly model and optimize all desired features. Therefore, the research has to create a combination of the different methodologies, and pick the desired features that are presented in the different studies.

First, the papers by S. Ann [14], Y. Jung [15], and P. Lu et al. [43] [42] demonstrate promising results for the optimization methods used. In terms of closed-loop applicability, computational efficiency, and sensitivity analysis, the indirect optimization methods applied in these studies stand out as particularly effective. As such, the indirect approach presented in these papers provides a solid foundation for selecting an appropriate optimization method for this research.

Furthermore, the paper by G. Dukeman et al. [60] offers a useful strategy for incorporating staging into the optimization process. Meanwhile, P. Lu et al. [43] present a strong method for implementing variable thrust as a control input. A combination of insights from these papers forms a strong starting point for developing the methodology to implement control variables within this research.

Additionally, the same work by P. Lu et al. [43] provides a comprehensive framework for modeling all the desired features of the booster dynamics while applying an indirect optimization method that performs well across the evaluated features.

In conclusion, the indirect optimization method discussed in Section B.2 proves to be well-suited for achieving the desired features of this research, presented in table B.1. The approach presented by P. Lu et al. [43] serves as a strong example of how this method can be effectively implemented. Furthermore, the control variable features described in Section B.3 are well addressed in the previously mentioned literature, providing valuable guidance for incorporating them into the current study.

B.6.2. Key considerations

Several key considerations must be addressed for the continuation of the research. First, it is crucial to account for gimbal lock. This can significantly impact simulations depending on the type of reference frame and dynamic model used. When gimbal lock is a concern, careful selection of the initial conditions for the booster simulation is essential. Incorrect initial conditions can lead to singularities, which may

result in the failure of the simulation. Singularities are especially common in coordinate systems such as spherical coordinates. In such cases, slight adjustments to the initial conditions might be necessary to avoid these issues.

The implementation of certain features can limit the ability to include others, especially when specific performance of the optimization method is required. For example, introducing staging as a control variable may cause the solution to become discretized, due to the sudden drop in mass when a stage is jettisoned. This can make it difficult to find a single converged solution to the optimal control problem. In other words, certain features can influence each other and affect the overall outcome of the optimization.

Additionally, the nonlinearity of aerodynamic forces is an important factor to consider. This nonlinearity can complicate the use of indirect optimization methods, as it may prevent the problem from being solved analytically. To address this, a homotopy procedure can be employed. This approach gradually increases the complexity of the rocket's dynamic model and aerodynamics within the simulation. By doing so, it helps identify the limitations of the indirect optimization method and allows for the determination when a hybrid optimization approach might be required.

C

MATLAB Code-snippets

C.1. Optimal Angle of Attack Computation

```
function alpha = Optimal_AoA(params, mk, Phi, Vr_nd, rho, CN_alpha, CA_alpha, alpha_prev)
    params.homotopy_pathcon = 1;

    alpha_max = deg2rad(params.alpha_max);              % hard AoA cap
    alpha_step_max = deg2rad(params.alpha_step_max_deg);

    % --- dimensionless thrust & speeds ---
    T      = params.T_1 / (mk * params.g0);             % dimensionless thrust
    Vr_mag = norm(Vr_nd);                               % nondim relative speed magnitude

    % --- Newton solve for unconstrained alpha_star ---
    alpha_star = 0;
    for iter = 1:10
        C_N = CN_alpha * alpha_star;
        C_A = CA_alpha + C_N * sin(alpha_star);

        % dC_A/dalpha and d^2C_A/dalpha^2 (chain rule applied)
        C_A_alpha       = CN_alpha * sin(alpha_star) + CN_alpha * alpha_star * cos(
            alpha_star);
        C_A_alpha_prime  = 2*CN_alpha * cos(alpha_star) - CN_alpha * alpha_star * sin(
            alpha_star);

        qbar_nd  = 0.5 * rho * Vr_mag^2 * params.S_ref;   % nondimensional dynamic pressure *
             area term

        N        = qbar_nd * C_N / (mk*params.g0);
        N_alpha  = qbar_nd * CN_alpha / (mk*params.g0);
        A        = qbar_nd * C_A / (mk*params.g0);
        A_alpha  = qbar_nd * C_A_alpha / (mk*params.g0);
        A_alpha2 = qbar_nd * C_A_alpha_prime / (mk*params.g0);

        F  = tan(Phi - alpha_star)*(T - A + N_alpha) - (A_alpha + N);
        dF = -tan(Phi - alpha_star)*A_alpha - A_alpha2 - N_alpha ...
             - (1 / (cos(alpha_star - Phi)^2)) * (T - A + N_alpha);

        alpha_star = alpha_star - F / (dF + 1e-12);
    end

    % --- hard cap first ---
    alpha_star = min(max(alpha_star, -alpha_max), alpha_max);

    % --- dynamic-pressure * alpha constraint ---
    Vr_dim_mag = Vr_mag * params.V_scale;          % dimensional speed magnitude
    q_dyn      = 0.5 * rho * Vr_dim_mag^2;         % [Pa] if rho in kg/m^3 and V_scale in m/s

    if abs(q_dyn * alpha_star) <= params.Q_alpha
        alpha_constrained = alpha_star;            % constraint inactive
```

```
    else
        % project to active constraint, with homotopy on the projection
        alpha_q = params.Q_alpha / q_dyn;
        alpha_constrained = sign(alpha_star) * min(alpha_q, alpha_max) * params.
            homotopy_pathcon ...
                        + (1 - params.homotopy_pathcon) * alpha_star;
    end

    % --- per-step rate limiting (max Δ|alpha| per node step) ---
    if nargin < 8 || isempty(alpha_prev)
        alpha_ratelimited = alpha_constrained;
    else
        delta = alpha_constrained - alpha_prev;
        if delta >  alpha_step_max, delta =  alpha_step_max; end
        if delta < -alpha_step_max, delta = -alpha_step_max; end
        alpha_ratelimited = alpha_prev + delta;
    end

    % --- final safety clip to alpha_max ---
    alpha = min(max(alpha_ratelimited, -alpha_max), alpha_max);
end
```

C.2. Analytical Vacuum Solution - Burn Arc Propagation

```
function [R_tot_1, V_tot_1, pV_tot_1, pR_tot_1] = trajectory_tot_stage1(params, Lambda_opt)
    % Settings
    N = params.N;  % Number of time steps
    tau_final =  params.t_burn_1 / params.T_scale;
    tau_vec = linspace(0, tau_final, N+1);

    % Storing of values
    r_store = zeros(3, N+1);
    v_store = zeros(3, N+1);
    pV_tot_1 = zeros(3, N+1);
    pR_tot_1 = zeros(3, N+1);

    % Initial conditions
    r0 = params.r_0;
    v0 = params.v_0;
    X0 = [r0; v0];
    pV0 = Lambda_opt(1:3);
    pR0 = Lambda_opt(4:6);
    I = eye(3);

    % Precompute thrust integrals incrementally
    m0 = params.m0_1;
    m_dot = params.m_dot_1;
    g0 = params.g0;
    T = params.T_1;

    Ic = zeros(3,1);
    Is = zeros(3,1);

    for i = 1:N+1
        tau = tau_vec(i);
        % Costate rotation
        Omega_tau = [cos(tau)*I, sin(tau)*I; -sin(tau)*I, cos(tau)*I];
        Lambda_tau = Omega_tau * [pV0; -pR0];
        pV_tau = Lambda_tau(1:3);
        I_p = pV_tau / norm(pV_tau);
        m_tau = m0 - m_dot * tau;
        T_tau = T / (m_tau * g0);

        Ic_dot = zeros(3, 5);
        Is_dot = zeros(3, 5);
        h = tau / 4;
        for k = 1:5
            Ic_dot(:,k) = I_p * cos(tau) * T_tau;
            Is_dot(:,k) = I_p * sin(tau) * T_tau;
        end

        tau_grid = linspace(0, tau, 5);
        h_local = tau / 4;
        Ic_dot = zeros(3, 5);
        Is_dot = zeros(3, 5);

        for k = 1:5
            t_k = tau_grid(k);
            Omega_k = [cos(t_k)*I, sin(t_k)*I; -sin(t_k)*I, cos(t_k)*I];
            Lambda_k = Omega_k * [pV0; -pR0];
            pV_k = Lambda_k(1:3);
            I_p_k = pV_k / norm(pV_k);
            m_k = m0 - m_dot * t_k;
            T_k = T / (m_k * g0);
            Ic_dot(:,k) = I_p_k * cos(t_k) * T_k;
            Is_dot(:,k) = I_p_k * sin(t_k) * T_k;
        end

        Ic = (2 * h_local / 45) * (7*Ic_dot(:,1) + 32*Ic_dot(:,2) + 12*Ic_dot(:,3) + 32*
            Ic_dot(:,4) + 7*Ic_dot(:,5));
        Is = (2 * h_local / 45) * (7*Is_dot(:,1) + 32*Is_dot(:,2) + 12*Is_dot(:,3) + 32*
            Is_dot(:,4) + 7*Is_dot(:,5));

```

```
68          % Propagate state
69          Omega_tau = [cos(tau)*I, sin(tau)*I; -sin(tau)*I, cos(tau)*I];
70          Gamma_tau = [sin(tau)*I, -cos(tau)*I; cos(tau)*I, sin(tau)*I];
71          X_tau = Omega_tau * X0 + Gamma_tau * [Ic; Is];
72          r_store(:, i) = X_tau(1:3);
73          v_store(:, i) = X_tau(4:6);
74
75          Lambda_tau = Omega_tau * [pV0; -pR0];
76          pV_tau = Lambda_tau(1:3);
77          pR_tau = -Lambda_tau(4:6);
78
79          pV_tot_1(:, i) = pV_tau;
80          pR_tot_1(:, i) = pR_tau;
81      end
82
83      % Convert to dimensional
84      R_tot_1 = r_store;
85      V_tot_1 = v_store;
86  end
```

C.3. Analytical Vacuum Solution - Coast Arc Propagation

```
r0 = Lambda_opt(8:10) * params.L_scale;
v0 = Lambda_opt(11:13) * params.V_scale;
pV0 = Lambda_opt(14:16);
pR0 = Lambda_opt(17:19);

t0 = 0;
t_final = Lambda_opt(7)*params.T_scale;               % Total coast time [s]
tau_final = Lambda_opt(7);   % Dimensionless coast time
dt_step = t_final / params.N;                % Spacing between points [s]
dtau_step = dt_step / params.T_scale;

%%
% Angular momentum:
h_vec = cross(r0, v0); % Vector
h_mag = norm(h_vec);

r_mag_0 = norm(r0);

% Eccentricity
e_vec = (cross(v0, h_vec)/params.mu) - r0/r_mag_0;
e = norm(e_vec);

% Semi major axis (from Vis-viva)
a = (params.mu/2) / ((params.mu/r_mag_0) -  (norm(v0)^2/2));

% Mean motion (average angluar rate of motion along a keplerian path)
n = sqrt(params.mu / a^3);

p = h_mag^2/params.mu;

% Unit vectors
e_unit = e_vec / norm(e_vec);
h_unit = h_vec / norm(h_vec);
p_unit = cross(h_unit, e_unit);

% Rotation matrix from PQW to inertial
R = [e_unit, p_unit, h_unit];  % 3x3 matrix

% True Anomaly at t0
% theta_rad_0 = acos(((h_mag^2/params.mu) - r_mag_0) / (r_mag_0 * e));
theta_rad_0 = atan2(dot(r0, p_unit), dot(r0, e_unit));

% Eccentric anomaly at t0
E_0 = 2 * atan(tan(theta_rad_0/2)/ sqrt((1+e)/(1-e)));

% Mean anomaly at t0
M_0 = E_0 - e*sin(E_0);

% Delta Mean anomaly: (elliptical orbit)
dM = n * (t_final - t0);

% New Mean Anomaly (elliptical orbit)
M_coast = M_0 + dM;

% Time vector for coast phase
t_vec = t0:dt_step:t_final;
N_points = length(t_vec);

%% Computation Eccentric anomaly

% Preallocate arrays for position and velocity vectors
r_log = zeros(3, N_points);
v_log = zeros(3, N_points);

R_tot_coast = zeros(3, N_points);
V_tot_coast = zeros(3, N_points);

% Convergence threshold
tol = 1e-6;  % Convergence criterion
```

```
max_iter = 1000;  % Safety stop to prevent infinite loop

% Iteration counter
iter = 0;

if e > 1
    disp("coast is hyperbolic, so this function need to be added")
end

% Loop over time and compute position/velocity at each time
for i = 1:N_points
    t_i = t_vec(i);

    % Mean anomaly at time t_i
    M_i = M_0 + n * (t_i - t0);

    % Solve Kepler's equation for E_i using Newton-Raphson
    E_i = pi;  % Initial guess
    iter = 0;
    while true
        E_new = M_i + e * sin(E_i);
        err = abs(E_new - E_i);
        E_i = E_new;
        iter = iter + 1;
        if err < tol
            % disp('Coast solver converged')
            break;
        elseif iter > max_iter
            % disp('Coast solver did not converge')
            break;
        end
    end

    % True anomaly _i
    theta_i = 2 * atan(sqrt((1+e)/(1-e)) * tan(E_i/2));

    % Radius r_i
    r_mag_i = (h_mag^2 / params.mu) / (1 + e*cos(theta_i));

    % Position in PQW
    r_pqw_i = r_mag_i * [cos(theta_i); sin(theta_i); 0];

    % Velocity in PQW
    v_r_i = (params.mu / h_mag) * e * sin(theta_i);
    v_theta_i = (params.mu / h_mag) * (1 + e * cos(theta_i));
    v_pqw_i = [v_r_i * cos(theta_i) - v_theta_i * sin(theta_i);
               v_r_i * sin(theta_i) + v_theta_i * cos(theta_i);
               0];

    % Rotate to inertial frame
    r_log(:, i) = R * r_pqw_i;
    v_log(:, i) = R * v_pqw_i;

    R_tot_coast(:,i) = r_log(:, i) / params.L_scale;
    V_tot_coast(:,i) = v_log(:, i) / params.V_scale;
end

%% Propagate costates
pV_tot_coast = zeros(3, N_points);
pR_tot_coast = zeros(3, N_points);

tau_vec = 0:dtau_step:tau_final;

for i = 1:N_points
    t = tau_vec(i);

    [pV_tau, pR_tau] = propagate_costates(params, pV0, pR0, t);

    pV_tot_coast(:, i) = pV_tau;
    pR_tot_coast(:, i) = pR_tau;

```

```
141 end
142
143
144 function [pV_tau, pR_tau] = propagate_costates(params, pV0, pR0, t)
145         I = eye(3);
146         Omega_tau = [cos(t)*I, sin(t)*I; -sin(t)*I, cos(t)*I];
147         Lambda_0_tau = [pV0; -pR0];
148         Lambda = Omega_tau * Lambda_0_tau;
149         pV_tau = Lambda(1:3);
150         pR_tau = -1 * Lambda(4:6);
151
152 end
```

D

Analytical Expressions of Environmental Models

This appendix presents the fitted functions and polynomial expressions used in the trajectory model.

D.1. Polynomial Fit for Temperature ($T(h)$)

$$
\begin{aligned}
T(h) = {} & \frac{645992072660383}{25353012004564588029934064107 52}h^{11} - \frac{785634642447301}{49517601571415210995964968 96}h^{10} \\
& + \frac{1651385259214261}{38685626227668133590597632}h^{9} - \frac{7852444447601263}{12089258196146291747061 76}h^{8} \\
& + \frac{5801440336565631}{9444732965739290427392}h^{7} - \frac{86149751180923}{2305843009213693952}h^{6} \\
& + \frac{843415748573519}{576460752303423488}h^{5} - \frac{5175102167916653}{144115188075855872}h^{4} \\
& + \frac{4650464721232555}{9007199254740992}h^{3} - \frac{4044800269305997}{1125899906842624}h^{2} \\
& + \frac{3067147986819553}{1125899906842624}h + \frac{5001053079678687}{17592186044416}
\end{aligned}
\tag{D.1}
$$

D.2. Fitted Expression for Normal Force Coefficient ($C_{N_\alpha}(M)$)

$$
C_{N_\alpha}(M) = 0.1022190 \, \exp\left(-\frac{1.5169762M^2 - 2.3813996}{2}\right) + \frac{0.0778752}{(0.729975\,M)^{11.5134} + 1} + 0.1480923 \tag{D.2}
$$

D.3. Fitted Expression for Axial Force Coefficient ($C_{A_\alpha}(M)$)

$$
C_{A_\alpha}(M) = 0.2839477 \, \exp\left(-\frac{1.8889370M^2 - 2.6367670}{2}\right) - \frac{0.0625142}{(1.047788\,M)^{23.6306} + 1} + 0.6242156 \tag{D.3}
$$

D.4. Polynomial Fit for Atmospheric Pressure ($P(h_p)$)

$$\begin{aligned}\ln P(h_p) = &\frac{7803750488859883}{154742504910672534362390528}h_p^6 - \frac{4649946300278655}{604462909807314587353088}h_p^5 \\ &- \frac{3772826104695695}{3777893186295716170956 8}h_p^4 + \frac{562808444335809}{922337203685477580 8}h_p^3 \\ &- \frac{6419106927691759}{2305843009213693952}h_p^2 - \frac{4053351155335035}{36028797018963968}h_p \\ &+ \frac{1621591562666981}{140737488355328}\end{aligned} \tag{D.4}$$

$$P_{\text{ambient}} = \exp(\ln P(h_p)) \tag{D.5}$$

D.5. Derivatives of Coefficients with Respect to Mach number

Derivative Normal Force Coefficient C_N

$$\begin{aligned}\frac{dC_N}{dM} = \frac{dC_{N_\alpha}}{dM} = &- \frac{1.2580 \times 10^{28}\, M\, \exp\left(\frac{1.3406\times10^{15}}{1.1259\times10^{12}} - \frac{6.8319\times10^{15}M^2}{9.0072\times10^{12}}\right)}{8.1130 \times 10^{19}} \\ &- \frac{2.5566 \times 10^{12} \left(\frac{29199\,M}{40000}\right)^{\frac{52567}{5000}}}{3.90625 \times 10^{12} \left[\left(\frac{29199\,M}{40000}\right)^{\frac{57567}{5000}} + 1\right]^2}\end{aligned} \tag{D.6}$$

Derivative Normal Axial Coefficient C_A

$$\begin{aligned}\frac{dC_A}{dM} &= \frac{dC_{A_\alpha}}{dM} + \frac{dC_{N_\alpha}}{dM} \\ &= \frac{3.5346 \times 10^{43} \left(\frac{4718817646363277\,M}{4503599627370496}\right)^{\frac{1592486901987043}{70368744177664}}}{2.2836 \times 10^{43} \left[\left(\frac{4718817646363277\,M}{4503599627370496}\right)^{\frac{1662855646164707}{70368744177664}} + 1\right]^2} \\ &\quad - \frac{2.1757 \times 10^{31}\, M\, \exp\left(\frac{2.9687\times10^{15}}{2.2518\times10^{12}} - \frac{8.5070\times10^{15}M^2}{9.0072\times10^{12}}\right)}{4.0565 \times 10^{19}} \\ &\quad - \frac{1.2580 \times 10^{28}\, M\, \exp\left(\frac{1.3406\times10^{15}}{1.1259\times10^{12}} - \frac{6.8319\times10^{15}M^2}{9.0072\times10^{12}}\right)}{8.1130 \times 10^{19}} \\ &\quad - \frac{2.5566 \times 10^{12} \left(\frac{29199\,M}{40000}\right)^{\frac{52567}{5000}}}{3.90625 \times 10^{12} \left[\left(\frac{29199\,M}{40000}\right)^{\frac{57567}{5000}} + 1\right]^2}\end{aligned} \tag{D.7}$$

D.6. Derivatives of Coefficients with Respect to Angle of Attack

$$\frac{dC_A}{d\alpha} = C_{N_\alpha}(M)\left[\sin(\alpha) + \alpha\cos(\alpha)\right] \tag{D.8}$$

$$\frac{d^2C_A}{d\alpha^2} = C_{N_\alpha}(M)\left[2\cos(\alpha) - \alpha\sin(\alpha)\right] \tag{D.9}$$

$$\frac{dC_N}{d\alpha} = C_{N_\alpha}(M) \tag{D.10}$$

E

Reference Trajectory Data

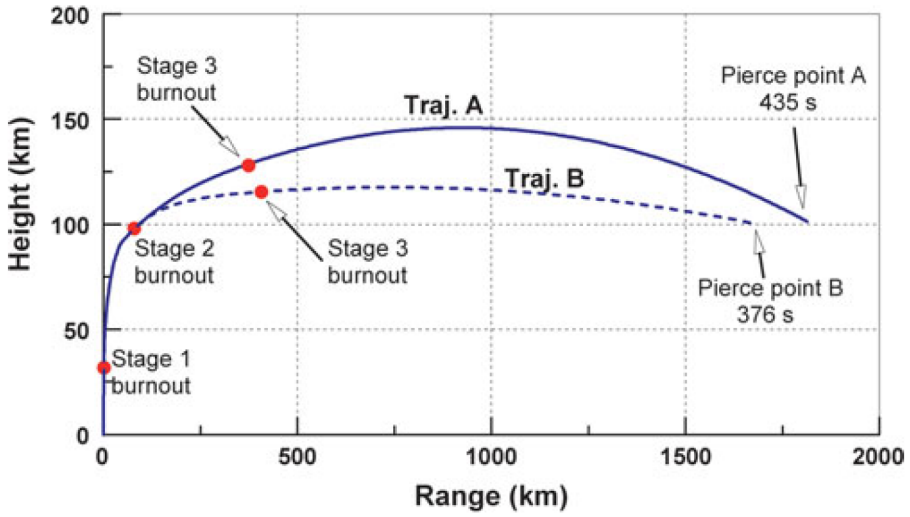


Figure E.1: Reference trajectories of the boost phase of the HTV-2 [70].
(Note: vertical scale is stretched w.r.t the horizontal scale)