Reachability Analysis to Design Zero-Wait Entry Guidance

Orbital Aborts to Enhance Mission Safety

Alejandro González Puerta

Master of Science Thesis

TU Delft Delft University of Technology

Challenge the future

Reachability Analysis to Design Zero-Wait Entry Guidance

Orbital Aborts to Enhance Mission Safety - March 5, 2017

by

Alejandro González Puerta

Student ID: 4176340

in partial fulfillment of the requirements for the degree of

Master of Science

in Aerospace Engineering

at Delft University of Technology,

to be defended publicly on March 17th, 2017 at 09:30 AM.

Student number:	4176340	
Supervisor:	Dr. ir. E. Mooij	
Thesis committee:	Prof. dr. ir. P. N. A. M. Visser,	TU Delft
	Dr. ir. M. Voskuijl,	TU Delft
	Dr. ir. E. Mooij,	TU Delft
	MSc. ir. C. Yabar Valles,	European Space Agency
	Dr. ir. M. Sagliano,	DLR

Cover image based on the work from:
`http://spaceflight.nasa.gov/gallery/images/shuttle/sts-127/hires/s127e006561.jpg`.

An electronic version of this thesis will be made available at `http://repository.tudelft.nl/` on February 28th, 2018.

TU Delft Delft University of Technology

PREFACE

The thesis presented here is the final step in my pursue for the Master of Science degree at the Faculty of Aerospace Engineering, Delft University of Technology. It has been a long and difficult journey that I could not have completed without the help and support of numerous people.

First and foremost I would like to thank my supervisor Erwin Mooij for his weekly support, words of advice and the occasional joke. During the past year, Erwin has shown a tremendous patience and was always there to point me in the right direction. Next, I would like to thank Celia Yabar Valles and Guillermo Ortega from the European Space Agency for giving me the opportunity to tackle such a thrilling and motivating challenge. Furthermore, I want to thank Dominic Dirkx and René Hoogendoorn for joining me in numerous C++ bug-squashing adventures. A special thanks goes to Marco Sagliano for his kind help deciphering the inner workings of his AMPI algorithm. Special thanks goes to Francesco Cremaschi from ASTOS Solutions for his unattended help and for providing an academic ASTOS license to work with. Finally, I would like to show my gratitude to the International Space Safety Foundation (ISSF) for showing faith in this research project by awarding me with the Graduate Fellowship Grant.

I am very grateful to all the people that I have met along the road who have made Delft feel more like home. In particular, I want to thank Beatriz for being there for both the ups and downs of this journey.

Esta tésis está dedicada a mi familia, ya que sin su apoyo y su afecto este viaje no habría sido posible. ¡Muchas gracias!

A. Gonzalez Puerta
Delft, March 2017

ABSTRACT

The goal of this thesis is to develop and test a dedicated guidance system for abort entry missions, focusing on flights that originate from a low Earth orbit. The assignment tackled in this thesis was proposed by the European Space Agency (ESA) and it intends to be a valuable contribution to the PRIDE entry vehicle development program.

The role of traditional guidance systems is to ensure that the vehicle is brought from a nominal entry interface point (EIP) to the neighborhood of a designated landing site at an acceptable energy state. Such systems are designed in such a way that the landing site can be safely reached even if off-nominal conditions are present at the EIP or during the descent flight. A common aspect of traditional guidance systems is that they heavily rely on prior knowledge of the nominal entry trajectory and often require significant planning time prior the de-orbit burn. Due to the nature of abort scenarios, not only is prior knowledge of the entry conditions usually unavailable, but also there is often limited time to plan and execute any entry procedures. This is particularly relevant for manned missions, where a long decision time may compromise the crew's safety. In addition, the range of entry conditions that may be encountered in an abort situation is significantly large, meaning that planning a trajectory to an alternative landing site may be required if the vehicle needs to be brought to safety in due time. For all these reasons, the development of an autonomous adaptive guidance system for abort entry missions is more than justified.

The first task of a guidance system is to plan the steering commands to be followed by the entry vehicle. Such planning must result in a trajectory that satisfies multiple system constraints and possibly minimizes a certain objective. Such trajectory optimization process is generally done on the ground, since traditional software can take hours to converge to a solution even when using a high-performance computer. Due to the limited computational power of current flight hardware, performing on-board trajectory optimization is not feasible, especially in an abort situation. This issue is tackled using the concept of on-board trajectory generation via Adaptive Multivariate Pseudospectral Interpolation (AMPI) by Sagliano et al. (2016), which attempts to achieve the optimality and accuracy of solutions obtained on the ground with a feasible computing speed. The enabler concept behind this technology is that interpolation algorithms are orders of magnitude faster than the existing optimization methods. The AMPI-based trajectory interpolator feeds off a large database of optimal trajectories computed off-board, where the selection of the appropriate database sub-space is made according to the projected entry conditions at the moment of the abort. Due to the large range of entry conditions encountered in abort scenarios, the trajectory database size may become unfeasible. Consequently, the AMPI method is reduced to bivariate interpolation and thus uses a database that simply covers off-nominal latitude and longitude values. Dispersions in the remaining entry state variables are addressed by a combination of a Linear Quadratic Regulator (LQR) tracking law and a simple lateral guidance implementation.

The developed guidance system is capable of planning a reference trajectory in the order of a few milliseconds, proving its capability to run on-board. Furthermore, the system can handle multiple abort scenarios, since the trajectory planner is capable of selecting the most appropriate reference trajectories from the database based on information provided by the navigation system. Thorough testing of the system showed that while the constraint compliance is marginally affected by the density of the trajectory database, the position dispersions at the terminal point are strongly correlated with the chosen density. Results show that increasing the spacing between the stored EIPs in the database increases the mean and spread of the terminal position dispersions. Despite this, the AMPI algorithm employed in the trajectory planner allows for a loss-less compression of the database. Additional Monte Carlo campaigns showed that the compression factor does not have a significant impact on the dispersions at the terminal point. Consequently, manageable database sizes can be achieved even for fine spacings, ranging from only 18 MB for a parameter spacing of 5 deg to 900 MB for a spacing of 0.1 deg.

In summary, the results shown in this thesis serve as solid proof that a lightweight autonomous adaptive abort guidance system can be successfully implemented by means of combining a trajectory planner based on trajectory interpolation and an LQR tracker. Despite this, the robustness of the system could be improved by revising the choice for landing sites or by enabling the system to re-plan the trajectory when in close proximity to the terminal point.

NOMENCLATURE

LATIN SYMBOLS

a	acceleration	m/s^2
$\boldsymbol{A}$	system matrix	-
$\boldsymbol{B}$	input matrix	-
C	aerodynamic force coefficient	-
C_D	drag coefficient	-
C	interpolation condition matrix	-
C_L	lift coefficient	-
$\boldsymbol{C}$	output matrix	-
$\boldsymbol{C}$	transformation matrix	-
C_m	pitching moment coefficient	-
$\boldsymbol{C}$	trajectory database information matrix	-
D	aerodynamic drag force	N
e	eccentricity of the reference ellipsoid	-
e	normalized energy	-
$\boldsymbol{e}$	extremal trajectory point	-
E	energy	J
f	ellipsoidal flattening parameter	-
$\boldsymbol{f_{int}}$	low-density interpolated function vector	-
$\tilde{\boldsymbol{f}}_{\boldsymbol{int}}$	high-density interpolated function vector	-
F	force	N
g	gravitational acceleration modulus	m/s^2
$\boldsymbol{g}$	dynamics constraint vector	-
h^*	geodetic altitude	m
h	geocentric altitude	m
h	step size	s
J	performance index	-
J_2	J_2 term of the gravity field	-
$\boldsymbol{k}$	Runge-Kutta slope vector function	-
$\boldsymbol{K}$	feedback matrix	-
L_{bf}	vehicle body flap length	m
L/D	lift-to-drag ratio	-
L	aerodynamic lift force	N
L_N	Legendre polynomial of N^{th} degree	-
L_{ref}	vehicle reference length	m

L_z	thermal lapse rate	K/m
M	Mach number	-
m	vehicle mass	kg
M	molecular mass of air	kg/mol
n_g	aerodynamic load	g
N_A	Avogadro's number = $6.0221409 \cdot 10^{23}$	particles/mol
N	sum of the number density of a gas at a given altitude	particles/m^3
$\boldsymbol{p}$	numerical optimization parameter vector	-
$\boldsymbol{p}$	trajectory database parameter vector	rad
P_{FRP}	low-density to high-density conversion matrix	-
p	pressure	Pa
$\boldsymbol{P}$	Riccati equation coefficient matrix	-
q_c	convective heat flux	W/m^2
$\boldsymbol{q}$	quadrature function vector	-
$\bar{q}$	dynamic pressure	N/m^2
Q	heat load	J/m^2
$\boldsymbol{Q}$	Riccati equation weight matrix	-
$\boldsymbol{r}$	position vector	m
r	normalized radial position	-
R	radius	m
R_N	Flipped Radau-Legendre polynomial of N^{th} degree	-
$\boldsymbol{R}$	Riccati equation coefficient matrix	-
R^*	universal gas constant = 8.31432	J/mol/K
s	range-to-go	m
$\boldsymbol{s}$	trajectory database parameter subspace vector	rad
S_{bf}	vehicle body flap area	m^2
S	aerodynamic side force	N
S_{ref}	vehicle reference area	m^2
t	time	s
T	temperature	K
T_M	molecular temperature of the US76 Model	K
$\boldsymbol{u}$	control vector	rad
u, v, w	Cartesian velocity coordinates along the X, Y, Z axes	m/s
$\boldsymbol{V}$	Vandermonde matrix	-
V_g	groundspeed	m/s
V_i	inertial speed	m/s
w	weighting factor	-
W_{ref}	vehicle reference width	m
$\boldsymbol{x_0}$	interpolation off-nominal point	rad

x	root of a function	-
$\boldsymbol{x}$	state vector	-
X, Y, Z	X, Y, Z axes	-
x, y, z	Cartesian position coordinates along the X, Y, Z axes	m
z	geopotential altitude	m

Greek Symbols

α	angle of attack	rad
β	angle of sideslip	rad
γ	flight-path angle	rad
γ	ratio of specific heats for air = 1.4	-
δ	derivative approximation	-
δ_{bf}	body flap deflection	rad
δ	geocentric latitude	rad
δ^*	geodetic latitude	rad
δz	auxiliary ellipsoid model parameter	m
ϵ	tolerance	-
$\boldsymbol{\eta}$	numerical state approximation vector	-
λ	US76 parameter = 0.01875	km^{-1}
μ	gravitational parameter	m^3/s^2
σ	bank angle	rad
ρ	atmospheric density	kg/m^3
ρ_w	covariance between meridional and zonal wind velocity components	-
τ	geocentric longitude	rad
τ^*	geodetic longitude	rad
$\boldsymbol{\tau}$	low-density pseudotime vector	-
$\tilde{\boldsymbol{\tau}}$	high-density pseudotime vector	-
$\boldsymbol{\Phi}$	increment vector function	-
$\boldsymbol{\varphi}$	boundary constraint vector	-
$\boldsymbol{\varphi}$	bivariate interpolation auxiliary vector	-
χ	heading angle	rad
$\boldsymbol{\psi}$	bivariate interpolation auxiliary vector	-
$\boldsymbol{\Omega}$	rotational velocity vector	rad/s
ω	rotational velocity modulus	rad/s

Subscripts

0	sea level
0	clean aerodynamic coefficient
0	initial value of quantity

E	Earth
N	vehicle nose
bf	vehicle body flap
cmd	commanded variable
f	final value of quantity
flt	in-flight value
i	i^{th} element in a discrete series
l	lower bound of a quantity
m	moment reference center
m	midpoint of a quantity
max	maximum quantity
msd	measured quantity
n	last element in a discrete series
nom	nominal quantity
prd	predicted quantity
ref	reference quantity
req	required quantity
u	upper bound of a quantity

ACRONYMS AND ABBREVIATIONS

AMPI	Adaptive Multivariate Pseudospectral Interpolation
BSM	Bisection Method
CFD	Computational Fluid Dynamics
CMC	Ceramic Matrix Composite
CRV	Crew Return Vehicle
DRS	Descent and Recovery System
EIP	Entry Interface Point
ESA	European Space Agency
ESTEC	European Space Research and Technology Centre
FRP	Flipped Radau-Legendre Polynomial
FRV	Future Reentry Vehicle
HD	High Density
ISS	International Space Station
ISSF	International Space Safety Foundation
IVP	Initial-Value Problem
IXV	Intermediate eXperimental Vehicle
LD	Low Density
NASA	National Aeronautics and Space Administration
NOAA	National Oceanic and Atmospheric Administration

OBC	On-Board Computer
PRIDE	Programme for Reusable In-orbit Demonstrator in Europe
SM	Secant Method
SPHYNX	Subscale Precursor Hypersonic X Vehicle
TAEM	Terminal Area Energy Management
US76	US Standard Atmosphere 1976
USAF	US Air Force
UTC	Coordinated Universal Time
WGS84	World Geodetic System 1984

Reference Frames

$\mathscr{F}_A$	aerodynamic frame
$\mathscr{F}_B$	body frame
$\mathscr{F}_I$	inertial frame
$\mathscr{F}_R$	rotating frame
$\mathscr{F}_T$	trajectory frame
$\mathscr{F}_V$	vertical frame

Notation

$\dot{\boldsymbol{x}}$	time derivative of vector $\boldsymbol{x}$
$\boldsymbol{x^j}$	vector in coordinates of reference frame j
$\boldsymbol{C^{-1}}$	inverse of matrix $\boldsymbol{C}$
$\boldsymbol{C^T}$	transpose of matrix $\boldsymbol{C}$
$\boldsymbol{C_{2,1}}$	transformation matrix from $\mathscr{F}_1$ to $\mathscr{F}_2$
$\boldsymbol{C}[i,j]$	element in the i$^{\text{th}}$ row and j$^{\text{th}}$ column of matrix $\boldsymbol{C}$
f'	derivative of function f
δe	infinitesimal deviation of quantity e from its nominal value
Δe	finite deviation of quantity e from its nominal value

Table of Contents

1 INTRODUCTION

Entry is the process that space-traveling objects experience when they penetrate the atmosphere of a celestial body. During reentry, most of the kinetic and potential energy associated with these objects is dissipated by atmospheric drag. As a result, such objects endure extreme temperatures and forces that may result in their obliteration. The reentry process is not tied to natural bodies alone, but is also common among man-made crafts. These crafts are commonly known as reentry vehicles and they have proved essential throughout the history of spaceflight. Missions involving planetary landers, crew recovery, satellite servicing or sample return would have been impossible without such vehicles. To illustrate the punishment that reentry vehicles experience during descent, Figures 1.1 and 1.2 show pictures of the European Space Agency's (ESA) Intermediate eXperimental Vehicle (IXV) before and after the flight, respectively.

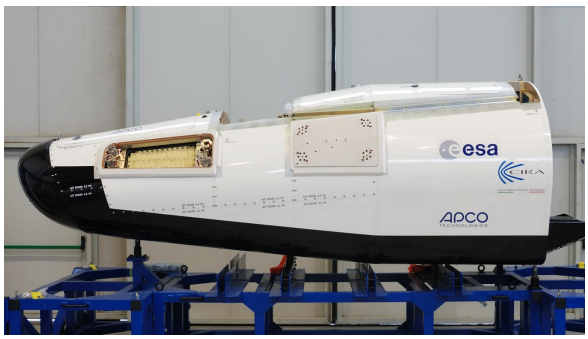

Figure 1.1: Photo of the IXV prior reentry [a].

Figure 1.2: Photo of the IXV after reentry. Evident paint job and TPS damage is shown [b].

1.1 MISSION PLANNING & ENTRY GUIDANCE

To ensure the survivability of a reentry vehicle and its payload, the changes in the vehicle's altitude and velocity are carefully controlled during the descent. Such control is achieved by adjusting the vehicle's orientation and consequently the magnitude of the aerodynamic forces acting on it. The actual magnitude of these adjustments is planned in advance as part of the trajectory-analysis process. Such analysis not only produces a set of attitude-command profiles, but also generates a region in the altitude-velocity plane where the vehicle is guaranteed to survive, the so-called the entry corridor. An example of such corridor is shown in Figure 1.3.

The bounds of this entry corridor are typically established by the maximum allowable aerodynamic load and dynamic pressure that the vehicle's structure and payload can handle, the maximum heating rate and heat load that the vehicle's Thermal Protection System (TPS) can cope with and the minimum deceleration required for the vehicle to be captured by the atmosphere. The trajectory-planning process must also adjust the entry interface point (EIP) and the attitude commands, such that the vehicle ends up in the neighborhood of the landing site. Such neighborhood is determined by the start of the Terminal Area Energy Management (TAEM) phase, whose role is to line up the vehicle with runway. In the scope of this report, the start of the TAEM phase determines the end of the guided entry and descent. An example trajectory showing the EIP and the start of the TAEM phase is shown in Figure 1.4.

[a] `http://i2.wp.com/www.therefinedgeek.com.au/wp-content/uploads/2014/03/ESA-IXV-Full-Scale-Prototype.jpg`, [Accessed On: 18-04-2016]

[b] `https://upload.wikimedia.org/wikipedia/commons/thumb/e/ef/Demonstrateur_corps_portant_IXV_apres_son_vol_de_fevrier_2015_DSC_0037.jpg/862px-Demonstrateur_corps_portant_IXV_apres_son_vol_de_fevrier_2015_DSC_0037.jpg`, [Accessed On: 18-04-2016]

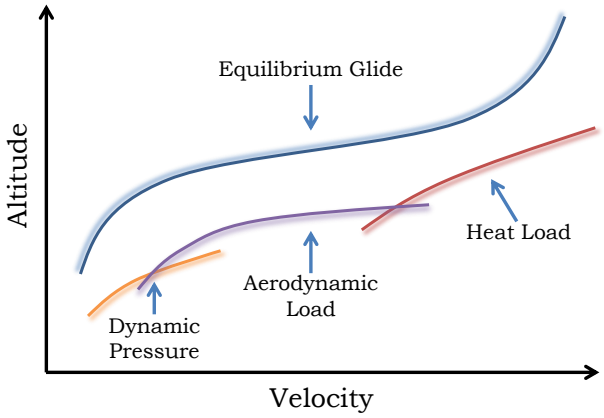

Figure 1.3: Sketch of a typical entry corridor. Here, the vehicle is only allowed to fly within the four boundaries shown. Adapted from Xue and Lu (2010)

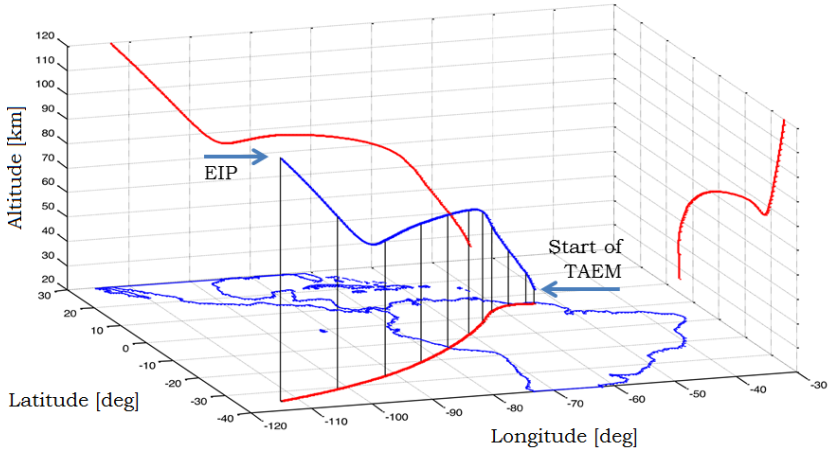

Figure 1.4: Example reentry trajectory over South America. The EIP is located at approximately 120 km altitude, -115 deg longitude and -35 deg latitude. The TAEM phase starts at approximately 25 km altitude, -62 deg longitude and 5 deg latitude. Figure adapted from (Mooij, 2014)

In addition to the trajectory-planning process, reentry vehicles are required to have a guidance system which ensures that the nominal trajectory is followed closely. The guidance system updates the planned attitude command profiles in basis of actual velocity and position data and thus removes any deviations from the nominal trajectory. Failure to include a guidance system poses a risk on the mission, since the vehicle may leave the entry corridor ultimately leading to the violation of multiple trajectory constraints.

Although reentry vehicles mostly end up flying a nominal trajectory, one should account for those trajectories that are required to be flown in case an unplanned event occurs. Such trajectories are referred to *abort trajectories* and they must be part of the trajectory-analysis process. Failing to do so, may result in the need of flying trajectories that could lead to mission failure. Abort trajectories differ significantly from each other, where such differences are driven by the abort scenario that triggered them, the mission phase when the abort occurs and the capabilities of the vehicle flying the abort trajectory (Muratore, 2009). According to Muratore, such abort scenarios include launcher structural failure, degraded launcher propulsion performance, emergency crew recovery from the International Space Station (ISS) or aborts due to primary landing site unavailability, amongst others. The discussed scenarios can be grouped into the following categories:

- **Launch Aborts:** Those abort scenarios that occur during the powered ascent, prior reaching a stable orbit. The mission profile of launch abort scenarios is highly dependent on the state of the launch vehicle at the moment of the abort calling, since it dictates whether further powered ascent towards orbit is possible or a gliding flight towards an emergency landing site is necessary.
- **Orbital Aborts:** Those abort scenarios that originate once the reentry vehicle is on a stable operational orbit. Orbital aborts begin with a de-orbit burn that puts the reentry vehicle in a descent path towards the EIP. Once atmospheric flight begins, the vehicle is guided towards the neighborhood of an emergency

landing site.

- **Descent Aborts:** Those abort scenarios triggered during the atmospheric flight between the EIP and the terminal point at the start of the TAEM phase. The mission profile of a descent abort could involve re-routing to an alternative landing site due to primary site unavailability.

A commonality among the three discussed groups is that all the scenarios require some form of guidance system that steers the vehicle to safety after the abort calling. In addition, the capability of reaching a safe terminal point is highly dependent on the vehicle position and velocity at the moment of the abort calling. Furthermore, the vehicle architecture has a large influence on the expected abort mission profile and consequently the design of the abort guidance system. Despite these similarities, the abort scenarios are still very different from each other due to the large differences in the initial vehicle state at the abort calling. It becomes clear that each of the categories discussed above is a research topic on its own. Consequently, the work presented in this report will only consider orbital aborts.

1.2 STATE-OF-THE-ART

In the past, guidance systems have been designed in such a way that they accept limited vehicle state errors at a few nominal EIPs, such as those described in the work of Harpold and Graves (1979), Saraf et al. (2004) and Xue and Lu (2010). In such systems, on-board switching from one nominal EIP to another would be limited by any preliminary mission analysis and would likely require significant in-orbit wait time to schedule a de-orbit burn that brings the vehicle to one of the limited EIPs available. For these reasons, the available guidance systems are not fully suitable for abort scenarios, where a long planning and execution time may compromise the mission and crew safety.

Consequently, an adaptive abort guidance system must accept a wide range of EIPs, only limited by the geometry of the operational orbit at which the abort is called. Furthermore, such system could be designed in such a way that global coverage is guaranteed, which would ensure that a landing site can be safely reached by immediately executing a de-orbit burn right after the abort calling. In summary, adaptive abort guidance system are characterized by the following capabilities:

1. Capability of accepting widely spread EIPs, ideally guaranteeing global coverage.
2. Capability of selecting a targeted emergency landing site taking into account the vehicle capabilities or site preference.
3. Performing all operations automatically and in a timely fashion.

To ensure such capabilities, the system's trajectory planner architecture must be revisited. As one might expect, the complexity of abort trajectory-planning lies behind the contradictory nature of the problem. Nowadays, two main methods exist that attempt to tackle such a problem: on-board trajectory optimization and on-board trajectory interpolation.

The first method consists of the generation of optimal reentry trajectories based on actual the vehicle state and capabilities at the time of the abort. Although this method involves relatively little planning and benefits from a great flexibility, it requires the use of a high-performance on-board computers that are beyond the current available technology. Despite this, recent attempts have been made to simplify the optimization problem, in such a way that current flight hardware could execute the optimization algorithm (Wang et al., 2006).

The second method interpolates a reentry trajectory from a optimal trajectory database generated with proven software. Such interpolation takes into account the vehicle state at some initial point and finds the trajectory that fits best in the generated database (Sagliano et al., 2016). According to Sagliano et al., such a method benefits from significantly less computational effort at the expense of a more limited flexibility. In terms of abort trajectory-planning, this limitation requires the generation of a sufficiently large trajectory database that accounts for all possible abort scenarios.

The author believes that an abort trajectory planner based on on-board trajectory interpolation is a strategy worth researching, since it benefits from the fact that is capable of running in current flight hardware and uses proven software building blocks, such as off-board trajectory optimization and interpolation algorithms. Once the planned trajectories are computed, these are fed to the trajectory tracker, which safely steers the vehicle during descent.

1.3 RESEARCH CONTEXT & METHODOLOGY

Given the discussion above, it becomes clear that orbital aborts could significantly enhance mission safety. However, at the time of writing of this report no dedicated guidance system exists that enables such capability, mainly due to the lack of flexibility and significant planning time that current systems require. Thus, the following research question is posed:

How can a reentry vehicle return from low-Earth orbit in an autonomous, reliable and timely manner while considering the limitations of current flight hardware?

The main research question is broken down into the following sub-questions, with the goal of structuring the research methodology:

1. How are the different orbital abort scenarios characterized in terms of EIP position and velocity?
2. How are the different orbital abort scenarios characterized in terms of the distribution of emergency landing sites?
3. What design challenges do such abort scenarios pose on the guidance system?
4. How effective is the proposed guidance system in tackling such challenges?
5. How does the choice for emergency landing sites influence on the system's safety?
6. How do the different guidance system design parameters influence on its performance?

The above questions are answered in this thesis by applying the following research strategy. First, a set of mission requirements is agreed with ESA which refines the scope of the mission and the vehicle to be used. It is established that the orbital abort scenarios are modeled from the moment the vehicle reaches the EIP, assuming that a separate system times and executes the de-orbit burn that places the vehicle at the EIP with the nominal entry state defined by the mission requirements. Although the propulsive forces at the de-orbit burn and the Keplerian flight prior the EIP are not modeled, the time-shift of the de-orbit burn is taken into account. This is achieved by assuming that changes in the timing lead to different placements of the EIP in latitude and longitude due to the ground-track shifting effect. Next, a reachability study is performed to "connect" the set of possible EIPs to the emergency landing sites established in the mission requirements. Such study is performed by solving multiple optimization problems using the AeroSpace Trajectory Optimization Software (ASTOS) commercial package, where the initial and final conditions of the problem are systematically changed to probe multiple EIPs and landing sites. Once the abort opportunities are established, a guidance system architecture is developed and implemented in C++. This guidance system uses the "flown" EIP to select the most appropriate landing site and trajectory to follow, based on the information provided by the reachability study. The guidance system issues angle of attack and bank angle commands and assumes an ideal control system, thus ignoring rotational dynamics and actuator effects. In addition to the guidance system, an entry simulator is developed which is used to test the performance of the system when subjected to errors in the EIP state. Finally, the impact of multiple design parameters are tuned to minimize the impact of the guidance system on the On-Board Computer (OBC).

1.4 REPORT OUTLINE

This report is layed out as follows: Chapter 2 summarizes the reference mission case and lists the requirements to be fulfilled by the guidance system. In Chapter 3, the fundamental physics principles and models that govern the reentry mechanics are discussed. In Chapter 4, the theory behind the algorithms that build the abort guidance system is presented. Chapter 5, discusses the underlying mathematics of the numerical methods implemented in the guidance software. Next, the reentry mechanics theory and the abort guidance theory is assembled into a number of software modules, which are combined into a complex architecture as presented in Chapter 6. Such chapter presents as well the numerous test used to verify that the developed software functions as expected. One of such software modules requires the assembly of a trajectory database, that maps a number of EIPs to a set of landing sites. The information stored in such database is evaluated in Chapter 7. Next the guidance system is tested with the goal of assessing its performance, where the results of such tests are discussed in Chapter 8. Finally, Chapter 9 summarizes all the findings of this thesis and presents a number of recommendations for future research.

2 Mission Heritage & Mission Case

The goal of this chapter is to discuss the overall mission case to be tackled in this thesis. To begin with, Section 2.1 gives a summary of previous missions that had an abort trajectory component. In Section 2.3 lists and evaluates the ESA-established requirements that build up the actual mission case. Finally, Section 2.4 describes in detail the reference vehicle to be used as part of the mission case of this thesis.

2.1 Vehicle Heritage

A good example for abort trajectories are those planned for the launch of the U.S. Space Shuttle and the HL-20, which were designed to guarantee the safety of the crew and the vehicle in case of launch failure. In line with the discussion of Section 1.3, the shape of these abort trajectories depended strongly on the vehicle capabilities and launch stage at the time of abort initiation (Muratore, 2009; Naftel and Talay, 1993).

Both the Shuttle and the HL-20 featured a hypersonic lift-to-drag ratio (L/D) in the order of 1.0 (Joosten, 1985; Stone and Piland, 1993), which brings numerous mission design benefits when compared to traditional ballistic vehicles with lower L/D values. To begin with, the experienced deceleration throughout the reentry is lower, which is highly desired whenever deconditioned, sick or injured crew needs to be transported (Stone and Piland, 1993). Furthermore, higher L/D values lead to a high cross-range capability which enables greater accessibility to a wider selection of landing sites. In turn, a higher cross-range reduces the in-orbit wait time to access a particular landing site, where such capability is highly desired when dealing with incapacitated crew members.

Although both vehicles had a similar L/D, the vehicle designs are substantially different: the Shuttle featured a wing planform and the HL-20 was a lifting body. Due to the long turn-around times and refurbishment costs of the Shuttle, NASA's interest towards the simpler but equally capable lifting body designs grew. Such designs feature reduced operational refurbishment requirements while keeping some of the capabilities of winged vehicles, such as low entry accelerations, runway landings and once-per-day return from orbit (Stone and Piland, 1993). In fact, the lifting body design of the HL-20 served as a baseline for NASA's X-38 program, which became the continuation of the HL-20 program throughout the 1990s (Muratore, 2009). The goal of the X-38 program was to provide the know-how and technical expertise necessary to develop a Crew Return Vehicle (CRV) which would serve as a lifeboat for the ISS capable of returning the entire crew within a single launch and within a short lead time.

Within the X-38 program, four test vehicle were envisaged which included three atmospheric demonstrators and one orbital/reentry demonstrator (Klaedtke et al., 1999). The V-131 was the first atmospheric vehicle and its main purpose was to prove the lifting-body to parafoil transition capability, which completed successfully after two drop tests. Note that the V-131 was designed to be 80% in scale relative to the future CRV. The V-132 demonstrator was identical in scale and shape to the V-131, but included movable control surfaces that aimed to validate autonomous flight prior parafoil deployment. The V-133 was the first full-scale vehicle and thus aerodynamically fully representative (Klaedtke et al., 1999). The aim of the V-133 was to verify the aerodynamic shape the CRV and the developed control laws. Finally, the V-201 was space-rated and featured Deorbit Propulsion Stage which would jettison prior reentry. The aim of the V-201 was to prove the whole mission profile, including flight-operations, reentry, atmospheric maneuverability and landing by parafoil (Klaedtke et al., 1999). Note that due to the X-38 program cancellation, the V-133 and the V-201 vehicles were never flown.

The X-38 program was an example of international collaboration between NASA and ESA, where multiple vehicle components were designed and manufactured in Europe, such as the body-flaps, rudders, the Ceramic Matrix Composite (CMC) leading edges and nose cap, the landing skids and elements of the Guidance, Navigation and Control (GNC) system, amongst others (Klaedtke et al., 1999). After the X-38 program cancellation, ESA initiated the development of the Subscale Precursor Hypersonic X Vehicle (SPHYNX), which was conceived as a 1:3 scaled version of the X-38, hence the similarities shown by Figures 2.1 and 2.2. By scaling down the vehi-

cle, the SPHYNX program could benefit from the abundant heritage from the X-38 developed in Europe (Gerard and Tumino, 2005). The goal of SPHYNX was to demonstrate European reentry capabilities as well as validating the multiple technologies developed for the X-38. The development of SPHYNX concluded in 2003 once a feasibility consolidation study was completed at system level Gerard and Tumino (2005).

Nowadays, the interest towards small lifting body reentry vehicles with dedicated abort capabilities is still present. In line with such interest, ESA developed the IXV, a lifting-body technology demonstrator which aimed to consolidate existing knowledge and expertise thus serving as a stepping-stone for the development of future European reentry vehicles (Kerr et al., 2012). Similarly to SPHYNX, the IXV features two independent body-flaps for aerodynamic roll and pitch control. Due to the absence of rudders on the IXV, yaw control is provided by the on-board Reaction Control System (RCS) thrusters. Note that such thrusters can be combined with the body-flaps to provide additional control torque. The IXV was launched in February 2015 into a sub-orbital trajectory by a VEGA-C launcher. The IXV was recovered intact after splashdown in the Pacific ocean, allowing post flight-inspection and data analysis (Zaccagnino et al., 2011).

Following the success of the IXV, ESA began the development of the future Programme for Reusable In-orbit Demonstrator in Europe (PRIDE) vehicle with the goal of developing an affordable reusable European space transportation system (Yabar Valles, 2016). Similarly to the IXV and SPHYNX, PRIDE is conceived as a lifting-body reentry vehicle which uses a combination of body-flaps and RCS thrusters for attitude control. The mission profile of the PRIDE vehicle is depicted in Figure 2.3, which begins with the launch of PRIDE on a VEGA-C launcher. Once in orbit, PRIDE will deploy its solar array allowing orbital operations to begin. When orbital operation conclude, the solar array is discarded and the vehicle begins its reentry after a de-orbit burn. When the vehicle reaches the right altitude and speed, a parafoil is deployed for the final landing phase. At the time of writing of this report, PRIDE is conceived to land on skids on a conventional runway although helicopter mid-air retrieval is also an option (Yabar Valles, 2016).

In partnership with ESA, the goal of this thesis assignment is to plan a set of abort trajectories for a Future Reentry Vehicle (FRV) and to design an abort guidance system the handles multiple abort scenarios, where the FRV is expected to have nominal mission profile similar to that of PRIDE.

Figure 2.1: A photography of the X38 (V-132) in flight [a]

.

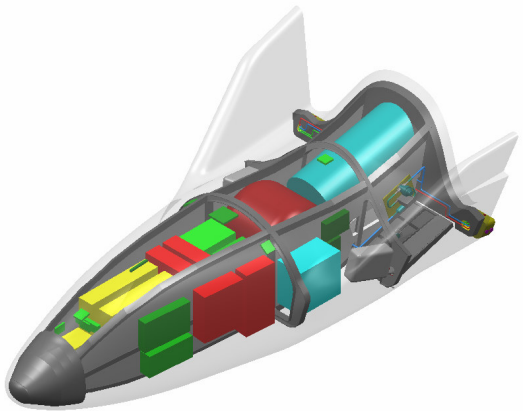

Figure 2.2: A render of the SPHYNX vehicle. Source: (Gerard and Tumino, 2005)

.

2.2 ORBITAL ABORT MISSION PROFILE

The goal of this section is summarize the mission profile envisaged for an orbital abort scenario and to discuss how the work to be performed in this thesis fits in such a framework. The reader is referred to Figure 2.4 for a schematic of such mission profile.

An orbital abort mission begins at the moment the actual abort is called for, which could occur at any point throughout the operational orbit. Once the abort mission begins, the deorbitation guidance computes the length and timing of the de-orbit burn that lowers the perigee altitude. To do so, it uses previously known information regarding the landing site reachability, in such a way that the de-orbit burn "injects" the vehicle

[a] `http://www.spaceflightinsider.com/wp-content/uploads/2015/03/ISS-International-Space-Station-X-37-Crew-Return-Vehicle-NASA-image-posted-on-SpaceFlight-Insider.jpg`, [Accessed On: 18-04-2016]

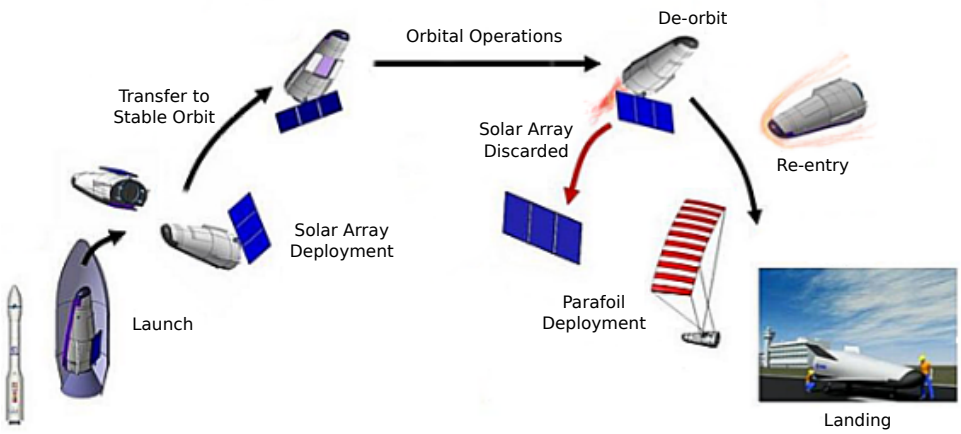

Figure 2.3: Simplified mission profile of the PRIDE and FRV vehicles. Adapted from: (Yabar Valles, 2016)

into a path towards an EIP that allows to safely reach one of the emergency landing sites. Once the deorbitation guidance determines a solution to reach an acceptable EIP, the vehicle performs a ballistic flight along the operational orbit until the first de-orbit burn is commanded. Next, the vehicle performs a de-orbit burn, where the duration of such burn primarily depends on the capabilities of the propulsion sub-system. Please note that the deorbitation guidance could command multiple burns with the goal of reducing the errors at the EIP (Yabar Valles, 2016). Once the final engine firing has been performed, the vehicle flies on a ballistic trajectory towards the EIP, where sufficient air density builds up to consider atmospheric flight. At this point, the abort reentry guidance is enabled, usually triggered by a minimum deceleration threshold as detected by the on-board accelerometers. Once enabled, the abort reentry guidance issues angle-of-attack and bank angle commands that bring the vehicle from the EIP to the terminal point, where appropriate altitude and speed conditions must be met for the DRS phase to begin (Kerr et al., 2012). Finally, the parafoil is deployed and the vehicle flies towards the designated landing site.

Given the framework discussed above, this thesis deals with the development of the abort reentry guidance and the study of the landing site reachability which are both critical elements of the system architecture. Consequently, the development of the abort deorbitation guidance and the final phase parafoil guidance are left for future work.

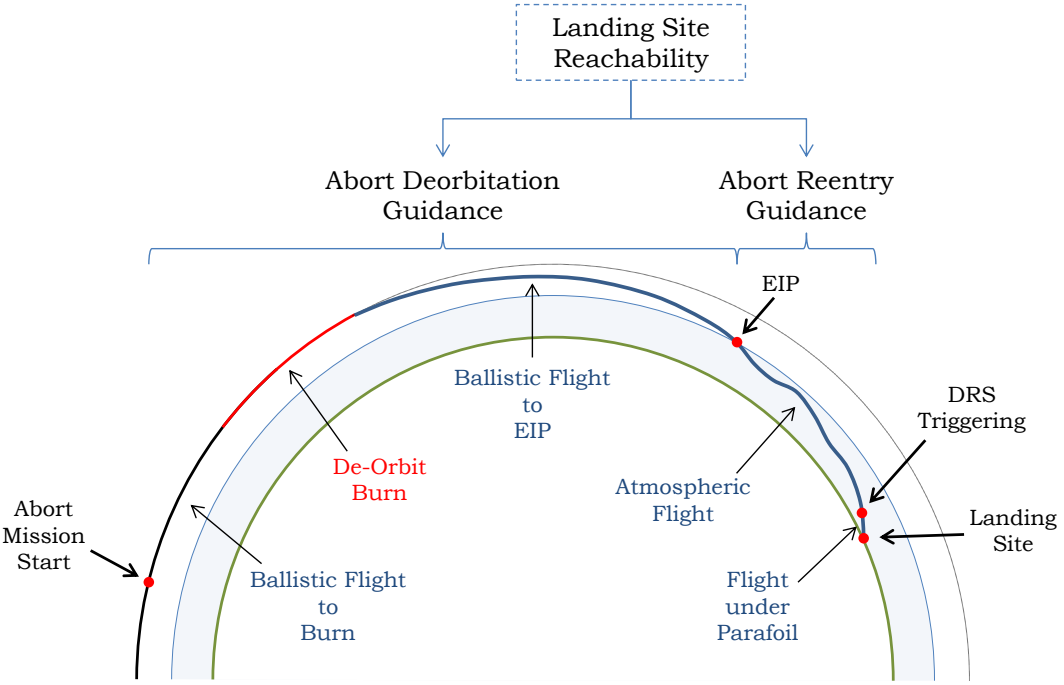

Figure 2.4: Orbital abort mission profile as tackled in this thesis.

2.3 MISSION REQUIREMENTS & CONSTRAINTS

The abort mission case that forms the thesis assignment is established by means of a set of mission requirements established by ESA (Yabar Valles, 2016). The goal of this section is to list and evaluate these requirements to establish a common baseline. Please note that some requirements were eliminated or modified to refine the scope of the thesis proposal. All changes presented here were agreed during the kick-off meeting of this work. Additionally, several system requirements were derived from the mission requirements.

- ~~**REQ-01:** The operational orbit shall be circular up to 500 km and in between 37 deg and 52 deg inclination. A minimum altitude of 300 km shall be considered.~~

  Reason: Large variations in entry altitudes and operational orbit inclinations are not considered due to schedule limitations of the thesis project. Consequently, this thesis assumes an operational orbit at 500 km altitude and an inclination of 52 deg as agreed in the kick-off meeting. As will be discussed in Chapter 9, the architecture of guidance system developed in this thesis can be adapted in future work to accept this requirement.

- ~~**REQ-02:** The spacecraft shall be launched from a Vega-C from Kourou.~~

  Analysis: This thesis proposal only considers aborts from orbit and during descent.

- **REQ-03:** The mass of the FRV shall be similar to the one of the SPHYNX vehicle.

  Analysis: The SPHYNX vehicle has a maximum allowable mass of 550 kg.

- **REQ-04:** The aerodynamic coefficients of the FRV shall be the ones of the SPHYNX vehicle.

- ~~**REQ-05:** The vehicle has 2 flap actuators like the IXV in order to trim the vehicle on the longitudinal (symmetrical deflections) and lateral (unsymmetrical deflections) axes during the re-entry phase. The control around the yaw axis is obtained using the thrusters control capability.~~

  Reason: Rotational dynamics are not considered in this study.

- ~~**REQ-06:** The vehicle shall have TBC reaction control thrusters to control the vehicle both in orbit and during re-entry. The single thruster type chosen is TBC.~~

  Reason: Rotational dynamics are not considered in this study.

- ~~**REQ-07:** It shall be assumed that the FRV has similar cross-range capability as the IXV, about 500 km to either side of the orbital path during the re-entry. This cross-range capability increases the landing latitude by about 4 deg.~~

  Reason: The cross-range capability is fixed by the aerodynamic database of the SPHYNX vehicle.

- **REQ-08:** The vehicle shall have the capability to abort or change trajectory during entry descent and landing.

  Analysis: This requirement imposes the need for an adaptive on-board trajectory planner.

- **REQ-09:** A runway landing shall be preferred, but an open field landing may be also considered.

  Analysis: This requirement allows for additional landing locations, should it become unfeasible to reach none of the landing sites specified in **REQ-10** and Table 2.1.

- **REQ-10:** The primary landing site shall be Santa Maria in Azores (36.974 deg latitude, -25.165 deg longitude).

- **REQ-11:** Other possible landing sites shall be selected when an abort or a change of trajectory is needed. These landing sites shall be compliant with the following characteristics:

  – The runway shall be at least 3 km long.

  – The closest town to the runway shall be at least 2 km away.

  Analysis: Although all of the runways of the airfields and airports specified in **REQ-10** and Table 2.1 are at least 3 km long, some of them are within a 2 km radius from a populated area. Further discussion at the kick-off meeting of this thesis lead to the acceptance of this requirement, despite the proximity to populated areas.

- ~~**REQ-12:** The EIP radius shall be considered of an orbital radius of 6498 km.~~

Reason: The specified EIP radius in this requirement would be equivalent to an altitude of 127 km, assuming a spherical shape model for Earth. During the kick-off meeting, the baseline altitude was changed to 120 km, as indicated by **REQ-S-07**. For simulation purposes, this requirement demands the atmosphere model to be defined up to this altitude, as denoted in **REQ-S-09**. As discussed above, large deviations in altitude are left for future research.

- **REQ-13:** Assuming an impulsive de-orbit manoeuvre of 125 m/s from 500 km orbit and the EIP radius specified above, the following approximate final conditions shall apply:
  - Resulting perigee radius: 6444 km.
  - Relative entry flight-path angle: -1.28 deg.
  - Relative entry velocity: 7.64 km/s.

  Analysis: The EIP conditions defined in this requirement correspond to an operational orbit at 500 km altitude and 51.83 deg inclination. Lower orbit inclination values would lead to slower relative entry velocities, reaching a minimum of 7.56 km/s for a 37 deg. Should the operational orbit altitude decrease to the minimum of 300 km altitude, the relative entry velocity decreases to 7.44 km/s and the flight path angle becomes steeper, reaching a value of -1.82 deg. Please note that this analysis assumes that the maneuver of 125 m/s is kept the same. However as discussed above, large variations in the operational orbit inclination and entry altitude are not considered. For this reason, during the kick-off meeting it was agreed that the entry flight-path angle and the entry velocity would be kept to the values specified here.
- **REQ-14:** The conditions for safe initiation at Descent and Recovery System (DRS) shall be as follows:
  - Altitude from 10 km to 13 km.
  - Vertical speed less than 140 m/s.
  - Mach number less than 1.

  Analysis: The DRS phase is in charge of bringing the vehicle to safety by means of a parafoil (Yabar Valles, 2016). The conditions specified in this requirement indicate the termination point of the abort guidance system discussed in this thesis. Thus, the DRS phase will not be part of the thesis project.
- ~~**REQ-15:** The vehicle shall try to minimize the propellant used for the trajectory changes.~~

  Reason: Propulsive maneuvers are not considered as part of this study.
- **REQ-16:** The time shall not be optimized since one orbit period (500 km circular orbit) is already less than 2 hours.
- ~~**REQ-17:** The vehicle shall have 4x90N thrusters arranged purely in one direction for the de-orbit thrust.~~

  Reason: Propulsive maneuvers are not considered as part of this study.
- **REQ-18:** All possible FRV trajectories shall contemplate failures, abort scenarios, changes on the landing site or many other aspects in which a modification of the final position is required.

  Analysis: This requirement imposes the need for an adaptive on-board trajectory planner.
- **REQ-19:** The trajectory optimization shall be done using the Aerospace Trajectory Optimization Software (ASTOS).

  Analysis: Trajectory optimization theory is a research topic on its own, justifying the use of proven tools to alleviate the software development and testing effort made in this thesis. Using external validated software such as ASTOS for the generation of optimal trajectories, enables the thesis project to focus on trajectory guidance and planning.
- **REQ-20:** Optimal and adaptive guidance shall be used.
- **REQ-21:** The vehicle shall be able to assume a failure and simulate another landing.

  Analysis: This requirement imposes the need for an adaptive on-board trajectory planner.

The above list of requirements is expanded with a number of system requirements with the goal of fully defining the mission case. Please note that **REQ-S-01** to **REQ-S-05** are directly obtained from the IXV's mission planning.

- **REQ-S-01:** The spacecraft shall withstand a maximum aerodynamic load of 3 g (Kerr et al., 2012).
- **REQ-S-02:** The spacecraft shall withstand a maximum dynamic pressure of 6000 Pa. (Zaccagnino et al., 2011)
- **REQ-S-03:** The spacecraft shall withstand a maximum heating rate of 670 kW/m^2 (Kerr et al., 2012).
- **REQ-S-04:** The spacecraft shall withstand a maximum heat load of 460 MJ/m^2. (Zaccagnino et al., 2011)
- **REQ-S-05:** The DRS shall be triggered within a 10 km radius from the designated landing site. (Zaccagnino et al., 2011)
- **REQ-S-06:** The reentry guidance shall terminate at the moment of DRS triggering.
- **REQ-S-07:** The EIP shall be placed at 120 km altitude.
- **REQ-S-08:** Should the primary landing site become inaccessible, the vehicle shall land in one of the landing sites listed in Table 2.1.
- **REQ-S-09:** The atmosphere model shall be defined up to at least 120 km altitude.
- **REQ-S-10:** Wind shall not be considered in this study.

As a final note, although referred as "FRV" in the requirements document, the reference vehicle will be denoted as SPHYNX in the remainder of the report.

Table 2.1: Alternative landing sites for SPHYNX.

ID	Landing Site	Geodetic Latitude [deg]	Longitude [deg]
1	Gran Canaria Airport	27.933	-15.388
2	Ben Guerir Air Force Base	32.128	-7.878
3	Perth Airport	-31.938	115.967
4	Darwin Airport	-12.411	130.878
5	Brisbane Airport	-27.394	153.121
6	Sidney Airport	-33.940	151.176
7	Woomera Airfield	-31.145	136.825

The above requirements are examined in terms of how critical they become for the design of the abort guidance system. They are classified as follows:

- **Driving Requirements:** Those requirements that drive the design of the abort guidance system more than average.
  - **REQ-04:** As will be discussed in Section 2.4.1, the SPHYNX aerodynamic database is not complete for the entire $\alpha - M$ space, thus constraining the solution space of the reachability analysis, ultimately leading to a more limited landing coverage. In addition, the SPHYNX aerodynamic database exhibits fast-changing drag values in the transonic regime, which result in an unpredictable behavior when the vehicle flies through such region of the $\alpha - M$ space.
  - **REQ-13, REQ-S-07:** The initial conditions established in these requirements restrict the vehicle velocity and altitude at the moment of entry. Since the entry state is a strong driver of the shape of the resulting trajectory, the landing site reachability results shown in Chapter 7 are only valid for the entry altitude and velocity conditions established in these requirements.
  - **REQ-14:** The terminal conditions established in this requirement constrain the final vehicle state at the triggering of the DRS. Although the reachability results shown in Chapter 7 indicate that these condition are easily satisfied, widening the range of allowable values could enlarge the solution space.
  - **REQ-S-08:** The number of landing sites and their spatial distribution has a direct impact on the size and spread of the entry windows. By imposing a predefined set of sites, the orbital abort opportunities of the system are largely constrained.
- **Killer Requirements:** Those requirements that drive the design of the abort guidance system to an unacceptable level.

- **REQ-S-02:** Given the initial entry velocity and altitude established in **REQ-13** and **REQ-S-07**, the entry trajectories studied in this thesis may reach maximum dynamic pressure values that range between 8 kPa and 10 kPa. This yields the 6 kPa constraint too restrictive and is thus removed to avoid an empty solution space.
- **REQ-S-03:** As will be discussed in Section 2.4, the nominal nose radius of SPHYNX is too small to comply with the maximum allowable heat-flux established in this requirement. If the nominal nose radius of 0.12 m is used, the maximum heat-flux experienced by the vehicle often exceeds 1000 kW/m^2 given the entry velocity and altitude established in **REQ-13** and **REQ-S-07**. Consequently, the vehicle's nose radius is increased to 0.5 m to satisfy the constraint of 670 kW/m^2.
- **REQ-S-04:** The maximum allowable heat load is related to the duration of the atmospheric flight, where longer flight times lead to a larger heat accumulation. Enforcing a maximum heat-load constraint severely limits the solution space, thus this requirement is removed to allow for more orbital abort opportunities.

2.4 REFERENCE VEHICLE

Selection of a relevant reference vehicle is crucial, since the mass and aerodynamic properties of such vehicle play an important role in shaping the resulting entry trajectories. As discussed in Section 2.3, it was agreed with ESA to select SPHYNX as the reference vehicle to be used throughout this study, due to its similarities with the PRIDE vehicle. The dimensions of SPHYNX to be used in this thesis proposal are given in Table 2.2, some of which are indicated in the technical drawings given in Figures 2.5 and 2.6.

Table 2.2: Dimensions of the SPHYNX reference vehicle (European Space Agency, 2016)

m [kg]	R_N [m]	L_{ref} [m]	W_{ref} [m]	S_{ref} [m^2]	X_m [m]	Y_m [m]	Z_m [m]	L_{bf} [m]	S_{bf} [m]
550	0.5*	2.8042	1.6570	2.4080	1.5476	0	0.2682	0.4577	0.2080

Table 2.2 lists m as the vehicle's mass, R_N as the nose radius, L_{ref} as the reference length, W_{ref} as the reference width and S_{ref} as the reference area. Furthermore, X_m, Y_m and Z_m indicate the position of the moment reference center. The vehicle's body flap length and area are given by L_{bf} and S_{bf}, respectively. Note that the nose radius of the vehicle has been artificially increased to 0.5 m from the nominal value of 0.12 m. The reason behind such increase arises from the fact that the nominal nose radius value is simply to small to comply with the heat-flux constraint established in Section 2.3.

In addition to the vehicle dimensions, the aerodynamic database of SPHYNX provides the aerodynamic coefficients as a function of the vehicle's Mach number, the vehicle's attitude and the control surface deflections (European Space Agency, 2016). An example output of such database is shown in Figures 2.7 and 2.8, which give the L/D as a function of Mach number and angle-of-attack. Furthermore, such figures show that the L/D of SPHYNX is larger than 1.0 in a wide area of the $\alpha - M$ plane, justifying its suitability as a CRV due to the reasons stated in Section 2.1. In addition, L/D serves as a measure of the vehicle's aerodynamic efficiency, where flying at a higher L/D extends the vehicle achievable range. This is discussed in Section 7.2, where it is seen how trajectories which need to fly further downrange to reach the landing site attempt to maintain a high L/D throughout the reentry. Note that it is often desired to fly through a region of the $\alpha - M$ plane that maximizes drag, which is normally done to track the heat-flux constraint in the early portion of the flight.

2.4.1. ASSESSMENT OF THE AERODYNAMIC DATABASE

The goal of this section is to discuss some of the features of SPHYNX's aerodynamic database, since as will be seen in the remainder of this thesis, such features play a key role in some of the aspects of the abort guidance system.

To begin with, the unprocessed aerodynamic database is divided upon the hypersonic regime and the subsonic-supersonic regime, as shown in Figures 2.7 and 2.8. To ensure continuity along the Mach number space, the hypersonic and subsonic-supersonic data is merged into one single regime. Figures 2.7 and 2.8 show a number of isolines, which indicates regions of the $\alpha - M$ space of equal and constant L/D. The spacing of the isolines provides an indication of how fast L/D changes along a particular direction. For instance, Figure 2.7 shows that L/D increases rapidly when the angle-of-attack increases from 0 deg to 20 deg until $M \approx 4.5$. On the contrary,

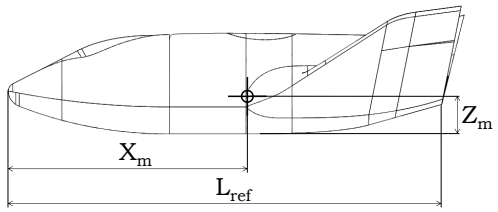

Figure 2.5: Side view of the SPHYNX reference vehicle (European Space Agency, 2016)

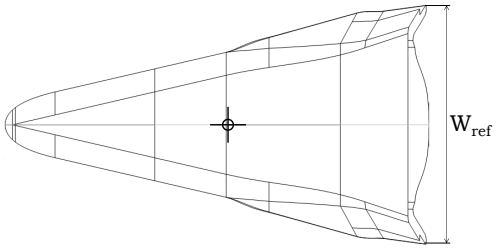

Figure 2.6: Bottom view of the SPHYNX reference vehicle (European Space Agency, 2016)

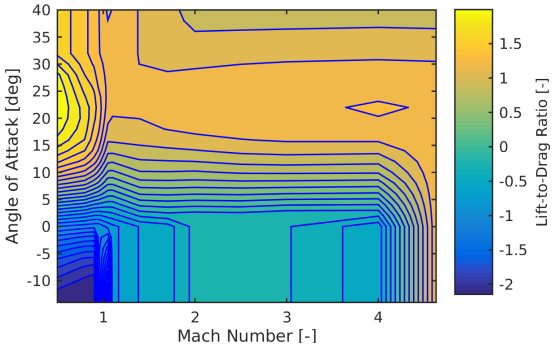

Figure 2.7: Lift-to-drag ratio of SPHYNX in the supersonic regime of the $\alpha - M$ space.

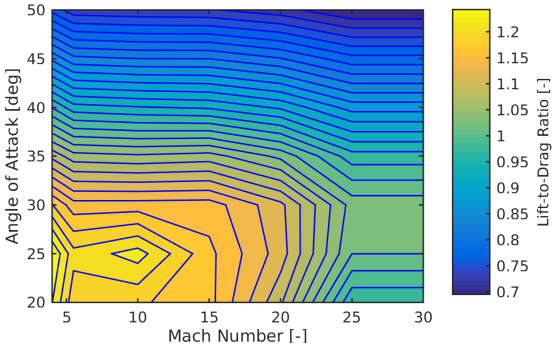

Figure 2.8: Lift-to-drag ratio of SPHYNX in the hypersonic regime of the $\alpha - M$ space.

L/D is relatively "stable" for angle-of-attack values larger than 20 deg, where $L/D \approx 1.2$. Furthermore, a local maximum is found at $M = 0.5$ and $\alpha = 22$ deg, where $L/D = 2.1$. Note that L/D decreases rapidly from this maximum as the flow transitions towards the supersonic regime. On the other hand, Figure 2.8 shows that the local maximum for L/D in the hypersonic regime is found at $M = 10$ and $\alpha = 25$ deg, where $L/D = 1.2$. In addition, Figure 2.8 shows that L/D tends to decrease rapidly from as the angle-of-attack increases. On the other hand, L/D remains approximately constant as the Mach number increases, which reflects the property of Mach number independence characteristic of hypersonic flows (Anderson Jr., 2006a). This phenomenon becomes more evident as the angle-of-attack increases, which causes such independence to be achieved at smaller Mach numbers.

As will be established in Section 2.3, this thesis makes use of a 3 degrees of freedom simulation and does not consider the presence of any wind. Note that the absence of wind results in an angle-of-sideslip β equal to zero. Since the total side-force coefficient C_S is primarily dependent on β, the side forces on the vehicle are zero (European Space Agency, 2016). Consequently, only the lift, drag and pitching moment coefficients are discussed in this section. Note that the aerodynamic database provides the clean form of such coefficients as well as the contributions of the body flap as a function of body flap deflection. In summary, the following aerodynamic coefficients are used:

$$C_L = C_{L,0}(M,\alpha) + \Delta C_{L,bf}\left(M,\alpha,\delta_{bf}\right) \tag{2.1}$$

$$C_D = C_{D,0}(M,\alpha) + \Delta C_{D,bf}\left(M,\alpha,\delta_{bf}\right) \tag{2.2}$$

$$C_m = C_{m,0}(M,\alpha) + \Delta C_{m,bf}\left(M,\alpha,\delta_{bf}\right) \tag{2.3}$$

where C_L is the total lift coefficient, $C_{L,0}$ is the clean lift coefficient and $C_{L,bf}$ is the lift coefficient contribution of the body flap deflection. Similarly, C_D is the total drag coefficient, $C_{D,0}$ is the clean drag coefficient and $C_{m,bf}$ is the drag coefficient contribution of the body flap deflection. Finally, C_m is the total pitching moment coefficient, $C_{m,0}$ is the clean pitching moment coefficient and $C_{m,bf}$ is the pitching moment coefficient contribution of the body flap deflection.

All the simulations performed in this thesis ensure that the SPHYNX vehicle is trimmed. Vehicle trimming is

achieved if the following two conditions are satisfied:

$$\frac{\partial C_m}{\partial \alpha} < 0 \tag{2.4}$$

$$C_m = 0 \tag{2.5}$$

where the first condition is a property of the vehicle design and it ensures that any disturbances in angle-of-attack result in a pitching moment that counteracts such disturbance, bringing the vehicle to its pitch equilibrium state. The second condition ensures that no pitching moment acts on the vehicle, and it is achieved by setting a body-flap deflection that counteracts the clean pitching moment coefficient. Such deflection is found by solving Equations (2.3) and (2.5) for every angle-of-attack and Mach number of the $\alpha - M$ space using the root-finding algorithm discussed in Section 5.1.2. The required body-flap deflections to trim the vehicle are shown in Figures 2.9 and 2.10. In general, increasing the angle-of-attack leads to smaller body-flap deflections for trim, which suggests that the magnitude of the pitching moment coefficient reduces with increasing angle-of-attack, as established by Equation (2.4). Figure 2.10 shows how the isolines group closer to each other in the subsonic-supersonic regime, thus indicating relatively fast changes in pitching moment and body-flap deflection in this region.

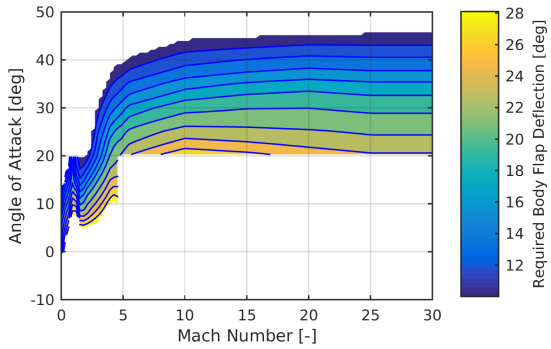

Figure 2.9: Elevator deflection required to trim the SPHYNX vehicle for the whole flight regime.

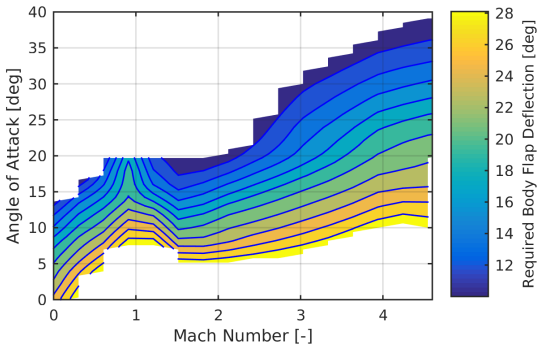

Figure 2.10: Elevator deflection required to trim the SPHYNX vehicle for the subsonic-supersonic regime.

Once the body-flap deflection required for trim is found, the total lift and drag coefficients are found as described by Equations 2.1 and 2.2, respectively. Close examination of Figures 2.9 and 2.10 reveal areas of $\alpha - M$ space where the body flap deflection is undefined. There are two reasons why this occurs:

1. **Unavailable Data:** The aerodynamic database of SPHYNX is not fully defined for entire $\alpha - M$ space. For instance, the subsonic-supersonic regime is only defined for Mach numbers in the $[0.0, 4.6]$ range and angle-of-attack values in the $[-10.0, 40.0]$ deg range. Similarly, the hypersonic regime is defined for Mach numbers in the $[4.6, 30.0]$ range and angle-of-attack values in the $[20.0, 50.0]$ deg range. Extrapolation of coefficient data in the non-defined areas is avoided due to the questionable validity of the produced coefficients.

2. **Non-trimmable Area:** In the remaining areas of the $\alpha - M$ space, the body flap deflection required to trim the vehicle exceed the maximum deflections allowed by the actuator mechanism. Although the body flap mechanism allows deflections in the $[5.0, 35.0]$ degree range, the maximum deflections are set to $[10.0, 30.0]$ to allow for aileron usage of the body flaps (European Space Agency, 2016).

In essence, the aerodynamic database constraints mentioned above delimit the angle-of-attack values that can be commanded as a function of Mach number, as shown in Figures 2.11 to 2.18. A key feature to highlight is the bottleneck created by such constraints in the supersonic regime for angle-of-attack values near 20 deg. As will be discussed in Section 7.2.2, this bottleneck strongly constrains the solution space of the reachability study.

Figures 2.11 and 2.12 show that L/D is still above 1.0 for most of the flight regime when the vehicle is trimmed, where higher L/D values are achieved as the Mach number reduces and the angle-of-attack is kept between approximately 15 deg and 30 deg. Figures 2.13 and 2.15 show that higher drag and lift values can be achieved by increasing the angle-of-attack, although this comes at the penalty of a reduced lift-to-drag ratio. As will be discussed in Section 7.2, this behavior is exploited in the early portion of the flight to track the heat-flux constraint defined by **REQ-S-03**. Figures 2.14 and 2.16 show the trimmed drag and lift coefficients in the subsonic-

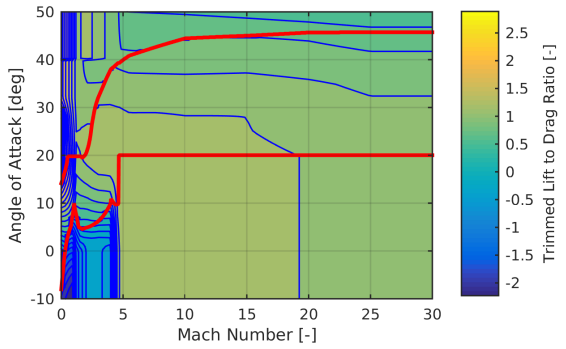

Figure 2.11: Trimmed lift-to-drag ratio of SPHYNX in the complete regime of the $\alpha - M$ space.

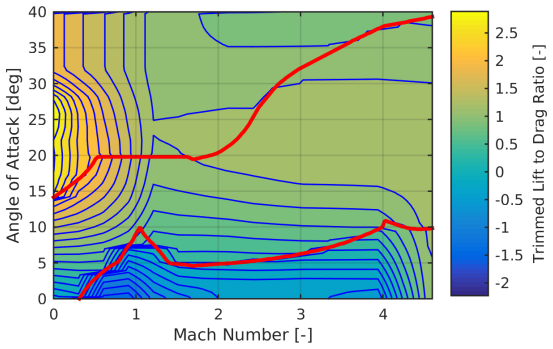

Figure 2.12: Trimmed lift-to-drag ratio of SPHYNX in the subsonic-supersonic regime of the $\alpha - M$ space.

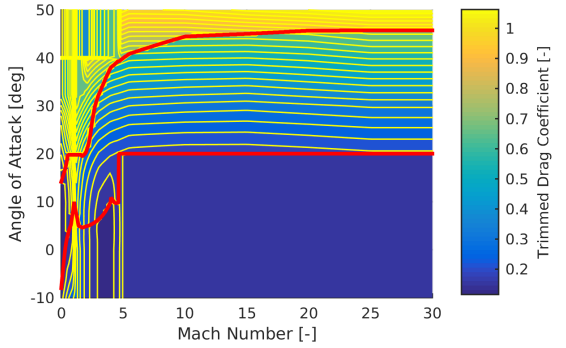

Figure 2.13: Trimmed drag coefficient of SPHYNX in the complete regime of the $\alpha - M$ space.

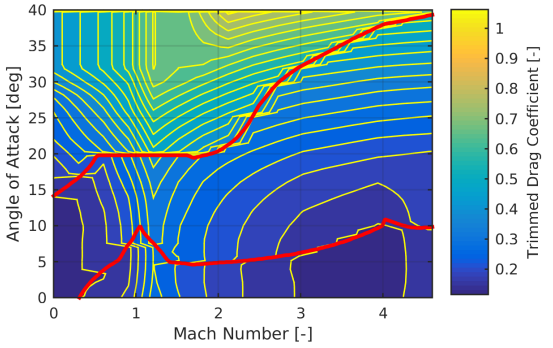

Figure 2.14: Trimmed drag coefficient of SPHYNX in the subsonic-supersonic regime of the $\alpha - M$ space.

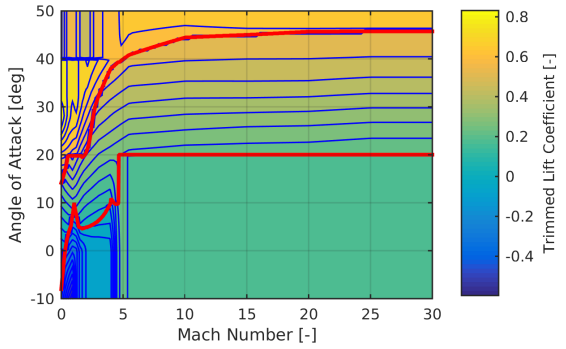

Figure 2.15: Trimmed lift coefficient of SPHYNX in the complete regime of the $\alpha - M$ space.

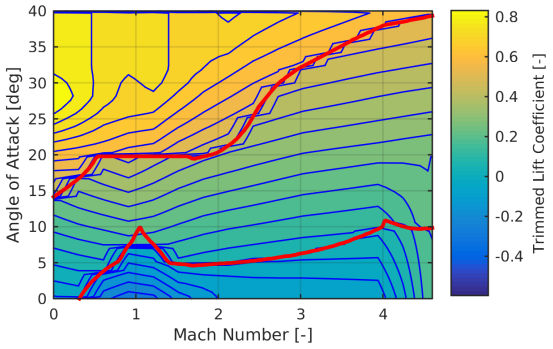

Figure 2.16: Trimmed lift coefficient of SPHYNX in the subsonic-supersonic regime of the $\alpha - M$ space.

supersonic regime, respectively. These two figures show how the isolines are closely packed in such regime, suggesting rapid changes in the aerodynamic forces, which will prove to be problematic as explained in Section 7.2.2. Finally, Figures 6.7 and 2.18 shows that the trimmed pitching moment coefficient is zero in the region constrained by the body-flap deflection limits, as expected. Note that outside these limits, the body-flap deflections are set to the maximum allowable values that reduce the pitching moment.

It is important to emphasize that the database described in this section uses the nominal vehicle geometry, in other words, it employs a nose radius of 0.12 m. As mentioned in Section 2.3, this thesis assumes a nose radius of 0.5 m, but does not modify the nominal database. Increasing the nose radius leads to a stronger wave drag and shifts downstream the transition point from laminar to turbulent flow (Anderson Jr., 2006b). Consequently, the the flow velocity gradients over the vehicle change, leading to changes in the distribution of aerodynamic heating and skin friction. This is a source of error that should be accounted for in future studies, where the aerodynamic database should be re-assessed by performing new Computational Fluid Dynamics (CFD) methods. As a final note, the aerodynamic database of SPHYNX does not provide derivative information and should

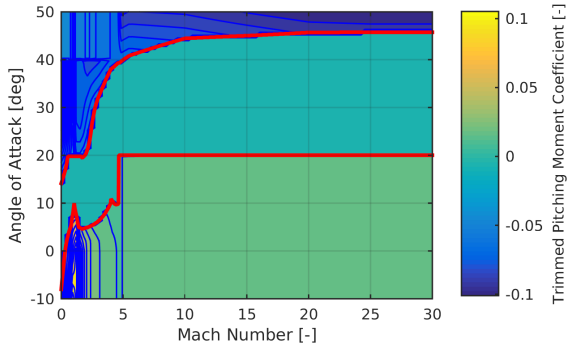

Figure 2.17: Trimmed pitching moment coefficient of SPHYNX in the complete regime of the $\alpha - M$ space.

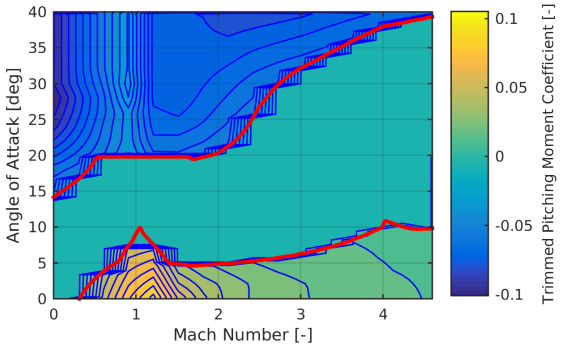

Figure 2.18: Trimmed pitching moment coefficient of SPHYNX in the subsonic-supersonic regime of the $\alpha - M$ space.

be thus computed with a finite difference method. This derivative information is required to compute the guidance gains, as will be discussed in Section 4.2.

3 FLIGHT MECHANICS

Understanding the motion of the SPHYNX vehicle during flight is an essential step in the process of developing a successful guidance system. The goal of this chapter is to provide all the tools and models necessary to provide a mathematical description of this motion. This area of study is referred as flight mechanics and it consists of two additional branches: *dynamics* and *kinematics*. On the one hand, dynamics focuses on the relations between the vehicle's velocity and the forces and torques that cause such velocity. On the other hand, kinematics studies the relations between the vehicle's velocity and its position. The SPHYNX vehicle is modelled as a point-mass with a given position with respect to the origin of a reference frame. It is assumed that the mass of the vehicle remains constant throughout the flight and that Earth rotates with a constant angular velocity. The forces acting on the point-mass and the resulting velocities and accelerations are modelled as vectors, whose magnitude and direction are related by Newton's Laws of Motion:

1. A body remains at rest or travels at a constant speed unless a force is applied to it.
2. A body of constant mass m experiencing a force F experiences an acceleration a related to F by $F = ma$.
3. Whenever a first body exerts a force F on a second body, the second body exerts a force $-F$ on the first body. F and $-F$ are equal in magnitude and opposite in direction.

Newton's Laws of Motion only apply in a non-rotating and non-accelerating (inertial) reference frame. The description of the forces experienced throughout the flight is simplified by defining multiple auxiliary reference frames, which are identified in Section 3.1. Ultimately, these flight forces need to be expressed in the inertial frame, requiring the transformations given in Section 3.2. Furthermore, the vehicle position and velocity in space, also known as the state, is given by the set of variables given in Section 3.3. Next, Section 3.4 identifies several environment models required to fully describe the forces experienced during the flight. The actual definition of these forces is given in Section 3.5. Section 3.6 puts together all of the tools and models in the set of equations that describe SPHYNX's motion during flight. Finally, Section 3.7 defines a number of path constraints commonly enforced during the trajectory planning process.

3.1 REFERENCE FRAMES

This section introduces the reference frames to be used in the derivation of the equations of motion. Each frame consists of three orthonormal axis, each of them pointing in a particular direction. All of the frames hereby presented are right-handed and have an independent notation.

Inertial Planetocentric Reference Frame, $\mathcal{F}_I$ The origin of the inertial planetocentric reference frame is located at the Earth's center of mass. The Z_I axis points north and is parallel to the Earth's rotational axis. The X_I axis points at the mean vernal equinox at a specific reference time. The Y_I is parallel to the Equatorian-plane thus completing the right-handed system. The J2000 system establishes a common reference time convention for $\mathcal{F}_I$, which defines the mean vernal equinox on January 1st, 2000 at 12 hrs (Tapley et al., 2004a). Note that Earth is accelerating due to its orbital motion around the Sun and presents precession and nutation motions as a consequence of third-body attractions (Wakker, 2015). However, these effects can be neglected given the purpose of this study and thus can yield $\mathcal{F}_I$ as an inertial frame.

Rotating Planetocentric Reference Frame, $\mathcal{F}_R$ The origin of the rotating planetocentric reference frame is also located at the Earth's center of mass. The Z_R axis also points north and is parallel to the Earth's rotational axis. The X_R axis point to the Greenwich meridian and the Y_R axis completes the right-handed system. The rotating planetocentric reference frame rotates about the Z_R axis with the Earth's angular velocity. The orientation of $\mathcal{F}_I$ and $\mathcal{F}_R$ coincides at the reference time.

Body Reference Frame, $\mathscr{F}_B$ The body reference frame is fixed to the vehicle and its origin is located at the vehicle's centre of mass. The X_B axis points forward toward the vehicle's nose and along its longitudinal axis. The Z_B axis points downwards and the Y_B axis points starboard, thus completing the right-handed system.

Vertical Reference Frame, $\mathscr{F}_V$ The origin of the vertical reference frame is also located at the vehicle's centre of mass. The Z_V axis points towards the Earth's geometric centre, parallel to the gravity vector. The X_V axis points in a northern direction and is parallel to a meridian plane. The Y_V completes the right-handed system.

Trajectory Reference Frame, $\mathscr{F}_T$ The trajectory reference frame has its origin located at the vehicle's centre of mass. The X_T axis points in the direction of the velocity vector relative to $\mathscr{F}_R$. The Z_T axis points downwards and lies on the vertical plane formed by the X_T axis and the Z_V axis. The Y_T completes the right-handed system.

Aerodynamic Reference Frame, $\mathscr{F}_A$ The origin of the aerodynamic reference frame is centered at the vehicle's centre of mass. The X_A axis points in the direction of the velocity vector relative to $\mathscr{F}_R$. The Z_A axis is collinear with the aerodynamic lift force vector but points in the opposite direction. The Y_A completes the right-handed system.

3.2 FRAME TRANSFORMATIONS

In this section, the transformations between the different auxiliary reference frames are presented. The goal of these transformations is to express a given vector in coordinates of different frames, without altering the direction and magnitude of the vector. All of the presented transformations are found in (Mooij, 2014) and they are all based on the formalism detailed in this section, referred as Euler angle rotations. Please note that other rotation formalisms exists, such as the Euler eigenaxis rotations, quaternions or the Gibbs-Rodrigues parameters (Wie, 2008). Although some of these formalism prove to be computationally more efficient and do not suffer from singularities, they are rather unintuitive and considered more appropriate for the representation of rotational dynamics and kinematics.

Figure 3.1 shows the reference frames $\mathscr{F}_1$ and $\mathscr{F}_2$ and the test vector $\boldsymbol{x}$. In this example, $\mathscr{F}_2$ is rotated about the X_1 axis by angle θ. By projecting the vector onto the axis of both reference frames and constructing the triangles shown in Figure 3.1, the following coordinate relation is found:

$$\begin{aligned} x_2 &= x_1 \\ y_2 &= a + b = y_1 \cos\theta + z_1 \sin\theta \\ z_2 &= c - d = -y_1 \sin\theta + z_1 \cos\theta \end{aligned} \tag{3.1}$$

where a, b, c, d are the auxiliary geometry variables shown in Figure 3.1 and x, y, z are the coordinates of the test vector $\boldsymbol{x}$. The above coordinate relation can be expressed in a matrix form, as shown in the following equation:

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \tag{3.2}$$

The transformation matrix just presented is referred as a unit-axis rotation matrix about the X axis. Similarly, rotation matrices for the Y axis and the Z axis can be derived. The complete set of unit-axis rotation matrices is shown in the following three equations:

$$\boldsymbol{C_x}(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \tag{3.3}$$

$$\boldsymbol{C_y}(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{3.4}$$

$$\boldsymbol{C_z}(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

Unit-axis rotation matrices have multiple properties. To begin with, unit-axis rotation matrices are capable of describing any combined rotation. This is achieved by multiplying multiple matrices in sequence. For example, assume that the arbitrary transformation of frame $\mathscr{F}_A$ to frame $\mathscr{F}_B$ can be achieved by a rotation over an angle θ_x about the X axis, followed by a rotation over an angle θ_y about the Y axis, followed by a rotation over an angle θ_z about the Z axis. The mathematical description of such transformation sequence is shown in the following equation:

$$\boldsymbol{C_{2,1}} = \boldsymbol{C_z}(\theta_z)\,\boldsymbol{C_y}(\theta_y)\,\boldsymbol{C_x}(\theta_x) \tag{3.6}$$

Furthermore, unit-axis rotation matrices are orthonormal which means that the inverse and transpose matrices are equal, as shown in Equation (3.7). Please note that the product of unit-axis rotation matrices is also orthonormal.

$$\boldsymbol{C_{1,2}} = \boldsymbol{C_{2,1}^T} = \boldsymbol{C_{2,1}^{-1}} \tag{3.7}$$

In essence, the orthonormality property is expected, since as mentioned above, frame transformations must not alter the magnitude and direction of the vector. Finally, the inverse of a rotation is found by inverting the order and signs of each individual transformation matrix, as shown in the following equation:

$$\boldsymbol{C_{A,B}} = \boldsymbol{C_x}(-\theta_x)\,\boldsymbol{C_y}(-\theta_y)\,\boldsymbol{C_z}(-\theta_z) \tag{3.8}$$

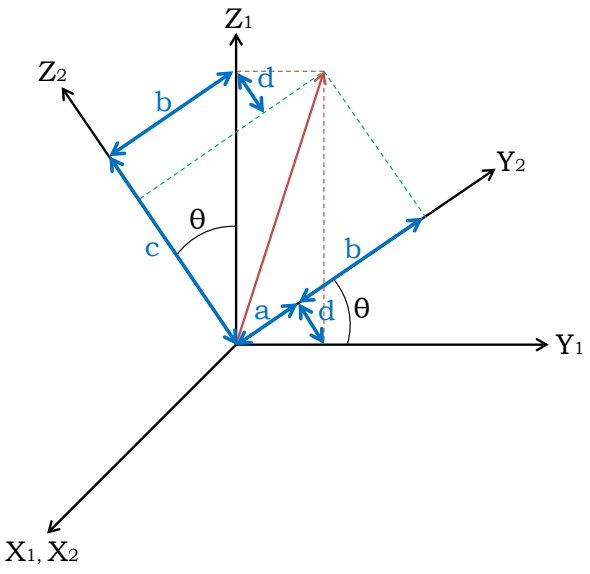

Figure 3.1: Geometry of the transformation between $\mathscr{F}_1$ and $\mathscr{F}_2$.

Next, a set of standard frame transformations is shown. Please note that it is possible to combine and invert the rotation matrices presented to produce additional frame transformations.

Rotating Planetocentric Frame to Inertial Planetocentric Frame The transformation from $\mathscr{F}_R$ to $\mathscr{F}_I$ consists of a single rotation about the Z_R axis. The rotation occurs over an angle $\omega_E t$, which measures how far the Greenwich meridian has rotated from its position at the reference epoch defined by the J2000 system. Here ω_E

is the rotational speed of Earth and t is the elapsed time from the reference epoch. Given the above definition, the Greenwich meridian is aligned with the vernal equinox at the initial simulation time t_0. The transformation of $\mathscr{F}_R$ to $\mathscr{F}_I$ is described by the matrix shown in Equation (3.9). The geometry of this transformation is shown on Figure 3.2.

$$\boldsymbol{C_{I,R}} = \boldsymbol{C_z}(-\omega_E t) = \begin{bmatrix} \cos\omega_E t & -\sin\omega_E t & 0 \\ \sin\omega_E t & \cos\omega_E t & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.9}$$

Vertical Frame to Rotating Planetocentric Frame The transformation from $\mathscr{F}_V$ to $\mathscr{F}_R$ consists of a $\left(\frac{\pi}{2}+\delta\right)$ rotation about the Y_V axis, followed by a $-\tau$ rotation about the Z_V axis. Here, τ is the geocentric longitude and δ is the geocentric latitude. The transformation of $\mathscr{F}_V$ to $\mathscr{F}_R$ is described by the matrix shown in Equation (3.10). c

$$\boldsymbol{C_{R,V}} = \boldsymbol{C_z}(-\tau)\,\boldsymbol{C_y}\left(\frac{\pi}{2}+\delta\right) = \begin{bmatrix} -\cos\tau\sin\delta & -\sin\tau & -\cos\tau\cos\delta \\ -\sin\tau\sin\delta & \cos\tau & -\sin\tau\cos\delta \\ \cos\delta & 0 & -\sin\delta \end{bmatrix} \tag{3.10}$$

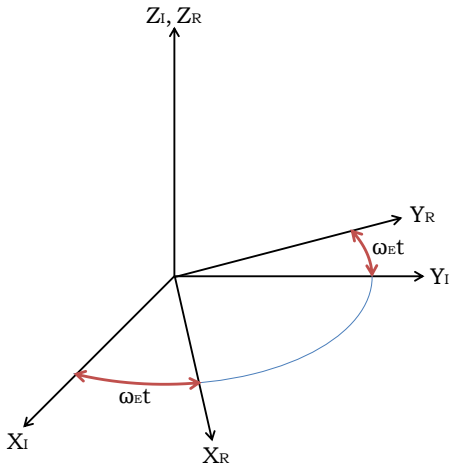

Figure 3.2: Geometry of the transformation between $\mathscr{F}_R$ and $\mathscr{F}_I$.

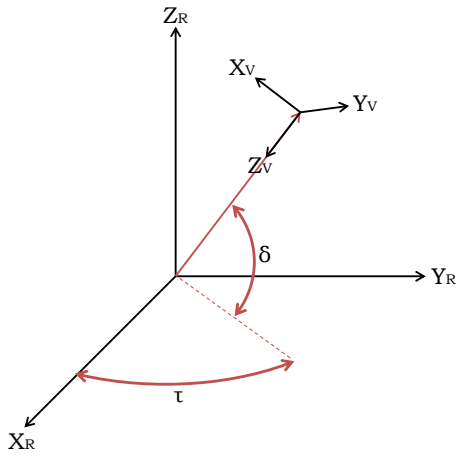

Figure 3.3: Geometry of the transformation between $\mathscr{F}_V$ and $\mathscr{F}_R$.

Trajectory Frame to Vertical Frame The transformation from $\mathscr{F}_T$ to $\mathscr{F}_V$ consists of a $-\gamma$ rotation about the Y_T axis, followed by a $-\chi$ rotation about the Z_T axis. Here, γ is the flight-path angle and χ is the heading angle. Such transformation is achieved using Equation 3.11. The geometry of this transformation is shown on Figure 3.4.

$$\boldsymbol{C_{V,T}} = \boldsymbol{C_z}(-\chi)\,\boldsymbol{C_y}(-\gamma) = \begin{bmatrix} \cos\chi\cos\gamma & -\sin\chi & \cos\chi\sin\gamma \\ \sin\chi\cos\gamma & \cos\chi & \sin\chi\sin\gamma \\ -\sin\gamma & 0 & \cos\gamma \end{bmatrix} \tag{3.11}$$

Aerodynamic Frame to Trajectory Frame The transformation from $\mathscr{F}_A$ to $\mathscr{F}_T$ consists of a single $-\sigma$ rotation about the X_A axis. Here, σ is the bank angle. The matrix that transforms the aerodynamic frame to the trajectory frame is given in Equation 3.12. The geometry of this transformation is shown on Figure 3.5.

$$\boldsymbol{C_{T,A}} = \boldsymbol{C_x}(\sigma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\sigma & \sin\sigma \\ 0 & -\sin\sigma & \cos\sigma \end{bmatrix} \tag{3.12}$$

Body Frame to Aerodynamic Frame The transformation from $\mathscr{F}_B$ to $\mathscr{F}_A$ consists of a $-\alpha$ rotation about the Y_B axis, followed by a β rotation about the Z_B axis. Here, α is the angle of attack and β is the angle of sideslip.

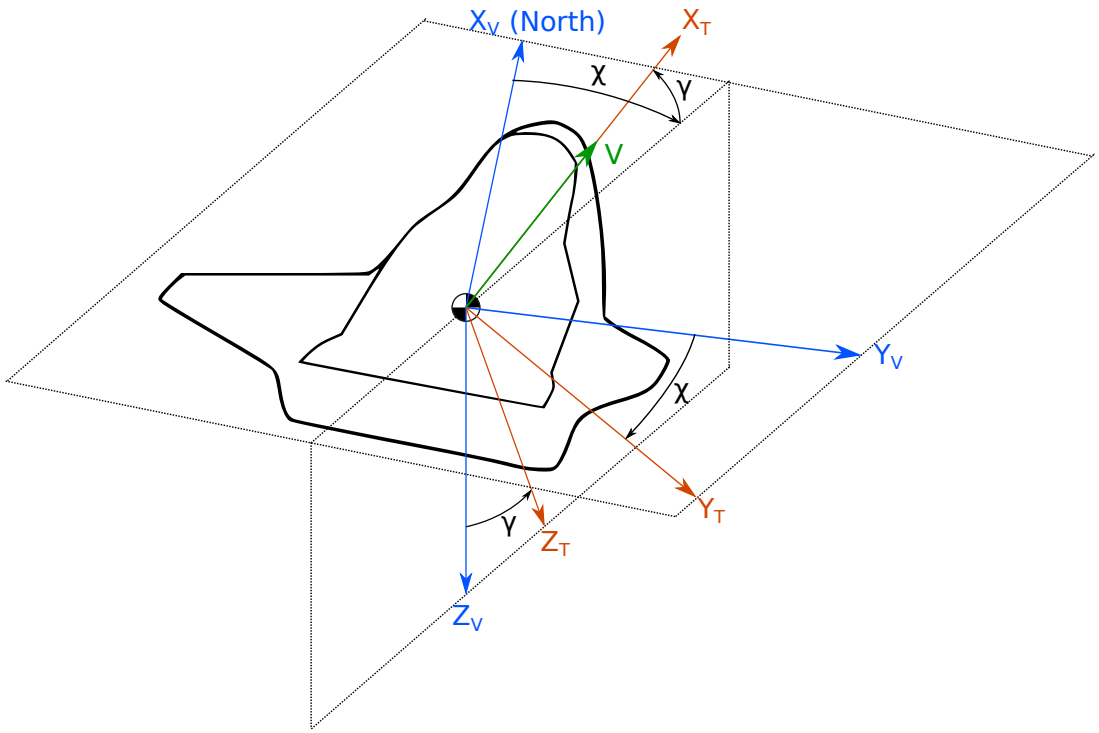

Figure 3.4: Geometry of the transformation between $\mathcal{F}_V$ and $\mathcal{F}_T$.

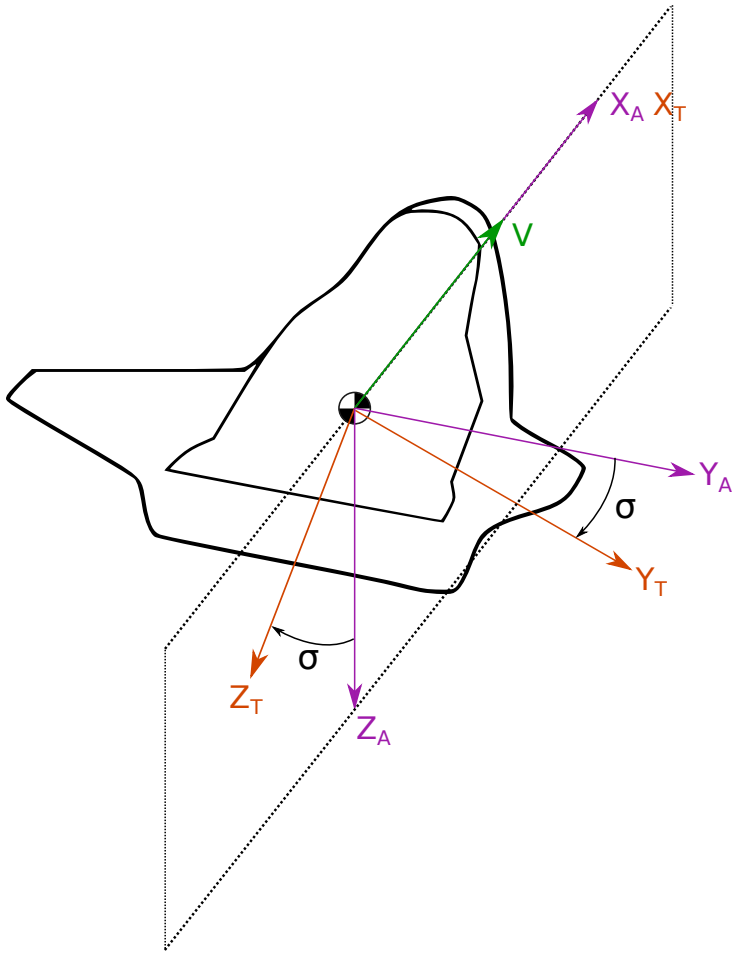

Figure 3.5: Geometry of the transformation between $\mathcal{F}_A$ and $\mathcal{F}_T$.

The associated transformation matrix is shown below:

$$\boldsymbol{C_{A,B}} = \boldsymbol{C_z}\left(\beta\right)\boldsymbol{C_y}\left(-\alpha\right) = \begin{bmatrix} \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \\ -\cos\alpha\sin\beta & \cos\beta & -\sin\alpha\sin\beta \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \tag{3.13}$$

Trajectory Frame to Body Frame The transformation from $\mathscr{F}_T$ to $\mathscr{F}_B$ consists of a $-\sigma$ rotation about the X_T axis, followed by a $-\beta$ rotation about the Z_T axis, followed by an α rotation about the Y_T axis. Here, α is the angle of attack, β is the angle of sideslip and σ is the bank angle. The associated transformation matrix is shown in the following equation:

$$\begin{aligned} \boldsymbol{C_{B,T}} &= \boldsymbol{C_y}\left(\alpha\right)\boldsymbol{C_z}\left(-\beta\right)\boldsymbol{C_x}\left(-\sigma\right) = \\ &= \begin{bmatrix} \cos\alpha\cos\beta & \sin\alpha\sin\sigma - \cos\alpha\sin\beta\cos\sigma & \cos\alpha\sin\beta\sin\sigma - \sin\alpha\cos\sigma \\ \sin\beta & \cos\beta\cos\sigma & -\cos\beta\sin\sigma \\ \sin\alpha\cos\beta & \cos\alpha\sin\sigma - \sin\alpha\sin\beta\cos\sigma & \cos\alpha\cos\sigma + \sin\alpha\sin\beta\cos\sigma \end{bmatrix} \end{aligned} \tag{3.14}$$

3.3 STATE VARIABLES

The state variables uniquely define the vehicle kinematics with respect to a specific reference frame. State variables are grouped in a state vector which contains information regarding the vehicle's position and velocity. The goal of this section is to present multiple state vector representations and to provide the required conversion steps that relate them. Please note that the SPHYNX vehicle is modelled as a point-mass, thus only translational kinematics are presented.

3.3.1. CARTESIAN STATE VARIABLES

The Cartesian state variables express the vehicle's position and velocity with respect to $\mathscr{F}_I$ and $\mathscr{F}_R$. The Cartesian velocity can also be expressed with respect to $\mathscr{F}_V$. The Cartesian state variables are the projections of the position and velocity vectors onto the three axis of the chosen reference frame. Such projections produce three unique coordinates. These coordinates are linearly combined to reconstruct the state vectors' length and direction. In this thesis, usage of the Cartesian state variables in $\mathscr{F}_I$ is predominant since they are used to write the equations of motion, which are part of a core software module, as will be described in Section 6.2. Despite this, Cartesian state variables in $\mathscr{F}_R$ and $\mathscr{F}_V$ are still used to perform the state variable transformations that interface the core simulation software with other supporting modules. Thus they are described here for the sake of clarity.

To begin with, the Cartesian position in $\mathscr{F}_I$ is given by the following equation:

$$\boldsymbol{r_I} = \begin{pmatrix} x_I \\ y_I \\ z_I \end{pmatrix} \tag{3.15}$$

The Cartesian velocity in $\mathscr{F}_I$ is obtained by differentiating the Cartesian position in $\mathscr{F}_I$ with respect to time, as shown in Equation (3.16). Since $\mathscr{F}_I$ is stationary, the magnitude of the Cartesian velocity in $\mathscr{F}_I$ is also known as the *inertial velocity*.

$$\boldsymbol{V_i^I} = \begin{pmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{z}_I \end{pmatrix} = \begin{pmatrix} u_i \\ v_i \\ w_i \end{pmatrix} \tag{3.16}$$

Similarly, the Cartesian position and the Cartesian velocity in $\mathscr{F}_R$ are given by Equations (3.17) and (3.18), respectively. Since $\mathscr{F}_R$ is rotating, the magnitude of the Cartesian velocity in $\mathscr{F}_R$ is referred the vehicle's *groundspeed*.

$$\boldsymbol{r_R} = \begin{pmatrix} x_R \\ y_R \\ z_R \end{pmatrix} \tag{3.17}$$

$$\boldsymbol{V_g^R} = \begin{pmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{z}_R \end{pmatrix} = \begin{pmatrix} u_g \\ v_g \\ w_g \end{pmatrix} \tag{3.18}$$

Finally, the Cartesian position and the Cartesian velocity in $\mathscr{F}_V$ are given by Equations (3.19) and (3.20), respectively. Since $\mathscr{F}_V$ is also rotating, the magnitude of the Cartesian velocity in $\mathscr{F}_V$ is simply the vehicle's *groundspeed* expressed in components of $\mathscr{F}_V$.

$$\boldsymbol{r_V} = \begin{pmatrix} x_V \\ y_V \\ z_V \end{pmatrix} \tag{3.19}$$

$$\boldsymbol{V_g^V} = \begin{pmatrix} u_{g,V} \\ v_{g,V} \\ w_{g,V} \end{pmatrix} \tag{3.20}$$

It is important to realize that *the representation of a vector is not related to the magnitude of the vector itself*: the inertial velocity and the groundspeed have different magnitudes but they can be expressed in the same frame. For this reason, all velocity and acceleration vectors presented in this report are assigned a superscript to indicate the frame in which they are represented.

3.3.2. SPHERICAL STATE VARIABLES

The spherical coordinate system defines the vehicle's position with respect to $\mathscr{F}_R$ using the radial distance R, the longitude τ and the latitude δ. In addition, the vehicle's velocity is expressed using the groundspeed V_g, the flight-path angle γ_g and the heading angle χ_g. Since no wind is considered in this thesis, the subscript 'g' that differentiates between groundspeed and airspeed will be dropped in the remainder of this report. However, V_g is still referred as the groundspeed to differentiate it from the inertial speed.

The longitude τ is measured positively to the east from the Greenwich meridian. Consequently, the value of τ is defined in the $(-180, 180]$ deg interval. The latitude δ is measured from the equator, where positive values correspond to northern-hemisphere latitudes and negative values correspond to the southern-hemisphere latitudes. Thus, the value of δ is defined in the $(-90, 90]$ deg interval. In addition, R is the radial distance from the center of Earth to the center of mass of the vehicle (Mooij, 2014).

Furthermore, γ is the angle between $\boldsymbol{V_g}$ and the local horizontal plane, where positive values indicate that $\boldsymbol{V_g}$ points in a direction above the plane and negative values below the plane. Thus, the value of γ is defined in the $(-90, 90]$ deg interval. Finally, χ is the angle between the projection of $\boldsymbol{V_g}$ on the local horizontal plane and the local north. Thus, the value of χ is defined in the $(-180, 180]$ deg interval, where 90 deg indicates that the vehicle moves due east, parallel to the equator (Mooij, 2014).

As will be discussed in Section 3.4.1, a reference ellipsoid provides a better approximation to the physical topography of Earth (Tapley et al., 2004b). Consequently, an alternate set of position coordinates is derived, which uses the geodetic latitude δ^*, the longitude τ and the height above the reference ellipsoid h^*. The relation between the spherical position coordinates and the geodetic position coordinates is given in Section 3.4.1.

A combination of geodetic position coordinates and spherical velocity coordinates are used to express the vehicle's state within all the guidance blocks and throughout all the plots shown in this thesis. The reason behind this is that such set of coordinates is easier to interpret than Cartesian state variables.

3.3.3. STATE VARIABLE CONVERSION

The goal this section is to introduce the required steps to convert state variables across the different representations mentioned in above. Please note that similarly to unit-rotation matrices, the conversions here presented can be combined to achieved any desired conversion. As discussed in the previous section, the widely

Cartesian (Rotating) to Cartesian (Inertial) The conversion of the Cartesian position in $\mathscr{F}_R$ to $\mathscr{F}_I$ is accomplished by means of the following transformation:

$$\boldsymbol{r_I} = \begin{pmatrix} x_I \\ y_I \\ z_I \end{pmatrix} = \boldsymbol{C_{I,R}} \boldsymbol{r_R} = \begin{bmatrix} \cos\omega_E t & -\sin\omega_E t & 0 \\ \sin\omega_E t & \cos\omega_E t & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_R \\ y_R \\ z_R \end{pmatrix} \tag{3.21}$$

To convert the groundspeed in $\mathscr{F}_R$ to the inertial velocity in $\mathscr{F}_I$, one must take into account the rotational

velocity of $\mathscr{F}_R$. This is referred as the transport theorem which relates the magnitudes of the inertial velocity and groundspeed vectors (Mooij, 2014). Such theorem is shown in the following relation:

$$\boldsymbol{V_i^R} = \begin{pmatrix} u_{i,R} \\ v_{i,R} \\ w_{i,R} \end{pmatrix} = \boldsymbol{V_g^R} + \boldsymbol{\Omega_{R,I}^R} \times \boldsymbol{r_R^R} = \begin{pmatrix} u_g \\ v_g \\ w_g \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \omega_E \end{pmatrix} \times \begin{pmatrix} x_R \\ y_R \\ z_R \end{pmatrix} \tag{3.22}$$

The previous equation yields the inertial velocity expressed in $\mathscr{F}_R$. Thus, the following transformation matrix is required to obtain the inertial velocity expressed in $\mathscr{F}_I$:

$$\boldsymbol{V_i^I} = \begin{pmatrix} u_i \\ v_i \\ w_i \end{pmatrix} = \boldsymbol{C_{I,R}} \boldsymbol{V_i^R} = \begin{bmatrix} \cos\omega_E t & -\sin\omega_E t & 0 \\ \sin\omega_E t & \cos\omega_E t & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} u_{i,R} \\ v_{i,R} \\ w_{i,R} \end{pmatrix} \tag{3.23}$$

Cartesian (Inertial) to Cartesian (Rotating) The Cartesian position in $\mathscr{F}_I$ is converted to $\mathscr{F}_R$ by means of the following transformation:

$$\boldsymbol{r_R} = \begin{pmatrix} x_R \\ y_R \\ z_R \end{pmatrix} = \boldsymbol{C_{R,I}} \boldsymbol{r_I} = \begin{bmatrix} \cos\omega_E t & \sin\omega_E t & 0 \\ -\sin\omega_E t & \cos\omega_E t & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_I \\ y_I \\ z_I \end{pmatrix} \tag{3.24}$$

To convert the inertial velocity in $\mathscr{F}_I$ to the groundspeed in $\mathscr{F}_R$, the transport theorem must be used. To do so, it is required to first compute the inertial velocity expressed in $\mathscr{F}_R$. This done by means of the following transformation:

$$\boldsymbol{V_i^R} = \begin{pmatrix} u_{i,R} \\ v_{i,R} \\ w_{i,R} \end{pmatrix} = \boldsymbol{C_{R,I}} \boldsymbol{V_i^I} = \begin{bmatrix} \cos\omega_E t & \sin\omega_E t & 0 \\ -\sin\omega_E t & \cos\omega_E t & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} u_i \\ v_i \\ w_i \end{pmatrix} \tag{3.25}$$

The transport theorem is finally applied, as shown in the next relation:

$$\boldsymbol{V_g^R} = \begin{pmatrix} u_g \\ v_g \\ w_g \end{pmatrix} = \boldsymbol{V_i^R} - \boldsymbol{\Omega_{R,I}^R} \times \boldsymbol{r_R} = \begin{pmatrix} u_{i,R} \\ v_{i,R} \\ w_{i,R} \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ \omega_E \end{pmatrix} \times \begin{pmatrix} x_R \\ y_R \\ z_R \end{pmatrix} \tag{3.26}$$

Cartesian (Rotating) to Spherical The radial distance can be interpreted as the magnitude of the Cartesian position expressed in $\mathscr{F}_R$, as shown in the following equation:

$$R = \sqrt{x_R^2 + y_R^2 + z_R^2} \tag{3.27}$$

In addition, the latitude and longitude are computed using projections of the $\boldsymbol{r_R}$ vector onto $\mathscr{F}_R$. The trigonometric relations of such projections are given in Equations (3.28) and (3.29). Please note that relations can be formed using cosine and tangent functions.

$$\sin\tau = \frac{y_R}{\sqrt{x_R^2 + y_R^2}} \tag{3.28}$$

$$\sin\delta = \frac{z_R}{R} \tag{3.29}$$

The groundspeed is computed using the components of the Cartesian velocity expressed in $\mathscr{F}_R$, as shown in the following equation:

$$V_g = \sqrt{u_g^2 + v_g^2 + w_g^2} \tag{3.30}$$

The flight-path angle and the heading angle are computed using projections of the $\boldsymbol{V_g}$ vector onto $\mathscr{F}_V$. The trigonometric relations of such projections are given in Equations (3.31) and (3.32). In this case, alternative

cosine and tangent relations can be found as well.

$$\sin\gamma = \frac{w_{g,V}}{V_g} \tag{3.31}$$

$$\sin\chi = \frac{v_{g,V}}{\sqrt{u_{g,V}^2 + v_{g,V}^2}} \tag{3.32}$$

Spherical to Cartesian (Rotating) As was seen in Section 3.2, τ and δ are used to transform $\mathscr{F}_V$ onto $\mathscr{F}_R$. In essence, spherical state variables can be converted to $\mathscr{F}_R$ Cartesian state variables by means of a transformation matrix, as shown in the following relation:

$$\boldsymbol{r_R} = \begin{pmatrix} x_R \\ y_R \\ z_R \end{pmatrix} = \boldsymbol{C_{R,V}}\boldsymbol{r_V} = \begin{bmatrix} -\cos\tau\sin\delta & -\sin\tau & -\cos\tau\cos\delta \\ -\sin\tau\sin\delta & \cos\tau & -\sin\tau\cos\delta \\ \cos\delta & 0 & -\sin\delta \end{bmatrix} \begin{pmatrix} 0 \\ 0 \\ -R \end{pmatrix} = \begin{pmatrix} R\cos\tau\cos\delta \\ R\sin\tau\cos\delta \\ R\sin\delta \end{pmatrix} \tag{3.33}$$

To obtain the groundspeed expressed in $\mathscr{F}_R$, one first needs to use γ_g and χ_g to compute the groundspeed expressed in $\mathscr{F}_V$, as shown in the following relation:

$$\boldsymbol{V^V} = \begin{pmatrix} u_{g,V} \\ v_{g,V} \\ w_{g,V} \end{pmatrix} = \begin{bmatrix} V_g\cos\gamma\cos\chi \\ V_g\cos\gamma\sin\chi \\ V_g\sin\gamma \end{bmatrix} \tag{3.34}$$

Next, the groundspeed expressed in $\mathscr{F}_R$ can be finally obtained by means of the following transformation:

$$\boldsymbol{V_g^R} = \begin{pmatrix} u_g \\ v_g \\ w_g \end{pmatrix} = \boldsymbol{C_{R,V}}\boldsymbol{V_g^V} = \begin{bmatrix} -\cos\tau\sin\delta & -\sin\tau & -\cos\tau\cos\delta \\ -\sin\tau\sin\delta & \cos\tau & -\sin\tau\cos\delta \\ \cos\delta & 0 & -\sin\delta \end{bmatrix} \begin{pmatrix} u_{g,V} \\ v_{g,V} \\ w_{g,V} \end{pmatrix} \tag{3.35}$$

The coordinate transformations discussed above are primarily used to interface the state propagator with the guidance and trajectory plotting blocks of the software architecture, as shown in Figures 3.6 and 3.7. The reader is referred to Chapter 6 for a thorough description of such software blocks.

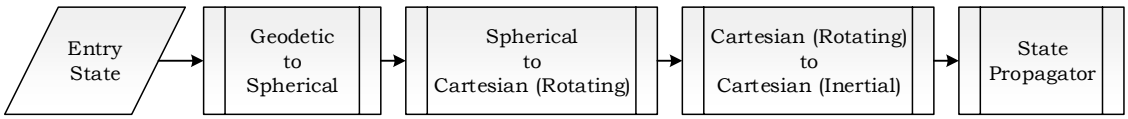

Figure 3.6: Transformation of the entry vehicle state in geodetic coordinates to Cartesian inertial state variables as required by the state propagator.

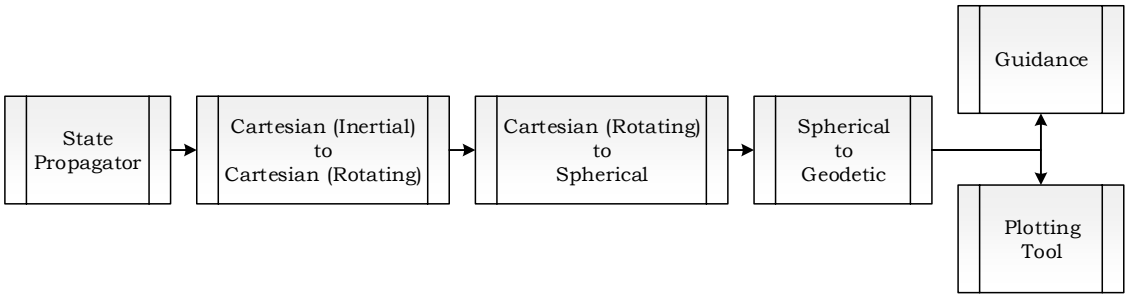

Figure 3.7: Transformation of the vehicle state in Cartesian inertial state variables as computed by the state propagator to geodetic coordinates.

3.4 ENVIRONMENT MODELS

Environment models provide location-based information concerning the planet shape and the atmospheric properties. The availability of such information is essential in the derivation of the equations of motion.

3.4.1. EARTH SHAPE MODEL

Precise knowledge of the Earth's shape model is crucial when determining accurate position information. Furthermore, any position error propagates to other location-dependent models, potentially leading to an unacceptable total error. In the field of planet shape modeling, two top-level options are available: a *perfect-sphere* model and an *ellipsoidal* model. As the name indicates, the perfect-sphere model assumes that the Earth has no flattening at the poles. Position information in the perfect-sphere model is denoted using geocentric spherical coordinates. Alternatively, the ellipsoidal models do account for the oblateness of Earth. In this case, position information is denoted using geodetic spherical coordinates.

The choice between either model boils down to a trade-off of accuracy against computational effort: according to Montenbruck and Gill (2001a), the error made when using the perfect-sphere model can become up to 12 arcminutes for intermediate latitudes, where such latitudes are defined as those in the neighborhood of ±45 deg. Figure 3.10 shows that such error translates to approximately 20 km on the ground, which can lead to missing the landing site entirely. According to **REQ-S-08**, several of the landing sites are indeed at those intermediate latitudes, thus favoring the use of an ellipsoidal shape model. One of the most common ellipsoidal models is the one present in the World Geodetic System 1984 (WGS84), which establishes an ellipsoid's flattening parameter f of $3.32761797 \cdot 10^{-3}$ and an equatorial radius R_E of 6,378,136 m (NIMA, 2000). Figure 3.9 shows the altitude error as a function of geocentric latitude when assuming a spherical shape mode instead of the WGS84 model. Such figures shows that the spherical shape model underestimates the altitude at the equator with an error of approximately -7 km. Due to the flattening of the reference ellipsoid at the poles, the spherical shape model overestimates the altitude at high latitudes. Figure 3.9 shows that such overestimation can amount up to 14 km. Figure 3.10 shows the projection of the latitude error on the ellipsoid surfaces as a consequence of using the spherical shape model instead of WGS84. As discussed above, the error is larger in the latitudes in the neighborhood of ±45 deg.

Despite the use of an ellipsoidal shape model, geocentric spherical velocity coordinates are used throughout this thesis for representation purposes, as well as for multiple frame transformation. The transformation of geocentric spherical coordinates to geodetic spherical coordinates often requires an iterative process (Wakker, 2015), which may prove computationally expensive. To increase the computational speed, the approximate direct method of Montenbruck and Gill (2001a) is used. In such method, the geodetic latitude is computed by means of the geocentric latitude and the flattening parameter, as shown by the following equation:

$$\delta^* = \delta + f \sin(2\delta) \tag{3.36}$$

where δ^* is the geodetic latitude, δ is the geocentric latitude and f is the flattening parameter. Note that due to the symmetry of the model, the geocentric longitude is equivalent to the geodetic longitude thus requiring no additional calculation steps. To complete the set of geodetic coordinates, it is necessary to compute the height above the reference ellipsoid. To do so, the auxiliary parameters given by the next four equations need to be determined first:

$$e = \sqrt{1 - (1-f)^2} \tag{3.37}$$

$$N = \frac{R_E}{\sqrt{1 - e^2 \sin^2 \delta^*}} \tag{3.38}$$

$$z = R_E \frac{(1-f)^2 \sin \delta^*}{\sqrt{1 - f(2-f) \sin^2 \delta^*}} \tag{3.39}$$

$$\Delta z = N e^2 \sin \delta^* \tag{3.40}$$

where e is the eccentricity of the reference ellipsoid and N, z and Δz are auxiliary parameters are the auxiliary parameters depicted in Figure 3.8. Finally, the height above the reference ellipsoid h^* is computed using the following equation:

$$h^* = \sqrt{R + \Delta z^2 + 2 z \Delta z} - N \tag{3.41}$$

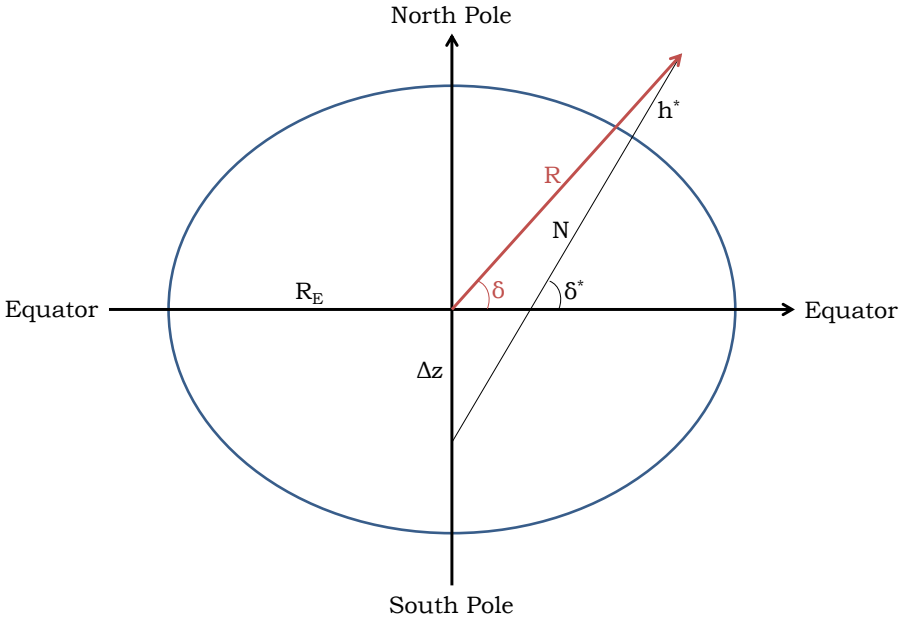

Figure 3.8: Geometry of the Earth ellipsoid. Based on the work from (Montenbruck and Gill, 2001b).

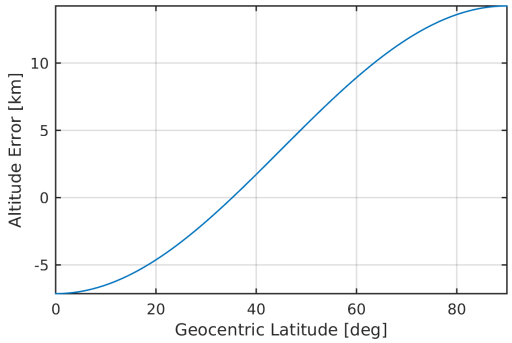

Figure 3.9: Altitude error with respect to the WGS84 ellipsoid.

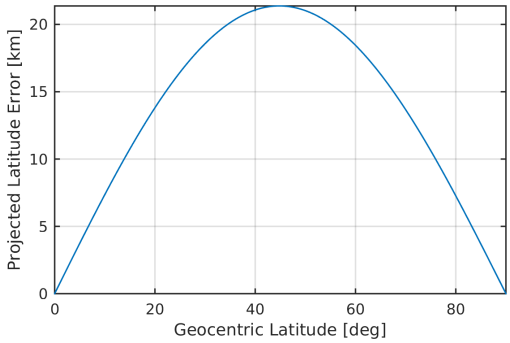

Figure 3.10: Projection on the ellipsoid surface of the latitude error with respect to the WGS84 ellipsoid.

3.4.2. GRAVITY FIELD MODEL

The complexity of the Earth's gravity field model can vary from simple models in which the field strength is approximated to be constant, to complex models in which the field strength is position dependent. In this study, a gravity field model with a spatial dependency is considered. Such dependency originates from the inhomogeneous distribution of the Earth's mass, where the main perturbing force results from the accumulation of mass over the planet's equator (Tapley et al., 2004c). The following four equations express in $\mathcal{F}_R$ the gravitational acceleration that a body experiences due such inhomogeneous distribution of mass (Mooij, 1997):

$$g_x = -\mu_E \frac{x_R}{R^3} Q(z_R, R) \tag{3.42}$$

$$g_y = -\mu_E \frac{y_R}{R^3} Q(z_R, R) \tag{3.43}$$

$$g_z = -\mu_E \frac{z_R}{R^3} \left(1 + \frac{3J_2}{2} \left(\frac{R_E}{R}\right)^2 \left(3 - 5\frac{z_R^2}{R^2}\right)\right) \tag{3.44}$$

$$Q(z_R, R) = 1 + \frac{3J_2}{2} \left(\frac{R_E}{R}\right)^2 \left(1 - 5\frac{z_R^2}{R^2}\right) \tag{3.45}$$

where g_x, g_y and g_z are the gravitational acceleration values acting parallel to the axis of $\mathcal{F}_R$, μ_E is the Earth's gravitational parameter, R is the radial distance and J_2 is the constant that quantifies the effect of the oblateness of the mass distribution on the gravity field. The J_2 is determined to be 1.082627 $\times 10^{-3}$ and the most

prominent impact of such term is a secular variation of the longitude of the ascending node Ω (Wakker, 2015). According to Wakker (2015), the mean rate of change of such secular variation in Ω is estimated with the following equation:

$$\dot{\Omega}_{mean} = -\frac{3}{2} J_2 R^2 \sqrt{\frac{\mu_E}{R_0^7}} \cos i_0 \tag{3.46}$$

where $\dot{\Omega}_{mean}$ is the mean rate of change of the longitude of the ascending node and R_0 and i_0 are the radius and inclination of the unperturbed circular orbit, respectively. According to the previous equation, a vehicle flying at 120 km altitude in a 52 deg inclined circular orbit would experience a mean node progression rate of approximately -0.26 deg/hour. As will be discussed in Section 7.2, the reentry trajectories explored in this thesis have flight-times that range from 20 min to 90 min, approximately. These flight-times are sufficiently long for the J_2 term to have a significant effect, thus justifying its consideration in the gravitational acceleration expression.

3.4.3. ATMOSPHERE MODEL

Atmosphere models provide location-based air density and air temperature values. These values are critical to the correct assessment of the aerodynamic forces that the vehicle experiences during reentry. Atmosphere models are classified under two groups:

- **Standard Models:** The air properties are defined only by the geodetic altitude. Examples of these models are the isothermal-barotropic atmosphere and the US Standard Atmosphere 1976 (US76).
- **Reference Models:** The air properties are defined by the complete set of geodetic spherical coordinates. Furthermore, these models may include temporal and seasonal variations. A good example of a reference model is the US Global Reference Atmospheric Model (GRAM) series.

Although reference atmosphere models provide more accurate air property values, their software implementation may prove difficult (Justus and Johnson, 1999). Thus, only standard atmosphere models are considered in this report. As highlighted above, two main options are available under the group of standard atmospheres.

Exponential Atmosphere This atmosphere model assumes that the temperature remains constant throughout the atmosphere and that the density depends purely on pressure (Mooij, 2014). The exponential atmosphere is also referred as the isothermal-barotropic atmosphere. The constant temperature assumption translates to a constant speed of sound assumption, which is highly undesired in the study of reentry trajectories.

US Standard Atmosphere 1976 The US76 atmosphere model uses various data tables to define air density and temperature (NASA and USAF and NOAA, 1976). Furthermore, the US76 takes into account variations in the gravitational acceleration and molecular mass of the air (NASA and USAF and NOAA, 1976). The accuracy of the US76 atmosphere model with respect to the other models is significantly better at higher altitudes.

Given the framework and goals of this thesis, model accuracy is highly valued over model simplicity. Consequently, the US76 Standard Atmosphere is selected as the preferred atmosphere model. Such model divides the atmosphere into different layers according to their geopotential altitude z, where each layer has a particular temperature profile. The atmosphere layers are grouped into the lower layers and the upper layers. The reason behind this grouping is that computation of the temperature profiles of the upper layers is different than for the lower layers.

The lower layers extend up to a geodetic altitude of 80 km and they are defined using linear temperature profiles. The temperature within each of the lower layer varies according to a certain thermal lapse rate, that remains constant throughout the layer. Note that the parameters required to compute the air properties in the lower layers are found in the following table:

For the sake of consistency of this report, Table 3.1 indexes the start each of the lower layers according to the geodetic altitude h. The conversion between h^* and z is given in the following equation:

$$z \approx \frac{R_{ref} h^*}{R_{ref} + h^*} \tag{3.47}$$

Table 3.1: Parameters for the lower layers of the US76 atmosphere model (NASA and USAF and NOAA, 1976)

Layer	Geodetic Altitude [km]	Base Temperature [K]	Lapse Rate [K/km]	Base Pressure [Pa]
0	0	288.15	-6.5	1.013E+05
1	11	216.65	0.0	2.263E+04
2	20	216.65	1.0	5.475E+03
3	32	228.65	2.8	8.680E+02
4	47	270.65	0.0	1.109E+02
5	51	270.65	-2.8	6.694E+01
6	72	214.65	-2.0	3.956E+00
7	86	186.946	N/A	3.734E-01

Furthermore, the US76 model introduces the molecular scale temperature T_M to define the temperature of each layer. This is done to simplify the altitude dependency of the air temperature and air composition. The relation between T_M, the temperature T, the air molecular mass at sea level M_0 and the air molecular mass M is shown in the following relation:

$$T = T_M \frac{M}{M_0} \tag{3.48}$$

As mentioned above, the variation of T_M with altitude for the lower layers is quantified by the thermal lapse rate L_{z_i}. Such variation is computed using the linear relation shown in the following equation:

$$T_M = T_{M_i} + L_{z_i}(z - z_i) \tag{3.49}$$

The above relations are combined with the hydrostatic equation to find the pressure throughout the lower layers, as shown by the next two equations. The first equation is used to find the pressure in the isothermal layers and the second equation to find the pressure in those layers with a non-zero thermal lapse rate:

$$p = p_i \exp\left(\frac{-g_0 M_0 (z - z_i)}{R^* T_{M_i}}\right) \tag{3.50}$$

$$p = p_i \left(\frac{T_{M_i}}{T_{M_i} + L_{z_i}(z - z_i)}\right)^{\frac{g_0 M_0}{R^* L_{z_i}}} \tag{3.51}$$

Once the temperature and pressure profiles have been devised, the density is computed using the ideal gas law presented in the following relation:

$$p = \frac{\rho R^* T}{M} \tag{3.52}$$

The temperature profiles for the upper atmosphere layers are shown in Equations (3.53) to (3.55). To begin with, the US76 model corrects the M values of the 7^{th} layer from an altitude of 80 km onwards. Such correction is achieved using the tabulated data given in Table 8 from (NASA and USAF and NOAA, 1976). The 8^{th} layer is consists of an isothermal region that reaches an altitude of 91 km. The temperature profile of the 9^{th} layer is modelled with an ellipse. The 10^{th} layer begins at an altitude of 110 km and is modelled with a linear gradient. Finally, the 11^{th} layer begins at 120 km and is modelled with an exponential function.

$$T = 186.8673\ K \tag{3.53}$$

$$T = T_c + A\sqrt{1 - \left(\frac{h - h_8}{b}\right)^2} \tag{3.54}$$

$$T = T_9 + L_9 (h - h_9) \tag{3.55}$$

$$T = T_{ref} - (T_{ref} - T_{10}) \exp\left[-\lambda \frac{(h - h_{10})(R_{ref} + h_{10})}{R_{ref} + h_{10}}\right] \tag{3.56}$$

The computation of the upper layer temperature profiles is achieved using the parameters shown in Table B.1. Once the temperature profiles have been determined, the pressure is computed using Equation (3.57). Here, the total number density is computed by summing the number densities of the species indicated in Table VIII

in (NASA and USAF and NOAA, 1976).

$$p = \frac{NR^* T}{N_A} \tag{3.57}$$

The density is finally computed using the ideal gas law. Please note that the molecular mass needs to be corrected as well when computing the density of the upper layers. Figures 3.11 and 3.12 show the temperature and pressure profiles as defined by the US76 standard atmosphere. For sake of clarity, a summary of the different layers is given in Figure B.1.

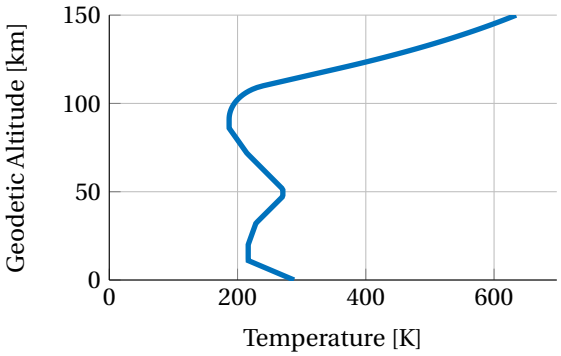

Figure 3.11: Temperature profile of the US76 standard atmosphere.

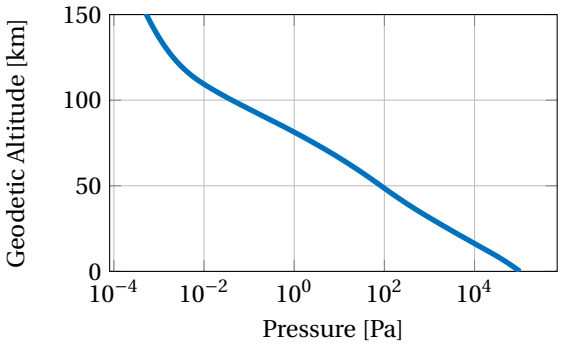

Figure 3.12: Pressure profile of the US76 standard atmosphere.

As a final note, the speed of sound throughout each layer can be computed using the following equation:

$$a = \sqrt{\frac{\gamma R^* T}{M}} \tag{3.58}$$

where γ is the ratio of specific heats, which takes the typical value of 1.4 for air (NASA and USAF and NOAA, 1976). Knowledge of the speed of sound at a particular altitude is key to determining the Mach number, which is required for the computation of vehicle aerodynamic coefficients. The Mach number is given by the following equation:

$$M = \frac{V_a}{a} \tag{3.59}$$

3.5 EXTERNAL FORCES

The external forces determine the accelerations that the vehicle experiences during flight. In this study, two main types of external forces are studied: aerodynamic forces and gravitational forces. As indicated in Sections 3.5.1 and 3.5.2, both types of external forces rely heavily on the environment models described in Section 3.4.

3.5.1. AERODYNAMIC FORCES

The external aerodynamic force that the vehicle experiences depends on a set of aerodynamic coefficients C_D, C_S, C_L and a reference area S_{ref}. These parameters are vehicle dependent and must be specified in the reference vehicle's aerodynamic database. The external aerodynamic forces are easily described in $\mathcal{F}_{AA}$, as shown in the following equation:

$$\boldsymbol{F}_{\boldsymbol{A}}^{\boldsymbol{AA}} = \begin{pmatrix} -D \\ -S \\ -L \end{pmatrix} = \begin{pmatrix} -C_D \bar{q} S_{ref} \\ 0 \\ -C_L \bar{q} S_{ref} \end{pmatrix} \tag{3.60}$$

Equation 3.60 introduces the dynamic pressure $\bar{q}$. Such pressure depends on the vehicle's groundspeed V_g and the air density, as shown in Equation 3.61. Such relation shows the dependency of the aerodynamic forces on the selected atmosphere model. Note that dynamic pressure actually depends on airspeed but since wind is not considered in this thesis, groundspeed is used instead.

$$\bar{q} = \frac{1}{2} \rho {V_g}^2 \tag{3.61}$$

The external aerodynamic force is also dependent on the vehicle attitude. The reason behind this is that the aerodynamic coefficients are a function of the angle of attack, the angle of side-slip and the Mach number. Such dependency is shown in Equation 3.62 and it must be fully specified by the reference vehicle's aerodynamic database.

$$C_D = C_D\left(\alpha_a, M, \delta_{bf,r}, \delta_{bf,l}\right) \qquad C_S = 0 \qquad C_L = C_L\left(\alpha_a, M, \delta_{bf,r}, \delta_{bf,l}\right) \tag{3.62}$$

where $\delta_{bf,r}$ and $\delta_{bf,l}$ are the SPHYNX's right and left body flap trim deflections, respectively. As stated in Section 2.4.1, the side-force coefficient is assumed to be zero at all times due to the fact that wind is not considered. Finally, in this thesis the body flaps of the SPHYNX vehicle are only used for trimming. Thus, the aerodynamic coefficients are functions of the combined body flap deflection, given in the next equation:

$$\delta_{bf} = \frac{\delta_{bf,r} + \delta_{bf,l}}{2} \tag{3.63}$$

3.5.2. GRAVITATIONAL FORCES

The gravitational force that the vehicle experiences is a direct consequence of the selected gravitational model. Such relation is given in Equation (3.64), which yields the gravitational force expressed in $\mathscr{F}_R$. The gravitational acceleration $\boldsymbol{g}^{\boldsymbol{R}}$ is found using Equations (3.42) to (3.45).

$$\boldsymbol{F}_{\boldsymbol{g}}^{\boldsymbol{R}} = m\boldsymbol{g}^{\boldsymbol{R}} = \begin{pmatrix} mg_x \\ mg_y \\ mg_z \end{pmatrix} \tag{3.64}$$

3.6 EQUATIONS OF MOTION

The goal of this section is to present a set of equations that describe the vehicle's motion throughout reentry. The equations of motion are numerically integrated in an entry simulator, which is used to evaluate the performance of the abort guidance algorithm. Although the state variable selection does not affect the shape of the simulated trajectory, it has a significant impact on the simulation run-time.

In this thesis, the equations of motion are written in terms of Cartesian-inertial state variables since such set of equations is free of discontinuities and results in fast integration times due to the absence of trigonometric terms. The kinematic and dynamic equations of motion with Cartesian-inertial state variables are given by the following two relations, respectively:

$$\dot{\boldsymbol{r}}_{\boldsymbol{I}} = \boldsymbol{V}_{\boldsymbol{i}}^{\boldsymbol{I}} \tag{3.65}$$

$$\dot{\boldsymbol{V}}_{\boldsymbol{i}}^{\boldsymbol{I}} = \boldsymbol{g}^{\boldsymbol{I}} + \frac{1}{m}\boldsymbol{F}_{\boldsymbol{A}}^{\boldsymbol{I}} \tag{3.66}$$

Equation 3.66 relates the external forces acting on the vehicle to variations in the vehicle's velocity. Equation 3.65 relates the vehicle's velocity to variations in the vehicle's position. By solving both equations simultaneously, one can fully describe the vehicle's motion. The external forces present in Equation 3.66 consist of the sum of the gravitational forces and the aerodynamic forces. Since the gravitational force is pre-multiplied by $1/m$, the resulting term is simply the gravitational acceleration. Furthermore, the terms on the right hand side of the dynamic equation need to be transformed to the inertial frame, as shown by the following two relations:

$$\boldsymbol{g}^{\boldsymbol{I}} = \boldsymbol{g}^{\boldsymbol{R}} \tag{3.67}$$

$$\boldsymbol{F}_{\boldsymbol{A}}^{\boldsymbol{I}} = \boldsymbol{C}_{\boldsymbol{I},\boldsymbol{AA}}\boldsymbol{F}_{\boldsymbol{A}}^{\boldsymbol{AA}} = \boldsymbol{C}_{\boldsymbol{I},\boldsymbol{R}}\boldsymbol{C}_{\boldsymbol{R},\boldsymbol{V}}\boldsymbol{C}_{\boldsymbol{V},\boldsymbol{B}}\boldsymbol{C}_{\boldsymbol{B},\boldsymbol{AA}}\boldsymbol{F}_{\boldsymbol{A}}^{\boldsymbol{AA}} \tag{3.68}$$

Please note that the gravitational force term does not require a frame transformation due to the symmetric nature of the J_2 term. As a final note, the computation of the aerodynamic forces present in Equation 3.68 requires knowledge of the groundspeed. Thus, one needs to apply the transport theorem at every time-step to obtain the groundspeed in $\mathscr{F}_I$, as shown in the following equation:

$$\boldsymbol{V}_{\boldsymbol{g}}^{\boldsymbol{I}} = \begin{pmatrix} u_{g,I} \\ v_{g,I} \\ w_{g,I} \end{pmatrix} = \boldsymbol{V}_{\boldsymbol{i}}^{\boldsymbol{I}} - \boldsymbol{\Omega}_{\boldsymbol{R},\boldsymbol{I}}^{\boldsymbol{I}} \times \boldsymbol{r}_{\boldsymbol{I}} = \begin{pmatrix} u_i \\ v_i \\ w_i \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ \omega_E \end{pmatrix} \times \begin{pmatrix} x_I \\ y_I \\ z_I \end{pmatrix} \tag{3.69}$$

A major disadvantage of the equations of motion in Cartesian-inertial state variables is that they are difficult to interpret. To tackle such issue, the output state produced by the numerical integration is converted to spherical state variables whenever a simple interpretation of the vehicle state is required. The main software modules that benefit from usage of spherical state variables are the guidance modules and the plotting modules.

3.7 TRAJECTORY CONSTRAINTS

The goal of this section is to present a number of functions that constrain the trajectories to be flown by SPHYNX. These functions are known as trajectory constraints and as mentioned in Chapter 1, they are essential to ensure the safety of the vehicle and its payload during descent.

Heat Flux The heat flux constraint defines the maximum heating rate per unit area that the vehicle can sustain as a consequence of aerothermodynamic heating. This constraint originates from the maximum temperature that the wall of the vehicle's TPS can handle to ensure thermodynamic equilibrium (Mooij, 2014). According to **REQ-S-03**, the SPHYNX vehicle shall withstand a maximum heating rate of 670 kW/m^2. A first approximation to this heat flux can found using the method explained below. Such method is a so called hot-wall approximation, which computes a wall temperature T_w by equating the incoming convective flux with the outgoing radiative flux q_{rad} for an assumed adiabatic wall temperature T_{aw} (Mooij, 2014).

$$q_c = \left(5.28137 \cdot 10^{-5}\right) \sqrt{\frac{\rho}{R_N}} V_g^{3.15} \left[1 - \frac{T_w}{T_{aw}}\right] \tag{3.70}$$

$$q_{rad} = \epsilon \sigma T_w^4 \tag{3.71}$$

Here, q_c is the convective heat flux at the vehicle's stagnation point, R_N is vehicle's nose radius, ϵ is the emissivity of the applied material and σ is the Stefan-Bolzmann constant. Note that Equation (3.70) scales with atmospheric density and groundspeed, suggesting that the heat flux constraint will prevent the vehicle from flying in the lower atmosphere layers at high speeds. The above equations can be combined together into the relation given below, which is fed the selected root-finding method from Section 5.1. The root-finding method is used to compute the wall temperature, which is then used to obtain the convective heat flux using Equation (3.70).

$$\left(5.28137 \cdot 10^{-5}\right) \sqrt{\frac{\rho}{R_N}} V_g^{3.15} \left[1 - \frac{T_w}{T_{aw}}\right] - \epsilon \sigma T_w^4 = 0 \tag{3.72}$$

Heat Load The heat load constraint defines the maximum amount of heat energy per unit area that the vehicle has accumulated over its entire flight. Such constraint originates and drives the design of the TPS, where lower heat load values result in lighter designs (Mooij, 2014). According to **REQ-S-04**, the SPHYNX vehicle shall withstand a maximum heat load of 460 MJ/m^2. The accumulated heat load is computed by integrating the incoming convective heat flux, as shown in the following equation:

$$Q = \int_{t_0}^{t_f} q_c \, dt \tag{3.73}$$

where Q is the accumulated heat load, t_0 is the initial time and t_f is the total flight time. Note that by limiting the heat flux received by the vehicle, the heat load is also decreased.

Aerodynamic Load The aerodynamic load constraint defines the maximum aerodynamic acceleration experienced during descent. Such constraint is generally driven by the mechanical load that the vehicle's structure can handle. The aerodynamic load value is often normalized to vehicle's weight at sea level, as shown in the following equation:

$$n_g = \frac{\rho V_g^2 S_{ref}}{2 m g_0} \sqrt{C_D^2 + C_L^2} \tag{3.74}$$

Here n_g is the aerodynamic load and S_{ref} is the vehicle reference area and g_0 is the Earth's gravitational acceleration at sea level. According to **REQ-S-01**, the SPHYNX vehicle shall withstand a maximum aerodynamics load of 3 g.

Dynamic Pressure The dynamic pressure constraint limits the maximum force per unit area that the vehicle must endure as a consequence of the incoming high-speed airflow. Such constraint arises from the need of limiting the mechanical stresses on the vehicle's structure. According to **REQ-S-02**, the SPHYNX vehicle shall withstand a maximum dynamic pressure of 6000 Pa. The dynamic pressure was already introduced in Equation 3.61, but is repeated here for consistency:

$$\bar{q} = \frac{1}{2}\rho {V_g}^2 \tag{3.75}$$

Please note that the dynamic pressure and the heat flux constraints can be conveniently combined into a single constraint, as shown in the following equation:

$$\frac{n_g}{\bar{q}} = \frac{S_{ref}}{m g_0}\sqrt{C_D^2 + C_L^2} \tag{3.76}$$

4 ADAPTIVE ABORT GUIDANCE THEORY

The role of a guidance system is to regulate the vehicle's position and velocity during descent. Such system ensures that the vehicle does not violate any trajectory constraints and that the landing site is reached with an appropriate energy state. This is achieved by scheduling angle-of-attack and bank-angle commands throughout the descent, which modulate the magnitude of the aerodynamic forces and tilt the lift vector in the local vertical plane, respectively.

The architecture of a guidance system can be classified under two different blocks: the *trajectory planner* and the *trajectory tracker*. On the one hand, the trajectory planner is in charge of determining the magnitude of the attitude commands and schedule these as a function of a particular independent variable. Such planning takes into account the vehicle state at the EIP, the trajectory constraints and the desired vehicle state at the terminal point. On the other hand, the trajectory tracker ensures that the reference attitude profiles produced by the planner are carefully followed during the flight, despite any disturbances or errors on the vehicle state at the EIP.

This chapter begins with Section 4.1 discusses the implementation of the trajectory planner researched in this thesis, which is based around a novel interpolation algorithm. Next, the details behind the trajectory tracker of the adaptive abort guidance system are covered in Section 4.2. Finally, Section 4.3 explores the implementation of the lateral guidance scheme employed in this thesis.

4.1 ON-BOARD TRAJECTORY INTERPOLATION

The goal of this section is to discuss the theory behind the generation of reference trajectories on-board via the method of Adaptive Multivariate Pseudospectral Interpolation (AMPI) presented by Sagliano et al. (2016). Before introducing the details behind this algorithm, the reader is advised to examine the following figure, where a top level view of the AMPI method is shown. In such figure, the algorithm receives two inputs: a set of "physics-following" trajectories and the on-board vehicle state the EIP. The set of "physics-following" trajectories refers to those trajectories that have been produced by an off-board trajectory optimization software. They are referred as "physics-following" because they have been computed by solving a set of equations of motion, and thus represent trajectories that a reentry vehicle would encounter in a potential flight scenario. As the name suggests, the on-board vehicle state at the EIP makes reference to the set of vehicle state variables that would be experienced at the EIP in an actual flight, which would be provided by the vehicle's navigation subsystem. The role of AMPI is to produce a "mathematical" reference trajectory via interpolation, where the "physics-following" trajectories are used as the supporting nodes of such interpolation and the on-board vehicle state at the EIP provides the interpolation evaluation point. At this stage, it is important to emphasize that the output of the AMPI algorithm does not represent a formal solution of the equations of motion, but does provide a good real-time approximation (Sagliano et al., 2016). As will be discussed in the remainder of this section, there is a number of AMPI settings and design choices that impact the quality of such approximations, as well as the computational burden of the algorithm.

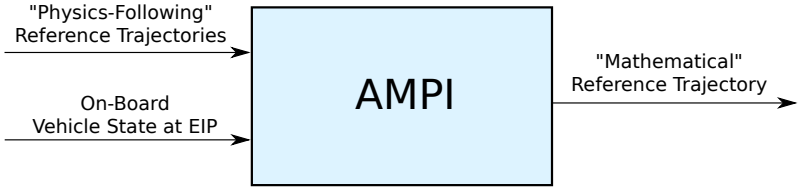

Figure 4.1: Sketch the concept behind on-board trajectory generation via AMPI.

The AMPI algorithm is a five stage process, where the first two stages are performed off-board and the remaining three are executed on-board, as shown in Figure 4.2. The algorithm begins with the discretization of all

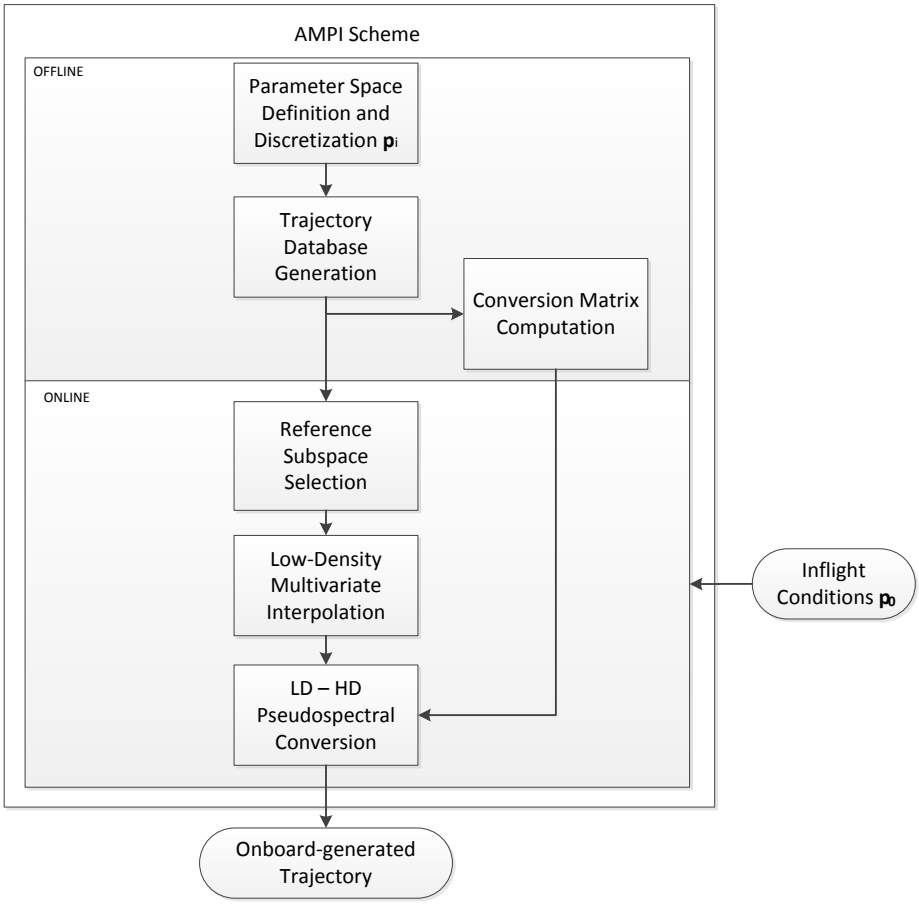

Figure 4.2: Architecture of the AMPI block. Source: (Sagliano et al., 2016)

the parameters that are expected to be off-nominal and that can be provided during the flight by the vehicle's navigation system. As mentioned above, in the framework of reentry guidance, these parameters are essentially the vehicle state variables at the EIP, which will be later used as the interpolation evaluation point. Once such discretized parameter space has been established, a database of optimal trajectories is generated using all the possible entry state variable combinations described by the parameter space. The generation of the database is done off-board, where proven optimization software and high computing power is available. Next, the trajectory database is sampled along the independent variable using a low-density (LD) grid, which effectively reduces its size to comply with stringent on-board storage requirements. Once the compressed database has been generated, it is stored on-board for further use. The third step in the AMPI algorithm is done on-board and it aims at determining the trajectory database sub-space to be used during interpolation based on the information given by the navigation system. This is done to ensure that the closest supporting reference trajectories to the interpolation point are selected. The fourth step is the actual interpolation step, where the interpolated reference trajectory is produced using the LD information stored in the database. Finally, the low-density interpolated reference trajectory is converted to a high-density (HD) solution, which can be later used by the trajectory tracker.

4.1.1. DEFINITION AND DISCRETIZATION OF THE PARAMETER SPACE

As mentioned in the previous section, the AMPI algorithm starts with the definition and discretization of the parameter space. This parameter space can be fully characterized according to three characteristics: the *number of parameters*, the *parameter spacing* and the *parameter range*. These three characteristics ultimately influence both the size of the trajectory database to be generated and stored on-board as well as the fidelity of the interpolated solutions produced by the AMPI algorithm. Such influence is described below:

- **Number of Parameters:** A large number of parameters will yield a trajectory database that produces high fidelity interpolated solutions. As will be seen in the next section, this is a direct consequence of having more "supporting information" available during the interpolation process. Unfortunately, the number

of parameter linearly increases the size of the trajectory database.

- **Parameter Spacing:** For any given parameter, the spacing between the reference profiles stored in the database determines how far the supporting information is to the interpolated value. Logically, the smaller this spacing becomes, the closer the supporting information available is, and thus the higher fidelity will be obtained during interpolation. However, by reducing the parameter spacing, the size of the trajectory database will grow.
- **Parameter Range:** For any given parameter, the parameter range determines the minimum and maximum values that such parameter can take. By increasing the parameter range, the trajectory database size increase but at the same time allows for further off-nominal values.

Given the discussion above, it becomes clear that the selection of the number of parameters, parameter spacing and the parameter range becomes a trade-off between the fidelity of the interpolated solutions and the size of trajectory database to be generated. For instance, Sagliano et al. parametrize the full six-variable entry state and work with a database containing 3^6 trajectories. By encapsulating the full entry state in the database, Sagliano et al. ensure that the interpolated trajectories resulting from such database are of high-fidelity, resembling those that would be obtained when solving the equations of motion. They are able to do this due to the fact that the parameter range that they use is limited, compared to the range that would be required in an adaptive abort guidance system. As will be later explained in Chapter 7, if full global coverage is desired, the resulting database size may become unfeasible if a full six-parameter space is selected. For this reason, the trajectory planner researched in this thesis will feed off a database featuring only two parameters: *geodetic latitude* and *longitude*. Consequently, the multivariate interpolation featured in (Sagliano et al., 2016) reduces to a simple bivariate interpolation. In mathematical terms, the parameter space employed in this thesis is formed by the following two parameters:

$$\boldsymbol{p^1} = \tau \subset \left[p_1^1, p_2^1\right] = [-180.0, 180.0] \text{ deg} \tag{4.1}$$

$$\boldsymbol{p^2} = \delta^* \subset \left[p_1^2, p_2^2\right] = [-52.0, 52.0] \text{ deg} \tag{4.2}$$

where $\boldsymbol{p^1}$ is the first database parameter and refers to the entry longitude and $\boldsymbol{p^2}$ is the second database parameter which refers to the entry geodetic latitude. For further clarity, it is important to highlight the notation used here, where the superscript indicates the referred database parameter and the subscript indicates the particular bound of such parameter. For example, p_1^2, refers to the lower bound of the second database parameter (geodetic latitude). In this example, the entry longitude is discretized from -180 deg to 180 deg, where the entry geodetic latitude is discretized -52 deg from to 52 deg, thus encapsulating all the EIPs originating from the operational orbit specfied in Chapter 2. Note that the two database parameters form a 2-dimensional rectangular grid, as shown in the following equation:

$$\mathscr{P} = \boldsymbol{p^1} \times \boldsymbol{p^2} \tag{4.3}$$

To illustrate the concept of the parameter space and its relation to the trajectory database, please examine the following figure. Here, a $[4 \times 7]$ grid represents the trajectory database. Every point in this grid is equivalent to an item in the trajectory database and thus contains trajectory information that complies with the equations of motion. In fact, each of these "points" is essentially an EIP from which a set of optimal attitude control profiles originate, which in turn bring the vehicle from that particular EIP to one of the listed landing sites.

4.1.2. SELECTION OF THE REFERENCE SUBSPACE

Once the parameter space has been established and the trajectory database has been generated, the parameter subspace enclosing the interpolation evaluation point $\boldsymbol{x_0}$ needs to be found. As mentioned above, $\boldsymbol{x_0}$ is constructed on-board, and it contains the vehicle's longitude and geodetic latitude at the expected EIP, as indicated in the following equation:

$$\boldsymbol{x_0} = \left[x_0^1, x_0^2\right] = \left[\tau_{flt}, \delta_{flt}^*\right] \tag{4.4}$$

$$x_0^i \subset \left[s_1^i, s_2^i\right] \tag{4.5}$$

The parameter subspace $\mathscr{P}_s$ is conformed by the vectors $\boldsymbol{s^1}$ and $\boldsymbol{s^2}$, as shown in the following three equations. Such subspace delimits the area of supporting information that is closest to the interpolation evaluation point, highlighted in Figure 4.3. The size of this area is determined by the parameter spacing defined in Section 4.1.1. It now becomes clear that decreasing the parameter spacing results in a smaller subspace area, ultimately yield-

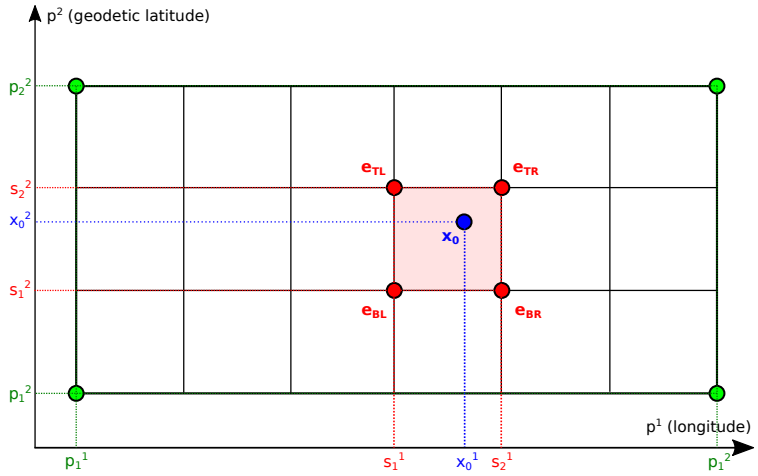

Figure 4.3: Sketch showing the methodology employed during bivariate interpolation.

ing the more accurate interpolated reference trajectories. This is a natural consequence of bringing the vertices of such area closer to the interpolation evaluation point, where the vertices contain the supporting information that complies with the equations of motion.

$$\mathscr{P}_s = \boldsymbol{s^1} \times \boldsymbol{s^2} \tag{4.6}$$

$$\boldsymbol{s^i} = \left[s_1^i, s_2^i\right] \tag{4.7}$$

$$s^i \subset \left[p_1^i, p_2^i\right] \tag{4.8}$$

For the sake of simplicity, the trajectories that delimit the parameter subspace enclosing the interpolation evaluation point are referred to as "extremals". The naming convention follows the relative grid positioning of such extremals, as shown in the following equation:

$$\boldsymbol{e_{TL}} = \left[s_1^1, s_2^2\right] \tag{4.9}$$

$$\boldsymbol{e_{TR}} = \left[s_2^1, s_2^2\right] \tag{4.10}$$

$$\boldsymbol{e_{BL}} = \left[s_1^1, s_1^2\right] \tag{4.11}$$

$$\boldsymbol{e_{BR}} = \left[s_2^1, s_1^2\right] \tag{4.12}$$

4.1.3. LOW-DENSITY NODE DETERMINATION

Once the four extremals have been selected as described in the previous section, it is required to sample the trajectory information that they contain. Each extremal contains a number of trajectory variables produced by the off-board optimization software, such as time, altitude, angle-of-attack, bank-angle, etc. These variables need to be sampled along an independent variable in such a way that they produce a low-density grid where interpolation is to be performed. This process can be visualized using the sketch shown in Figure 4.4. Here, the extremals lay on the plane of the page and the associated trajectory information, $f(t)$, is found in a direction that points into such plane. The sampling along the independent variable, chosen to be t in this example, is depicted as "slices" of this third dimension. At this stage, it is important to emphasize that choice for the independent variable is not enforced, thus another variable such as energy, can be selected instead of time.

Irrespectively of the choice for the independent variable, the four extremals will differ in trajectory length, which is obvious when considering that they represent entirely different solutions to the equation of motion. This is a key aspect to understand, since to make interpolation possible, the independent variables of each extremal need to be converted to a "pseudo" independent variable with common bounds. In the remainder of this report, such common independent variable is referred to as pseudotime τ, which is defined from -1 to 1 as shown in the following equation:

$$\boldsymbol{\tau} \subset [-1, 1] \tag{4.13}$$

The conversion between the actual time and the pseudotime is done by the following two equations. Once again, note that this conversion is valid for any other independent variable since the "pseudo" independent

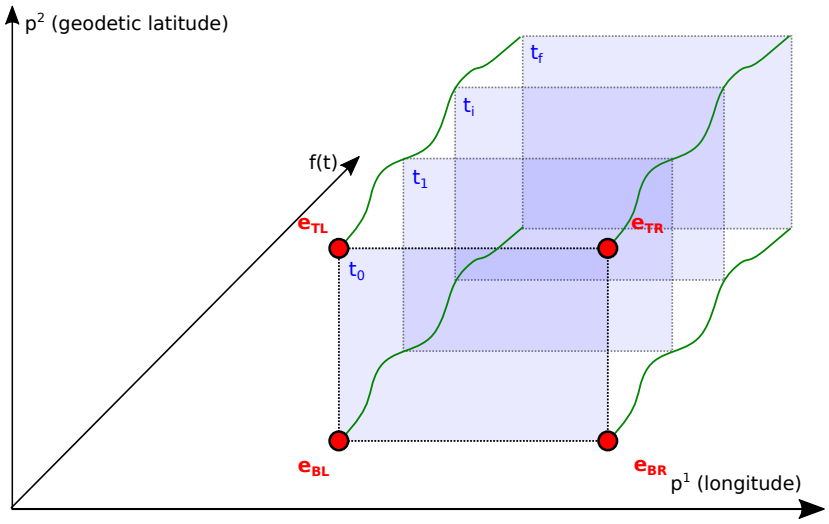

Figure 4.4: Sketch showing the low-density sampling of the trajectory information contained in the database.

variable always spans from -1 to 1.

$$t = \frac{t_f - t_0}{2}\tau + \frac{t_f - t_0}{2} \tag{4.14}$$

$$\tau = \frac{2}{t_f - t_0}t - \frac{t_f + t_0}{t_f - t_0} \tag{4.15}$$

Similarly to other trajectory variables, the pseudotime is sampled in a low-density grid, where each sampling point is referred to as a "node". The number of low-density nodes N_{LD} is a key parameter that determines the quality of the interpolated reference trajectory. On the one hand, if too few nodes would be used, the sampling would not be thorough enough and the interpolated solution would not capture well the entry dynamics of the four extremals. On the other hand, if too many points are used, the database compression may result marginal. Note that the number of low-density nodes is a parameter to be determined as part of this research, which is discussed in Chapter 8. Either way, the pseudotime is sampled into a vector of N_{LD} elements, as shown in the next equation:

$$\boldsymbol{\tau} = \left[\tau_0, \tau_1, \ldots, \tau_{N_{LD}}\right] \tag{4.16}$$

In addition to the number of low-density nodes, the spacing between them is a key parameter that influences how accurate the interpolated solution is. A known curse of polynomial interpolation is the so-called Runge's phenomenon, which leads to artificial oscillation towards the ends of the interval when the degree of the interpolating polynomial becomes too large and the nodes are equally spaced. The presence of such oscillations would result in an interpolated reference trajectory that artificially deviates from the extremals shortly after the EIP and just before the terminal point, leading to unwanted dispersions and possible the violation of the trajectory constraints. Sagliano et al. (2016) tackle this issue by distributing the low-density nodes according to the roots of the Flipped Radau-Legendre Polynomials (FRPs), which are defined according to the following two equations:

$$L_N(\tau) = \frac{1}{2^N N!}\frac{d^N}{d\tau^N}\left[\left(\tau^2 - 1\right)^2\right] \tag{4.17}$$

$$R_N(\tau) = L_N(\tau) - L_{N-1}(\tau) \tag{4.18}$$

where L_N is the Legendre polynomial of N^{th} degree and R_N is the FRP polynomial of N^{th} degree. An example of how the low-density nodes would be distributed is shown in the following figure, where an FRP of 45^{th} degree is shown. Such figure clearly shows that the method proposed by Sagliano et al. increases the number of low-density nodes towards the end of the interval, effectively tackling the discussed Runge's phenomenon.

4.1.4. LOW-DENSITY MULTIVARIATE INTERPOLATION

Once the pseudotime vector has been constructed and the database has been sampled, it is finally possible to interpolate the extremals. As mentioned in Section 4.1.2, using a 2-dimensional parameter space reduces

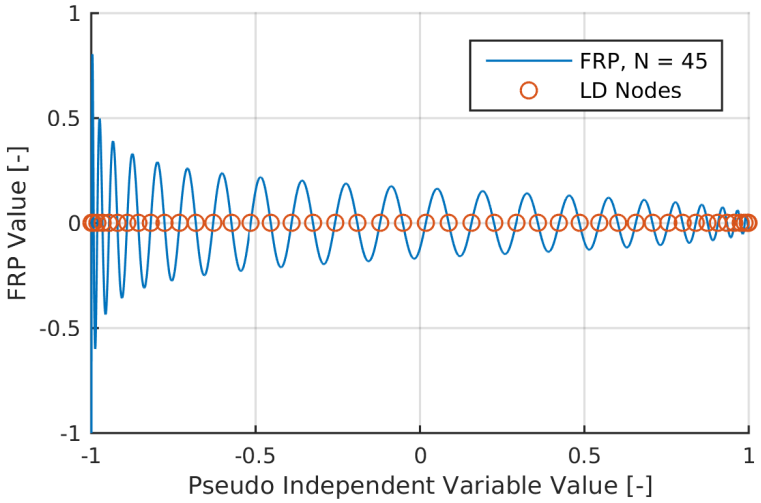

Figure 4.5: Example of a 45$^{\text{th}}$ degree FRP along with the resulting LD nodes.

the multivariate interpolation to bivariate interpolation. The trajectory interpolation discussed in this thesis is based on the bilinear interpolation scheme due to its simplicity. Furthermore, using a scheme that uses higher order basis functions, such as cubic spline interpolation, would require additional extremals to support the interpolation process. However, as Figure 4.3 reveals, these additional extremals would not necessarily improve the fidelity of the interpolated solution, since they would be located further away from the interpolation evaluation point as compared to the extremals in Equations 4.9 to 4.12.

The first step in the process of bilinear interpolation is to assemble the extremal function to be interpolated f into matrix form, as shown in the following equation:

$$\boldsymbol{C_k} = \begin{bmatrix} f(\tau_k, \boldsymbol{e_{BL}}) & f(\tau_k, \boldsymbol{e_{TL}}) \\ f(\tau_k, \boldsymbol{e_{BR}}) & f(\tau_k, \boldsymbol{e_{TR}}) \end{bmatrix} \tag{4.19}$$

where $\boldsymbol{C_k}$ is the matrix containing the database information evaluated at τ_k and $f(\tau_k, \boldsymbol{e_i})$ is the function to be interpolated evaluated at τ_k on extremal $\boldsymbol{e_i}$. Note that C_k is recomputed for every low-density node specified in Section 4.1.3. The next step is to assemble the projection matrices $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$, which are given in the next equation:

$$\boldsymbol{\phi}\left(x_0^1, \boldsymbol{s^1}, \boldsymbol{s^2}\right) = \frac{1}{\left(s_2^1 - s_1^1\right)\left(s_2^2 - s_1^2\right)} \begin{bmatrix} s_2^1 - x_0^1 \\ x_0^1 - s_1^1 \end{bmatrix} \tag{4.20}$$

$$\boldsymbol{\psi}\left(x_0^2, \boldsymbol{s^2}\right) = \begin{bmatrix} s_2^2 - x_0^2 \\ x_0^2 - s_1^2 \end{bmatrix} \tag{4.21}$$

Finally, the bilinear interpolation at is computed through a matrix multiplication, as indicated in the following equation:

$$f_{int}(\tau_k, \boldsymbol{x_0}) = \boldsymbol{\phi}^T C \boldsymbol{\psi} \tag{4.22}$$

where $f_{int}(\tau_k, \boldsymbol{x_0})$ is the interpolant of the extremal function f, evaluated at τ_k and $\boldsymbol{x_0}$. As a final note, it is important to emphasize that Equations 4.19 to 4.22 assume that the interpolation of any database variable f throughout the pseudotime vector $\boldsymbol{\tau}$ only depends on the relative position of $\boldsymbol{x_0}$ with respect to the extremals introduced in Equations 4.9 to 4.12. The interpolated function can be thus assembled into a vector as shown in the following equation:

$$\boldsymbol{f_{int}}(\tau, \boldsymbol{x_0}) = \left[f_{int}(\tau_0, \boldsymbol{x_0}), f_{int}(\tau_1, \boldsymbol{x_0}), \ldots, f_{int}\left(\tau_{N_{LD}}, \boldsymbol{x_0}\right)\right] \tag{4.23}$$

4.1.5. LOW-DENSITY TO HIGH-DENSITY PSEUDOSPECTRAL CONVERSION

The last stage of the AMPI algorithm is to convert the low-density interpolated solution obtained in the previous section into a high-density solution that can be used by the trajectory tracker. To do so, a high-density

pseudotime vector is first constructed as given in the following equation:

$$\tilde{\boldsymbol{\tau}} = \left[\tilde{\tau}_0, \tilde{\tau}_1, \ldots, \tilde{\tau}_{N_{HD}}\right] \tag{4.24}$$

where $\tilde{\boldsymbol{\tau}}$ is the high-density pseudotime vector and N_{HD} is the number of high-density nodes. Similarly to the low-density pseudotime vector, $\tilde{\boldsymbol{\tau}}$ spans from -1 to 1. The spacing of the nodes is not critical in this case, since Runge's phenomenon is no longer present due to the fact that no interpolation is made at this stage. The LD-HD conversion is made effective by constructing an approximating function that uses the low-density nodes as support points and the separation of high-density nodes and low-density nodes as basis functions. This is shown in the following equation:

$$\tilde{f}_{int}(\tilde{\tau}_m) = \sum_{i=0}^{N_{LD}} f_{int}(\tau_i) \prod_{\substack{k=0 \\ k\neq i}}^{N_{LD}} \frac{\tilde{\tau}_m - \tau_k}{\tau_i - \tau_k}, \; m = 0, 1, \ldots, N_{HD} \tag{4.25}$$

where $\tilde{f}_{int}(\tilde{\tau}_m)$ is the high-density interpolated solution at node $\tilde{\tau}_m$. The above operation can be assembled into matrix form as shown by the following two equations:

$$\tilde{\boldsymbol{f}}_{\boldsymbol{int}} = \boldsymbol{f}_{\boldsymbol{int}} \boldsymbol{P}_{\boldsymbol{FRP}} \tag{4.26}$$

$$\boldsymbol{P}_{\boldsymbol{FRP}} = \begin{bmatrix} \prod_{k=1}^{N_{LD}} \frac{\tilde{\tau}_0 - \tau_k}{\tau_0 - \tau_k} & \cdots & \prod_{k=1}^{N_{LD}} \frac{\tilde{\tau}_{N_{HD}} - \tau_k}{\tau_0 - \tau_k} \\ \cdots & \cdots & \cdots \\ \prod_{k=0}^{N_{LD}-1} \frac{\tilde{\tau}_0 - \tau_k}{\tau_{N_{LD}} - \tau_k} & \cdots & \prod_{k=0}^{N_{LD}-1} \frac{\tilde{\tau}_{N_{HD}} - \tau_k}{\tau_{N_{LD}} - \tau_k} \end{bmatrix} \tag{4.27}$$

where $\tilde{\boldsymbol{f}}_{\boldsymbol{int}}$ is the high-density interpolated solution and $\boldsymbol{P}_{\boldsymbol{FRP}}$ is the LD-HD conversion matrix. Note that $\boldsymbol{P}_{\boldsymbol{FRP}}$ only contains information about the relative spacing of the high-density nodes with respect to the low-density nodes.

A practical example of the on-board trajectory generation process via AMPI is shown in Figures 4.6 to 4.8. First, the AMPI algorithm receives from the navigation subsystem the geodetic latitude and longitude of the flown EIP, represented by the black cross in Figure 4.6. Given such information, the algorithm searches through a trajectory database stored on-board for optimal trajectories whose EIPs enclose the flown EIP. These optimal trajectories provide the supporting information for interpolation and they are obtained off-board using an optimization algorithm as discussed in Chapter 7. Please note that such trajectories not only contain the angle-of-attack and bank-angle controls, but also a number of dependent variable profiles, such as the vehicle state and the aerothermal load profiles. In fact, Figure 4.6 shows the groundtracks of the four neighboring trajectories to the flown EIP. The groundtracks overlap as they converge towards the targeted landing site, which in this example is the Santa Maria airport in the Azores archipelago, as shown in Figure 4.7. In this example the four supporting optimal trajectories, referred earlier in this section as "extremals", originate from EIPs that are placed at 10 deg from each other, enclosing a flown EIP placed at 18 deg in geodetic latitude and -130 deg in longitude.

The AMPI algorithm is capable of interpolating any function as long as supporting information is available in the optimal trajectory file. The algorithm uses the relative position of the flown EIP with respect to the extremal EIPs to compute the interpolant of the desired database function. For instance, Figure 4.8 shows how the bank-angle interpolant relates to the extremal bank-angle profiles. In this example, the flown EIP is located in the center of an "interpolation cell" similar to that shown in Figure 4.3. Consequently, the bank-angle interpolant resembles the "average" of the four extremal bank-angle profiles. Should the flown EIP in this example have been placed closer to one of the extremal trajectories, the resulting bank-angle interpolant would have had acquired a similar shape to that of the approached extremal. In fact, if the flown EIP is placed over one of the extremal EIPs, the resulting interpolant should resemble the shape of that particular extremal. As will be discussed in Section 6.3.2, this behavior is exploited to verify the software block responsible for the interpolation. As a final note, it is important to emphasize that due to the nature of interpolation, the resulting interpolants are smoothen as compared to the supporting extremals, as shown in Figure 4.8.

In this thesis, 190 LD nodes and 4000 HD nodes are used as a baseline. As will be shown in Chapter 7, database entries range from 2000 to 5000 seconds, thus 4000 HD nodes lead to an adequate spacing of 0.5 to 1.25 seconds. It is important to emphasize that the transformation discussed in section is a loss-less process, enabling the on-board storage of the database in its LD form. The compression factor of the database can be simply computed

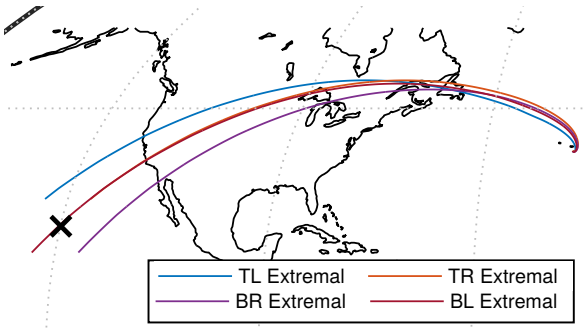

Figure 4.6: Extremal trajectories used in the interpolation of the example bank-angle profile.

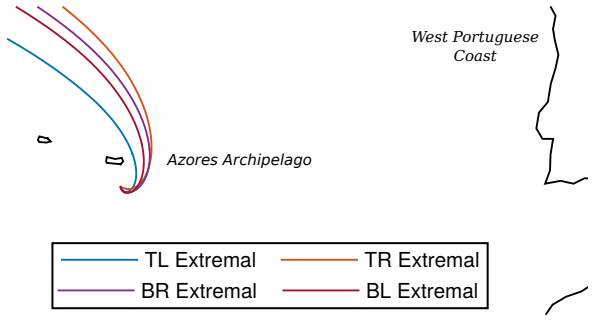

Figure 4.7: Zoomed view of final phase of the extremal trajectories used in the interpolation of the example bank-angle profile.

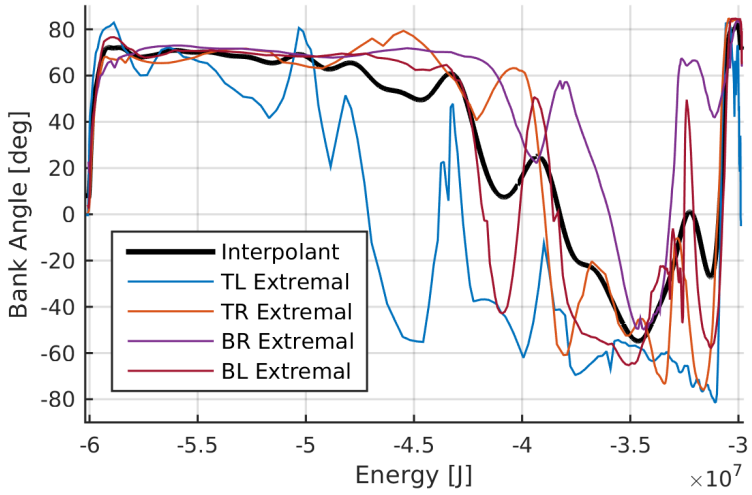

Figure 4.8: Example of bank-angle interpolation using energy as the independent variable.

by the next equation. Given the baseline compression factor is 0.0475, a potential database of 500 MB could be reduced to approximately 24 MB.

$$\kappa = \frac{N_{LD}}{N_{HD}} \tag{4.28}$$

Finally, the size of the database is estimated according to the following relation:

$$S_{DB} = N_t S_{F,HD} \frac{N_{LD}}{N_{HD}} \tag{4.29}$$

where S_{DB} is the compressed size of the database, N_t is the number of trajectories in the database and $S_{F,HD}$ is the average file size for a single trajectory. Note that using 4000 HD nodes, the average file size becomes approximately 300 kB.

In summary, the goal of this section was to introduce the theory behind the on-board generation of reference trajectories via AMPI. The method described is largely based on the work of (Sagliano et al., 2016), although simplified to a database of two parameters. The method described effectively builds the trajectory planner of the abort guidance system researched in this thesis, since it is capable of computing reference trajectories independent of the EIP encountered at the start of the flight.

4.2 LINEAR QUADRATIC REGULATOR THEORY

The goal of this section is to discuss the details behind the implementation of the trajectory tracker of the guidance system researched in this thesis. Such trajectory tracker is needed to ensure that the reference trajectory produced by the trajectory planner discussed in Section 4.1 is accurately flown, despite the presence of deviations in the vehicle state at the EIP or throughout the flight. The trajectory tracker implemented in this thesis

is based on the Linear Quadratic Regulator (LQR) theory, which is well known and has been applied in the past to reentry guidance such as in the studies of (Mooij, 2014), (Dukeman, 2002) and (Webb and Lu, 2016). In this section, an LQR trajectory tracker based on the work of (Mooij, 2014) is presented.

The purpose of LQR theory is to obtain a feedback control law for a linearized system by minimizing a performance criterion. The LQR tracker presented here uses linearized longitudinal dynamics with respect to time to generate the required angle-of-attack and bank-angle commands. Similarly to the work of Mooij (2014), the LQR tracker presented in this thesis is derived from the equations of motion in spherical coordinates. Furthermore, the state deviation vector $\delta\boldsymbol{x}$ proposed here consists of the groundspeed ΔV, flight-path angle $\Delta\gamma$, radius ΔR, and traveled range Δs, as shown by the following three equations. Additionally, the control deviation vector $\Delta\boldsymbol{u}$ is formed by the angle-of-attack $\Delta\alpha$ and the bank-angle $\Delta\sigma$. As shown in the following equation, the rate of change in state deviations $\Delta\dot{\boldsymbol{x}}$ as a function of $\Delta\boldsymbol{x}$ and $\Delta\boldsymbol{u}$. These deviations are computed with respect to the nominal state $\boldsymbol{x_{nom}}$ and the nominal controls $\boldsymbol{u_{nom}}$, which are provided by the trajectory planner discussed in Section 4.1.

$$\Delta\dot{\boldsymbol{x}} = \boldsymbol{A}\Delta\boldsymbol{x} + \boldsymbol{B}\Delta\boldsymbol{u} \tag{4.30}$$

where $\boldsymbol{A}$ is the system matrix which determines how the rate of change of state deviation is affected by the current state deviations. Furthermore, $\boldsymbol{B}$ is the input matrix, which determines how $\Delta\dot{\boldsymbol{x}}$ is affected by the current control deviations. The full mathematical expression of Equation 4.30 is given by the following equation:

$$\begin{bmatrix} \Delta\dot{V} \\ \Delta\dot{\gamma} \\ \Delta\dot{R} \\ \Delta\dot{s} \end{bmatrix} = \begin{bmatrix} a_{VV} & a_{V\gamma} & a_{VR} & a_{Vs} \\ a_{\gamma V} & a_{\gamma\gamma} & a_{\gamma R} & a_{\gamma a} \\ a_{RV} & a_{R\gamma} & a_{RR} & a_{Rs} \\ a_{sV} & a_{s\gamma} & a_{sR} & a_{ss} \end{bmatrix} \begin{bmatrix} \Delta V \\ \Delta\gamma \\ \Delta R \\ \Delta s \end{bmatrix} + \begin{bmatrix} b_{V\alpha} & b_{V\sigma} \\ b_{\gamma\alpha} & b_{\gamma\sigma} \\ b_{R\alpha} & b_{R\sigma} \\ b_{s\alpha} & b_{s\sigma} \end{bmatrix} \begin{bmatrix} \Delta\alpha \\ \Delta\sigma \end{bmatrix} \tag{4.31}$$

Note that the traveled range is defined by the following equation:

$$s = V_g \cos\gamma \tag{4.32}$$

For the sake of clarity, the elements of the system matrix are found in Equations 4.33 to 4.42. Each element is denoted by a_{ij}, which correlates the rate of change of deviations in the state variable i and the current deviation in the state variable j. Note that any variables with the subscript nom indicate nominal values that are retrieved from the reference trajectory.

$$a_{VV} = -\frac{1}{mV_{nom}}\left(M_{nom}\frac{\partial C_d}{\partial M}q_{nom}S_{ref} + 2D_{nom}\right) \tag{4.33}$$

$$a_{V\gamma} = -g_{nom}\cos\gamma_{nom} \tag{4.34}$$

$$a_{VR} = \frac{2g_{nom}}{R_{nom}}\sin\gamma_{nom} \tag{4.35}$$

$$a_{\gamma V} = \frac{1}{V_{nom}}\left(-\dot{\gamma}_{nom} + 2\frac{V_{nom}}{R_{nom}}\cos\gamma_{nom}\right) + \frac{\cos\sigma_{nom}}{mV_{nom}^2}\left(M_{nom}\frac{\partial C_l}{\partial M}q_{nom}S_{ref} + 2L_{nom}\right) \tag{4.36}$$

$$a_{\gamma\gamma} = -\left(\frac{V_{nom}}{R_{nom}} - \frac{g_{nom}}{V_{nom}}\right)\sin\gamma_{nom} \tag{4.37}$$

$$a_{\gamma R} = \frac{\cos\gamma_{nom}}{R_{nom}}\left(2\frac{g_{nom}}{V_{nom}} - \frac{V_{nom}}{R_{nom}}\right) \tag{4.38}$$

$$a_{RV} = \sin\gamma_{nom} \tag{4.39}$$

$$a_{R\gamma} = V_{nom}\cos\gamma_{nom} \tag{4.40}$$

$$a_{s\gamma} = V_{nom}\sin\gamma_{nom} \tag{4.41}$$

$$a_{Vs} = a_{\gamma s} = a_{RR} = a_{Rs} = a_{sV} = a_{sR} = a_{ss} = 0 \tag{4.42}$$

Similarly, the elements of the input matrix are given in Equations 4.43 to 4.46. Here, the matrix elements are denoted by b_{ij}, which correlates the rate of change of deviations in the state variable i and the current deviation

in the control variable j.

$$b_{V\alpha} = -\frac{1}{m}\frac{\partial C_d}{\partial \alpha} q_{nom} S_{ref} \tag{4.43}$$

$$b_{\gamma\alpha} = \frac{\cos\sigma_{nom}}{mV_{nom}}\frac{\partial C_l}{\partial \alpha} q_{nom} S_{ref} \tag{4.44}$$

$$b_{\gamma\sigma} = -\frac{L_{nom}}{mV_{nom}}\sin\sigma_{nom} \tag{4.45}$$

$$b_{R\alpha} = b_{s\alpha} = b_{V\sigma} = b_{R\sigma} = b_{s\sigma} = 0 \tag{4.46}$$

Note that the equations just presented introduce partial derivatives of the lift and drag coefficients with respect to Mach number and angle-of-attack. As stated in Section 2.4, the aerodynamic database of the SPHYNX vehicle does not provide derivative information, thus the partial derivatives just given need to be computed via numerical differentiation. Such partial derivatives are approximated using the central difference scheme, as shown in the following four equations:

$$\frac{\partial C_d}{\partial \alpha} \approx \frac{C_d\left(M_{nom}, \alpha_{nom} + \frac{1}{2}h_\alpha\right) - C_d\left(M_{nom}, \alpha_{nom} - \frac{1}{2}h_\alpha\right)}{h_\alpha} \tag{4.47}$$

$$\frac{\partial C_l}{\partial \alpha} \approx \frac{C_l\left(M_{nom}, \alpha_{nom} + \frac{1}{2}h_\alpha\right) - C_l\left(M_{nom}, \alpha_{nom} - \frac{1}{2}h_\alpha\right)}{h_\alpha} \tag{4.48}$$

$$\frac{\partial C_d}{\partial M} \approx \frac{C_d\left(M_{nom} + \frac{1}{2}h_M, \alpha_{nom}\right) - C_d\left(M_{nom} - \frac{1}{2}h_M, \alpha_{nom}\right)}{h_M} \tag{4.49}$$

$$\frac{\partial C_l}{\partial M} \approx \frac{C_l\left(M_{nom} + \frac{1}{2}h_M, \alpha_{nom}\right) - C_l\left(M_{nom} - \frac{1}{2}h_M, \alpha_{nom}\right)}{h_M} \tag{4.50}$$

where $\partial C_d/\partial\alpha$ is the partial derivative of the drag coefficient with respect to the angle-of-attack, $\partial C_l/\partial\alpha$ is the partial derivative of the lift coefficient with respect to the angle-of-attack, $\partial C_d/\partial M$ is the partial derivative of the drag coefficient with respect to the Mach number and $\partial C_l/\partial M$ is the partial derivative of the lift coefficient with respect to the Mach number. In addition, h_α and h_σ refer to the central difference step size for angle-of-attack and Mach number, respectively. Note that a small step size is desired to retrieve a good approximation to the partial derivatives, which is particularly important in the supersonic region where the aerodynamic response of SPHYNX is rapidly changing, as shown in Figures 2.14 and 2.16. Doing otherwise would cause Equations (4.47) to (4.50) to overestimate the magnitude of the partial derivatives, ultimately leading to poor tracking performance. The values for such step sizes are adjusted by trial and error and are shown in the following two equations:

$$h_\alpha = 0.006 \text{ deg} \tag{4.51}$$

$$h_M = 0.03 \tag{4.52}$$

As mentioned above, the goal of the LQR theory is to define a feedback control law. This law computes the required control deviations as a function of a feedback matrix $\boldsymbol{K}$ and the state deviations. The feedback control law is shown in the next equation:

$$\Delta\boldsymbol{u} = -\boldsymbol{K}\Delta\boldsymbol{x} \tag{4.53}$$

The elements of the feedback matrix are given in the next equation:

$$\boldsymbol{K} = \begin{bmatrix} k_{\alpha,V} & k_{\alpha,\gamma} & k_{\alpha,R} & k_{\alpha,s} \\ k_{\sigma,V} & k_{\sigma,\gamma} & k_{\sigma,R} & k_{\sigma,s} \end{bmatrix} \tag{4.54}$$

where $k_{i,j}$ quantifies the required deviations in control u_i given the errors in the state variable x_j. The feedback matrix is computed by minimizing the performance criterion presented in the following equation:

$$J_{LQR} = \int_{t_0}^{\infty} \left(Q_V \Delta V^2 + Q_s \Delta s^2 + R_\alpha \Delta\alpha^2 + R_\sigma \Delta\sigma^2\right) dt \tag{4.55}$$

where Q_i refers to the weight associated to the deviation in the state variable i and R_i refers to the weight associated to the deviation in the control variable i. A unique solution to this minimization is found by solving Equa-

tions (4.56) to (4.57). The theoretical details behind this system equations can be found in (Wie, 2008).

$$\boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}\boldsymbol{P} + \boldsymbol{Q} = \boldsymbol{0} \tag{4.56}$$

$$\boldsymbol{K} = \boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P} \tag{4.57}$$

where $\boldsymbol{Q}$ is the weight matrix of state deviation and $\boldsymbol{R}$ is the weight matrix of control deviations. The elements of such matrices are given in the following two equations:

$$\boldsymbol{Q} = \begin{bmatrix} Q_V & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & Q_s \end{bmatrix} \tag{4.58}$$

$$\boldsymbol{R} = \begin{bmatrix} R_\alpha & 0 \\ 0 & R_\sigma \end{bmatrix} \tag{4.59}$$

Note that the diagonal elements in the matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ can be found using Bryson's rule, which is stated in the following equation (Dukeman, 2002):

$$Q_V \Delta V_{max}^2 = Q_s \Delta s_{max}^2 = R_\alpha \Delta\alpha_{max}^2 = R_\sigma \Delta\sigma_{max}^2 \tag{4.60}$$

where the variables denoted with subscript max refer to the maximum allowable deviations in such variable. Note that these maximum allowable deviations are simply control law design parameters and do not necessarily impose hard constraints (Dukeman, 2002). Finally, the baseline maximum allowable deviations for the tracked variable are given below:

$$\Delta V_{max} = 50.0 \text{ m/s} \tag{4.61}$$

$$\Delta s_{max} = 10.0 \text{ m} \tag{4.62}$$

$$\Delta\alpha_{max} = 0.5 \text{ deg} \tag{4.63}$$

$$\Delta\sigma_{max} = 5.0 \text{ deg} \tag{4.64}$$

Note that the values given above merely establish a baseline and thus may need to be tuned as discussed in Chapter 8. Once Equations (4.56) to (4.59) are solved and the feedback matrix is known, the commanded angle-of-attack and the commanded bank-angle magnitude are given by the following two equations:

$$\alpha = \alpha_{nom} + \Delta\alpha \tag{4.65}$$

$$|\sigma| = |\sigma_{nom}| + \Delta\sigma \tag{4.66}$$

It is important to emphasize that controlling the four state variables with two control variables has proven to be unfeasible. For this matter, the weights of the flight-path angle and radius are set to zero, in a way that infinite errors are allowed for such variables but are still part of the linearized dynamic equations. Despite this, Figures 4.9 to 4.16 show that the angle-of-attack and bank-angle gains associated to the four states are non-zero throughout the entry profile. This indicates that errors in the flight-path angle and the radial distance are still accounted for when computing the necessary control deviations.

In addition, note that the elements of the input matrix that relate the control inputs to the rate of change of the traveled range error are zero, indicating that the traveled range cannot be controlled directly. Instead, the traveled range is indirectly controlled by means of the groundspeed and the flight-path angle, as shown in Equation (4.32). As will be discussed in Section 6.3.3, the LQR tracker presented in this section tracks well both altitude and groundspeed, but shows significant dispersions in the final distance to the landing site. This behavior suggests that direct tracking of the traveled range could be beneficial, which could be achieved by combining the traveled range kinematic expression in Equation (4.32) with an additional state variable, in a similar way that altitude and flight-path angle are combined into the "pseudo-altitude" variable in (Mooij, 2017).

The gains shown in Figures 4.9 to 4.16 are obtained from a nominal reference trajectory with the entry conditions established in Table 6.4 and are used in Section 6.3.3 to test the performance of the trajectory tracker. Please note that each trajectory stored in the database has its own set of gains and are thus part of the trajectory interpolation scheme discussed in Section 4.1. Although the gains are shown here as a function of time for the sake of clarity, the interpolation uses the vehicle total energy as the independent variable. Using energy

instead of time allows to schedule the gains according to a monotonic variable that represents well the entry dynamics. Although the gains shown in Figures 4.9 to 4.16 exhibit relatively fast changes in time, this was not considered an issue since this effect is mitigated thanks to the smoothing nature of interpolation. Furthermore, such behavior mostly occurs in the first 1500 s after entry, where the vehicle is in the upper atmosphere and where aerodynamic control authority is limited.

Please note that the tracker presented here only deals with the vehicle's longitudinal dynamics. Consequently, a separate lateral guidance logic is required to provide the sign of the commanded bank-angle.

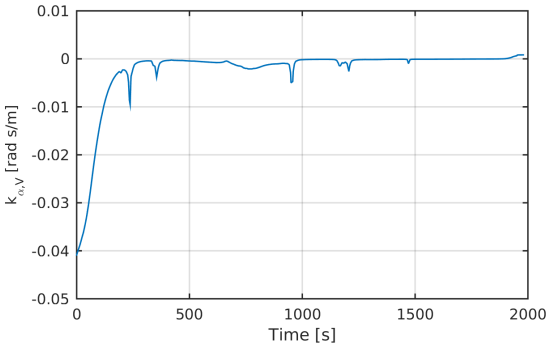

Figure 4.9: Angle-of-attack gain due to errors in groundspeed, given a nominal trajectory with the entry conditions established in Table 6.4.

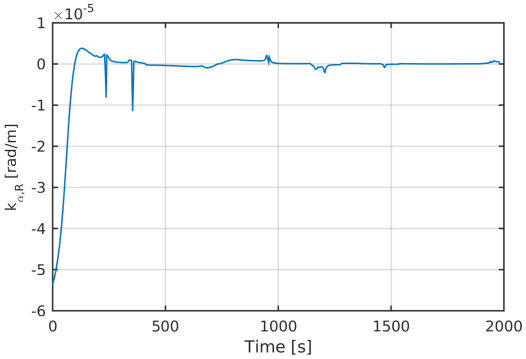

Figure 4.10: Angle-of-attack gain due to errors in radial distance, given a nominal trajectory with the entry conditions established in Table 6.4.

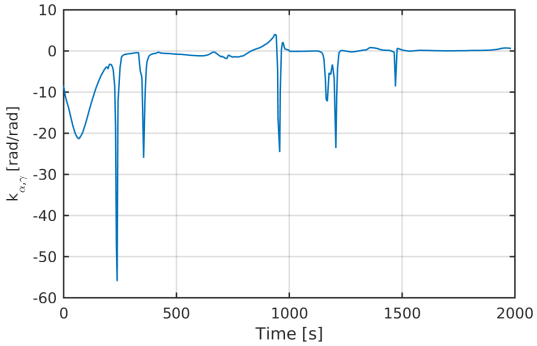

Figure 4.11: Angle-of-attack gain due to errors in flight-path angle, given a nominal trajectory with the entry conditions established in Table 6.4.

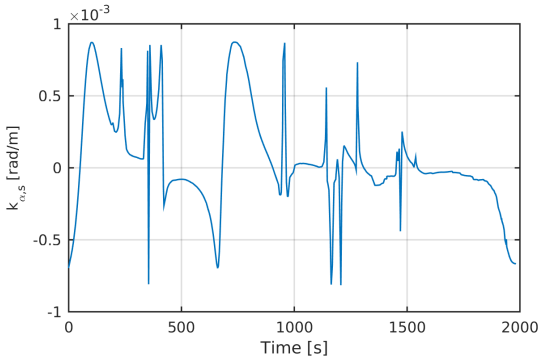

Figure 4.12: Angle-of-attack gain due to errors in traveled distance, given a nominal trajectory with the entry conditions established in Table 6.4.

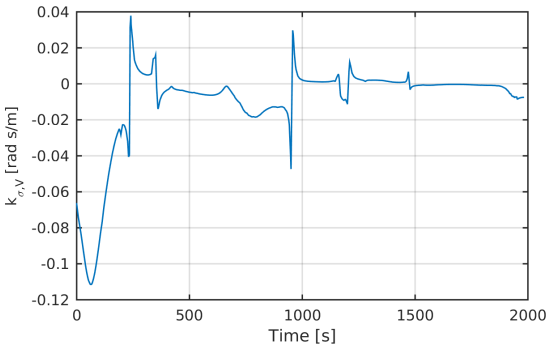

Figure 4.13: Bank-angle gain due to errors in groundspeed, given a nominal trajectory with the entry conditions established in Table 6.4.

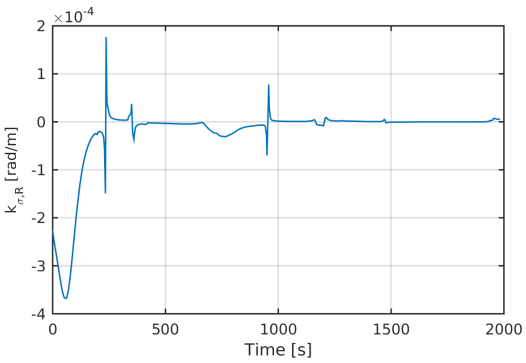

Figure 4.14: Bank-angle gain due to errors in radial distance, given a nominal trajectory with the entry conditions established in Table 6.4.

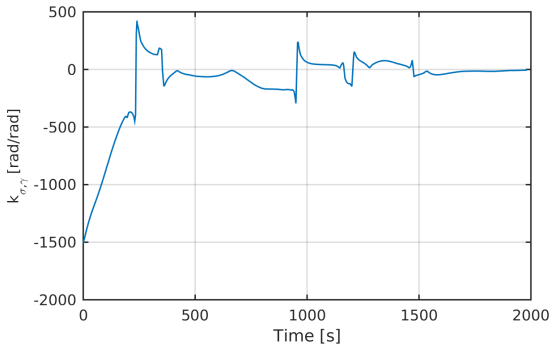

Figure 4.15: Bank-angle gain due to errors in flight-path angle, given a nominal trajectory with the entry conditions established in Table 6.4.

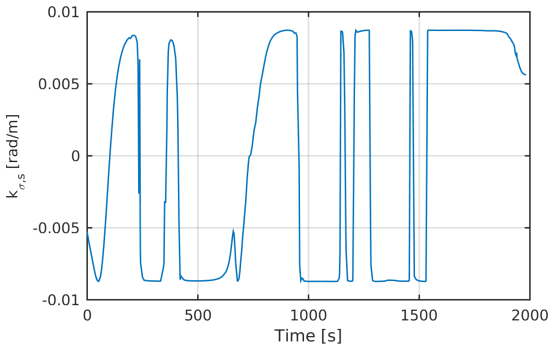

Figure 4.16: Bank-angle gain due to errors in traveled distance, given a nominal trajectory with the entry conditions established in Table 6.4.

4.3 LATERAL GUIDANCE

The goal of this section is to discuss the lateral guidance implemented in the abort guidance system developed in this thesis. The role of such guidance is to keep the vehicle from deviating too far in the cross-track direction from its nominal path. The lateral guidance is essential to ensure that the vehicle flies towards the targeted landing site. The lateral guidance implemented in this thesis is based on the work from Mooij (2014), which tracks the heading angle error and ensures that it is kept within some predefined bounds.

To begin with, the heading angle to the target landing site χ_T is computed using the following equation, where the geometry employed to derive such relation is shown in Figure 4.17.

$$\chi_T = \tan^{-1}\left(\frac{\sin(\tau_T-\tau)\sin\left(\frac{\pi}{2}-\delta_T\right)}{\cos\left(\frac{\pi}{2}-\delta_T\right)\sin\left(\frac{\pi}{2}-\delta\right)-\cos(\tau_T-\tau)\cos\left(\frac{\pi}{2}-\delta\right)\sin\left(\frac{\pi}{2}-\delta_T\right)}\right) \tag{4.67}$$

where τ is the vehicle longitude, δ is the vehicle latitude, τ_T is the longitude of the landing site and δ_T is the latitude of the landing site. The heading angle error χ_e at any point throughout the trajectory is simply the difference between the vehicle's current heading angle and the heading angle to the target:

$$\chi_e = \chi - \chi_T \tag{4.68}$$

According to the previous equation, a zero heading error indicates that the vehicle is traveling along the great circle arc that connects the current vehicle position and the target landing site position. Furthermore, a positive heading error indicates that the vehicle deviates in the starboard direction from such arc. On the contrary, a negative heading error would indicate a deviation in the port direction. The heading error can be controlled by reversing the sign of bank-angle, which effectively mirrors the lift vector with respect to the vertical plane. A positive bank-angle results in a lift vector whose horizontal projection points starboard. Similarly, a negative bank-angle leads the horizontal projection to point port. Consequently, enforcing the signs of the vehicle's heading error and commanded bank-angle to be opposite, the vehicle is steered in such a way that it flies towards the landing site:

$$\chi_e \text{sign}(\sigma) < 0 \tag{4.69}$$

In addition, the magnitude of the heading error is controlled to avoid an excessive number of bank reversals. This is done by establishing a heading error dead band (HEDB), which constructs a profile that determines the maximum allowable heading error χ_{db} as a function of some independent variable. An example of such HEDB is shown in Figure 4.18, where the χ_{db} profile is plotted against the distance to the target. As discussed by Mooij (2014), the impact of the HEDB becomes critical the closer the vehicle gets to the landing site. A too tight HEDB could lead to unnecessary bank reversals, where a wide HEDB could lead to the vehicle missing the landing site interface. In addition, the design of the HEDB profile should be tuned in such a way that the last bank reversal places the vehicle at the landing interface with a heading error that meets the required value. To do so, the so-called low-distance HEDB is computed. Since no such requirement was set for this thesis, the tuning of the HEDB is not implemented. Thus, a constant coarse HEDB of ±25 deg was established by means of trial and error, which has proved to be effective in steering the vehicle towards the landing site as discussed in Section

6.3.3. In summary, the lateral guidance discussed here commands bank reversals as long as the conditions in Equations 4.69 and 4.70 are met.

$$|\chi_e| > |\chi_{db}| \tag{4.70}$$

As a final note, bank reversals must be executed with a finite bank-angle rate which is determined by the capabilities of the vehicle's control system. In the case of a lifting body such as SPHYNX such capability is strongly influenced by the flight regime, which determines the effectiveness of the elevons. Furthermore, Mooij (2014) discusses how such finite bank-angle rate introduces flight-path angle errors at bank reversals, which ultimately propagate forward in the trajectory and must be thus handled by the longitudinal guidance. The effects discussed here are considered beyond the scope of this thesis, thus any bank reversals executed by the abort guidance system are instantaneous and thus modeled with an infinite bank-angle rate.

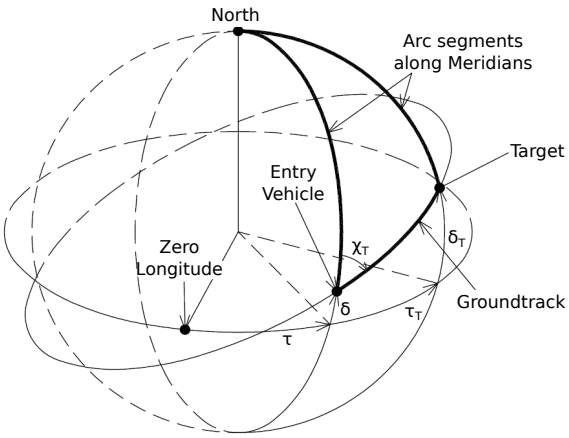

Figure 4.17: Sketch showing the geometry used to compute the heading angle error. Adapted from (Mooij, 2014)

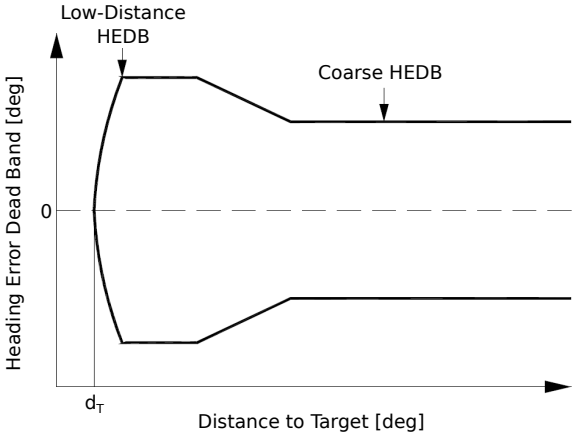

Figure 4.18: Sketch showing a typical HEDB profile. Adapted from (Mooij, 2014)

5 NUMERICAL METHODS

In this chapter, an overview of the numerical methods to be used throughout the thesis assignment is given. These numerical methods are classified according to their goal and are later linked to their particular usage in the thesis.

5.1 ROOT-FINDING

The goal of the root-finding methods is to determine the zeroes of a given input function $f(x)$. Root-finding methods are often used to solve non-linear functions, which cannot be solved symbolically and are commonplace in entry problems.

5.1.1. BISECTION METHOD

The bisection method (BSM) is the simplest of all root-finding methods. It exploits Bolzano's theorem, which states that given a continuous function $f(x)$, a zero is guaranteed to exist within the interval $[x_l, x_u]$ as long as $f(x_l)$ and $f(x_u)$ have opposite signs. Root-finding methods exploiting this theorem are called bracketing methods. The BSM iteratively reduces the interval width until a predefined tolerance ϵ is obtained. The first step in the BSM is to compute the interval midpoint x_m and its corresponding function value $f(x_m)$, as shown in Equation (5.1). This operation bisects the original interval into sub-intervals $[x_l, x_m]$ and $[x_m, x_u]$.

$$x_m = \frac{1}{2}(x_l + x_u) \tag{5.1}$$

The second step is to determine whether the midpoint x_m serves as the function's root, which is true once the interval width is smaller than the required tolerance, as shown in Equation (5.2). If the tolerance requirement is not met, the method proceeds to the next iteration and either $[x_l, x_m]$ or $[x_m, x_u]$ is selected as the new interval to bisect. To guarantee convergence, the new interval must yield opposite function signs at its boundaries. The BSM has a linear order of convergence and its convergence is guaranteed.

$$\epsilon < x_u - x_l \tag{5.2}$$

The BSM is employed to find the body flap deflections of the SPHYNX vehicle that zero Equation (2.3), effectively trimming the vehicle. Furthermore, the BSM is used to find the roots of the FRPs that establish the LD nodes discussed in Section 4.1.3.

5.1.2. SECANT METHOD

Similarly to the BSM, the Secant Method (SM) is a bracketing method that exploits Bolzano's theorem by searching for roots within an interval where zero crossings are guaranteed. The SM iteratively reduces the search interval until the required tolerance ϵ is achieved. Contrary to the BSM, the SM does not define the root of $f(x)$ as the interval midpoint. Instead, the root of $f(x)$ is set as the x-intercept of the secant lines going through the last root guesses x_i and x_{i-1}, meaning that the root may not necessarily be bracketed. The SM requires two initial root guesses that should be close to the true root. An expression for the new root guess is given by the following equation:

$$x_{i+1} = x_i - f(x_i)\frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})} \tag{5.3}$$

Convergence of the SM is not guaranteed and its order of convergence is better than linear but worse than quadratic, also known as superlinear convergence. The SM is used to find the zero of Equation (3.72), which is

used to find the wall temperature at which the convective heat flux and the radiative heat flux are in equilibrium.

5.2 INTEGRATION

The purpose of a numerical integrator is to find a solution to a system of differential equations that satisfies a set of initial conditions (Burden and Faires, 2011). This problem is commonly referred in literature as an Initial-Value Problem (IVP). Numerical integrators are required in the thesis assignment to find a solution to the equations of motion given in Section 3.6, which describe the time evolution of the vehicle state. Nowadays, multiple numerical integrators exists and these vary in complexity and performance. To fully understand the reasons behind these differences, a short theoretical introduction must be given first. To begin with, the mathematical formulation of the IVP is given in the next equation:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}), \qquad t \in \left[t_0, t_f\right], \qquad \boldsymbol{x}(t_0) = \boldsymbol{x_0} \tag{5.4}$$

where $\boldsymbol{x}$ is the state vector, t is the independent variable, $\boldsymbol{f}$ is the system of differential equations to be solved and $\boldsymbol{x_0}$ is the initial state vector. In this case, the independent variable is time which is defined in the interval $\left[t_0, t_f\right]$. The outcome of a numerical integrator is an approximation $\boldsymbol{\eta}(t, \boldsymbol{x})$ to the exact solution of the IVP, as shown in the following relation:

$$\boldsymbol{x}(t_i + h) = \boldsymbol{x_{i+1}} \approx \boldsymbol{x_i} + h\boldsymbol{\Phi}(t, \boldsymbol{x}) = \boldsymbol{\eta}(t, \boldsymbol{x}) \tag{5.5}$$

In the previous equation, $\boldsymbol{x}(t_i + h)$ is the predicted state a step forward in time h, which is conveniently denoted by $\boldsymbol{x_{i+1}}$, where n indicates the current stepsize. The approximation of $\boldsymbol{x_{i+1}}$ is computed using the current state $\boldsymbol{x_i}$, the current stepsize h and an increment function $\boldsymbol{\Phi}(t, \boldsymbol{x})$. The shape of increment function is uniquely defined by each integrator scheme and it may contain stepsize and state information at multiple time instants.

Please note that Equation (5.5) is applied the entire independent variable domain in a sequential manner, ultimately yielding x as a function of t. The error of a numerical integrator is defined as the difference between the exact solution value and the numerical prediction. This error can be evaluated at each stepsize, as shown in the next equation:

$$\epsilon = \left|\boldsymbol{x}(t_i + h) - \boldsymbol{\eta}(t, \boldsymbol{x})\right| \tag{5.6}$$

The magnitude of this error largely depends on the stepsize size, where smaller step sizes lead to smaller errors. However, decreasing the step size requires an increase in the number of steps to be taken, as a consequence of the fixed time domain. This leads to a higher computational effort which may lead to unfeasible computational times. This issue can be tackled by selecting a different integrator scheme, which could potentially achieve a similar accuracy with a larger step size.

To illustrate the above discussion, Figure 5.1 shows the solution of a one-dimensional IVP in a numerical fashion. Here, the $f(t)$ is the derivative, $x(t)$ is the solution to the IVP, and t is the independent variable. This figure clearly shows how decreasing the stepsize leads to a reduction in the numerical error. As a final note, one can classify the different numerical integrators based on where the step-size and derivative information is obtained. Depending on where the value of the derivative is obtained, integration algorithms can be classified under two groups: *single-step methods* and *multi-step methods*. As will be highlighted in Section 6, at the time of writing of this report there were no multi-step methods available in the main software library used for this thesis. Consequently, only single-step methods are discussed in this section. Single-step methods are those which use derivative information from a single step (Burden and Faires, 2011). When propagating x_i to x_{i+1} in Figure 5.1, a single-step method would only use information from the interval $[t_2, t_3]$.

5.2.1. CLASSICAL RUNGE-KUTTA METHODS

The Runge-Kutta (RK) is a commonly used family of single-step methods based on the slopes at various points within the integration step (Montenbruck and Gill, 2001b). In these methods, the increment function is computed as a weighted mean of these slopes using a set of b_i coefficients, as indicated in Equation (5.7). Note that the points taken within the integration step are referred as the stages s of the RK method.

$$\boldsymbol{x_{i+1}} = \boldsymbol{x_i} + h\sum_{i=1}^{s} b_i \boldsymbol{k_i} \tag{5.7}$$

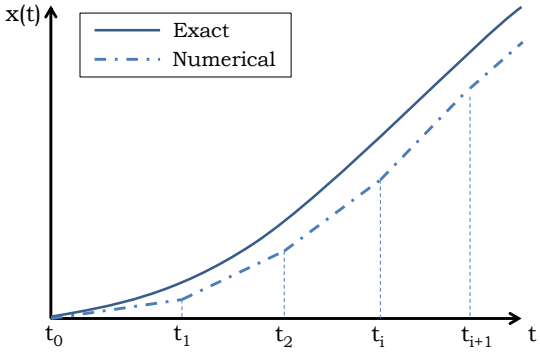

Figure 5.1: Sketch showing the basis behind numerical integration.

The slope functions used in the RK methods use a generic formulation indicated by Equations (5.8) and (5.9). The position of the stage within the interval is weighted with a set of c_i coefficients, also known as the nodes.

$$\boldsymbol{k_1} = \boldsymbol{f}\left(t_i + c_1 h, \boldsymbol{x_i}\right) \tag{5.8}$$

$$\boldsymbol{k_i} = \boldsymbol{f}\left(t_i + c_i h, \boldsymbol{x_i} + h\sum_{j=1}^{i-1} a_{ij}\boldsymbol{k_j}\right), \qquad (i = 2,3,\ldots,s) \tag{5.9}$$

Equation (5.9) introduces the set of parameters a_i, which are computed such that the order of the local truncation error is high as possible (Montenbruck and Gill, 2001b). Note that the a_i coefficients must satisfy the following relation:

$$c_i = \sum_{j=1}^{i-1} a_{ij}, \qquad (i = 2,3,\ldots,s) \tag{5.10}$$

Although increasing the number of stages tends to decrease the error made in the integration, this significantly increases the computational effort. According to Burden and Faires (2011), RK methods suffer from the so called Butcher boundary, which limits the best possible local truncation error that the method can achieve as a function of the number of function evaluations per step. Such boundary implies that methods of order five with a smaller step size are preferred over high-order methods with a large step size.

The RK4 method is a four stage method that is commonly used thanks to its simplicity and its moderate accuracy. In accordance to the discussion above, the RK4 method is right at Butcher boundary, thus implying a good balance between the method's accuracy and speed. The increment function and the slope functions of the RK4 method are shown in the following five equations:

$$\boldsymbol{\Phi_{RK4}} = \frac{1}{6}\left(\boldsymbol{k_1} + 2\boldsymbol{k_2} + 2\boldsymbol{k_3} + \boldsymbol{k_4}\right) \tag{5.11}$$

$$\boldsymbol{k_1} = \boldsymbol{f}\left(t_i, \boldsymbol{x_i}\right) \tag{5.12}$$

$$\boldsymbol{k_2} = \boldsymbol{f}\left(t_i + h/2, \boldsymbol{x_i} + h\boldsymbol{k_1}/2\right) \tag{5.13}$$

$$\boldsymbol{k_3} = \boldsymbol{f}\left(t_i + h/2, \boldsymbol{x_i} + h\boldsymbol{k_2}/2\right) \tag{5.14}$$

$$\boldsymbol{k_4} = \boldsymbol{f}\left(t_i + h, \boldsymbol{x_i} + h\boldsymbol{k_3}\right) \tag{5.15}$$

The RK4 method is the method of choice to integrate the equations of motion presented in Section 3.4. The step size is set to 0.5 seconds. Note that RK4 suffers from the inability of controlling the actual interpolation error made at each step. This is a direct consequence of using a fixed step-size and can result in the violation of the truncation error tolerance if the chosen step-size is too large for particular time. On the other hand, the chosen step-size can prove too small, leading to the waste of computational power. To tackle this problem, *embedded* RK methods were devised, which are capable of adapting the step-size when necessary. However, Embedded RK methods do not strictly evaluate the state derivative function in a monotonous fashion which turns out problematic when implementing a guidance scheme.

5.3 INTERPOLATION

Interpolation is the process of constructing a function which passes through specified data points (Dwight, 2013). The general expression defining the interpolation problem is given in the following relation:

$$p(\boldsymbol{x_i}) = f_i, \qquad i = (1, 2, \ldots, n) \tag{5.16}$$

where p is the interpolating function, also known as the interpolant, $\boldsymbol{x_i}$ are the grid nodes and f_i are the function values at such nodes.

5.3.1. LINEAR SPLINE INTERPOLATION

Linear spline interpolation is one of the most basic interpolation methods available. It is based on the construction of linear segments between each of the points in the data set. The connection of these segments creates a piecewise function that is continuous at the data points. Such function is referred as a spline where its mathematical notation is given in the next equation:

$$s(x) = \begin{cases} s_0(x) & \text{if } x \in [x_0, x_1] \\ s_1(x) & \text{if } x \in [x_1, x_2] \\ \vdots & \vdots \\ s_{n-1}(x) & \text{if } x \in [x_{n-1}, x_n] \end{cases} \tag{5.17}$$

where $s(x)$ is the spline, $s_i(x)$ are the spline segments, x_i are the grid points and n is the number of grid points. In linear spline interpolation, the slope of each segment is determined by the function values at the boundaries of the segment, as shown in the following relation:

$$s_i(x) = f_i + \frac{f_{i+1} - f_i}{x_{i+1} - x_i}(x - x_i), \qquad (i = 0, 1, \ldots, n) \tag{5.18}$$

where f_i are the function values at the grid points. Linear spline interpolation provides a continuous function that passes through all the points in the data set. However, the generated spline is non-differentiable at the segment joints, which can result in convergence problems when executing optimization routines that employ such spline.

5.3.2. CUBIC SPLINE INTERPOLATION

Similarly to its linear counterpart, cubic spline interpolation constructs a piecewise continuous function by using multiple segments. However, as the name indicates, in this case the segments consists of third degree polynomials. Although the method becomes slightly more extensive, cubic spline interpolation produces a spline that is differentiable at the segment joints. The shape of each segment is given by the following equation:

$$s_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \tag{5.19}$$

The a_i, b_i, c_i and d_i coefficients in Equation (5.19) depend on the function's second derivative at the segment joints, denoted here as M_i. This dependency is shown in next four equations:

$$a_i = \frac{M_{i+1} - M_i}{6h_i} \tag{5.20}$$

$$b_i = \frac{M_i}{2} \tag{5.21}$$

$$c_i = \frac{f_{i+1} - f_i}{h_i} - \frac{h_i}{3}M_i - \frac{h_i}{6}M_{i+1} \tag{5.22}$$

$$d_i = f_i \tag{5.23}$$

where $h_i = x_{i+1} - x_i$. Since the second derivatives are unknown, they are found by imposing that the first derivatives of adjacent segments match at the corresponding segment boundaries. This is expressed by the following equation:

$$s_i'(x_i) = s_{i-1}'(x_i) \tag{5.24}$$

The details behind the determination of M_i are found in (Dwight, 2013). Finally, the definition of the cubic

splines is completed by assuming the second derivatives at the first and last nodes to be zero.

Note that other interpolation methods exist, aside from the ones presented above. For instance, *polynomial interpolation* attempts to fit a certain polynomial to a data set, where the resulting interpolant is a simple, continuous and differentiable function. Although simple, polynomial interpolation introduces artificial oscillations in the interpolant and does not ensure that such interpolant passes through all the datapoints. In addition, *Hermite spline interpolation* extends cubic spline interpolation using derivative information to construct each of the segments, thus yielding a more accurate approximation to the original function. The main issue behind Hermite splines is that the required derivative information is often not available, which is the case given the data sets to be interpolated for the thesis assignment.

In the scope of the thesis assignment, interpolation is used to construct functions from the following data sets:

- **SPHYNX's Aerodynamic Database:** The aerodynamic database is provided in tabular form, thus data points are only available at discrete nodes. Due to the continuous nature of aerodynamics, information is often required in between the available nodes, thus justifying the need for interpolation. As identified in Equations (2.1) to (2.3), the aerodynamic coefficient information can be a function of up to three independent variables. As will be discussed in Chapter 6, multi-linear spline interpolation is available in the software library used in this thesis. For this reason, linear splines are used to interpolate the aerodynamic database. It is important to emphasize that the optimization software used in this thesis is automatically smooths any problematic discontinuities caused by the linear interpolation of the database.
- **Extremal Trajectory Interpolation**: As discussed in Section 4.1.4, bi-linear spline interpolation serves as the core of the trajectory planner used by the guidance system of this thesis.
- **US76 Atmosphere Model:** As will be explained in Section 6, the atmosphere model available in the main software library comes in the form of tabulated data, thus requiring the use of an interpolation technique. The atmospheric properties are interpolated using Cubic spline interpolation, due to the one-dimensional nature of the data set.

5.4 OPTIMIZATION

The goal of this section is to introduce the basics of numerical optimization. Numerical optimization is the process of minimizing (or maximizing) a certain function using numerical methods. Such methods are commonly used in trajectory optimization problems due to their non-linear and highly constrained nature. The discussions in this section are kept short, since in the scope of the thesis assignment, optimization is regarded as a tool rather than an area of research.

As discussed in Section 3.6, the dynamics of a reentry vehicle are described by a set of differential equations which depend on a set of state and control variables. According to Betts (1998), such equations can be generalized using the mathematical notation shown in the following equation:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{p}, t\right) \tag{5.25}$$

where $\dot{\boldsymbol{x}}$ is the time derivative of the state vector, $\boldsymbol{f}$ is the vector containing the functions that describe the dynamics, $\boldsymbol{x}$ is the state vector, $\boldsymbol{u}$ is the control vector and $\boldsymbol{p}$ is a set of parameters. Furthermore, the vehicle dynamics are completed by defining initial boundary and final boundary constraints, whose general mathematical notation is shown in the following two equations:

$$\boldsymbol{\psi_{0l}} \leq \boldsymbol{\psi}\left(\boldsymbol{x}(t_0), \boldsymbol{u}(t_0), \boldsymbol{p}, t_0\right) \leq \boldsymbol{\psi_{0u}} \tag{5.26}$$

$$\boldsymbol{\psi_{fl}} \leq \boldsymbol{\psi}\left(\boldsymbol{x}(t_f), \boldsymbol{u}(t_f), \boldsymbol{p}, t_f\right) \leq \boldsymbol{\psi_{fu}} \tag{5.27}$$

where $\boldsymbol{\psi}$ is the constraint function vector, $\boldsymbol{\psi_{0l}}$ and $\boldsymbol{\psi_{0u}}$ store the minimum and maximum values allowed for the initial boundary constraints and $\boldsymbol{\psi_{0u}}$ and $\boldsymbol{\psi_{fl}}$ store the minimum and maximum values allowed for the final boundary constraints. Note that the expressions above allow the boundary constraints to take a range of values and are thus of *inequality* type. Despite this, it is still possible to enforce a particular value on the boundary constraints, as indicated by the following notation:

$$\boldsymbol{\psi_0} = \boldsymbol{\psi}\left(\boldsymbol{x}(t_0), \boldsymbol{u}(t_0), \boldsymbol{p}, t_0\right) \tag{5.28}$$

$$\boldsymbol{\psi_f}\boldsymbol{\psi}\left(\boldsymbol{x}(t_f), \boldsymbol{u}(t_f), \boldsymbol{p}, t_f\right) \tag{5.29}$$

where $\boldsymbol{\psi_0}$ stores the enforced value for the initial boundary constraints and $\boldsymbol{\psi_0}$ stores the enforced value for the final boundary constraints. The notation just given forms constraints of the *equality* type. In addition to the initial and final boundary constraints, the solution to the dynamic equations must satisfy a set of path constraints while keeping the state and the controls within certain bounds. The notation of these additional constraints in shown in the next three equations:

$$\boldsymbol{g_l} \leq \boldsymbol{g}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{p}, t\right) \leq \boldsymbol{g_u} \tag{5.30}$$

$$\boldsymbol{x_l} \leq \boldsymbol{x}(t) \leq \boldsymbol{x_u} \tag{5.31}$$

$$\boldsymbol{u_l} \leq \boldsymbol{u}(t) \leq \boldsymbol{u_u} \tag{5.32}$$

where $\boldsymbol{g}$ is the path constraint function vector and $\boldsymbol{g_l}$ and $\boldsymbol{g_u}$ are the minimum and maximum values allowed for the path constraints, respectively. Once the dynamics and the relevant constraints are defined, the numerical optimization method will attempt to find the set of controls $\boldsymbol{u}$ that minimize the performance index J given in the following equation:

$$J = \int_{t_0}^{t_f} \boldsymbol{q}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{p}, t\right) dt + \varphi\left(\boldsymbol{x}(t_0), \boldsymbol{x}\left(t_f\right), t_0, t_f, \boldsymbol{p}\right) \tag{5.33}$$

where $\boldsymbol{q}$ is a set of quadrature functions that approximates the derivative and φ are scalar functions that depend only on the initial and final state. Since the above formulation of the performance index includes a time integral and boundary-dependent scalar functions, it is referred as a Bolza problem. Examples of such performance indexes are the trajectory's cross-range, the dispersions at the terminal boundary, the total accumulated heat load, etc.

6 SIMULATION SOFTWARE

The goal of this chapter is to review the different software tools to be used throughout this thesis. First, an overview of the different software packages is given in Section 6.1. Next, Section 6.2 discusses how all the tools and packages are combined into a software architecture. Finally, Section 6.3 covers the verification of such architecture.

6.1 SOFTWARE OVERVIEW

6.1.1. ABORT TRAJECTORY GENERATION & OPTIMIZATION

In accordance with **REQ-19**, the ASTOS tool shall be used to generate the optimal trajectories that comprise the database that feeds the trajectory planner. ASTOS is a commercial package that comprises of multiple optimization methods and a set of environment, vehicle and dynamic models that are combined in a powerful user interface[a]. The purpose of such a tool is to provide a "black box" environment capable of generating optimal trajectories according to a set of input parameters. The ASTOS package has been used in the past to generate optimal trajectories Ancarola (2002); Castellini et al. (2010). Since the focus of the thesis assignment is to design and evaluate an abort guidance system, the use of such readily available software provides significant advantages in terms of development time.

NUMERICAL OPTIMIZERS OF ASTOS

Once the optimal control problem is properly defined, it is fed to an optimizer that will attempt to minimize the established performance index by adjusting the control vector. Optimization methods can be classified under two main categories, depending on the strategy used to find the extremes of the performance index.

- **Indirect Methods:** The performance index is augmented with a set of adjoint equations, which constraint the problem's dynamics. By doing such augmentation, optimal solutions obtained by indirect methods are guaranteed to satisfy the established constraints. This strategy is commonly known as Pontryagin's minimum principle and further details can be found in (Betts, 1998).
- **Direct Methods:** The performance index is reduced at every iteration by determining a search direction in a so called non-linear programming problem. In direct methods, the constraints on the problem's dynamics are enforced in the process of determining the search direction.

In general, indirect methods are able to compute the performance index with a greater accuracy than direct methods. Despite this, indirect methods require a closer initial guess than direct methods for the method to converge. This means that the region of convergence for indirect methods is considerably smaller than for direct methods. Furthermore, indirect methods require good knowledge of the problem dynamic and mathematics of optimization. On the other hand, direct methods have a larger region of convergence and are mathematically simpler. In addition to the direct-indirect classification, numerical optimization methods can be grouped as well depending on the strategy used to compute the performance index and the constraints (Betts, 1998). The following two groups are featured in the ASTOS software:

- **Multiple Shooting Methods:** The performance index and the constraints are obtained by solving an IVP. Once the method is fed with a set of initial conditions and a control history, a numerical method propagates the initial conditions until a desired set of final conditions is met.
- **Collocation Methods:** These methods define multiple nodes throughout the variable space in which the state, the dynamics and the controls are discretized. Given an initial guess for the controls and the

[a] *ASTOS*® Model Library, ASTOS Solutions, [Accessed On: 22-03-2016], `https://www.astos.de/products/astos/model_lib`

state, the a set of quadrature rules are used to correct for the problem dynamics and constraints. The performance index is computed based on the information available throughout the nodes.

Multiple shooting methods are very convenient whenever the control history can be easily defined as a function of some state-derived variable (Betts, 1998). On the other hand, multiple shooting methods suffer from the propagation of errors throughout the optimal trajectory. Furthermore, the use of computationally intensive numerical integrators can lead to unfeasible solving times. Collocation methods do not suffer from these issues, but they benefit significantly from an accurate guess, which is often unknown. The ASTOS tool has a number of optimization methods available, where the name and class of each methods is summarized in Table 6.1.

Table 6.1: Optimization methods of ASTOS.

TROPIC	SOS	PROMIS	CAMTOS
Direct Collocation	Direct Collocation	Direct Multiple Shooting	Hybrid

6.1.2. RE-ENTRY SIMULATOR & ABORT GUIDANCE

Part of the thesis work involves the set-up of a reentry simulator which accurately models the reentry dynamics of SPHYNX. This simulator is then used to test the abort guidance system that ensures the compliance of the requirements specified in Section 2.3. Both the entry simulator and the guidance system are built by implementing into a computer program the theory discussed in Chapters 3 and 4. The development process of such computer program starts by selecting an appropriate programming language.

In the world of scientific computing, the choice boils down to either MATLAB or C++. On the first hand, MATLAB is commonly available for university students and staff, meaning that it benefits from extensive worldwide support. Furthermore, it enables a rapid algorithm deployment, it has many readily-available numerical methods and it has a powerful graphing engine. On the other hand, experience shows that the run-time of MATLAB-based simulations can become unfeasible as the complexity of the simulations increases. Finally, the MATLAB programming environment is a costly proprietary software that is often unavailable to the general public.

Alternatively, the computer program can be developed in C++, which benefits from significantly faster runtimes and the wide availability of open-source libraries. An example of such library is the TU Delft Astrodynamics Toolbox (TUDAT), which offers a series of common verified libraries used in space mission design. Despite its advantages in code execution speed and library availability, the deployment time of algorithms in C++ can prove significantly longer than in MATLAB. This is not considered an issue due to the immediate TUDAT support available directly from its developers at TU Delft. Thus, C++ is selected as the primary programming language for the development of the entry simulator and guidance system programs.

Most of the packages offered in TUDAT that are used are summarized in Table 6.2. Such packages are linked to the particular software modules where they are used and to the section in this report where the theory of such packages is discussed.

6.2 SOFTWARE ARCHITECTURE

The software architecture is represented by a set of blocks which represent either a software routine, input data or output data. The top-level architecture consists of an ASTOS block, three interpolator blocks, a main block and a plotting block, as shown in Figure 6.1. In this architecture, the main block uses the outputs from the ASTOS block and the interpolator blocks to simulate the trajectories flown by SPHYNX given a set of initial conditions. The simulated trajectories are then fed to the plotting block, which interprets the main block's output data.

As explained in Section 6.1.1, the role of the ASTOS block is to provide a large trajectory database that covers sufficient in-orbit abort scenarios. This database is then fed to a mission manager that determines a reference trajectory based on the vehicle state at the time of the abort. Once the reference trajectory is determined, it is fed to the entry simulator which computes SPHYNX's trajectory past the EIP. Note that the trajectory database, the aerodynamic database interpolant and the wind-model interpolant only need to be computed once, meaning that if the main block needs to be run multiple times, computational time can be saved.

Table 6.2: List of readily available TUDAT functions and models.

	Functionality	Usage	Covered
Flight Dynamics	Frame Transformations	*Entry Simulator*	Section 3.2
	State Variable Conversions	Guidance Interface	Section 3.3
	Geodetic Variable Conversions	Guidance Interface	Section 3.4.1
	Aerodynamic Accelerations	State Derivative Function	Section 3.5.1
	Gravitational Accelerations	State Derivative Function	Section 3.5.2
	EOM in Cartesian State Variables	State Propagator	Section 3.5.2
Environment Models	J_2 Central Gravity Model	Gravity Model	Section 3.4.2
	Tabulated US76 Standard Atmosphere	Atmosphere Model	Section 3.4.3
	Elliptical Earth Shape Model	Planet Shape Model	Section 3.4.1
Numerical Methods	Bisection Root-Finder	Trimming Function	Section 5.1.1
	Secant Root-Finder	Radiative Heat-Flux Function	Section 5.1.2
	Multi-Linear Interpolation	SPHYNX Database Interpolation	Section 5.3
		Extremal Trajectory Interpolation	
	Runge-Kutta 4 Integrator	State Propagator	Section 5.2
Miscellaneous	Celestial Constants		Appendix A
	Physical Constants		Appendix A
	Mathematical Constants		
	Unit Conversions		

The mission manager block architecture consists of three sub-routines, as shown in Figure 6.2. First, the EIP selector block feeds the desired EIP to a trajectory subspace selector block, whose role is to identify the set of optimal trajectories that originate near this EIP, as discussed in Section 4.1.2. The EIP is selected manually, although the possibility of selecting a random point that lays within the database coverage is also possible. Once the trajectory subspace is identified, the trajectory interpolator block obtains a reference trajectory based on the trajectory interpolation theory discussed in Chapter 4.

Once the reference trajectory is known, the reentry trajectory simulation can start, whose architecture is given in Figure 6.3. The first step is to establish the vehicle state at the EIP as the current state. Once the required guidance commands are known, the simulation proceeds with the state propagation which computes the vehicle at the future state. In every state derivative call, the current altitude is checked against a 10 km termination condition. If such condition is met, the simulation stops yielding the final flown trajectory. On the contrary, the future state is established as the current state and the whole procedure starts again.

The state propagator is essentially an implementation of the RK4 numerical integrator described in Section 5.2. This integrator shall receive a function that computes the state derivatives using the latest step and guidance commands. Such state derivative function is in essence, a systematic implementation of the equations of motion, as shown in Figure 6.4.

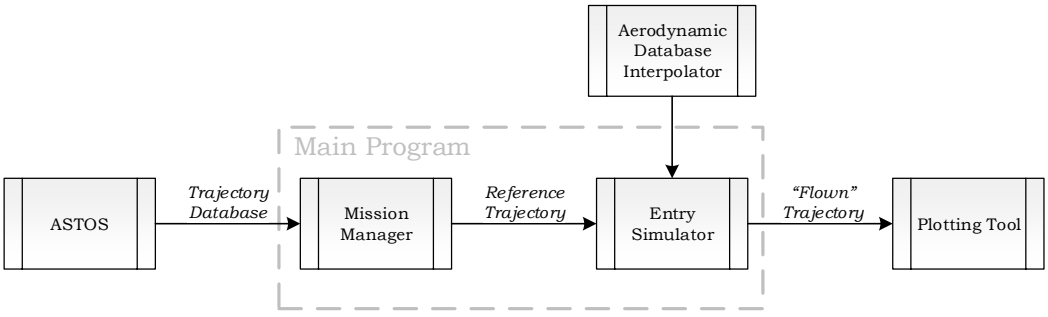

Figure 6.1: Flowchart of the software's top-level architecture.

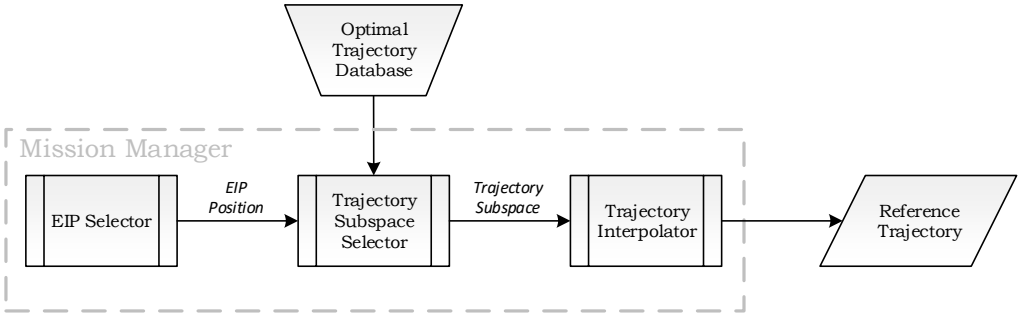

Figure 6.2: Flowchart of the mission manager architecture.

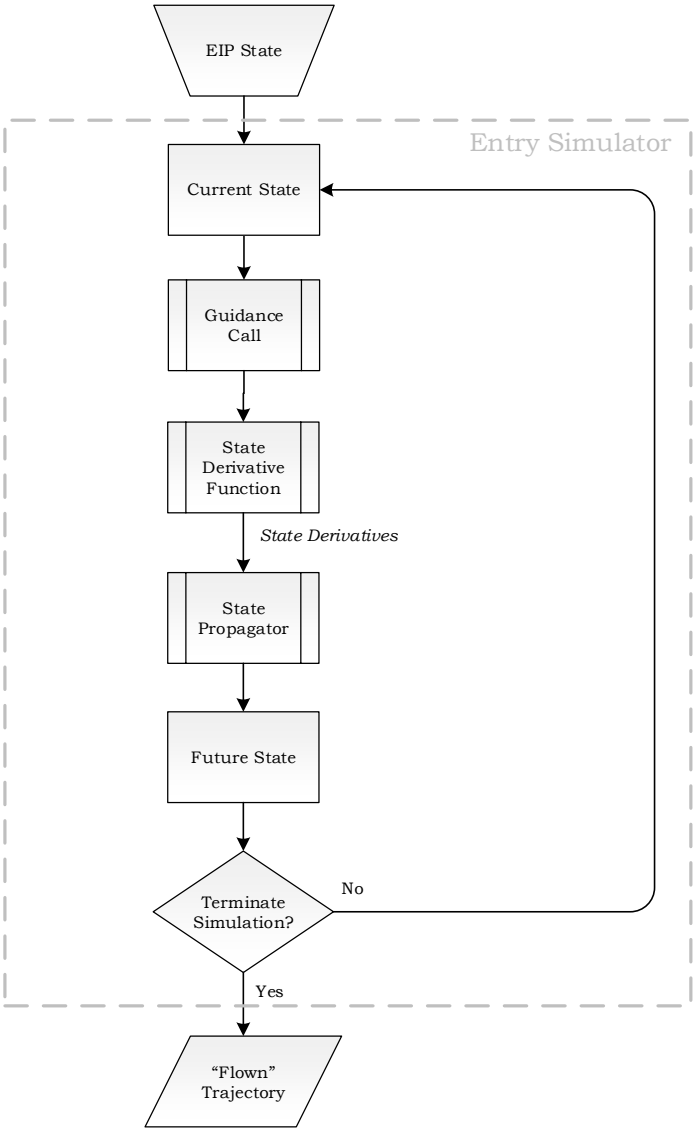

Figure 6.3: Flowchart of the entry simulator architecture.

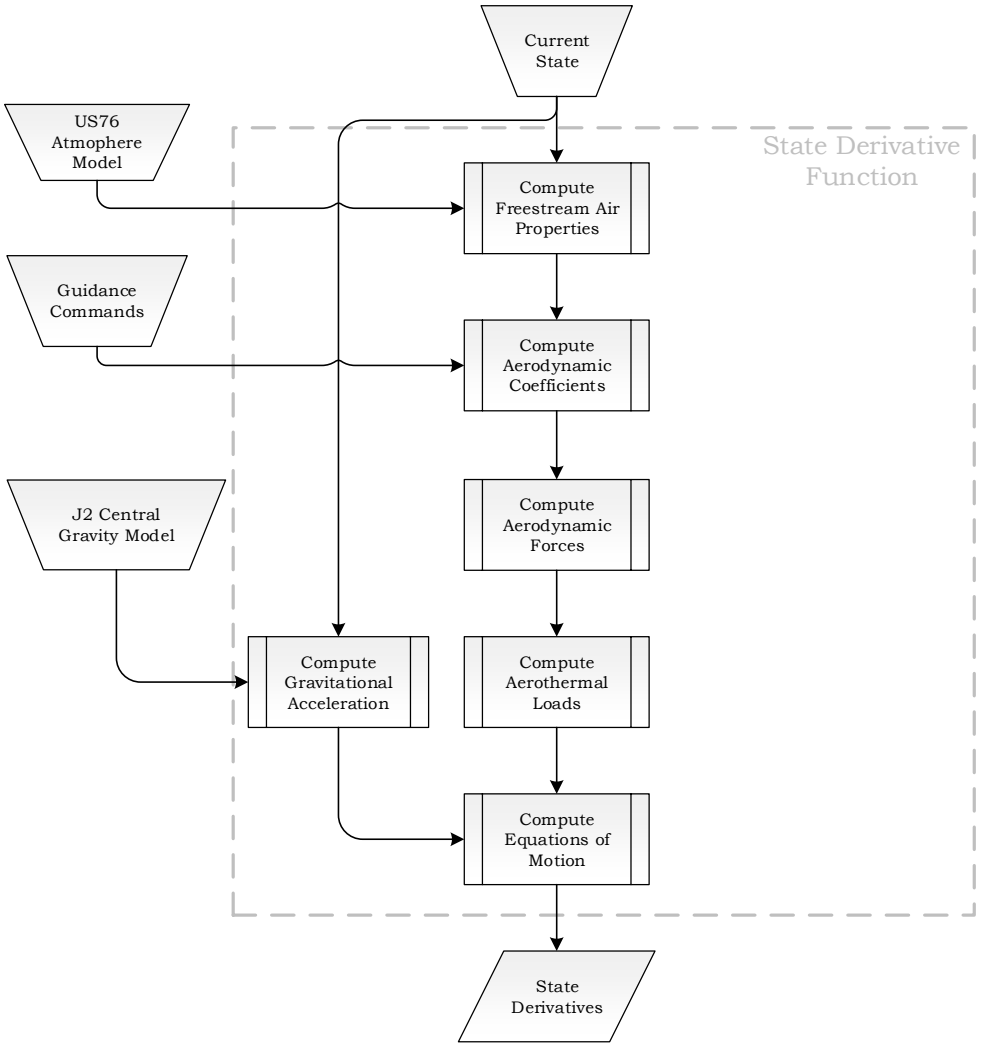

Figure 6.4: Flowchart of the state derivative function.

6.3 SOFTWARE VERIFICATION & VALIDATION

In Section 6.1.2, the need of developing new software as part of the thesis assignment was explained. As part of such process, the developer must ensure that the software performs what it is meant to do and that it satisfies the requirements established by the customer. This idea introduces the concepts of *verification* and *validation*, whose definitions are given below:

- **Verification:** Process to confirm that adequate specifications and inputs exist for any activity, and that the outputs of the activities are correct and consistent with the specifications and input. (ECSS Secretariat, 2009)
- **Validation:** Process to confirm that the requirements baseline functions and performances are correctly and completely implemented in the final product. (ECSS Secretariat, 2009)

Please note that the ASTOS tool used as part of the thesis assignment needs no verification nor validation. Furthermore, any standard libraries used as part of the software development process are expected to be readily verified.

As mentioned above, the goal of the verification process is to ensure that the developed software does what it was set to do. Previous experience showed that in large projects such as the thesis assignment, verification should be performed as an integral part of the software development process and shall be executed in a bottom-up approach. In other words, the verification of any software must start with its most basic constituents and it must end with the higher architectural level. Failure to do so may lead to carrying errors over to

later development stages, that can result in major schedule delays.

The complexity of the verification methods mainly depends on the architecture level being verified. In general, the lower level functions can be verified by testing against analytical solutions, problems with well known solutions or by simple inspection. As the complexity builds up towards the higher level blocks, the verification methods become more specific.

- **Entry Simulator:** The entry simulator should be capable of simulating an entry trajectory to a sufficient degree of accuracy. Note that the core of the entry simulator is readily available in TUDAT. Despite this, a number of additional functionalities were developed which need to be tested:
  - Vehicle Trimming Function:
    1. Check that the trimming function is capable of replicating the expected trimmable area of SPHYNX (European Space Agency, 2016).
    2. Check that the pitching moment coefficient in the computed trimmable area is zero.
  - Aerodynamic Database & Guidance Interface:
    1. Check that the entry simulator is capable of replicating a guided entry using the HORUS vehicle. The reference guided entry is produced using the external simulation by Mooij (1998).
    2. Check that the entry simulator is capable of replicating a guided entry using the SPHYNX vehicle. The reference guided entry is produced using the external simulator of ASTOS.
  - Aerothermodynamic Functions:
    1. Check that the aerodynamic load calculator, the heat flux calculator and the heat load calculator yield the expected output when tested against predictable profiles.
    2. Check that he aerodynamic load calculator, the heat flux calculator and the heat load calculator are capable of replicating the profiles produced by ASTOS.
- **Trajectory Planner:** The trajectory planner should generate an acceptable reference trajectory given a vehicle state.
  - Check that the trajectory planner is capable of selecting the closest neighboring trajectories to the fed EIP.
  - Check that the interpolated reference trajectory resembles the extremals whenever the EIP lays close to such extremals.
- **Trajectory Tracker:** The trajectory tracker should be capable of guiding a reentry vehicle throughout the entry trajectory while ensuring that no path constraints are violated.
  - Check that the trajectory tracker is capable of following a reference trajectory in which no constraints are violated.

6.3.1. ENTRY SIMULATOR VERIFICATION

VEHICLE TRIMMING FUNCTION TEST

The goal of this test is to verify that the function which computes the required body-flap deflection to trim the spacecraft does so correctly. The required body-flap deflection is computed throughout the vehicle's flight envelope, where $M \in [0,30]$ and $\alpha \in [-10,50]$ deg. The computed values are shown in Figure 6.5, which are then compared to those in the preliminary trim analysis provided with the database documentation (European Space Agency, 2016).

Figure 6.5 shows that the trim function is able to compute required body-flap deflection angles within the actuator deflection range of the SPHYNX vehicle. Furthermore, the $\delta_e \in [10,30]$ deg elevator deflection boundaries imposed in the documentation are reflected accordingly in the computed deflection map. Note that the preliminary trim analysis provided in the database documentation includes a region of extrapolated results that is not included in the trim analysis test nor in the vehicle model used throughout the thesis assignment. Thus, any disparities in elevator deflection that exist in the region of extrapolated data are ignored.

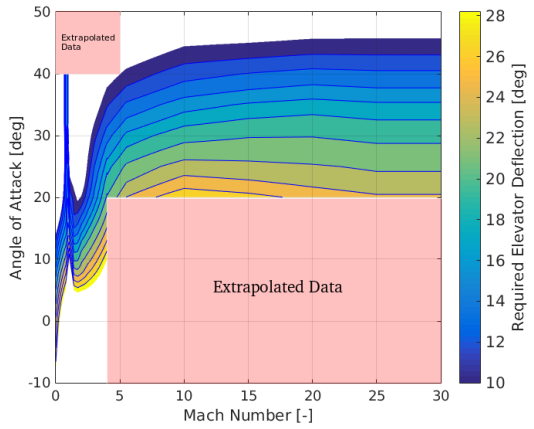

Figure 6.5: Required elevator deflections for trim as computed by the vehicle trimming function.

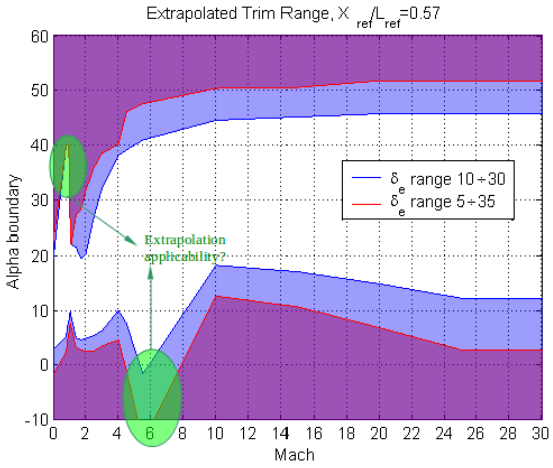

Figure 6.6: Required elevator deflections for trim as established in European Space Agency (2016).

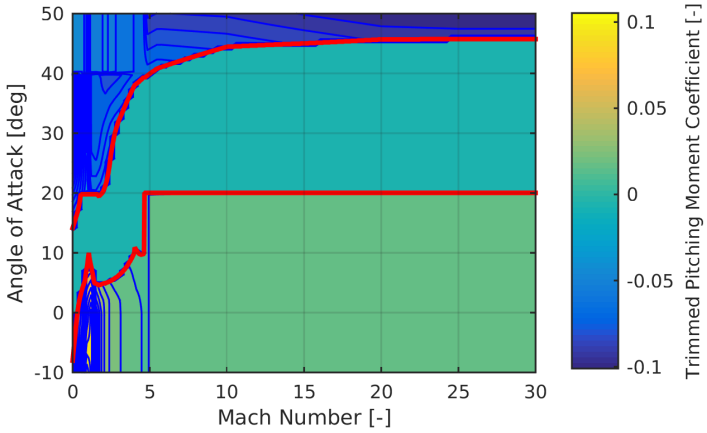

Figure 6.7: Trimmed pitching moment coefficient of SPHYNX in the complete regime of the $\alpha - M$ space.

An additional test is presented in Figure 6.7, where the pitching moment of SPHYNX in a trim-processed database is presented. It becomes evident that in the trimmable area, the pitching moment is zero thus proving the effectiveness of the trimming function.

In summary, the trim body-flap deflection function is able to replicate the results provided in the database documentation, thus yielding the function verified.

GUIDANCE INTERFACE & AERODYNAMIC DATABASE INTERFACE TESTS

The goal of this test is to verify the guidance interface and the aerodynamic database interface. The test is considered passed if the simulator is able to replicate reference guided trajectories generated with external validated simulators. The entry simulator is tested against two different vehicles and two different external simulators, with the goal of thoroughly testing the coded software. The vehicles tested are the SPHYNX vehicle used in this thesis and the HORUS vehicle described in Mooij (1995). The simulated entry trajectories were subjected to the same initial conditions as those provided with the reference trajectories, which are indicated in Tables 6.3 and 6.4. Furthermore, the environment models employed in both simulators are the same, with the exception of the atmosphere model: the external simulator in Mooij (1997) uses an analytical form of the US76 atmosphere model, whereas the TUDAT simulator uses a tabulated form of such model.

Table 6.3: Initial entry conditions for the guidance and aerodynamic database interface tests. Source: (Mooij, 1997)

Vehicle	Altitude	Longitude	Latitude	Groundspeed	Flight-Path Angle	Heading Angle
HORUS	119.96 km	-105.97 deg	-22.06 deg	7438 m/s	-1.42 deg	70.44 deg

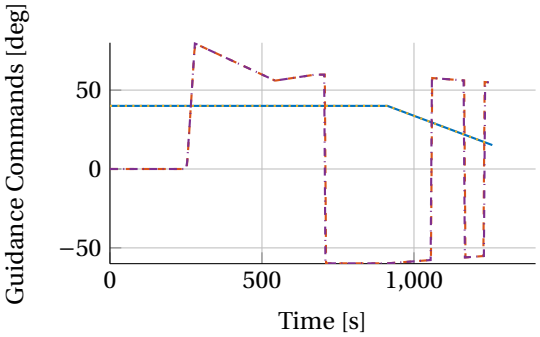

Figure 6.8: Comparison of the reference and simulated commanded angle of attack and bank angle profiles for the HORUS vehicle.

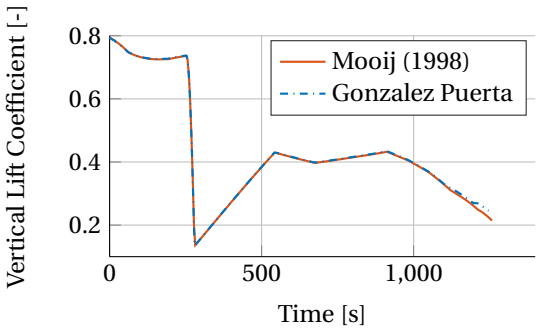

Figure 6.9: Comparison of the reference and simulated vertical lift coefficient profiles for the HORUS vehicle.

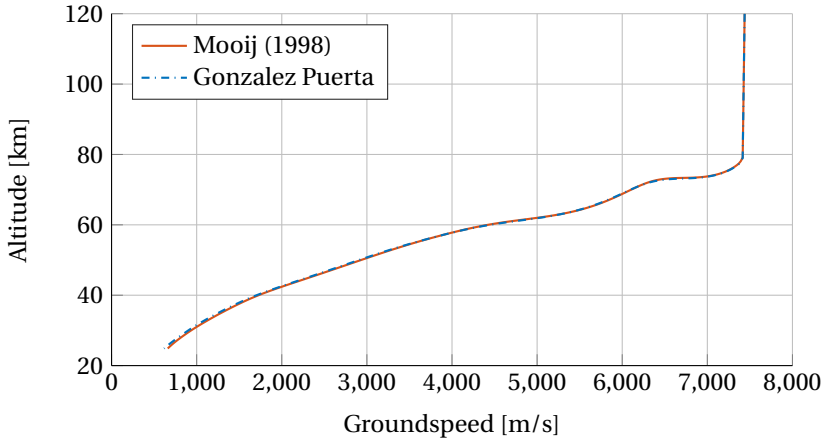

Figure 6.10: Comparison of the reference and simulated altitude profiles for the HORUS vehicle.

Figure 6.8 shows that the guidance interface is able to track the angle of attack and bank angle commands logged in the reference trajectory, thus verifying that such interface was implemented correctly. In addition, Figure 6.9 shows that the vertical lift coefficient produced by the simulation matches that of the reference trajectory, which verifies the implementation of the aerodynamic interface. Finally, Figure 6.10 serves as definite proof of the correct implementation of the guidance and aerodynamic database interfaces, since the altitude-groundspeed profile generated by the simulator is nearly identical to the one provided in the reference.

Although the simulated groundspeed-altitude profile is very close to the reference, minor disparities in altitude and groundspeed begin to accumulate early in the trajectory, as shown in Figure 6.11. The maximum absolute error in altitude is found to be 1075 m and is obtained at the end-point of the trajectory. In the case of groundspeed, the maximum absolute error is found to be 101 m/s. This error accumulation throughout the flight is reflected in the simulated groundtrack, whose absolute error is shown in Figure 6.12. Since the guidance interface and the aerodynamic database interface have been verified, the discussed disparities are likely to be related to the use of a tabulated atmosphere model instead of the analytical model employed in the external simulator. In conclusion, these disparities are considered acceptable, as long as they are quantified.

The guidance and the aerodynamic database interfaces are tested further by comparing the produced trajectories against the those produced by the verified simulator of ASTOS. The entry conditions of this test are presented in Table 6.4. Study of Figures 6.13 and 6.14 shows that the Entry Simulator is capable of replicating the guidance commands scheduled by ASTOS, thus verifying the guidance interface module. Furthermore, Figure 6.19 shows that the trajectory flown is identical to the one produced by ASTOS, thus verifying the aerodynamic database interface.

AEROTHERMODYNAMICS UNIT TEST

The goal of this test is to verify the functions that compose the aerothermodynamics module of the simulator. Such module contains four independent functions: an aerodynamic load calculator, an adiabatic wall temper-

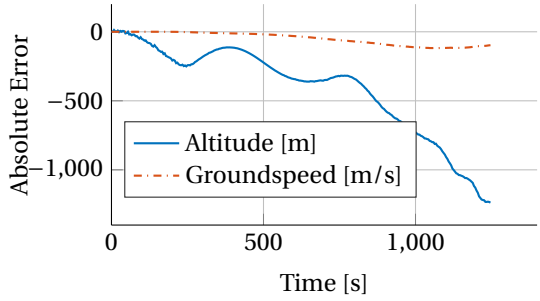

Figure 6.11: Altitude and groundspeed absolute error profiles generated by the simulation when compared against the HORUS entry.

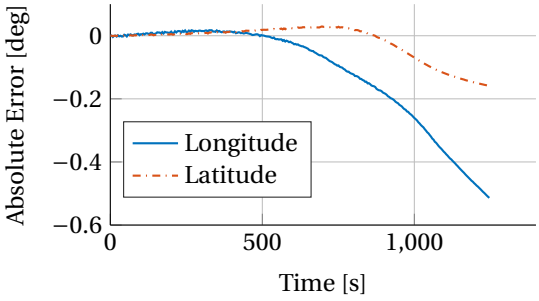

Figure 6.12: Longitude and latitude absolute error profiles generated by the simulation when compared against the HORUS entry.

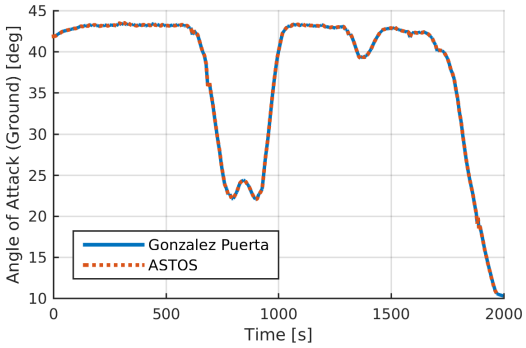

Figure 6.13: Comparison of the angle-of-attack profiles generated by the Entry Simulator and ASTOS.

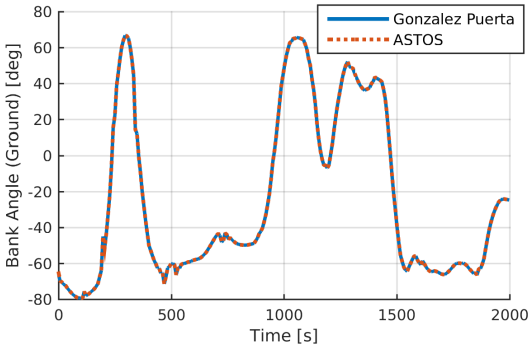

Figure 6.14: Comparison of the bank-angle profiles generated by the Entry Simulator and ASTOS.

Table 6.4: Initial entry conditions for the guidance interface test, aerodynamic interface test and aerothermodynamics unit test.

Vehicle	Altitude	Longitude	Latitude	Groundspeed	Flight-Path Angle	Heading Angle
SPHYNX	120 km	-130.0 deg	18.0 deg	7640 m/s	-1.28 deg	40.3 deg

Table 6.5: Vehicle parameters used in the aerothermodynamics unit test.

Vehicle Mass	Reference Area	Nose Radius	Wall Emissivity	C_D	C_S	C_L
1000 kg	10 m^2	1.0 m	0.85	1.0	1.0	1.0

ature calculator, a heat flux calculator and a heat load calculator. The functions are tested by exploring the behaviour of their outputs when subjected to predictable test cases. The calculators require a set of vehicle reference parameters, which are summarized in Table 6.5.

The aerodynamic load calculator is tested by computing such load against an increasing groundspeed test profile. The calculation assumes the constant aerodynamic coefficients specified in Table 6.5 and a constant air density of 1.225 kg/m^3. The resulting aerodynamic load profile shown in Figure 6.15 displays a quadratic behavior, as expected from Equation (3.74).

The adiabatic wall temperature calculator is tested by computing such temperature against an increasing Mach number test profile. The calculator assumes a constant ratio of specific heats of 1.4 and approximates the recovery factor using the laminar boundary layer over a flat plate theory explained in (Anderson Jr., 2006c). This provides a minor overestimation of the adiabatic wall temperature, since the recovery factor decreases with increasing Mach number (Anderson Jr., 2006d). The adiabatic wall temperature profile shown in Figure 6.16 shows that such temperature increases with Mach number, which is expected due to the higher kinetic energy of the flow. Furthermore, Figure 6.16 indicates that for a stagnant flow, the adiabatic wall temperature equals the flow's free-stream temperature.

The heat flux calculator is tested by computing such flux over an increasing Mach number of profile, while

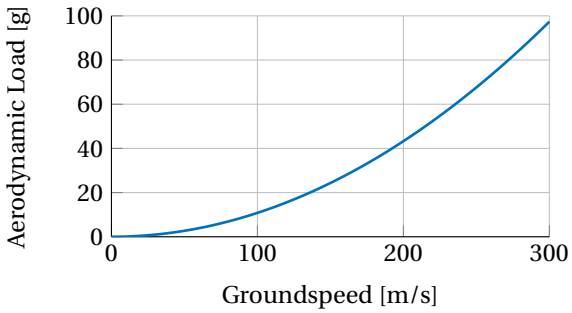

Figure 6.15: Aerodynamic load profile obtained using constant aerodynamic coefficients and a constant air density of 1.225 kg/m^3.

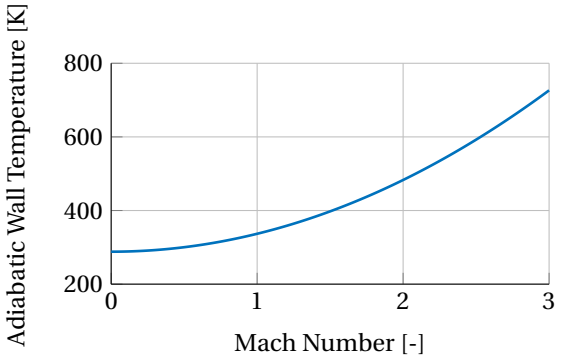

Figure 6.16: Adiabatic wall temperature profile obtained using a constant free-stream temperature of 288.15 K, a constant ratio of specific heats of 1.4 and a recovery factor of 0.845.

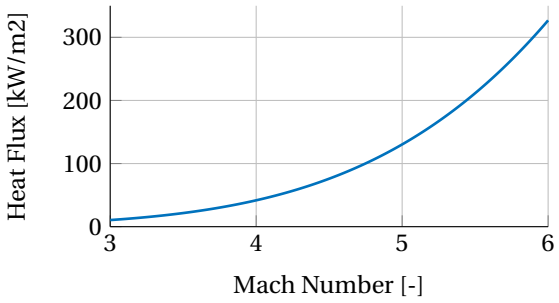

Figure 6.17: Heat flux profile obtained using a constant air density of 1.225 kg/m^3, a constant speed of sound of 340.294 m/s, a constant ratio of specific heats of 1.4, a recovery factor of 0.845 and a heat-flux constant of 5.28137e-5.

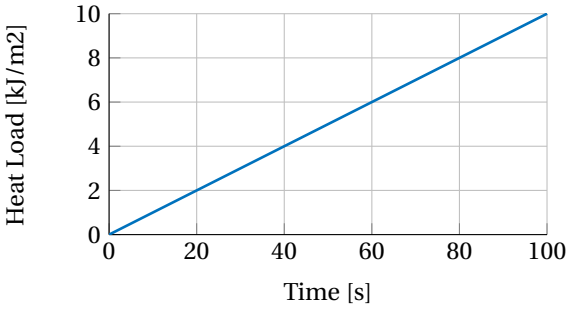

Figure 6.18: Heat load profile obtained using a constant heat flux of 100 W/m^2.

keeping the air density, the speed of sound, the ratio of specific heats and the recovery factor constant. Figure 6.17 shows with an increasing Mach number, the heat flux increases, which is expected as a result of the increasing kinetic energy of the flow.

The heat load calculator is tested by computing such load over time and using a constant heat flux profile. Figure 6.18 shows a linear increase in heat load over time, which is expected as the heat flux is assumed constant. In this case, the analytical expression for the heat load is simply $Q = \dot{q}t$. This implies that the slope of the produced heat load profile must equal the value of the established constant heat flux. Figure 6.18 indicates that the heat load calculator is able to replicate such behavior, since the slope of tested profile matches the incoming constant heat flux of 100 W/m^2.

In addition to the tests above, the aerothermodynamics module was verified by comparing simulated load profiles for a SPHYNX vehicle entry against the respective profiles computed with the ASTOS software. The initial conditions used for the following tests are summarized in Table 6.4. Figure 6.19 shows that the simulated altitude-groundspeed profile generated by the Entry Simulator closely matches the profile generated by ASTOS. Thus, it is expected that the aerodynamic load, heat flux and heat load profiles match as well. Indeed, Figures 6.20 to 6.22 show that the Aerothermodynamics Module is able to replicate the aerothermal load profiles given by ASTOS with a small error. In summary, the following maximum relative errors were obtained: 1.32% for the aerodynamic load profile, 1.98% for the heat flux profile and 1.18% error for heat load profile.

In summary, Figures 6.15 to 6.18 show that the functions included in the Aerothermodynamics Module behave as expected when computing predictable profiles. Furthermore, Figures 6.20 to 6.22 show that such functions generate load profiles with an acceptable error when subjected to a full re-entry simulation. All in all, the results shown here render the Aerothermodynamics Module verified.

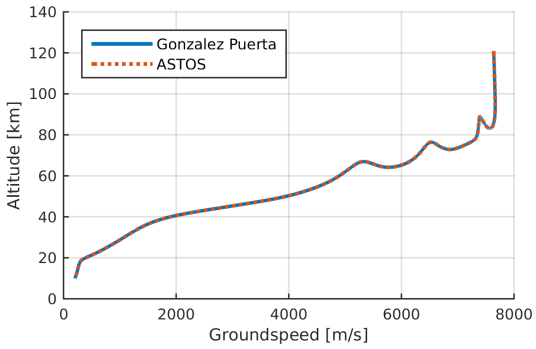

Figure 6.19: Comparison of the altitude-groundspeed profiles generated by the Entry Simulator and ASTOS.

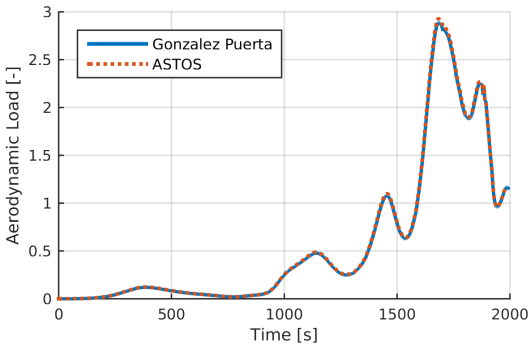

Figure 6.20: Comparison of the aerodynamic load profiles generated by the Entry Simulator and ASTOS.

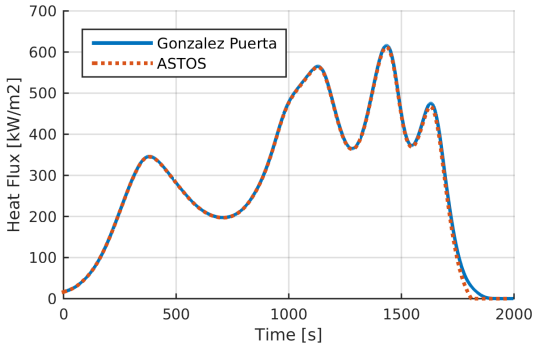

Figure 6.21: Comparison of the heat flux profiles generated by the Entry Simulator and ASTOS.

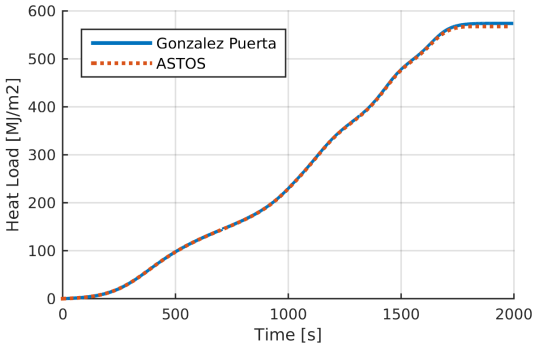

Figure 6.22: Comparison of the heat load profiles generated by the Entry Simulator and ASTOS.

6.3.2. TRAJECTORY PLANNER VERIFICATION

In this section, the AMPI-based trajectory planner presented in Section 4.1 is tested. This test is divided into three independent tests.

First, the trajectory subspace selector is tested. Such software module is fed with arbitrary EIPs that span the whole parameter space. The trajectory database selector was capable of selecting the four closest extremals to the EIP received. Whenever one or more extremals were missing as a consequence of the EIP being outside the database, the trajectory subspace selector would throw an error.

Second, the trajectory interpolation is tested by inspecting an interpolated trajectory that was produced using predictable extremal profiles. Note that the extremals supporting the interpolation done in this test have no physical meaning and do not belong to the trajectory database. The four extremal trajectories in this test are defined as follows:

$$e_{TL,test}(t) = t^{1.5} \tag{6.1}$$

$$e_{TR,test}(t) = t^{2.0} \tag{6.2}$$

$$e_{BL,test}(t) = 1.5 \tag{6.3}$$

$$e_{BR,test}(t) = 2.0 \tag{6.4}$$

where $e_{i,test}$ is the value of the extremal i and t is an arbitrary independent variable. The test performed here compares the results produced by the AMPI interpolator against a regular bi-variate interpolation performed using a proven MATLAB routine. Figure 6.23 shows that the produced interpolant by the AMPI algorithm is identical to the one produced by MATLAB as long as the independent variables have equal lengths. On the other hand, if the independent variables are different in length, the AMPI algorithm is able to capture such difference by extending the interpolation along the independent variable as shown in Figure 6.24. As expected, altering the length of one of the extremals impacts the value of the interpolant, as compared to the regular bi-variate interpolation performed by MATLAB.

Finally, the trajectory planner is tested by feeding an EIP that lays on one of the extremal trajectories of the

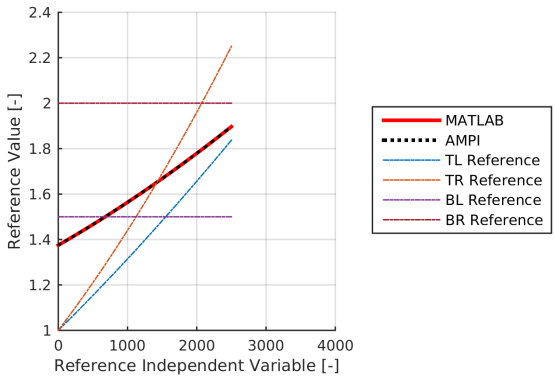

Figure 6.23: Example interpolation using extremals in which the reference independent variables have equal lenghts.

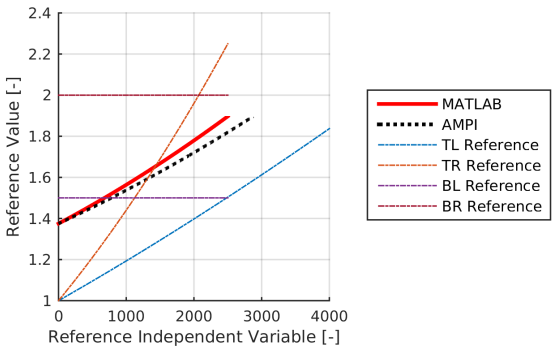

Figure 6.24: Example interpolation using extremals in which the reference independent variables have different lenghts.

database. This test is considered passed if the resulting interpolated trajectory resembles the extremal trajectory. Indeed, Figure 6.25 shows that the trajectory planner is capable of replicating the top-left extremal as long as the fed EIP lays over such extremal. Note that this final test also proves that correct working of the database subspace selector.

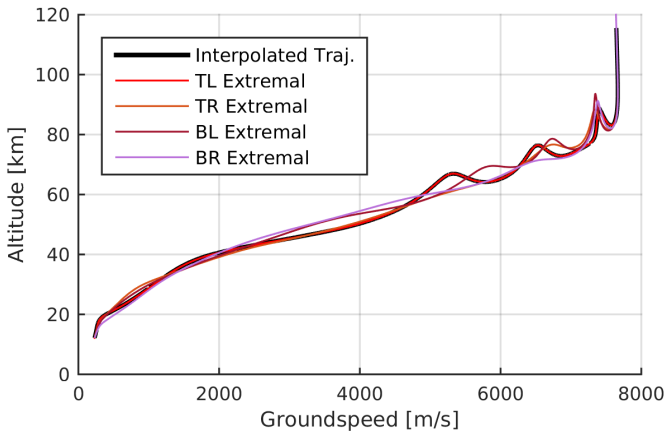

Figure 6.25: Interpolated altitude-groundspeed profile given an EIP that lays on the TL extremal.

6.3.3. TRAJECTORY TRACKER VERIFICATION

The goal of this test is to verify that the LQR tracker presented in Section 4.2 is capable of handling entry point errors effectively. To test this, a reference trajectory is fed produced and a Monte Carlo campaign with 1000 cases is run. Note that the reference trajectory used in this test is provided directly from ASTOS and no interpolation is performed. Doing otherwise would introduce artificial errors in the reference trajectory, thus invalidating the test. The entry conditions used in this test are given in Table 6.4. The characteristics of the applied dispersions are listed in Table 6.6.

Figure 6.26 reveals that the LQR tracker is capable of following the altitude-groundspeed profile in most of the cases, with the exception of a few cases where the deviations from such profile are more pronounced. This leads to a marginal violation of the heat flux constraint in six trajectories, as indicated by the overshoot values given in Table 6.7. Despite this, in all the other trajectories the LQR tracker ensures that the aerodynamic load and heat flux constraints are satisfied, as as shown in Figures 6.28 and 6.29. Although the performance of the LQR is accepted in its current state, the robustness of the system could be improved by further tuning of the guidance gains or changing the tracked variable from groundspeed and traveled range to groundspeed and pseudo altitude, as done in (Mooij, 2017). Such pseudo-altitude is a combination of the altitude and the flight-path angle, thus allowing direct tracking of both variables. In the current state of the system, the flight-path angle is tracked indirectly using the traveled range, as shown by Equation (4.32).

Figure 6.27 shows how all trajectories are able to fly towards the designated landing site. In addition, in those

Table 6.6: Monte Carlo dispersion used to test the LQR trajectory tracker.

Variable	Dispersion Type	Mean	Min/Max	3σ
EIP Altitude	Gaussian	0 m		150 m
EIP Longitude	Uniform		±0.1 deg	
EIP Latitude	Uniform		±0.1 deg	
EIP Groundspeed	Gaussian	0 m/s		20 m/s
EIP Flight Path Angle	Gaussian	0 deg		0.1 deg
EIP Heading Angle	Gaussian	0 deg		0.01 deg
Aerodynamic Coefficients	Gaussian	0		10 %
Vehicle Mass	Uniform		±1 %	
Air Density	Gaussian	0		10 %

trajectories where the heading error exceeds the HEDB limit, a bank reversal is commanded that corrects the targeted heading, thus proving the effectiveness of the lateral guidance. Figure 6.35 shows that the LQR tracker struggles to meet the 10 km distance requirement imposed by **REQ-S-06**, where more than half of the trajectories exceed such value. Again, this issue could be tackled by changing the tracked variables or performing a further tuning of the guidance gains. Despite this, Figures 6.32 and 6.33 show that the LQR tracker is capable of bringing the vehicle to the required vertical speed and Mach number ranges as specified by **REQ-14**. All the trajectories exhibit vertical speeds below the 140 m/s limit and a Mach numbers in the [0.0, 1.0] range.

Figures 6.30 and 6.31 show that the integrated angle-of-attack and bank-angle efforts are well bounded and no trajectories exhibit an excessive control effort. Finally, Figure 6.34 shows that the LQR test produced maximum heat-load values normally distributed about the nominal value of 573.9 MJ/m^2.

Table 6.7: Summary of the constraint compliance results obtained in the LQR tracker test.

Nr. $n_{L,max}$	**Mean** $n_{L,max}$ **[-]**	**Max.** $n_{L,max}$ **[-]**	**Nr.** q_{max}	**Mean** q_{max} **[kW/m^2]**	**Max.** q_{max} **[kW/m^2]**
0	N/A	N/A	6	702.5	730.8

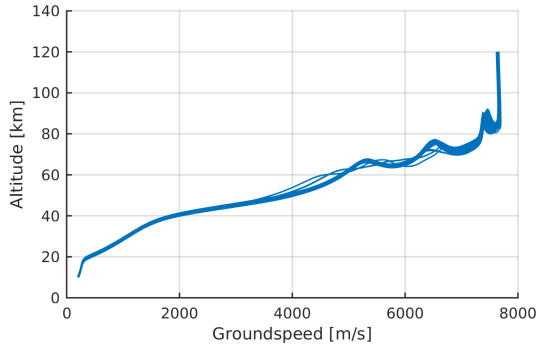

Figure 6.26: Monte Carlo altitude-groundspeed profiles used in the LQR tracker test.

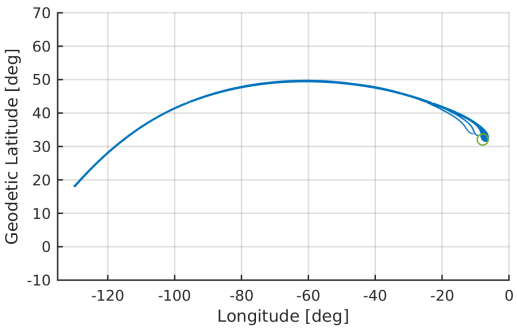

Figure 6.27: Monte Carlo trajectory groundtracks used in the LQR tracker test.

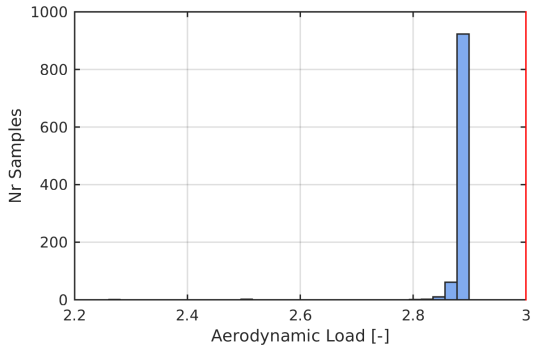

Figure 6.28: Histogram of the aerodynamic load as produced by the LQR tracker test. Required value: 3.0 g

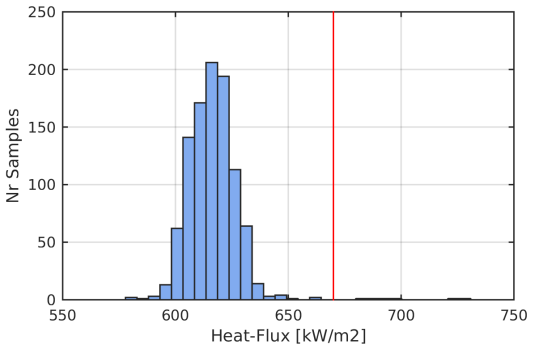

Figure 6.29: Histogram of the heat-flux as produced by the LQR tracker test. Nominal value: 670 kW/m^2

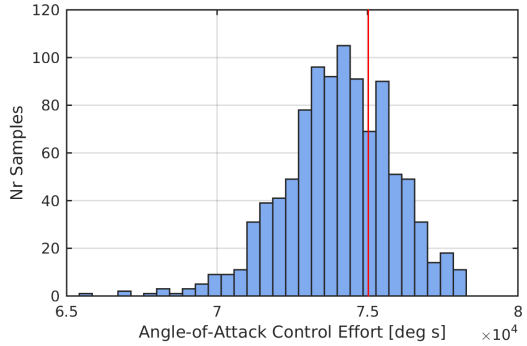

Figure 6.30: Histogram of the angle-of-attack effort as produced by the LQR tracker test. Nominal value: 75,028 deg s

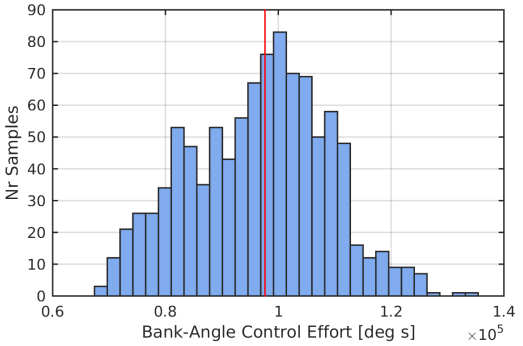

Figure 6.31: Histogram of the bank-angle effort as produced by the LQR tracker test. Nominal value: 97,672 deg s

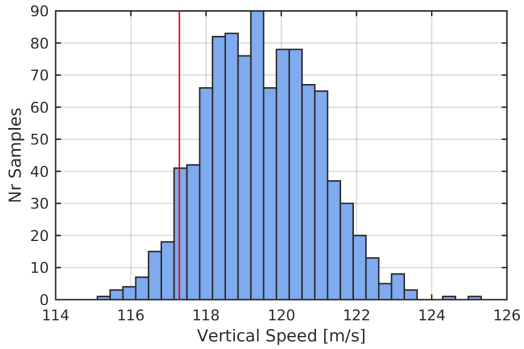

Figure 6.32: Histogram of the final vertical speed as produced by the LQR tracker test. Nominal value: 117.2 m/s

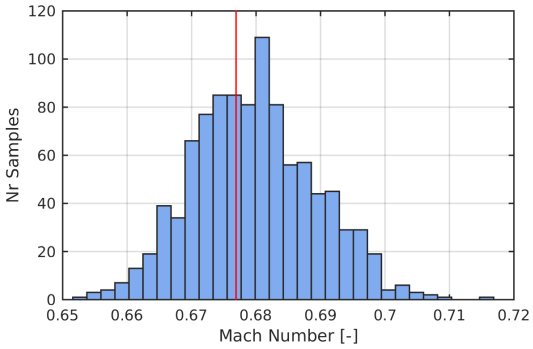

Figure 6.33: Histogram of the final Mach number as produced by the LQR tracker test. Nominal value: 0.677

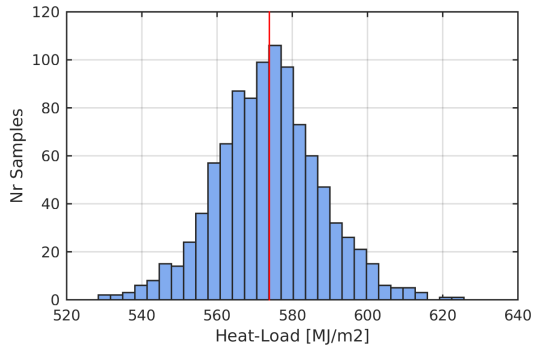

Figure 6.34: Histogram of the final heat load as produced by the LQR tracker test. Nominal value: 573.9 MJ/m^2

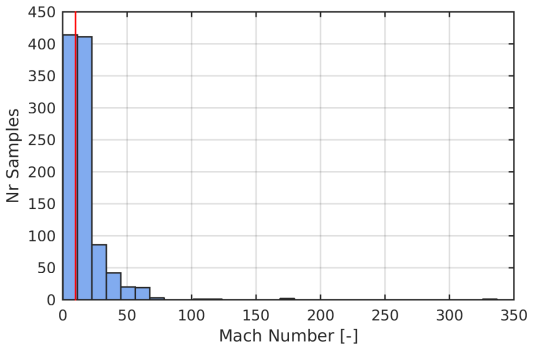

Figure 6.35: Histogram of the final distance to target as produced by the LQR tracker test. Required value: 10 km

7 REACHABILITY STUDY

In the framework of entry mission planning, a reachability study can be defined as the process of determining the landing footprint of a vehicle. Performing a thorough reachability study is essential to determine the possible abort opportunities of an entry vehicle. Most importantly, the results of such a study explicitly conforms the trajectory database required by the AMPI trajectory planner discussed in Chapter 4.

This chapter begins with Section 7.1 describes and justifies the configuration of the software used to generate the trajectory database. Section 7.2, summarizes the findings of the reachability study. Finally, Section 7.3 accounts for the limitations of this study and how these may impact the performance of the abort guidance system.

7.1 ASTOS SCENARIO SET-UP

The ASTOS scenario is composed by a number of settings that together describe a problem to be solved. In the scope of the present reachability study, such problem can be formulated into the following research question:

What are the attitude controls required to steer the re-entry vehicle from an arbitrary entry point to one of the designated emergency landing sites, while ensuring that the trajectory constraints are satisfied?

The scenario settings can be classified into four distinct groups: environment models, vehicle models, grid settings and optimization settings. Making the correct choice for such settings has a tremendous impact in obtaining a good number of feasible sample trajectories. In fact, as will be explained later in the present chapter, making the wrong choices may limit the scope of the trajectory database, potentially degrading the overall performance of the abort guidance system.

7.1.1. ENVIRONMENT & VEHICLE MODEL SETTINGS

As described in Section 3.4, the environment model settings assemble the framework that enables the solution of the equations of motion. For the purpose of coherence, the environment model settings defined in the ASTOS scenario must match those employed in the entry simulator described in Section 3.4. Doing otherwise would lead to differences in the reentry flight dynamics, ultimately invalidating the trajectory database. A summary of the environment model settings is given in Table 7.1. Please note that the trajectories computed in this reachability study are not influenced by wind, since it is not considered in this thesis.

Table 7.1: ASTOS Environment Model Settings.

Planet Shape	Gravity Field	Atmosphere	Wind
Ellipsoid	J2 Term Only	US76 Standard Atmosphere	No Wind

The vehicle model settings used in ASTOS contain information about the SPHYNX aerodynamics and dimensions. Similarly to the environment models, the settings used in ASTOS must match those in the simulator to have a coherent database. Such settings are thoroughly described in Section 2.4 and will not be repeated here.

7.1.2. CONSTRAINT SETTINGS

In every optimization problem, a number of constraints needs to be set. These constraints limit the values that a control, state or derived dependent variable can take. Optimization constraints not only define the nature of

the problem to be solved, but also narrow the solution space to a number of feasible solutions. As mentioned in Section 5.4, there are three main types of constraints: initial boundary constraints, final boundary constraints and path constraints. In the problem at hand, the optimization constraints are mainly imposed by the system requirements established in Section 2.3. In addition, new constraints are added to enforce the limitations of SPHYNX and to ensure that the EIPs tested are the result of a de-orbit burn at the operational orbit studied. All the imposed constraints are listed below, along with the motivation behind them. Table 7.2 summarizes all the constraints along with their type, origin and imposed values.

- **Entry Altitude, Latitude & Longitude:** These equality constraints specify the position of SPHYNX at the EIP. By systematically changing the latitude and longitude values, the vehicle entry windows can be studied. This is discussed in more detail in Section 7.1.5.
- **Entry Groundspeed, Flight-Path Angle, & Heading Angle:** Similarly, these equality constraints specify the velocity of SPHYNX at the EIP. While the initial altitude and flight-path angle are fixed, the initial heading angle depends on the chosen initial latitude as discussed in Section 7.1.5. This is done to fully capture the nature of the descent orbit.
- **Max. Distance to Target:** This constraint defines the terminal area around the landing site under study. Since SPHYNX must lay inside such area, this is set as an inequality constraint. This constraint specifies the latitude and longitude at which the terminal area is centered.
- **Terminal Altitude & Mach Number Ranges:** These inequality constraints limit the allowable altitude and groundspeed of SPHYNX at the terminal area.
- **Max. Terminal Vertical Velocity:** This inequality constraint limits the maximum allowable sink rate of the vehicle at the terminal area.
- **Max. Heat Flux Density & Max. Aerodynamic Load:** These inequality constraints limit the aerothermodynamic loads that the SPHYNX vehicle can experience during re-entry.
- **Max. Flight Path Angle:** This constraint is artificially imposed to limit the phugoid motion of the vehicle in the lower atmosphere. A positive value is allowed to enable solutions that use a skip entry approach to achieve more downrange.
- **Top & Bottom $\alpha - M$ Ranges:** These constraints enforce the limits of the aerodynamic database. This is done such that the output trajectories are trimmable and lay within the a validated region of the database, as discussed in Section 2.4.1.
- **Max. / Min. Bank-Angle:** The maximum allowable bank-angle of SPHYNX is set to ±85 deg to avoid entry trajectories in lift-down configuration.

7.1.3. OPTIMIZATION SETTINGS

Once all the environment models, vehicle models and optimization constraints have been set up, an optimizer needs to be configured. The first step is to set up the performance index to be minimized by such optimizer. In this thesis, the performance index is selected in such a way that the angle of attack and bank angle controls are as smooth as possible. This was found necessary for the trajectory interpolation to work, since doing otherwise would results in discontinuous control profiles at the extremals that are too dissimilar from each other. Furthermore, discontinuous control profiles at the extremals would introduce artificial oscillations at such discontinuities. To achieve such smoothing, the performance index is defined as the integral of control derivatives over time, as shown in the following equation:

$$J_{sm} = \int_{t_0}^{t_f} w_{sm} \boldsymbol{q}(\boldsymbol{u}(t))\, dt \tag{7.1}$$

where J_{sm} is the trajectory smoothing performance index and w_{sm} is the trajectory smoothing factor. In this thesis, the trajectory smoothing factor is set to $1.0 \cdot 10^{-4}$ and was found sufficient to produce smooth trajectories for the interpolation to succeed.

As discussed in Section 6.1.1, ASTOS offers four optimization methods: TROPIC, SOS, PROMIS and CAMTOS. By systematically launching multiple optimization tasks with varying entry conditions, it was determined that CAMTOS is the optimization scheme with the largest region of convergence for the problem at hand, despite it

Table 7.2: Summary of all the optimization constraints employed in the reachability study.

ID	Constraint	Value	Type	Sub-Type	Origin
1	Entry Altitude	120 km	EQ	IB	REQ-S-07
2	Entry Latitude	Variable	EQ	IB	Orbit Geometry
3	Entry Longitude	Variable	EQ	IB	Orbit Geometry
4	Entry Groundspeed	7640 m/s	EQ	IB	REQ-13
5	Entry Flight-Path Angle	-1.28 deg	EQ	IB	REQ-13
6	Entry Heading Angle	Variable	EQ	IB	Orbit Geometry
7	Max. Distance To Target	10 km	IQ	FB	REQ-S-05
8	Terminal Altitude Range	10 - 13 km	IQ	FB	REQ-14
9	Terminal Mach Number Range	0.0 - 1.0	IQ	FB	REQ-14
10	Max. Terminal Vertical Speed	-140 m/s	IQ	FB	REQ-14
11	Max. Heat Flux Density	670 kW/m^2	IQ	Path	REQ-S-03
12	Max. Aerodynamic Load	3.0	IQ	Path	REQ-S-01
13	Max. Flight Path Angle	1.0 deg	IQ	Path	Artificial Constraint
14	Top $\alpha - M$ Boundary	Variable	IQ	Path	European Space Agency (2016)
15	Bottom $\alpha - M$ Boundary	Variable	IQ	Path	European Space Agency (2016)
16	Max. Bank Angle	85 deg	IQ	Path	Artificial Control Limit
17	Min. Bank Angle	-85 deg	IQ	Path	Artificial Control Limit

being somewhat slower than the other methods. Furthermore, CAMTOS proved to be the only method that successfully enforced the $\alpha - M$ constraints, which are critical to ensure that the obtained solutions are bounded to the limits of the aerodynamic database. During the multiple optimization runs launched during the reachability study, CAMTOS achieved feasible solutions within 50 iterations. For this reason, maximum number of optimization iterations was limited to 50 with the goal of saving computational power.

7.1.4. GRID SETTINGS

All optimization methods require the set-up of a grid prior any further tasks are performed. The aim of this grid is to discretize the time vector into a number of points. In collocation methods, such points are referred to as collocation nodes. Each of these nodes indicates a time instant at which the controls and the states must be computed. During every optimization iteration, a new set of controls is computed such that the specified performance index is minimized (or maximized).

As one may expect, the settings of such a grid have a major influence in the performance of the optimizer. Grids are characterized by the number of collocation nodes chosen and the spacing between such nodes. On the one hand, a coarse grid may result in a not-fully-converged solution that presents inconsistencies in the dynamics and a non-optimized performance index. On the other hand, a fine grid may lead to an extremely long computational time or a solution that does not converge at all. Furthermore, the grid spacing should be reduced at those areas where the control space is limited or a path constraint is present.

For the problem at hand, it was found that the optimal number of collocation nodes is 500. A higher number of nodes simply lead to unfeasible computational times and non-converged solutions. Using less than 500 nodes resulted in converged solutions that deviated too far from the landing site. The reader may question how the solution could have converged, if the distance to the landing site was strictly set as a final boundary constraint, as described in Section 7.1.2. This is due to the nature of the CAMTOS optimization method: once the optimal controls are found, the trajectory is integrated using a multiple shooting method. The result of such integration determines the states and constraint values through time. Since multiple shooting methods work by solving new initial value problems at every collocation node, the resulting trajectory may very well satisfy the boundary constraints but it will do so at the cost of a non-realistic trajectory that presents discontinuities.

This method differs from the integrator employed in the Entry Simulator, which is a single shooting method. In single shooting, the entry conditions are simply propagated through time in a single arc, thus leading to realistic continuous trajectories. Fortunately, ASTOS allows the user to re-simulate the trajectory using a single shooting method, but it requires the grid to be sufficiently dense for the solution to match the "performance" of multiple shooting. The nature of this problem is depicted in Figure 7.1.

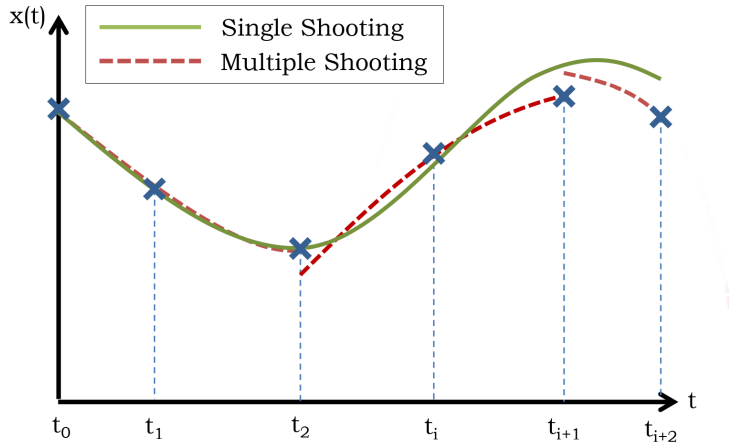

Figure 7.1: Comparison of single shooting integration vs. multiple shooting integration.

7.1.5. BATCH-ANALYSIS SET-UP

Once all the settings that define the problem have been set-up, an optimization task can be launched. The solution obtained after such a task is completed contains the attitude commands that steer the vehicle from a given entry point to a chosen landing site, in other words, a single trajectory.

At this point, the reader is asked to take a step back to put the problem at hand into perspective: the goal of the reachability study is to compute worldwide entry windows for all of the eight landing sites. Furthermore, the vehicle may either be on an ascending leg or a descending leg at the EIP, thus doubling the number of trajectories. This implies that a maximum of 16 optimal trajectories may be found for every latitude and longitude combination that the vehicle can take at the EIP. To illustrate this, assume that the studied EIPs originate from a 52 deg operational orbit, thus bracketing the possible entry latitudes to the [-52,52] range. The range of possible entry longitudes spans from -180 deg to 180 deg. For a coarse database parameter spacing of 10 deg and an average computational time of 8 minutes per trajectory on a Core^{TM} i7 5700HQ processor running at 2.70 GHz, the total computational time of the whole reachability study would amount to at least 33 full days. This figure does not include any post-processing tasks and does not account for the time lost when attempting to optimize an entry point where no solutions are found. Fortunately, ASTOS has the capability of systematically launching optimization tasks in a batch, which is crucial for the problem at hand. In this way, unattended optimization tasks can be launched overnight and the time wasted as a consequence of user interaction with the software is minimized. To aid in the completion of this reachability study, ESA allowed the author to run the optimization tasks on a high-performance Xeon®E5-2687W processor clocked at 3.10 GHz, where such computer is located at the European Space Research and Technology Centre (ESTEC). Still, it is important to design a robust batch process that quickly ignores entry points whenever no solutions are found. The goal of this section is to describe the framework of this batch process and how it is employed to complete the reachability study.

BATCH MODE SET-UP SCRIPT

The ASTOS batch loop functionality works by sequentially reading through a text file that stores all the pertinent settings required to launch an individual optimization task. Each line of such text file stores the entry conditions for a single EIP. It is thus possible to determine all the feasible trajectories to a particular landing site by systematically probing all the possible EIPs, while keeping the terminal conditions the same. A schematic of how this is done is shown in Figure 7.2.

As shown in Figure 7.2, the approach begins by probing all the possible EIP latitudes at the west-most possible EIP longitude (Step 1). This process is repeated for all the discretized longitude values (Step 2), ultimately

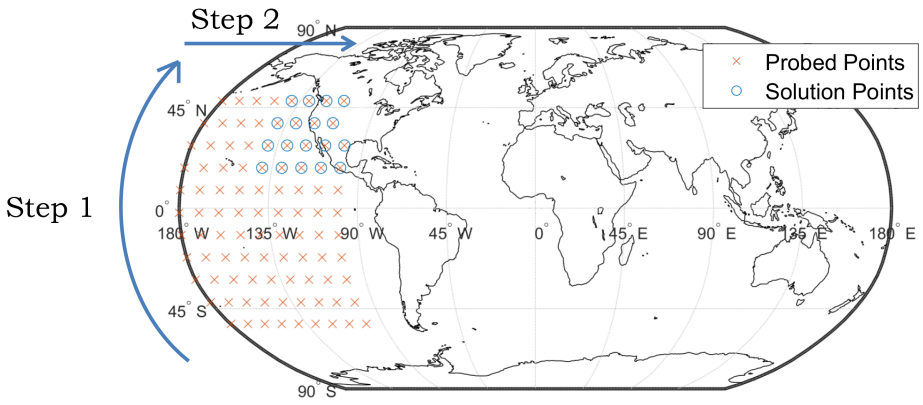

Figure 7.2: Strategy used to generate an entry window for an arbitrary landing site.

generating worldwide entry window. In Figure 7.2, probe EIPs are distributed with a spacing of 10 deg, running from -180 deg to -100 deg longitude and -52 deg to 52 deg in latitude. In this example, only 18 solution EIPs are found, which together delimit the entry window area. The text file with the settings that are fed to the ASTOS batch loop is generated with the logic present in Figure 7.3.

Figure 7.3 shows that the first step taken by the setup script is to discretize the search space. This is done according to the EIP spacing and latitude-longitude search limits provided by the user. The discretized search space provides all the possible latitude-longitude combinations in the probed area. Next, the heading angle at every probe point is computed implicitly using the following two equations, which are derived from the spherical triangle depicted in Figure 7.4.

$$\chi_{EIP} = \pi - \cos^{-1}\left(-\cos\left(\tau_{EIP}\right)\cos\left(\frac{\pi}{2} - i\right)\right) \tag{7.2}$$

$$\delta_{EIP} = f\left(\chi_{EIP}\right) = H\sin^{-1}\left(\frac{\pi}{2} - \left(\frac{\sin\left(\frac{\pi}{2} - i\right)}{\sin\left(\pi - \chi_{EIP}\right)}\right)\right) \tag{7.3}$$

$$H = \begin{cases} 1 & \text{if } \tau_{EIP} < \pi \\ -1 & \text{elsewhere} \end{cases} \tag{7.4}$$

where χ_{EIP} is the heading angle of the EIP, i is the orbit inclination and δ_{EIP} and τ_{EIP} are the latitude and longitude angles of the EIP, respectively. Equations (7.2) to (7.4) yield heading angle values smaller than 90 deg if the vehicle is on an ascending leg at the EIP. On the contrary, if the vehicle is on a descending leg, the obtained heading angle is larger than 90 deg.

In addition to the heading angle, a bank angle estimate is computed at every probe point. Such estimate is the constant bank angle required to bring the SPHYNX vehicle from the probed EIP to the longitude of the landing site. The estimated bank angle is computed by comparing the downrange to be travelled against the downrange capabilities of the SPHYNX vehicle. The downrange travelled by SPHYNX as a function of constant bank angle is shown in Figure 7.5. The computation of a bank angle estimate is necessary to provide the optimizer with a control initial guess that is sufficiently close to the true value. Doing otherwise often results in non-converged solutions. Finally, once all the settings are computed, the settings file is saved and later fed to the ASTOS batch loop.

BATCH MODE LOOP

The aim of this section is to describe the inner workings of the ASTOS batch mode loop and the motivation behind its architecture, which is depicted in Figure 7.6. The first step in such loop is to open the batch settings file created with the MATLAB script presented in Section 7.1.5 and begin reading such file one line at a time. As mentioned earlier, each line contains the settings required to initialize an optimization task for a single probe EIP. Next, these settings are used to simulate an initial guess trajectory that will serve as a starting point for the optimization process. Since the estimated bank angle in such settings is chosen such that the vehicle is brought to the longitude of the landing site, the end-point of the initial guess is expected to land in the neighborhood of the targeted site. If this is not the case, the initial guess will likely either have overshot the landing site or be too far from it in terms of downrange or crossrange. In this case, the optimization of the

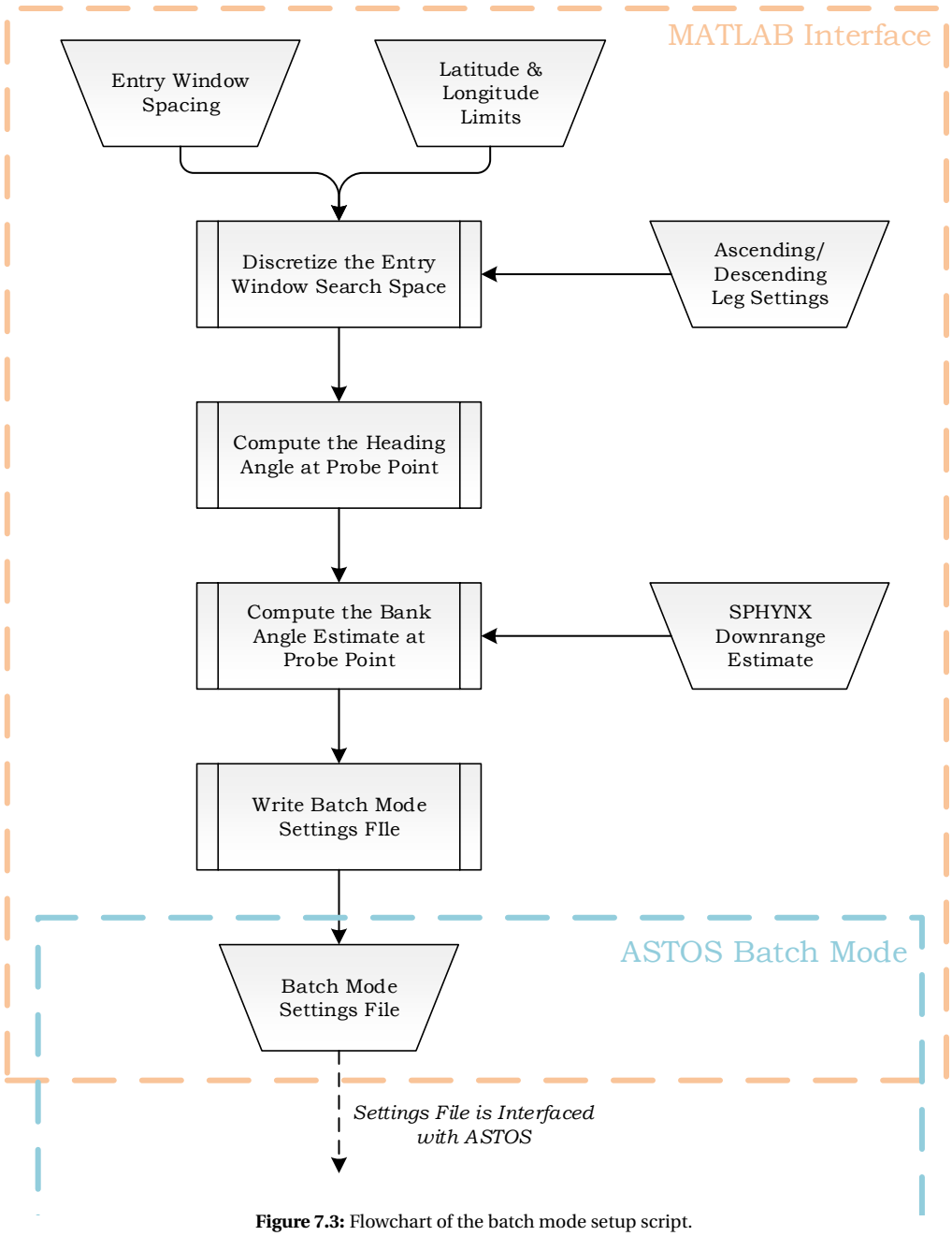

Figure 7.3: Flowchart of the batch mode setup script.

probed EIP will probably not yield a solution. Thus, the batch loop is set up such that the optimization of initial guess trajectories, whose end-point is not close to the landing site, is avoided. This is done with the goal of minimizing the waste of computational power, which is significant for EIPs where a solution could never be found. Although this strategy may rule out EIPs which could potentially yield a solution, the author believes that the savings in computational time are significant enough to justify the use of this method. The area of allowable end-points is delimited by a [-100, 50] deg longitude boundary and a [-30, 30] deg latitude boundary with respect to the landing site. By enlarging such area, the probability of neglecting potential solution points lowers at the expense of computational time. Since all the entry-window maps are visually inspected after completion, any missing solution points can be quickly identified and executed.

All in all, if the end-point of the initial guess lies within the boundaries, the batch loop will attempt to find a solution. It is sometimes that case that despite all the previous checks, the optimizer does not converge to a solution. Should this be the case, the active EIP is neglected and the next point in the file is read. In summary, the batch loop will only save feasible solution, ultimately populating the trajectory database.

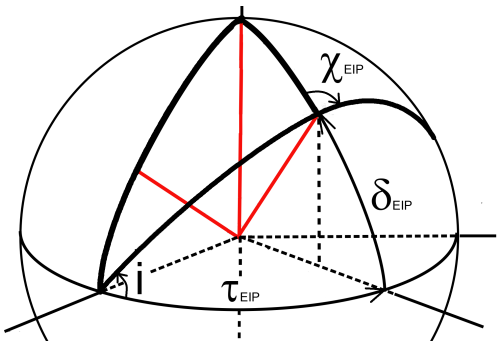

Figure 7.4: Comparison of single shooting integration vs. multiple shooting integration.

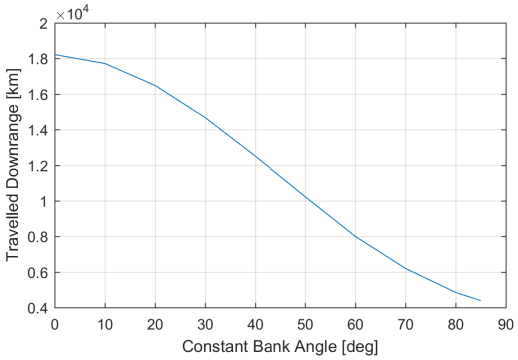

Figure 7.5: Estimated downrange capability of the SPHYNX vehicle as function of constant bank angle.

7.2 TRAJECTORY DATABASE ASSESSMENT

By systematically running the batch loop presented in Section 7.1.5 for all the landing sites, a trajectory database of feasible trajectories is created. The aim of this section is to assess such database, both in terms of the entry-window coverage obtained and the compliance with the imposed trajectory constraints.

7.2.1. COVERAGE RESULTS

In this section, the coverage of the trajectory database is assessed. Note that the term coverage refers to the entry-window area from which a particular landing site can be reached at nominal entry conditions and taking into account all of the path constraints. The entry-windows are presented in the form of "coverage maps", which display a point cloud with all the solution EIPs found in the reachability study. The entry-window is formed by the "area" which such point cloud draws on the map. Note that a coverage map is produced for every landing site established by **REQ-S-08**, as shown by Figures 7.7 to 7.22.

For instance, the coverage map for the primary landing site, Azores, is shown in Figure 7.7 and the associated trajectory groundtracks are shown in Figure 7.8. Note that the groundtracks presented in Figure 7.8 originate from the EIPs shown in Figure 7.7. Furthermore, close examination of the Azores entry-window shows two EIP clusters. On the one hand, the cluster that spans most of the Pacific Ocean is linked represents those EIPs that lay on an ascending leg, in other words, trajectories whose entry heading angle spans the [0, 90] deg range. On the other hand, te second cluster covers an area that spans the southern African continent, the Indian ocean, southeast Asia and Oceania. The "descending cluster" represent those EIPs laying on a descending leg. It is important to highlight how the ascending cluster is significantly closer to Azores than the descending cluster. This means that the trajectories originating from descending cluster travel notably further downrange than those in the ascending cluster.

Figure 7.7 also shows areas where no EIPs are found. The reason behind this is that trajectories originating from such areas would violate at least one of the path constraints or would exceed SPHYNX's downrange/crossrange capabilities. For instance, the area over North America and the northern Atlantic Ocean is very close to Azores, meaning that trajectories originating here would not have enough time to dissipated the vehicle energy, leading to the violation of the maximum aerodynamic load and maximum heat flux constraints. Another region of interest is the area spanning North Africa, Middle east and Central Asia. Since this area is located past the landing site and the vehicle flies a prograde orbit, it becomes obvious that the reason why no EIPs are found here is that such area leads to trajectories that exceed SPHYNX's downrange capability. Other areas of interest are the southern Atlantic Ocean and the northern Pacific Ocean east of the Japanese archipelago. As it becomes evident from the groundtracks, EIPs placed in these latter areas would result in trajectories where SPHYNX would need to travel significantly far in the crossrange direction to reach the Azores site, exceeding the crossrange-downrange envelope of the vehicle. Finally, the reason why no EIPs are found in latitudes outside the [52, -52] deg range is simply because the geometry of the operational orbit does not allow for it, which is a 52 deg inclined orbit.

Figures 7.9 to 7.22 show that the trajectories targeting Gran Canaria, Ben Guerir, Perth, Darwin, Brisbane and Woomera exhibit split ascending and descending clusters. These clusters are divided into two additional groups, where the first group approaches the landing site from a northern direction and the second group

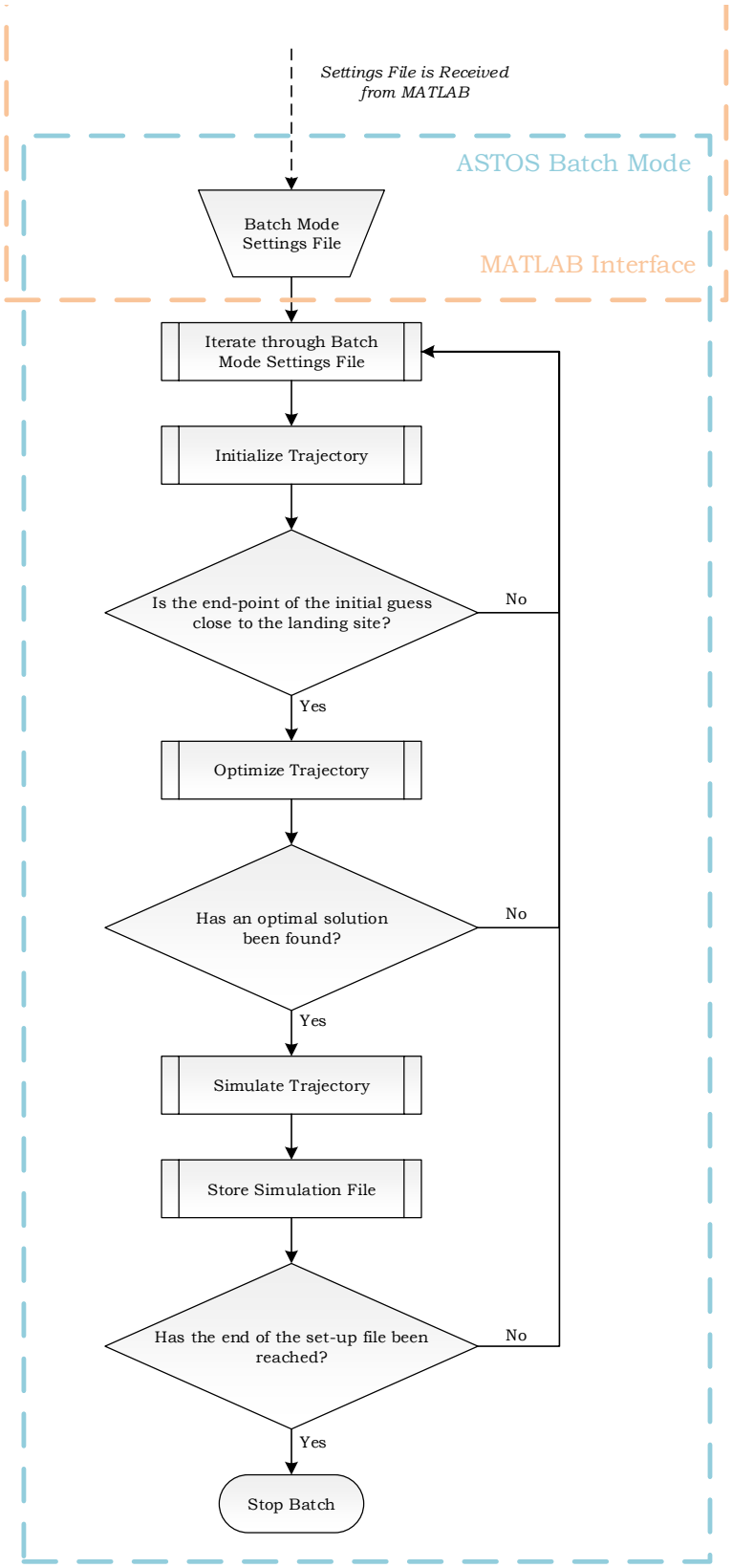

Figure 7.6: Flowchart of the batch mode algorithm.

makes a southern approach. A marginal trend exists where the separation of these two groups increases for those sites placed closer to the equator. In addition, the groundtracks originating from the ascending and the descending cluster eventually overlap, showing consistency among both clusters with the constraints of the longitudinal entry corridor and the downrange-crossrange capabilities of SPHYNX.

Following up on the discussion above, it is interesting to relate the ascending and descending clusters to the commanded bank angle profiles produced by ASTOS. To do so, four trajectories of the database targeting the Azores site are highlighted. These trajectories are representative of the whole database since they exhibit entry profiles that are common to all the targeted sites. As shown in Figures 7.23 and 7.24, SPHYNX displays two types of entries depending on how far the EIP is from the targeted landing site. If the EIP is close to the landing site, the vehicle performs a regular gliding entry, characterized by a direct shallow descent towards the targeted final altitude. On the contrary, if the EIP is far from the landing site, the vehicle performs a skipping entry characterized by a rebound in the upper atmosphere that leads to a sub-orbital flight followed by a regular gliding entry. By performing a skipping entry, the distance traveled by SPHYNX is significantly increased as shown by Figure 7.24, ultimately allowing to reach the targeted landing site. Figures 7.25 shows that the type of entry is primarily controlled by the bank-angle, where a near full-lift-up configuration during the first 400 seconds puts the vehicle in a skipping entry. If a gliding entry is to be performed, a large bank angle magnitude is commanded

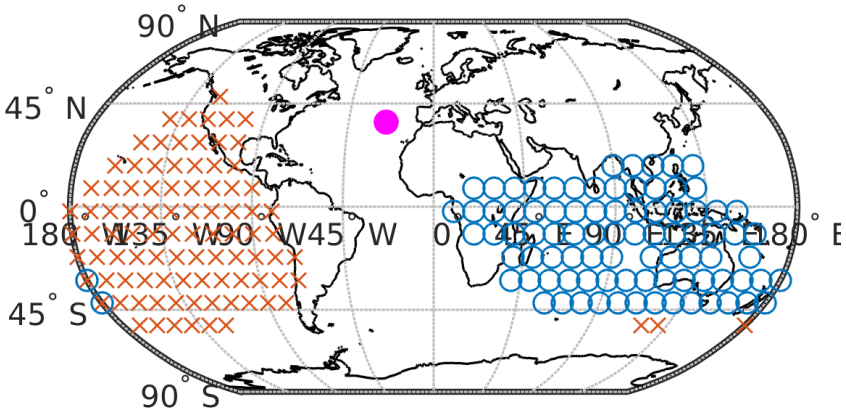

Figure 7.7: Entry window for the Azores site. *Orange Crosses: Ascending Cluster , Blue Circles: Descending Cluster , Filled Point: Site Location*

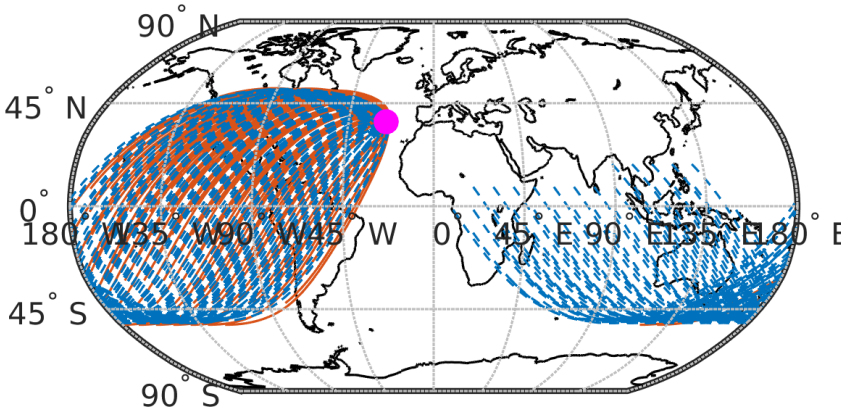

Figure 7.8: Entry trajectories towards the Azores site. *Orange Lines: Ascending Cluster , Blue Lines: Descending Cluster , Filled Point: Site Location*

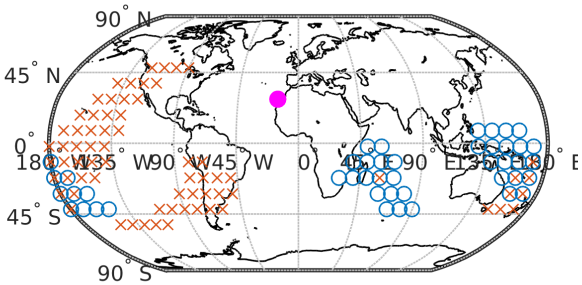

Figure 7.9: Entry window for the Gran Canaria site. *Orange Crosses: Ascending Cluster , Blue Circles: Descending Cluster , Filled Point: Site Location*

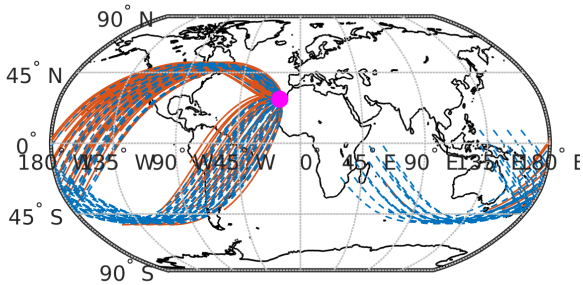

Figure 7.10: Groundtracks of trajectories targeting the Gran Canaria site. *Orange Lines: Ascending Cluster , Blue Lines: Descending Cluster , Filled Point: Site Location*

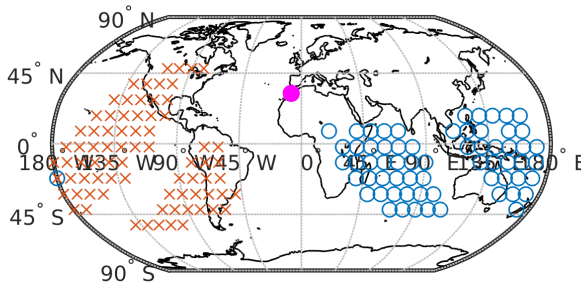

Figure 7.11: Entry window for the Ben Guerir site. *Orange Crosses: Ascending Cluster , Blue Circles: Descending Cluster , Filled Point: Site Location*

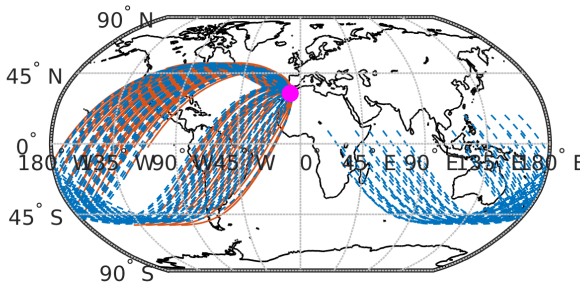

Figure 7.12: Groundtracks of trajectories targeting the Ben Guerir site. *Orange Lines: Ascending Cluster , Blue Lines: Descending Cluster , Filled Point: Site Location*

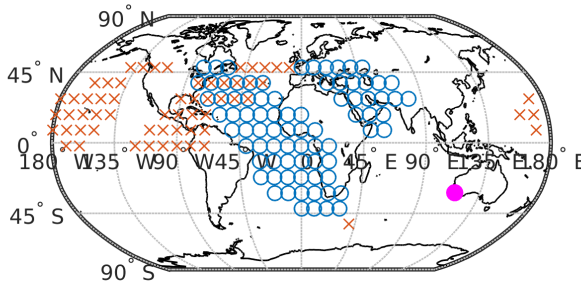

Figure 7.13: Entry window for the Perth site. *Orange Crosses: Ascending Cluster , Blue Circles: Descending Cluster , Filled Point: Site Location*

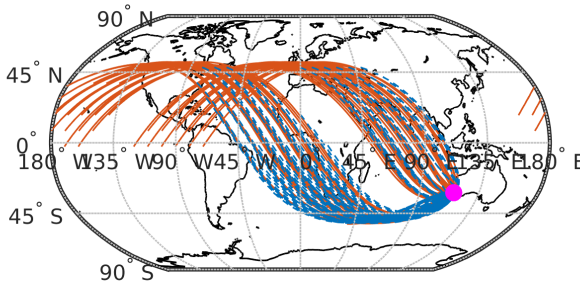

Figure 7.14: Groundtracks of trajectories targeting the Perth site. *Orange Lines: Ascending Cluster , Blue Lines: Descending Cluster , Filled Point: Site Location*

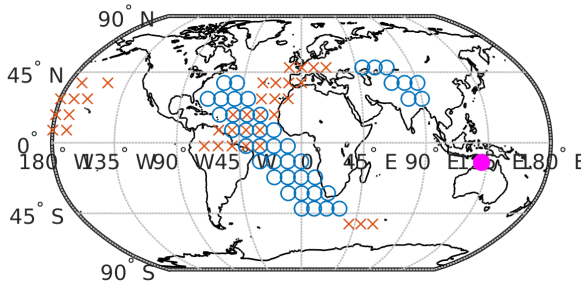

Figure 7.15: Entry window for the Darwin site. *Orange Crosses: Ascending Cluster , Blue Circles: Descending Cluster , Filled Point: Site Location*

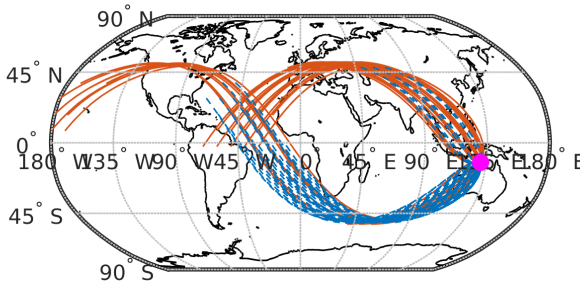

Figure 7.16: Groundtracks of trajectories targeting the Darwin site. *Orange Lines: Ascending Cluster , Blue Lines: Descending Cluster , Filled Point: Site Location*

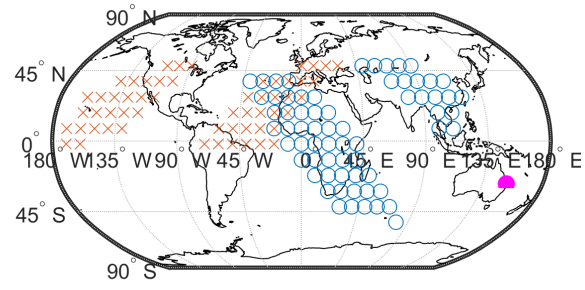

Figure 7.17: Entry window for the Brisbane site. *Orange Crosses: Ascending Cluster , Blue Circles: Descending Cluster , Filled Point: Site Location*

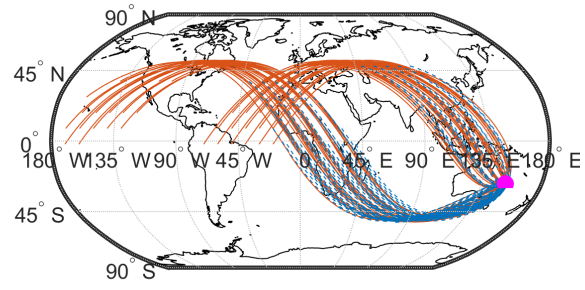

Figure 7.18: Groundtracks of trajectories targeting the Brisbane site. *Orange Lines: Ascending Cluster , Blue Lines: Descending Cluster , Filled Point: Site Location*

shortly after the EIP, effectively reducing the vertical component of the lift vector and thus allowing the vehicle to sink into the atmosphere. As discussed in Section 4.3, the bank-angle direction controls the crossrange error which becomes evident when comparing Figures 7.25 and 7.30. Here, the gliding entries are shown to have opposite bank-angle signs and thus show the vehicle to move in opposite crossrange directions. If the bank angle is negative, the trajectory is shown to curve in the "port" direction, whereas if the bank angle is positive,

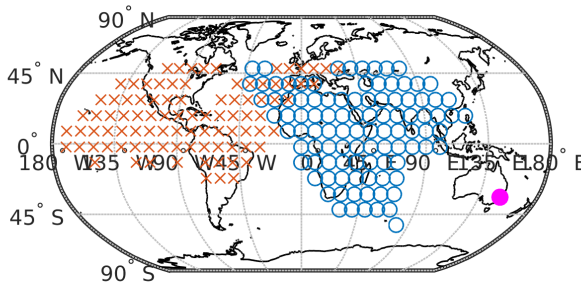

Figure 7.19: Entry window for the Sidney site. *Orange Crosses: Ascending Cluster , Blue Circles: Descending Cluster , Filled Point: Site Location*

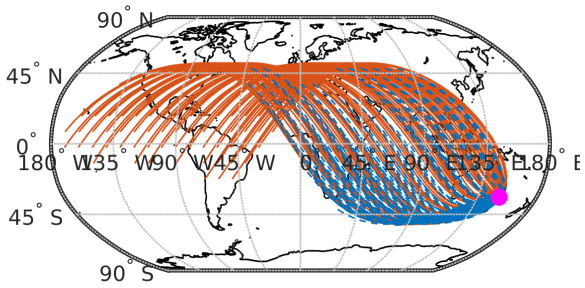

Figure 7.20: Groundtracks of trajectories targeting the Sidney site. *Orange Lines: Ascending Cluster , Blue Lines: Descending Cluster , Filled Point: Site Location*

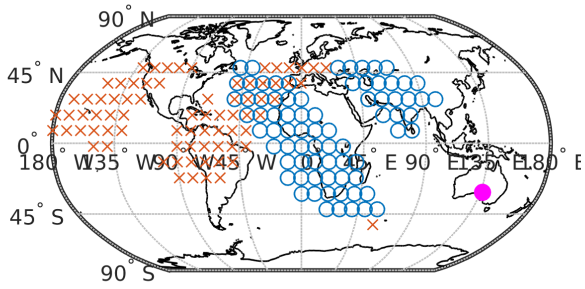

Figure 7.21: Entry window for the Woomera site. *Orange Crosses: Ascending Cluster , Blue Circles: Descending Cluster , Filled Point: Site Location*

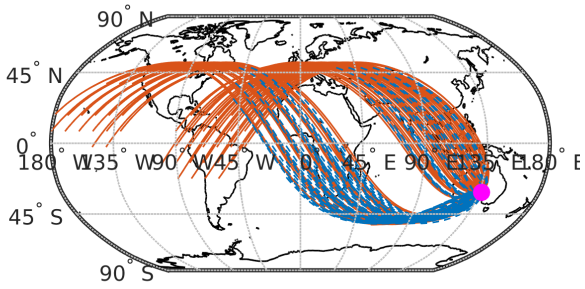

Figure 7.22: Groundtracks of trajectories targeting the Woomera site. *Orange Lines: Ascending Cluster , Blue Lines: Descending Cluster , Filled Point: Site Location*

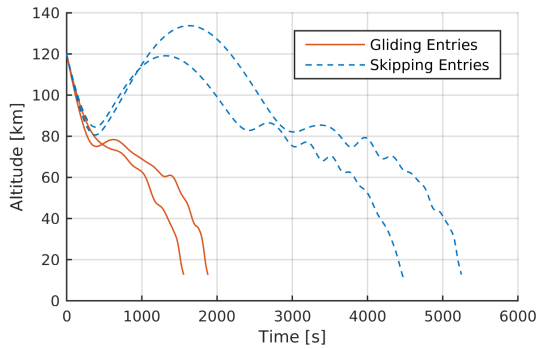

Figure 7.23: Altitude profiles for four example trajectories of the database that target the Azores site.

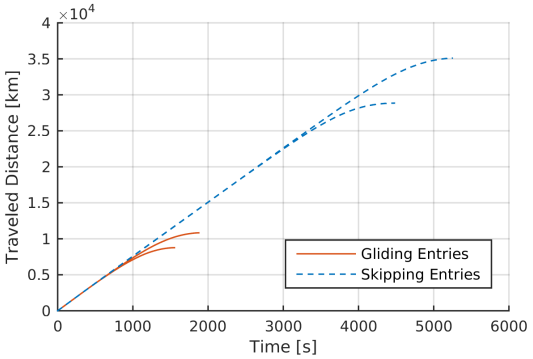

Figure 7.24: Traveled distance profiles for four example trajectories of the database that target the Azores site.

the trajectory curves in the "starboard" direction. On the other hand, the angle-of-attack is primarily used to modulate the lift and drag coefficients and thus the energy dissipation of energy. Comparison of Figures 7.26 and 7.28 shows that the angle-of-attack is increased to track the heat-flux constraint, which is particularly evident for the skipping entry. On the other hand, Figure 7.27 shows that the aerodynamic load in these four example trajectories is nowhere near the 3.0 g limit imposed by **REQ-S-01**. Still, it is relevant to highlight that this load becomes more sensitive to changes in the angle-of-attack once the vehicle is in the high density lower atmosphere. Finally, Figure 7.29 shows that skipping entries lead to an increase in the total heat load due the longer flight-time. Still, such increase is moderate since a significant fraction of the flight occurs outside the atmosphere where the experience heat-flux plateaus as a consequence of the low density.

GLOBAL TIME-COVERAGE

A global time-coverage assessment of the trajectory database is created by combining the ascending and descending clusters for all the landing sites, as shown in Figures 7.31 and 7.32. Inspection of both figures reveals that the database provides a global time-coverage of approximately 50%. This means that if a de-orbit burn is commanded immediately after the abort calling in the operational orbit, an EIP that guarantees access to one of the landing sites will be reached in 50% of the cases. It is important to emphasize that to ensure full

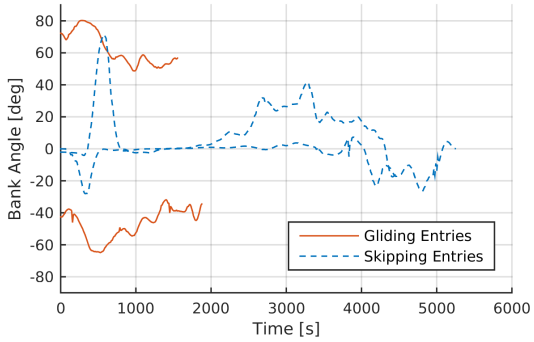

Figure 7.25: Bank-angle profiles for four example trajectories of the database that target the Azores site.

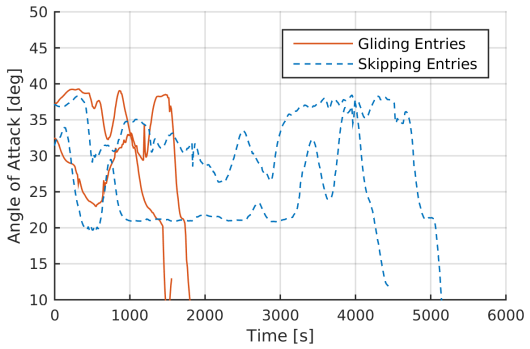

Figure 7.26: Angle-of-attack profiles for four example trajectories of the database that target the Azores site.

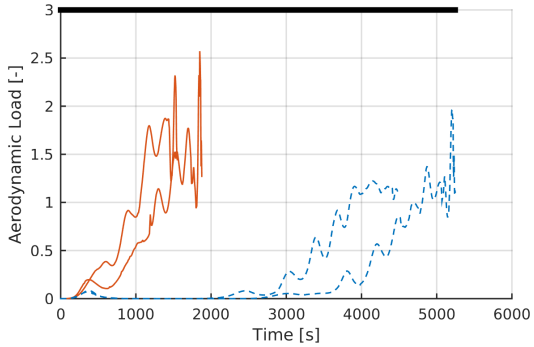

Figure 7.27: Aerodynamic load profiles for four example trajectories of the database that target the Azores site.

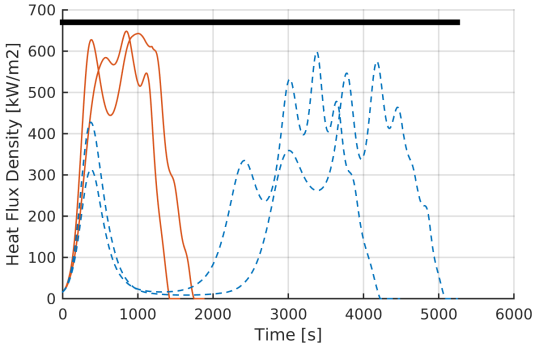

Figure 7.28: Heat-flux profiles for four example trajectories of the database that target the Azores site.

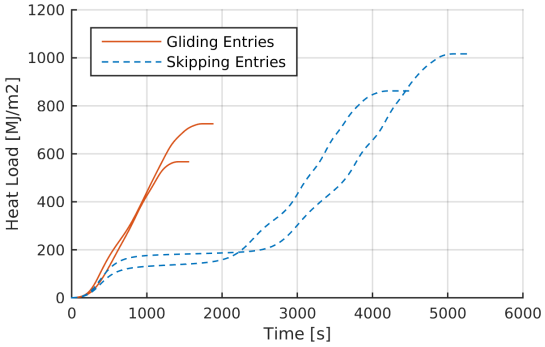

Figure 7.29: Heat-load profiles for four example trajectories of the database that target the Azores site.

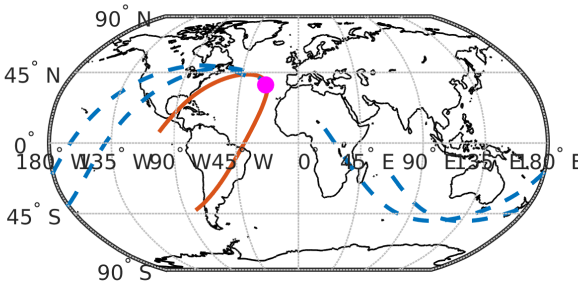

Figure 7.30: Example entry trajectories towards the Azores site. *Orange Lines: Ascending Cluster , Blue Lines: Descending Cluster , Filled Point: Site Location*

global coverage both the ascending and descending leg solutions must be available at every EIP. The area enclosed by the two black curves represents the region where no common ascending and descending EIPs are found. A rough estimate of the maximum in-orbit wait time can be made by computing time required for the groundtrack to shift across the boundaries of such area. Such approximation is given below:

$$t_{wait} \approx \frac{\zeta}{\omega_E} \tag{7.5}$$

where is the standby time in-orbit, is the required groundtrack shift and is angular velocity of Earth. In this case, the required groundtrack shift is approximate 150 deg which yields a maximum in-orbit wait-time of approximately 10 hours.

It should be emphasized that global coverage is highly dependent on the distribution of the emergency landing sites. Although the optimization of such distribution is beyond the scope of this thesis, a few patterns arise

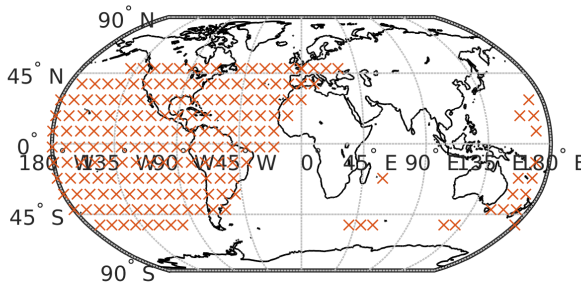

Figure 7.31: Global entry-window for trajectories with an EIP in an ascending leg.

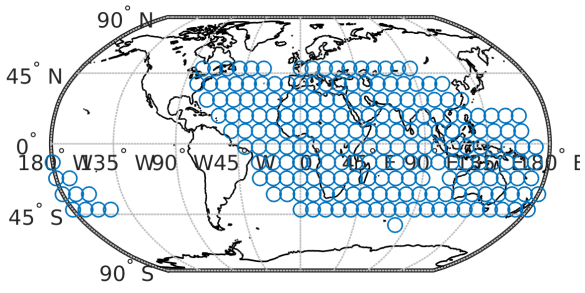

Figure 7.32: Global entry-window for trajectories with an EIP in an ascending leg.

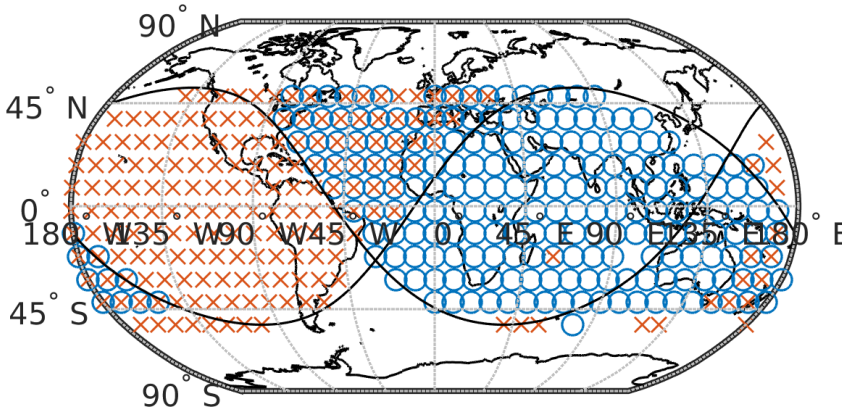

Figure 7.33: Global entry-window that combines the ascending and descending clusters.

from the coverage maps shown in Figures 7.7 to 7.22. To begin with, landing sites located far away from the equator appear to provide significantly larger entry-windows. For instance, the Darwin and Sidney sites belong both to the Australian group and exhibit a significant difference in the entry-window area. The Darwin site, located closer to the equator, has a far smaller entry-window than the Sidney site. A similar pattern arises when comparing the northern Atlantic sites, where Azores provides a larger entry-window than Gran Canaria, which is located further south. Given such reasoning, placing a new site in the Japanese archipelago could provide access to flights entering the atmosphere in a descending leg over the pacific ocean, thus enlarging the global entry-window. Similarly, a site near the Canada-US border could provide access to flight entering the atmosphere in an ascending leg over the eastern hemisphere.

SHARED COVERAGE

Comparison of Figures 7.7 to 7.22 reveals that overlapping entry-windows can be found for multiple landing sites. Any EIPs enclosed by such shared windows offer the possibility to re-target the vehicle to an alternative landing site. Such capability of the database is referred to as shared coverage, which is discussed in the remainder of this section.

The shared coverage maps with the Azores site are shown in Figures 7.34 to 7.39. Note that in the case of Azores, only a single EIP shared with Darwin is found, thus not shown here. The shared coverage results are summarized in Table 7.3, which details the percentage of shared EIPs per site. For instance, Azores shares 42% of its EIPs with Gran Canaria, 63% with Ben Guerir, 14% with Perth, 1% with Darwin, 23% with Brisbane, 44% with Sidney and 23% with Woomera. It is important to highlight that such table does not necessarily require symmetrical results due to the fact that all sites have a different total number of EIPs available. For example, even if Azores and Gran Canaria share 77 points, the total number of EIPs available for Azores is 182 and 114 for Gran Canaria, thus yielding the percentages indicated in Table 7.3.

Table 7.3 is split in four quadrants, where the top-left quadrant shows the percentage of shared EIPs among the northern Atlantic sites and the bottom-right quadrant shows the share among the Australian sites. The top-right and top-left quadrants give the percentage of shared EIPs across the two groups. For instance, the northern Atlantic sites share an average of 65.1% of the entry windows area with other sites of the same group. In the case of the Australian sites, the average percentage of shared window entry area among members of the

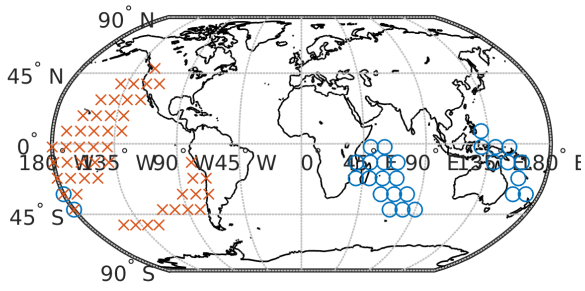

Figure 7.34: Azores' shared entry-window with Gran Canaria. *Orange Crosses: Ascending EIPs , Blue Circles: Descending Cluster*

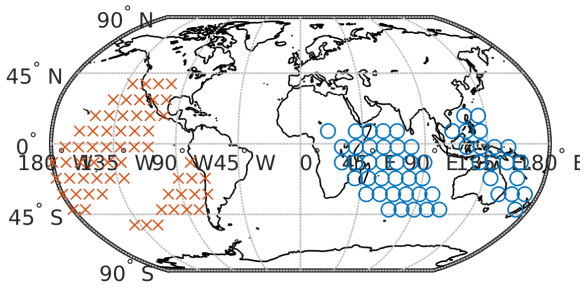

Figure 7.35: Azores' shared entry-window with Ben Guerir. *Orange Crosses: Ascending Cluster , Blue Circles: Descending Cluster*

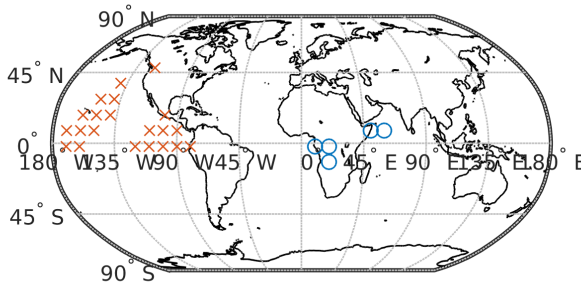

Figure 7.36: Azores' shared entry-window with Perth. *Orange Crosses: Ascending EIPs , Blue Circles: Descending Cluster*

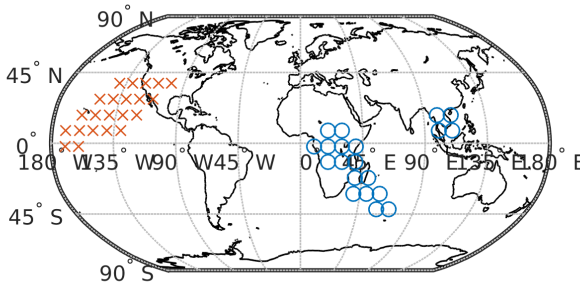

Figure 7.37: Azores' shared entry-window with Brisbane. *Orange Crosses: Ascending Cluster , Blue Circles: Descending Cluster*

same group is 51.9%. The lower percentage share of the Australian group is justified by the fact that the sites in such group are more widely spread than those in the northern Atlantic group. For instance, the Brisbane, Sidney and Woomera sites are relatively close to each other if compared to Perth and Darwin. Consequently, the entry window share among Brisbane, Sidney and Woomera averages at approximately 70.0%, where the share with Perth and Darwin drops to 43.4%. When the cross-group shared area is examined, an average of 20.5% is found. On the other hand, Perth and Darwin share approximately 48.5% of the entry window area. Please note that although this figure is relatively high even when the two sites are located far from each other. The reason behind this is that the total number of EIPs available for the Darwin site is significantly low, thus suggesting a high percentage share as discussed above.

It becomes clear that although entry-windows are expected to be mostly shared among members of the same landing group, there is a significant cross-group entry-window sharing, where landing sites of the other group can be reached by putting the vehicle in a long skip entry as discussed in Section 7.2.1. It is important to emphasize that all sites share 50% of its entry-window with at least one alternative site, proving the abort capabilities of the system. Note that re-routing to a different site can be simply done by selecting the appropriate trajectories from the database and no further modifications to the system need to be made. Furthermore, there are entry-window areas where more than one site can be reached. Given the above discussion, it becomes clear that while clustering the sites together increases the re-routing opportunities, it significantly decreases the global coverage.

Note that the ability to re-route to a different landing site is highly dependent on the remaining energy available, which is depleted as the vehicle descends from the EIP. For example, rerouting from Azores to Perth may only be possible within the first 500 seconds of the flight. In summary, the any analysis done in this section concerns the capability of routing prior the EIP is reached. Any analysis regarding how far down the trajectory any re-routing is possible is considered beyond the scope of this thesis.

7.2.2. CONSTRAINT COMPLIANCE

The goal of this section is to discuss the constraint compliance of the trajectories stored in the database. In general, there is a good agreement with such constraints, although there are few points to highlight.

Despite the fact that the bank angle dominates the shape of the reentry trajectory, angle of attack modulation plays an important role in controlling the constraint compliance. For instance, large angle of attack values are typically issued in the early portion of the flight with the goal of slowing the vehicle down in the higher at-

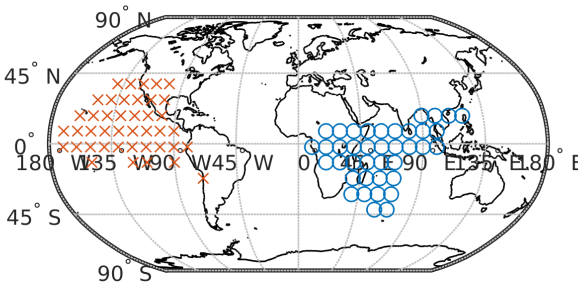

Figure 7.38: Azores' shared entry-window with Sidney. *Orange Crosses: Ascending EIPs , Blue Circles: Descending Cluster*

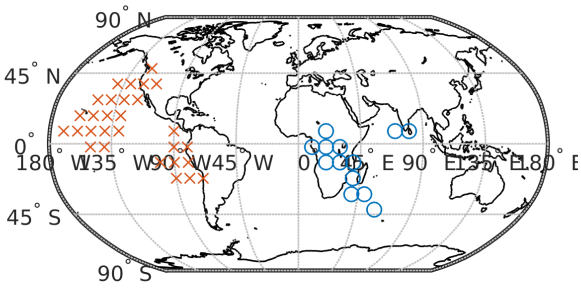

Figure 7.39: Azores' shared entry-window with Woomera. *Orange Crosses: Ascending Cluster , Blue Circles: Descending Cluster*

Table 7.3: Percentage of EIPs shared among landing sites.

Shared With:	**Azores**	**Gran Canaria**	**Ben Guerir**	**Perth**	**Darwin**	**Brisbane**	**Sidney**	**Woomera**
Azores [%]		42	63	14	1	23	44	23
Gran Canaria [%]	68		75	12	1	24	38	27
Ben Guerir [%]	82	61		11	2	24	45	24
Perth [%]	18	10	11		35	30	52	59
Darwin [%]	3	3	4	62		50	46	71
Brisbane [%]	34	23	28	34	32		90	72
Sidney [%]	42	22	33	38	19	58		62
Woomera [%]	28	20	22	54	36	59	79	

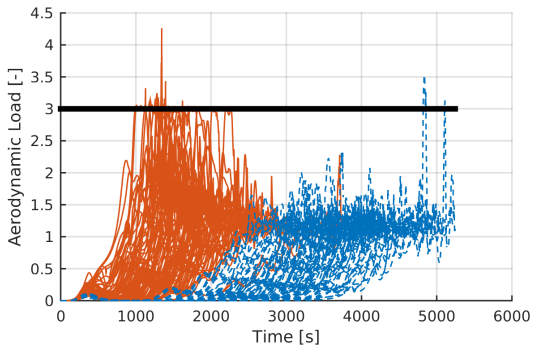

Figure 7.40: Aerodynamic load profiles of trajectories targeting the Azores site. *Orange Crosses: Ascending Cluster , Blue Crosses: Descending Cluster*

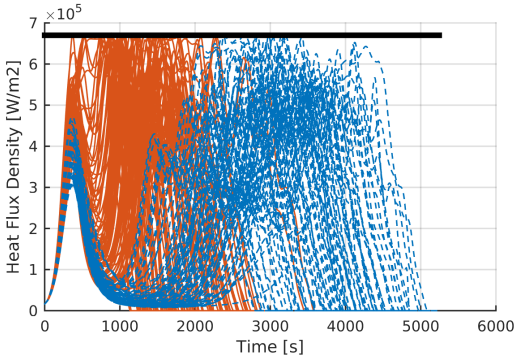

Figure 7.41: Heat flux profiles of trajectories targeting the Azores site. *Orange Lines: Ascending Cluster , Blue Lines: Descending Cluster*

mosphere as a consequence of the high lift and drag. Doing so tracks the heating rate in such a way that the maximum heat flux constraint is met. In a similar fashion , medium angle of attack values are commanded whenever there is sufficient aerodynamic pressure, which leads to a high lift-to-drag ratio, increasing the vehicle's range capabilities. It is important to emphasize that the angle of attack is constrained by the aerodynamic database boundaries, which establish an angle of attack range based on Mach number. As will be seen in the remainder of this section, the shape of such boundaries is a key factor in the compliance of the database trajectories with the imposed path constraints.

To begin with, there is a trend for a limited number of trajectories to violate the aerodynamic load constraint, as shown in Figure 7.40 for the Azores site. Figure 7.42 suggest that this issue arises due to the fact that such trajectories fly through the area of the $\alpha - M$ space where the aerodynamic lift coefficients change rapidly, which is discussed in Chapter 2.4. Since the aerodynamic coefficients and the value of the aerodynamic load are directly related as given in Equation (3.74), the mentioned rapid changes lead to sharp peaks in the aerodynamic load which the optimization software struggles to capture. On the contrary, the heat flux constraint is well captured by all the trajectories for all the landing sites, as shown by Figure 7.41. The heat flux is indirectly related to the aerodynamic coefficients as shown by Equation (3.70), thus any rapid changes do not reflect immediately in the heat flux profile. Consequently, changes in heat flux are sufficiently slow to be captured by the optimization software and thus kept within the constraints.

Although the aerodynamic database boundaries are generally satisfied, the supersonic region under Mach 5 is problematic as predicted in Section 2.4.1. The corridor narrows down significantly and the sharp boundaries cause the optimizer to struggle in such region. Such issue is shown in Figure 7.42 which shows the $\alpha - M$ profiles of trajectories targeting the Azores site, although such trend appears as well for the remaining landing sites. Note that optimizer in ASTOS extrapolates the aerodynamic database in those areas where the boundaries are surpassed. This extrapolation may not yield and accurate representation of the aerodynamic characteristics of the vehicle. Furthermore, the boundaries are constructed by defining the area of the aerodynamic database where there is no sufficient elevator deflection to trim the vehicle. Violating such boundaries puts the vehicle in a trajectory where it would be flying untrimmed and consequently induce angle of attack errors.

An example trajectory where a large aerodynamic load constraint overshoot occurs is show in Figure 7.44, which belongs to the group of trajectories that target the Azores landing site. Here, a peak over shoot of over 4.26 g is seen at approximately 1400 s into the flight. Comparison with Figure 7.45 shows that this overshoot occurs when the angle-of-attack is rapidly increased from 26 deg to 38 deg, which occurs when the vehicle flies just below Mach 5.0 as shown by Figure 7.46. As discussed in Section 7.2.1, this behavior shows how the aerodynamic load is particularly sensitive to changes in the angle-of-attack in the lower atmosphere. The key question here is why such rapid is increase in angle-of-attack is commanded in the first place. An answer to this question is found by examining Figure 7.47, which shows that the range-to-go at Mach 5.0 is below 120 km. At the altitude when this occurs, the vehicle is traveling at approximately 1600 m/s and consequently it needs to slow down fast enough to avoid overshooting the landing site. To achieve this, the optimizer commands a high angle-of-attack which increases the drag coefficient and consequently the rate of energy dissipation. The behavior described above is seen in most of the trajectories that violate the aerodynamic load constraint. The performance could be improved by running subsequent optimization tasks, which could modulate the angle-of-attack and bank-angle controls in such a way that the main deceleration phases is extended towards the hypersonic regime.

The 10 km constraint on the maximum landing distance is generally satisfied, although Figure 7.43 proves that a limited number of trajectories in every site terminate further from the landing site than stipulated by the landing accuracy constraint. This issue is related to the bottle neck of the aerodynamic database under Mach 7.5, which results in trajectories not having sufficient freedom to steer the vehicle towards the landing site.

The constraint compliance characteristics of the aerodynamic database are summarized in Table 7.4. It is shown that on average, approximately 7.4% of the trajectories stored overshoot the aerodynamic load constraint. The mean value of such overshoots ranges from 3.05 g to 3.24 g, which is moderately above the limit established in **REQ-S-01**. The peak aerodynamic load however, can reach up to 4.26 g which is considered unfeasible. Similarly, the number of heat-flux constraint overshoots are within the same order of magnitude at an average of 7.2%. The average and peak values however, are marginally above the 670 kW/m^2 constraint established by **REQ-S-03**. Finally, the percentage of trajectories that terminate outside the 10 km radius from the landing site as established by **REQ-S-05** is 2.6%, which is on average lower than the percentages of aerodynamic load and heat-flux constraint overshoots. The mean overshoot distance-to-target ranges between 12.2 km to 58.1 km and the maximum overshoot value reaches up to 113.8 km, where these values exceed significantly the 10 km limit.

Overall, it becomes clear that a limited number of trajectories stored in the database exceed the limits established by the system requirements. These trajectories must be identified and re-processed, since using them to generate reference interpolated trajectories as discussed in Chapter 4 would force the guidance system to track a trajectory that violates a number of constraints in the first place. As will be discussed in Chapter 8, the guidance performance is assesses using a small subset of the database, where the trajectories in such subset are ensured to satisfy all the constraints.

7.3 LIMITATIONS OF THE REACHABILITY STUDY

The goal of this section is to discuss the limitations encountered throughout the reachability study and how these impact both the scope of this thesis, as well as particular aspects of the abort guidance system.

To begin with, the most detrimental limitation encountered throughout this study is the significant optimization tuning and post-processing effort required to obtain feasible trajectories. Despite all the precautions taken, the batch loop discussed in Section 7.1.5 misses a number of solution points in the entry-window maps, which requires to manually search for these points and individually tune the optimization settings for each

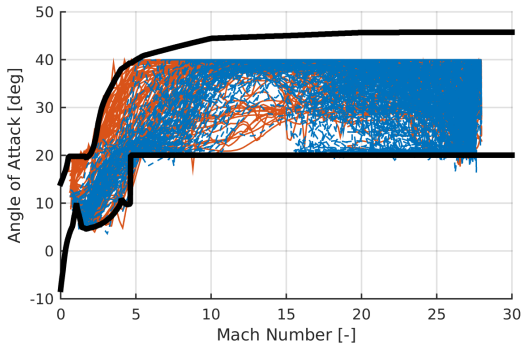

Figure 7.42: SPHYNX aerodynamic database compliance of trajectories targeting the Azores site. *Orange Lines: Ascending Cluster , Blue Lines: Descending Cluster*

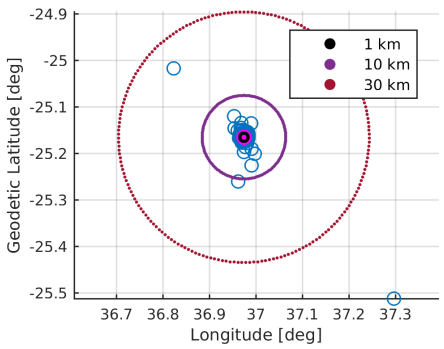

Figure 7.43: Landing accuracy of trajectories targeting the Azores site. *Orange Crosses: Ascending Cluster , Blue Circles: Descending Cluster*

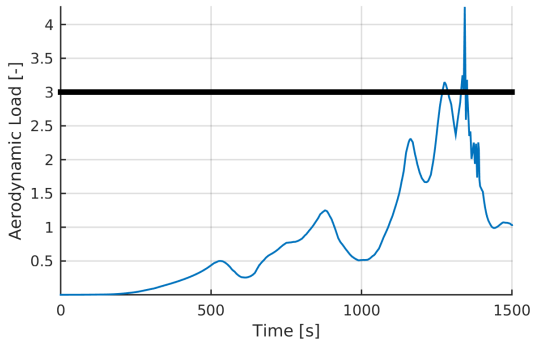

Figure 7.44: Aerodynamic load profile for the trajectory targeting the Azores site with the largest aerodynamic load overshoot.

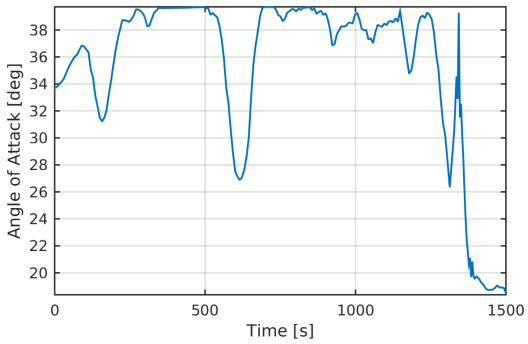

Figure 7.45: Angle-of-attack profile for the trajectory targeting the Azores site with the largest aerodynamic load overshoot.

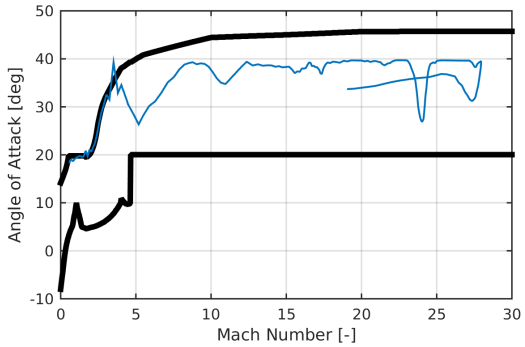

Figure 7.46: Aerodynamic database compliance for the trajectory targeting the Azores site with the largest aerodynamic load overshoot.

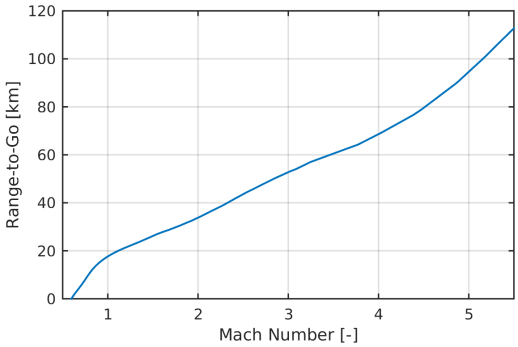

Figure 7.47: Range-to-go vs Mach number profile for the trajectory targeting the Azores site with the largest aerodynamic load overshoot.

point. Furthermore, many of the solution points obtained do not actually comply with the constraints established due to multiple shooting integration issue described in Section 7.1.4. This is particularly true for the SPHYNX aerodynamic database boundaries, which are hard to capture in the optimization process due to their discontinuous behaviors in the supersonic region. Consequently, it is necessary to identify these problematic points and launch new optimization runs with a refined grid, thus adding computational and post-processing time to the reachability study.

Even when using a 500 node fine grid, many solution points are still far from the landing site or violate the database constraints. At the time of writing of this report, no solution was found This is issue cannot be solved and thus proves a limitation of the ASTOS software. Thus, the devised abort guidance system must be able to detect whether the reference trajectory violates the database boundaries and apply instead the nearest angle of attack within the database. Since the deviations from the database are relatively small, this approach does

Table 7.4: Summary of the contraint compliance of all the trajectories stored in the database.

	Azores	**Gran Canaria**	**Ben Guerir**	**Perth**	**Darwin**	**Brisbane**	**Sidney**	**Woomera**
% $n_L > 3.0$	9.3	9.6	7.8	5.8	6.4	9.0	5.2	5.9
Mean n_L [-]	3.20	3.05	3.18	3.08	3.24	3.18	3.07	3.20
Max. n_L [-]	4.26	3.21	4.41	3.53	3.71	3.96	3.47	3.61
% $q_c > 670$ kW/m^2	6.6	7.0	7.9	7.1	10.3	5.7	6.2	6.6
Mean q_c [kW/m^2]	671.3	671.5	673.1	670.9	671.9	672.4	671.6	671.2
Max. q_c [kW/m^2]	675.5	675.6	677.8	672.8	674.4	676.9	676.7	677.7
% $s_{site} > 10$ km	1.6	2.6	1.4	2.1	3.8	2.5	2.6	3.9
Mean s_{site} [km]	28.9	28.9	28.9	33.5	22.3	12.2	58.1	27.8
Max. s_{site} [km]	52.8	52.8	47.1	47.0	42.2	13.9	113.8	52.8

not result in a significant error.

In addition to the tuning and post-processing effort described above, the computational time required to produce a single entry-window map amounts to over 12 hours on a high-end laptop with a Core^{TM} i7 5700HQ processor running at 2.70 GHz. It is important to mention that most of the computational power is "wasted" while probing non-solution EIPs, although required to produce the coverage maps discussed in Section 7.2.1. This issue was partly tackled by performing an initial low-fidelity run with 50 collocation nodes to narrow down the search area, which would serve as the foundation for a high-fidelity 500 node run. Despite this, the computational time is still a limiting factor, which led the author to redefine the scope of the reachability to reentry trajectories originating from a single operational orbit with an inclination of 52 deg.

Finally, at the time of writing of this thesis report, the author faced numerous stability issues with the ASTOS software, leading to schedule overruns which further justified the new scope of the study.

8 GUIDANCE PERFORMANCE

The goal of this chapter is to relate the performance results of the abort guidance system discussed in Chapter 4 to the trajectory database design parameters highlighted in Section 4.1.1.

8.1 PERFORMANCE IMPACT OF THE DATABASE PARAMETER SPACING

The goal of this section is to characterize the impact of the trajectory database parameter spacing on the terminal point dispersions and constraint compliance. To do so, the extremals delimiting the interpolation cell are systematically brought closer to a test EIP thus reducing the cell area and improving the fidelity of the interpolated solutions, as sketched in Figure 8.1. Due to the large coverage area that the database spans, it is unfeasible to recompute the entire database for smaller grid spacings due to the reasons stated in Section 7.3. Consequently, the tests performed in this section are restricted to a single arbitrary landing site and EIP, thus serving as a proof-of-concept. The selected EIP is placed at -125 deg longitude and 13 deg geodetic latitude, where the targeted landing site is the Ben Guerir Air Force Base. The extremals used in this first test and the corresponding database parameter spacings are given in Table 8.1, where the coarsest spacing is 10 deg and the finest spacing is 0.1 deg.

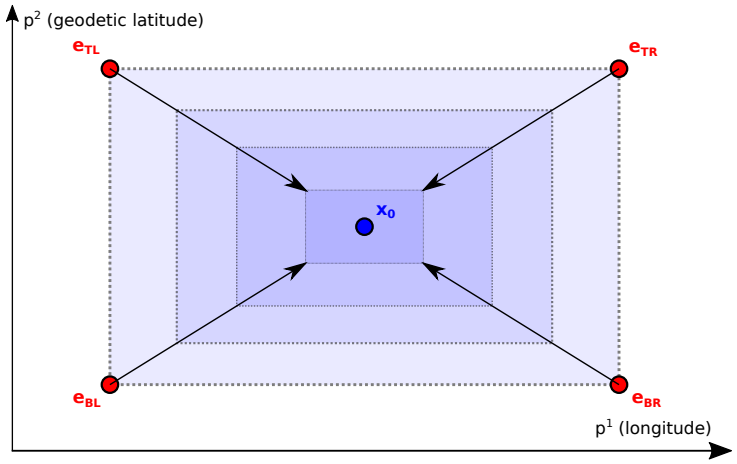

Figure 8.1: Sketch showing how the grid cell area reduces as a consequence of limiting the parameter spacing.

To get a better understanding of how the parameter spacing affects the performance impact of the guidance system, the reader is referred to Figures 8.2 to 8.7. Here, two guided entries are presented which originate at the same EIP, where the only difference is the size of grid cell on which the interpolation is performed. The tested entries feed off 10 deg and 1 deg spaced grid cells. Please note that significantly different grid spacings are selected with the goal of clearly showing what trends are to be expected as the spacing is reduced. In 8.2 to 8.7, the flown trajectory as guided by the abort guidance system is shown together with the reference interpolated trajectory and the supporting extremals. At this point it is important to recall that the extremals supporting the interpolation are actual solutions of the equations of motion. By comparing Figures 8.2 and 8.3, it becomes clear that both the flown trajectory and the interpolated trajectory are more tightly "enclosed" when a 1 deg spacing grid cell is used. As discussed in Section 4.1.1, this behavior provides a direct indication that the reference interpolated trajectory obtained using the finer grid spacing will provide a better approximation to the solution of the equations of motion. To support this, Figures 8.4 and 8.5 show the altitude profiles of the two entries discussed here. It becomes clear that the entry which feeds off a 10 deg cell shows extremals differ significantly in shape and trajectory length, contrary to the case where a 1 deg spacing is used, where the extremals show similar profiles. Consequently, the quality of the approximation to the solution of the equations

of motion will be poor, potentially leading to a trajectory that is not flyable in the first place. This is proven by Figures 8.6 and 8.7, which display the angle-of-attack profiles of the two tested entries. It becomes evident that the trajectory feeding of the 10 deg grid cell struggles to approximate the interpolated reference profile, simply because such reference is not compliant with the physics of the problem. On the contrary, the trajectory that feeds off a 1 deg cell is able to approximate the reference angle-of-attack profile from the start of the entry. In summary, Figures 8.2 to 8.7 demonstrate that decreasing the parameter spacing improves the quality of the approximated reference command profiles, potentially leading to a better guidance perfomance.

Now that the expected trend in guidance performance as a function of the grid spacing has been justified, six 1000-case Monte Carlo simulation campaigns are performed with the goal of establishing actual performance values. Note that dispersion characteristics used in the Monte Carlo campaigns are specified in Table 8.1. The values indicated in such table are similar to those given by Sagliano et al. (2016) and Sagliano et al. (2014), since the reference vehicle tested in both studies has similar characteristics to those of SPHYNX. Please note that the

Table 8.1: EIP settings of the extremals as a function of the database parameter spacing.

	Parameter Spacing:	**10.0 deg**	**5.0 deg**	**2.0 deg**	**1.0 deg**	**0.5 deg**	**0.1 deg**
Top-Left Extremal	τ_{EIP} [deg]	-130.0	-127.5	-126.0	-125.5	-125.05	-125.05
	δ^*_{EIP} [deg]	18.0	15.5	14.0	13.5	13.25	13.05
	χ_{EIP} [deg]	40.3	39.7	39.4	39.3	39.23	39.20
Top-Right Extremal	τ_{EIP} [deg]	-120.0	-122.5	-124.0	-124.5	-124.75	-124.95
	δ^*_{EIP} [deg]	18.0	15.5	14.0	13.5	13.25	13.05
	χ_{EIP} [deg]	40.3	39.7	39.4	39.3	39.23	39.20
Bottom-Left Extremal	τ_{EIP} [deg]	-130.0	-127.5	-126.0	-125.5	-125.05	-125.05
	δ^*_{EIP} [deg]	8.0	10.5	12.0	12.5	12.75	12.95
	χ_{EIP} [deg]	38.4	38.8	39.0	39.1	39.14	39.18
Bottom-Right Extremal	τ_{EIP} [deg]	-120.0	-122.5	-124.0	-124.5	-124.75	-124.95
	δ^*_{EIP} [deg]	8.0	10.5	12.0	12.5	12.75	12.95
	χ_{EIP} [deg]	38.4	38.8	39.0	39.1	39.14	39.18

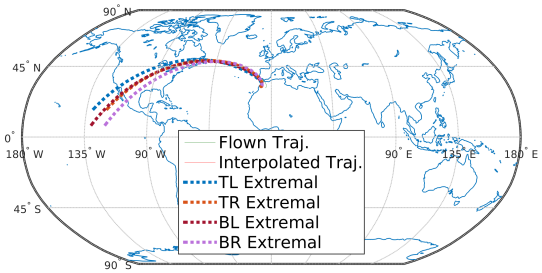

Figure 8.2: Entry window for the Brisbane site.

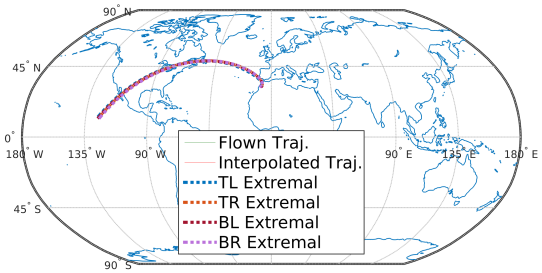

Figure 8.3: Groundtracks of trajectories targeting the Brisbane site.

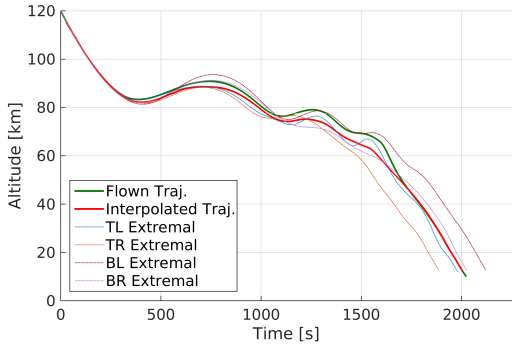

Figure 8.4: Altitude profiles for the test entry with extremals separated 10 deg apart from each other.

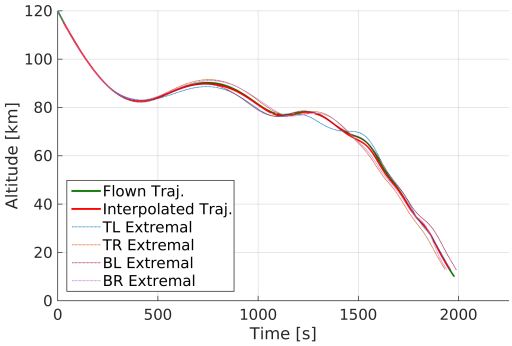

Figure 8.5: Altitude profiles for the test entry with extremals separated 1 deg apart from each other.

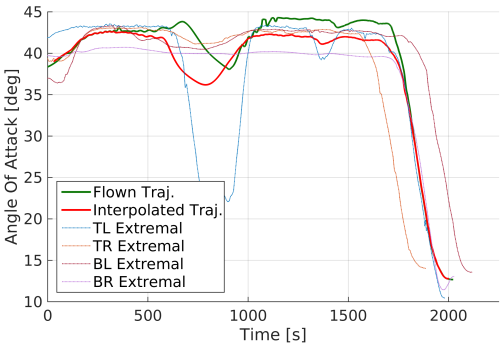

Figure 8.6: Angle-of-attack profiles for the test entry with extremals separated 10 deg apart from each other.

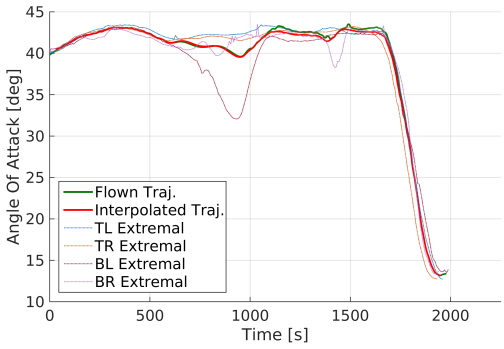

Figure 8.7: Angle-of-attack profiles for the test entry with extremals separated 1 deg apart from each other.

Table 8.2: Monte Carlo dispersion settings used to characterize the guidance performance as a function of the parameter spacing.

Variable	Dispersion Type	Mean	Min/Max	3σ
EIP Altitude	Gaussian	0 m		150 m
EIP Longitude	Uniform		$\pm\Delta p_1/2$ deg	
EIP Latitude	Uniform		$\pm\Delta p_2/2$ deg	
EIP Groundspeed	Gaussian	0 m/s		20 m/s
EIP Flight Path Angle	Gaussian	0 deg		0.1 deg
EIP Heading Angle	Gaussian	0 deg		0.01 deg
Aerodynamic Coefficients	Gaussian	0		10 %
Vehicle Mass	Uniform		$\pm$1 %	
Air Density	Gaussian	0		10 %

dispersion values retrieved from Sagliano et al. (2016) have been restricted due to the fact that the AMPI algorithm employed in this thesis is based on a 2-dimensional parameter space, as compared to the 6-dimensional parameter space used in such study which is expected to provide better dispersion robustness. Note that in Table 8.1, the six-variable entry state is dispersed along with the lift coefficient, the drag coefficient, the vehicle mass and the air density. Note that the entry geodetic latitude and the entry longitude are dispersed according to a uniform distribution whose maximum and minimum values are defined by the particular parameter spacing tested. This ensures that EIPs spanning the whole interpolation grid cell are probed. Finally, note that the six Monte Carlo campaigns are fed with the same seed to ensure a fair comparison between the parameter spacings tested.

Figures 8.8 to 8.13 show the groundtracks produced by the six Monte Carlo campaigns. Please note that such groundtracks originate from the left side of the figures and terminate in the neighborhood of the Ben Guerir landing site, marked in such figures by a green circle. Inspection of the left area of such figures reveals that the EIPs of the 1000 cases tested spread over the entire square grid cell, where the area of such cell is reduced as the parameter spacing is decreased from 10 deg to 0.1 deg. This feature is a direct consequence of the applied geodetic latitude and longitude dispersions, which attempt to probe the entire interpolation cell. In other words, the entry window coverage of a single interpolation cell is proportional to the parameter spacing. Another feature that can be highlighted from the groundtrack plots is that all of the trajectories manage to head towards the landing site, thus proving the effectiveness of the lateral guidance. Despite this, Figures 8.14 and 8.17 show significant dispersions in the "landing" accuracy. A trend is revealed which shows how decreasing the parameter spacing leads to a wider spread of the terminal position dispersions and an an increased mean value of such dispersions. The supporting data that supports these conclusions is presented in Table 8.3, where the standard deviation provides an indication of spread. Figures 8.10 to 8.13 show that a limited number of cases fall significantly further from the landing site as compared to all the other cases of the same Monte Carlo campaign. It is important that to emphasize that these trajectories are removed from the computation of the statistical values given in Table 8.3, but will be considered in the conclusions drawn in Chapter 9.

Contrary to the distance to target dispersions, the guidance algorithm effectively tracks the altitude and groundspeed in almost every case tested, as shown by Figure 8.18. In this figure, the altitude-groundspeed dispersions for the Monte Carlo batch feeding off a grid cell of 2 deg are shown. Here, three trajectories that differ signif-

Table 8.3: Dispersion properties of the terminal position as a function of parameter spacing, using 190 LD nodes.

Parameter Spacing:	**10.0 deg**	**5.0 deg**	**2.0 deg**	**1.0 deg**	**0.5 deg**	**0.1 deg**
Mean [km]	208.43	82.42	37.38	34.49	19.18	14.58
Standard Deviation [km]	114.69	45.08	27.66	22.17	20.35	17.50

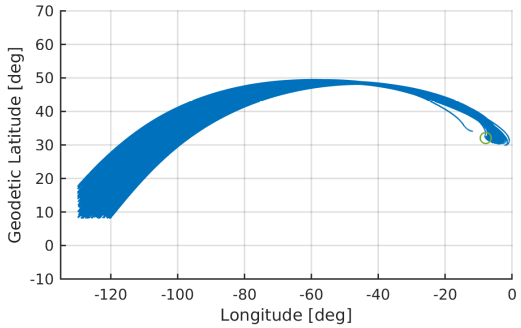

Figure 8.8: Groundtrack dispersions for a parameter spacing of 10 deg.

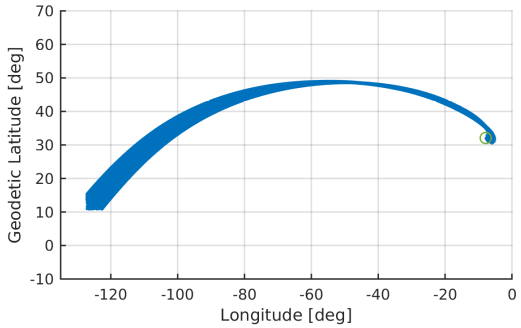

Figure 8.9: Groundtrack dispersions for a parameter spacing of 5 deg.

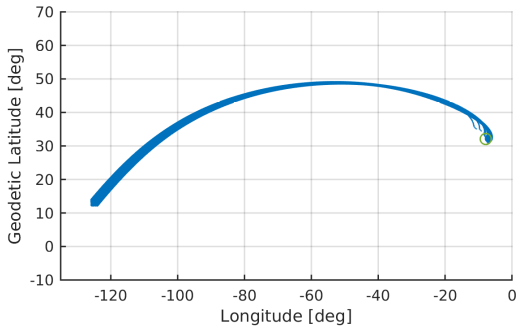

Figure 8.10: Groundtrack dispersions for a parameter spacing of 2 deg.

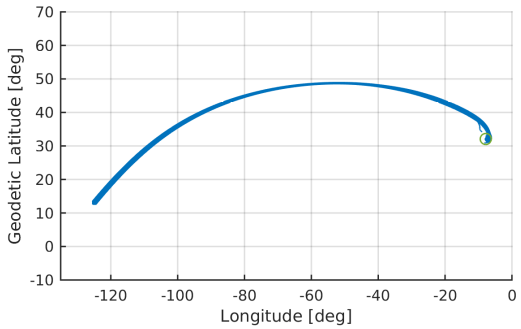

Figure 8.11: Groundtrack dispersions for a parameter spacing of 1 deg.

Table 8.4: Summary of the terminal velocity constraint compliance properties given a database sampled in 190 LD nodes.

$N_{LD} = 190$	**10.0 deg**	**5.0 deg**	**2.0 deg**	**1.0 deg**	**0.5 deg**	**0.1 deg**
Mean M_f [-]	0.669	0.647	0.664	0.663	0.665	0.666
Standard Deviation M_f [-]	0.010	0.006	0.006	0.007	0.011	0.014
Mean $V_{v,f}$ [m/s^1]	118	106	117	116	117	117
Standard Deviation $V_{v,f}$ [m/s^1]	3	7	2	1	2	4

icantly in from those in the main group are highlighted in red, which coincide with those trajectories that fall far excessively far from the landing site, as shown in Figure 8.10. Such issue suggests that the gains of the LQR tracker could require some further tuning and possibly a re-design, where direct tracking of the traveled range could provide a better performance, as discussed in Section 4.2. Despite this, the guidance algorithm is capable of bringing the vehicle to the required altitude and Mach number ran in every single case as specified by **REQ-14**, as summarized in Table 8.4. Here, a summary of the statistical characteristics of final Mach number M_f and the final vertical speed $V_{v,f}$ is given. Please note that all these quantities are retrieved at 10 km altitude, which is the termination condition of the entry simulator. It becomes evident that the parameter spacing does not make a significant impact neither on the final Mach number nor in the final vertical speed, since the mean and the standard deviation of both quantities remain practically unchanged. In fact, the standard deviation of both quantities is relatively small, suggesting robustness to dispersions.

Table 8.5 summarizes the overshoot characteristics of the aerodynamic load and heat-flux dispersions for all

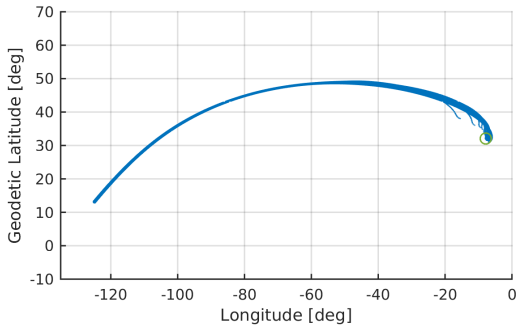

Figure 8.12: Groundtrack dispersions for a parameter spacing of 0.5 deg.

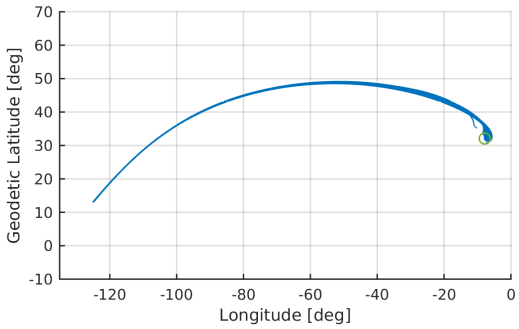

Figure 8.13: Groundtrack dispersions for a parameter spacing of 0.1 deg.

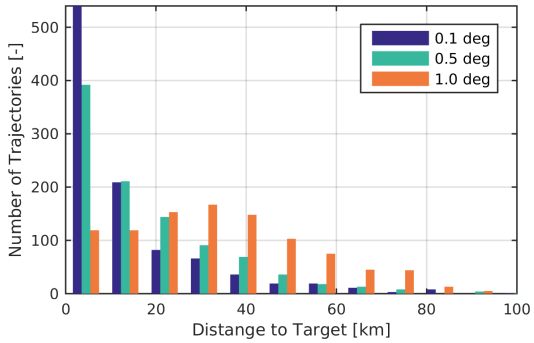

Figure 8.14: Histogram of the distance to target dispersions for parameter spacings of 0.1 deg, 0.5 deg and 1 deg.

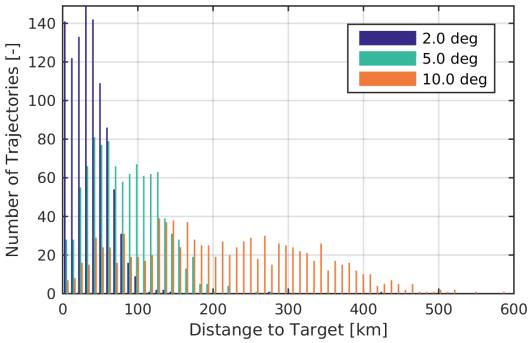

Figure 8.15: Histogram of the distance to target dispersions for parameter spacings of 2 deg, 5 deg and 10 deg.

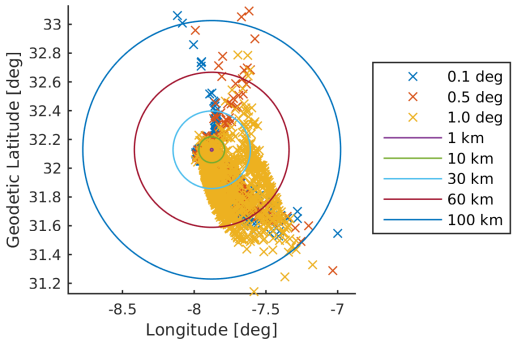

Figure 8.16: Projection of the distance to target dispersions for parameter spacings of 0.1 deg, 0.5 deg and 1 deg.

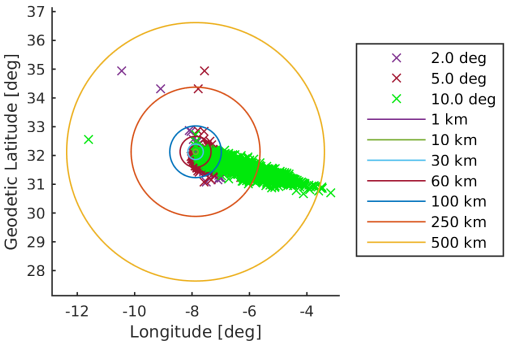

Figure 8.17: Projection of the distance to target dispersions for parameter spacings of 2 deg, 5 deg and 10 deg.

the parameter spacings tested, using 190 LD nodes. It is shown that the 10 deg spacing leads to worst performance in terms of aerodynamic load, which is violated in a third of the cases tested with a mean overshoot value of 3.42 g and maximum overshoot value of 4.11 g. Note that although the spacings of 5 deg to 1 deg result in a large number of overshoots, the mean and maximum values of such overshoots are significantly lower that for 10 deg and within acceptable bounds. Despite this, it becomes evident that the finest spacings of 0.5 deg and 0.1 deg lead to just three overshoot and very close to the maximum of allowable value of 3.0 g established by **REQ-S-01**.

Contrary to the aerodynamic load overshoot results, Table 8.5 reveals that the heat-flux constraint of 670 kW/m^2 established by **REQ-S-03** is complied with in most of the cases. Despite this, some of the tested cases show significantly high maximum heat-flux values, in particular the 10 deg and the 0.5 deg cases. These abnormally high constraint violations correspond to those trajectories that deviate too far from the altitude-groundspeed profile as highlighted above. Finally, it is important to highlight the large number of heat-flux overshoots that appear for the finest spacing. This is related to the fact that extremals that support the interpolation closely

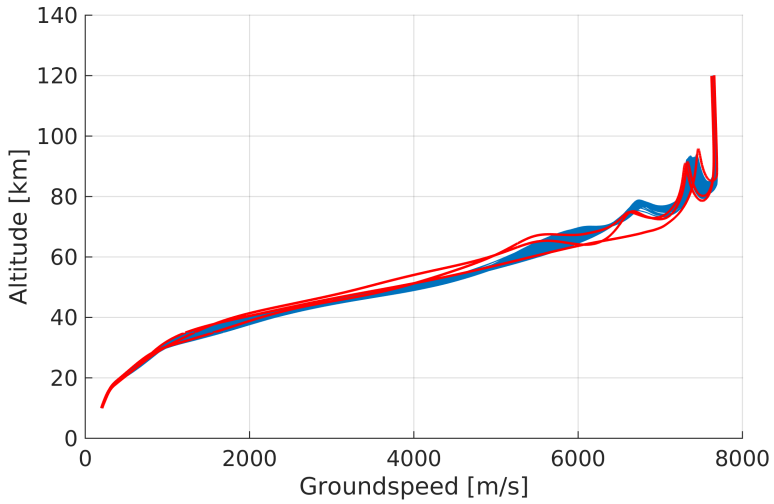

Figure 8.18: Altitude-Groundspeed dispersions for a parameter spacing of 2 deg and 190 LD nodes. Marked in red are example trajectories that differ significantly from the main group.

Table 8.5: Summary of the constraint overshoot characteristics given a database sampled in 190 LD nodes.

$N_{LD} = 190$	**10.0 deg**	**5.0 deg**	**2.0 deg**	**1.0 deg**	**0.5 deg**	**0.1 deg**
Nr. $n_{L,max}$	330	236	506	839	3	3
Mean $n_{L,max}$ [-]	3.42	3.03	3.06	3.09	3.00	3.02
Max. $n_{L,max}$ [-]	4.11	3.12	3.13	3.29	3.01	3.05
Nr. q_{max}	3	1	2	1	56	371
Mean q_{max} [kW/m^2]	754	673	736	693	681	678
Max. q_{max} [kW/m^2]	810	673	753	693	938	749

track the heat-flux constraint and consequently any errors in the reference profile could easily lead to the violation of the constraint. Despite this, the overshoot values are still within acceptable limits and are expected to improve with a revision of the LQR tracker.

Table 8.6 summarizes the statistical parameters for the total angle-of-attack control effort $\sum_{\alpha_c}$ and the total bank-angle control effort $\sum_{\sigma_c}$ are summarized. These are performance indices that provide an indication on how much the guidance system is stressed, where lower values are desired. The total control effort for angle-of-attack and bank-angle is estimated with the following equations (Mooij, 2017):

$$\sum_{\alpha_c} = \int_0^{t_f} \alpha(t)\, dt \tag{8.1}$$

$$\sum_{\sigma_c} = \int_0^{t_f} \sigma(t)\, dt \tag{8.2}$$

where $\sum_{\alpha_c}$ and $\sum_{\sigma_c}$ are the total angle-of-attack and bank-angle control efforts, respectively. Note that the total effort is considered due to intrinsic relation between the nominal and tracking commands (Mooij, 2017). The results shown in Table 8.6 shows that there is no clear trend between the parameter spacing and the control effort tackled by the guidance system. This comes at no surprise, since the maximum allowable deflection are capped by the values established in (4.63) and (4.64). Otherwise, the control effort of those entries that feed off a coarser grid would have been expected to be higher, since as mentioned earlier in this section, such coarse grids would lead to an unfeasible control profiles that stress the guidance system.

In summary, the constraint violation results are in line with the expected trend for the performance of the system as a function of the parameter spacing, where coarser spacings lead to reference trajectories that do not approximate the solutions of the equations of motion sufficiently well, thus forcing the guidance system to track a trajectory that is not flyable in the first place.

Table 8.6: Summary of the control effort properties given a database sampled in 190 LD nodes.

$N_{LD} = 190$	**10.0 deg**	**5.0 deg**	**2.0 deg**	**1.0 deg**	**0.5 deg**	**0.1 deg**
Mean $\sum_{\alpha_c}$ [deg s]	79,162	78,713	78,480	77,539	77,836	77,020
Max. $\sum_{\alpha_c}$ [deg s]	84,561	82,173	81,983	80,283	80,810	80,736
Std. $\sum_{\alpha_c}$ [deg s]	2,726	1,788	2,312	1,348	1,500	2,281
Mean $\sum_{\sigma_c}$ [deg s]	96,262	95,873	89,567	88,483	85,573	87,629
Max. $\sum_{\sigma_c}$ [deg s]	133,600	138,070	145,730	141,240	145,500	147,940
Std. $\sum_{\sigma_c}$ [deg s]	13,296	13,048	13,724	13,881	13,967	14,123

8.2 PERFORMANCE IMPACT OF THE LD NODE NUMBER

All the dispersions discussed so far employed a database sampled on 190 LD nodes. As discussed in Section 4.1.5, decreasing the number of nodes can effectively reduce the size of the database even further. However, a sufficient number of nodes needs to be used to ensure that the trajectory database captures the entry dynamics well enough to produce flyable reference trajectories. The goal of this section is to characterize the impact of the number of LD nodes used on the final dispersions and the compliance of the path constraints. To do so, identical Monte Carlo campaigns to those presented in the previous section are performed.

To begin with, Tables 8.7 and 8.8 show the statistical parameters of the distance to target dispersions for 140 LD nodes and 90 LD nodes, respectively. These dispersions are assembled into the histograms given in Figures 8.19 to 8.22. Comparison with Table 8.3 and Figures 8.14 and 8.15, it becomes evident that reducing the number of LD nodes down to 90 does not have a significant impact in the dispersion characteristics of the distance to target, where discrepancies of less than 6% are found.

In addition, Tables 8.9 and 8.10 show the overshoot characteristics for the six Monte Carlo campaigns while using 140 LD nodes and 90 LD nodes, respectively. Similarly to the final distance-to-target dispersions, it becomes clear that decreasing the number of LD nodes does not have a significant impact in the aerodynamic load and heat flux overshoot characteristics. In fact, similar trends to those shown for 190 LD nodes are seen. To begin with, a parameter spacing of 10 deg proves again to be too coarse, since the maximum aerodynamic load of 3.0 g is largely violated. Using 140 LD nodes leads to a maximum aerodynamic load of 4.59 g where using 90 LD nodes results in a maximum load of 4.27 g. Furthermore, spacings of 5 deg, 2 deg and 1 deg lead to a large reduction in the excess aerodynamic load, although still marginally above the limit, taking mean values that range from 3.04 g to 3.10 g. Once again, only grid spacing values of 0.5 deg and 0.1 deg provide a sufficient reduction in the number of trajectories that violate the aerodynamic load constraint.

In the case of the heat-flux constraint, reducing the number of LD nodes does not change the trends seen for 190 LD nodes. Comparison of Tables 8.5, 8.9 and 8.10 shows that the number of heat-flux overshoots is very small for spacing values of 10 deg, 5 deg, 2 deg and 1 deg. In addition, the corresponding mean and maximum values show that the magnitude of the overshoots are within acceptable bounds. Once again, spacing values of 0.5 deg and 0.1 deg lead to a large number of heat-flux constraint overshoots characterized with an unacceptable maximum values. This is again linked to the performance of the LQR tracker, which produces a limited number of trajectories that deviate too far from altitude-groundspeed corridor early in the trajectory. On the contrary, Tables 8.11 and 8.12 show that reducing the parameter spacing leads to trajectories that still comply with the terminal velocity requirements established in **REQ-14**. This is proven by the fact that all the vertical speed mean values are well below the maximum allowable speed of 140 m/s and the mean Mach number is within the subsonic range at 10 km altitude.

Finally, Tables 8.13 and 8.14 prove that the control effort for both the angle-of-attack and the bank-angle does exhibit a significant change when the number of LD nodes is reduced. Again, this is a direct consequence of the enforced maximum allowable control deviations.

Table 8.7: Dispersion properties of the terminal position as a function of parameter spacing, using 140 LD nodes.

Parameter Spacing:	**10.0 deg**	**5.0 deg**	**2.0 deg**	**1.0 deg**	**0.5 deg**	**0.1 deg**
Mean [km]	222.47	85.51	36.82	34.85	19.43	12.71
Standard Deviation [km]	129.22	47.28	26.00	22.38	20.52	13.60

Table 8.8: Dispersion properties of the terminal position as a function of parameter spacing, using 90 LD nodes.

Parameter Spacing:	**10.0 deg**	**5.0 deg**	**2.0 deg**	**1.0 deg**	**0.5 deg**	**0.1 deg**
Mean [km]	209.91	82.71	35.69	34.47	19.53	13.15
Standard Deviation [km]	118.15	45.03	23.70	21.26	21.90	13.59

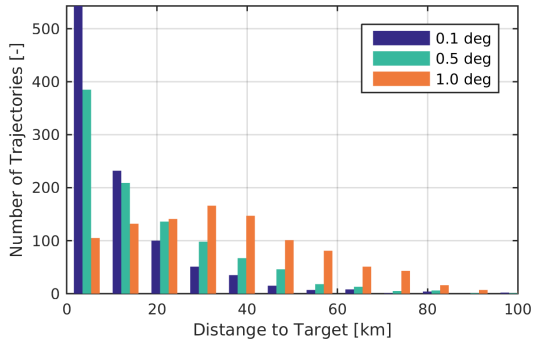

Figure 8.19: Histogram of the distance to target dispersions for parameter spacings of 0.1 deg, 0.5 deg and 1 deg. Interpolation using 140 LD nodes.

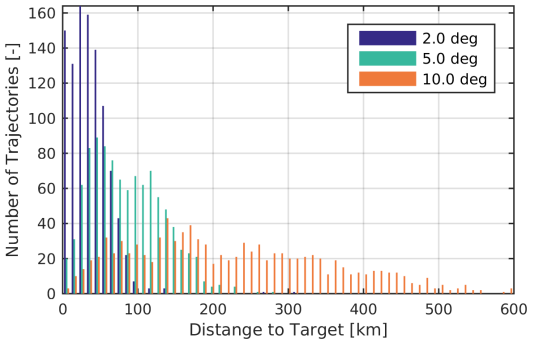

Figure 8.20: Histogram of the distance to target dispersions for parameter spacings of 2 deg, 5 deg and 10 deg. Interpolation using 140 LD nodes.

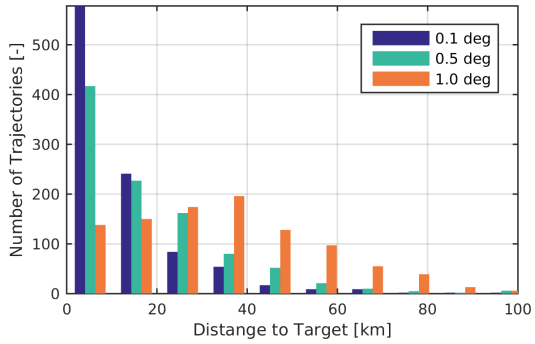

Figure 8.21: Histogram of the distance to target dispersions for parameter spacings of 0.1 deg, 0.5 deg and 1 deg. Interpolation using 90 LD nodes.

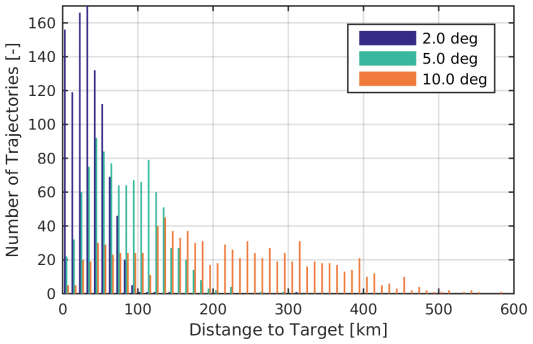

Figure 8.22: Histogram of the distance to target dispersions for parameter spacings of 2 deg, 5 deg and 10 deg. Interpolation using 90 LD nodes.

Table 8.9: Summary of the constraint overshoot characteristics given a database sampled in 140 LD nodes.

$N_{LD} = 140$	**10.0 deg**	**5.0 deg**	**2.0 deg**	**1.0 deg**	**0.5 deg**	**0.1 deg**
Nr. $n_{L,max}$	344	254	506	845	4	3
Mean $n_{L,max}$ [-]	3.52	3.04	3.06	3.10	3.01	3.01
Max. $n_{L,max}$ [-]	4.59	3.14	3.14	3.30	3.02	3.03
Nr. q_{max}	3	1	2	1	57	374
Mean q_{max} [kW/m^2]	723	680	694	696	680	678
Max. q_{max} [kW/m^2]	749	680	714	696	938	698

8.2.1. COMPRESSION OF THE TRAJECTORY DATABASE

In summary, it becomes clear that reducing the number of LD nodes down to 90 results in no significant change in the terminal point dispersions with respect to 190 LD nodes. Furthermore, no relevant differences are found in the aerodynamic load and heat-flux constraint overshoots. In addition, the final Mach number and the final vertical speed are within the requirements established by **REQ-14** and no difference in control effort are seen.

All in all, it is evident that 90 LD nodes are sufficient to capture the reentry dynamics. Consequently, the database size could be successfully reduced without posing a significant impact on the performance of the

guidance system. A summary of the database sizes for global coverage according to the parameter spacing and LD node number is given in Table 8.15. Such table shows that the database size ranges from 40.0 GB for a fine spacing of 0.1 deg and no compression, down to 9 MB for a spacing of 10 deg with the maximum compression. At this point, it is important to recall that the database compression used in this thesis is a loss-less process. In addition, the guidance system developed in this thesis was tested for speed and although the LD-to-HD decompression of the database proved to be the most expensive guidance task, it is still within a few milliseconds. All in all, it becomes clear that 90 LD nodes should be used to achieve the largest compression factor, since it

Table 8.10: Summary of the constraint overshoot characteristics given a database sampled in 90 LD nodes.

$N_{LD} = 90$	**10.0 deg**	**5.0 deg**	**2.0 deg**	**1.0 deg**	**0.5 deg**	**0.1 deg**
Nr. $n_{L,max}$	337	212	521	861	4	2
Mean $n_{L,max}$ [-]	3.45	3.04	3.07	3.10	3.01	3.01
Max. $n_{L,max}$ [-]	4.27	3.15	3.14	3.30	3.02	3.02
Nr. q_{max}	3	4	0	1	51	356
Mean q_{max} [kW/m^2]	716	691	N/A	742	686	679
Max. q_{max} [kW/m^2]	745	716	N/A	742	957	767

Table 8.11: Summary of the terminal velocity constraint compliance properties given a database sampled in 140 LD nodes.

$N_{LD} = 140$	**10.0 deg**	**5.0 deg**	**2.0 deg**	**1.0 deg**	**0.5 deg**	**0.1 deg**
Mean M_f [-]	0.670	0.647	0.660	0.663	0.658	0.665
Std. M_f [-]	0.011	0.006	0.008	0.0061	0.012	0.015
Mean $V_{v,f}$ [m/s^1]	118	106	115	116	114	117
Std. $V_{v,f}$ [m/s^1]	2	7	2	3	4	5

Table 8.12: Summary of the terminal velocity constraint compliance properties given a database sampled in 90 LD nodes.

$N_{LD} = 90$	**10.0 deg**	**5.0 deg**	**2.0 deg**	**1.0 deg**	**0.5 deg**	**0.1 deg**
Mean M_f [-]	0.672	0.649	0.663	0.664	0.663	0.666
Std. M_f [-]	0.011	0.006	0.008	0.007	0.009	0.014
Mean $V_{v,f}$ [m/s^1]	120	107	117	117	118	117
Std. $V_{v,f}$ [m/s^1]	3	7	2	1	3	4

Table 8.13: Summary of the control effort properties given a database sampled in 140 LD nodes.

$N_{LD} = 140$	**10.0 deg**	**5.0 deg**	**2.0 deg**	**1.0 deg**	**0.5 deg**	**0.1 deg**
Mean $\sum_{\alpha_c}$ [deg s]	78,618	78,691	78,679	77,056	77,325	78,767
Max. $\sum_{\alpha_c}$ [deg s]	84,546	82,174	81,976	80,267	80,801	81,318
Std. $\sum_{\alpha_c}$ [deg s]	3,366	1,768	2,126	1,547	1,971	902
Mean $\sum_{\sigma_c}$ [deg s]	97,057	95,990	89,591	88,906	86,184	86,980
Max. $\sum_{\sigma_c}$ [deg s]	140,920	137,130	144,550	142,230	138,630	123,570
Std. $\sum_{\sigma_c}$ [deg s]	13,749	13,842	13,794	14,023	14,283	12,754

Table 8.14: Summary of the control effort properties given a database sampled in 90 LD nodes.

$N_{LD} = 90$	**10.0 deg**	**5.0 deg**	**2.0 deg**	**1.0 deg**	**0.5 deg**	**0.1 deg**
Mean $\sum_{\alpha_c}$ [deg s]	79,143	78,882	79,844	77,523	78,105	78,915
Max. $\sum_{\alpha_c}$ [deg s]	84,663	82,167	81,975	80,257	80,833	81,639
Std. $\sum_{\alpha_c}$ [deg s]	2,708	1,445	787	993	1,021	996
Mean $\sum_{\sigma_c}$ [deg s]	96,658	95,105	89,749	88,178	86,190	87,179
Max. $\sum_{\sigma_c}$ [deg s]	136,140	133,880	129,310	133,730	132,670	123,680
Std. $\sum_{\sigma_c}$ [deg s]	12,787	13,251	12,686	13,959	13,316	12,403

poses no significant drawbacks on the guidance performance. Finally, according to Larson and Wertz (2005) the flash memory of commercially available radiation-hardened on-board computers can amount up to 16 GB, thus proving that the trajectory database can be stored on-board even if the finest grid spacing is used.

Table 8.15: Estimated database size values for 190, 140 and 90 LD nodes.

Parameter Spacing:	**10.0 deg**	**5.0 deg**	**2.0 deg**	**1.0 deg**	**0.5 deg**	**0.1 deg**
No Compression	400 MB	800 MB	2.0 GB	4.0 GB	8.0 GB	40.0 GB
$N_{LD} = 190$	19 MB	38 MB	95 MB	190 MB	380 MB	1.9 GB
$N_{LD} = 140$	14 MB	28 MB	70 MB	140 MB	280 MB	1.4 GB
$N_{LD} = 90$	9 MB	18 MB	45 MB	90 MB	180 MB	900 MB

9 Conclusions & Recommendations

This is the final chapter of this thesis and it summarizes all the findings made during the course of the project, as well as recommendations for future work.

9.1 Conclusions

This thesis assignment began with the following research question:

How can a reentry vehicle return from low-Earth orbit in an autonomous, reliable and timely manner while considering the limitations of current flight hardware?

To better answer this question, the following sub-questions are posed:

1. How are the different orbital abort scenarios characterized in terms of EIP position and velocity?

   All the orbital abort scenarios are translated to a set of initial entry conditions where the altitude, groundspeed and flight-path angle are kept to their nominal values and the latitude, longitude and heading angle take a wide range of values. Note that the nominal values of the altitude, groundspeed and flight-path angle are 120 km, 7640 m/s and -1.28 deg, respectively. The trajectory to be flown is primarily affected by the ground projection of the EIP and whether the vehicle reaches the EIP with a North-East heading (ascending leg) or a South-East heading (descending leg).

2. How are the different orbital abort scenarios characterized in terms of the distribution of emergency landing sites?

   The available landing sites are classified into two main groups: The Northern Atlantic group includes the Azores airport, the Gran Canaria airport and the Ben Guerir air force base. The Australian group is formed by the Woomera Airfield and the Perth, Darwin, Brisbane and Sidney airports. If the projected EIP lays on the Western hemisphere with a North-East heading, all the landing sites in the Northern Atlantic group are reached in relatively short descent due to the close proximity of the targeted sites. Furthermore, most of the sites in the Australian group are reached by putting the vehicle into a full-lift up configuration that puts SPHYNX in a skip-entry, thus extending its downrange capability. Similarly, if the projected EIP lays on the Eastern hemisphere with a South-East heading, the Australian and Northern Atlantic sites are reached by establishing no-skip and skip entries, respectively.

3. What design challenges do such abort scenarios pose on the guidance system?

   The large range of initial conditions that the abort guidance system needs to handle stresses the trajectory planner, which generates reference command profiles on-board via interpolation of an optimal-trajectory database. Such database shall cover all the possible EIP configurations and landing sites that the operational orbit permits. Consequently, a computational time equivalent to 33 full days on a modern laptop CPU is required to generate such database, which does not consider the significant man-power required to post-process the data. In addition, it is estimated that approximately 40 GB of storage space are required on-board to accommodate an un-processed high-density database.

4. How effective is the proposed guidance system in tackling such challenges?

   (a) *How is the computational effort required to produce the trajectory database managed?*

   A number of scripts that populate the database in an autonomous manner are developed. These scripts systematically probe multiple EIPs and prematurely abort any optimization tasks that are unfeasible. This method allows to minimize the time employed to interact with the optimization software while allowing simulation tasks to run overnight. In addition, the computing time is re-

duced further by running parallel optimization tasks on a high-performance server-grade CPU provided by ESA.

(b) *How is the impact that the system poses on the on-board computer minimized?*

The size of the database can be compressed using a loss-less algorithm without compromising performance. Such compression is determined by the low-density node number, where lower numbers offer a stronger compression factor. For instance, a 90 low-density node spacing reduces the database to 2.25% of its original size, ultimately yielding manageable database sizes that range from 9 MB to 900 MB depending on the density of the database.

It is found that the database decompression process of the stored database proves to be the most expensive routine within the guidance system. Still, thorough testing has shown that trajectories can be planned in the order of a few milliseconds, proving its capability to run on-board. Finally, the system can handle multiple abort scenarios by selecting the most appropriate emergency landing site given the information provided by the navigation system.

5. How does the choice for emergency landing sites influence on the system's safety?

(a) *How does the system perform in terms of providing zero-wait entry capability?*

Given the available landing sites, full global time-coverage is not guaranteed. This means that a safe landing site cannot be guaranteed irrespectively of the epoch at which the de-orbit burn is initiated. With the current sites, approximately 50% global time-coverage is provided, meaning that landing sites can be targeted in 50% of the time. This translates to a maximum in-orbit standby time of approximately 10 hours.

(b) *How is the system characterized in terms of providing in-flight abort opportunities?*

Results from the reachability study show that shared entry windows are found, since all sites share over 50% of its entry window with at least one alternative site. Although the ability to reroute to an alternative site largely depends on the total energy available, the results shown in this thesis show important re-routing capabilities prior reaching the EIP. Note that any re-routing operation can be achieved by simply selecting a different trajectory subspace within the database.

6. How do the different guidance system design parameters influence on its performance?

(a) *How does the database parameter spacing impact the system performance?*

It is found that the final distance-to-target dispersions are strongly affected by the parameter spacing. Results show that coarser spacings increase the mean and spread of such dispersions. In fact, none of the parameter spacings tested complies with the required 10 km distance. Furthermore, the parameter spacing significantly affects the compliance with the aerodynamic load constraint, where coarser grid spacings increase the mean and peak value of the overshoots. Given the current state of the system, spacing values of less than 0.5 deg are suggested since these result in a marginal violation of the maximum allowable values as established in the system requirements.

(b) *How does the low-density node number impact the system performance?*

It is found that the low-density node number leads to no significant differences in constraint compliance, final distance-to-target dispersions nor in the control effort. Note that the trajectories in the database were sampled with at least 90 low-density nodes, where reducing this number further could lead to unfeasible reference control profiles.

9.2 RECOMMENDATIONS

In addition to the conclusions drawn in the previous section, the following recommendations are suggested for future work:

- The SPHYNX vehicle aerodynamic database employed in this report was not complete and presents relatively rapid changes in the drag coefficient in the supersonic region. This issue results in a behavior of the vehicle in the supersonic region that is difficult to control. In addition, the absence of aerodynamic coefficient data in the hypersonic region creates a bottleneck in the $\alpha - M$ space that severely limits the capability of finding optimal trajectories. In further research, this aerodynamic database should be

revised and completed. It could also be a possibility to replace the vehicle entirely, for one of similar characteristics.

- The lateral guidance implemented in this thesis is based on a simple HEDB logic with constant ±25 deg dead band limits. Tuning of the dead band limits or perhaps replacing the HEDB logic by a dynamic lateral guidance such as the one presented by Zuojun and Lu (2004) could reduce the number of bank reversals towards the end of the trajectory.
- The bank reversals in this thesis are instantaneous and thus modeled with an infinite rate of change of bank angle. In future work, a finite rate should be implemented to assess the impact of the bank reversals in the flight-path error and the terminal dispersions.
- In a limited number of cases, the implemented guidance brings the vehicle excessively far from the landing site and puts the vehicle in a trajectory that results in large heat flux constraint violations. This is caused by failure of the LQR to track well the altitude and groundspeed in the early portions of the flight. More thorough testing of the LQR tracker and additional tuning of the guidance gains is suggested for future work. In addition, an LQR tracker where the traveled range is directly tracked could provide improve the robustness to dispersions.
- The robustness of the guidance logic could be improved by merging the interpolation-based trajectory planner with a predictor-corrector scheme. This method could potentially reduce the dispersion at the terminal point and even bring the vehicle to the neighborhood of a landing site. Such predictor-corrector guidance could be activated when the vehicle is in the proximity of the landing site. An example of successful merging of trajectory tracking and a predictor-corrector scheme is given in (Webb and Lu, 2016).
- A full system test where the choice for landing site is left free is recommended. A similar dispersions analysis would need to be made given the landing site selected prior entry. This would require to add an additional guidance module where a certain set of criteria is used to select the most appropriate site. Such criteria could include landing preference or the maximum foreseen aerothermal loads, among others.
- The choice for the emergency landing sites should be revised, since most of the given sites include international airports and are in close proximity to populated areas. In addition, the optimal placement of such sites should be studied, since it was found that some of the selected sites provide no additional landing coverage.

BIBLIOGRAPHY

Ancarola, B. P. [2002]. "Ariane 5 Performance Optimization for Interplanetary Mission", *AIAA Astrodynamics Specialist Conference and Exhibit*. 2002-4902.

Anderson Jr., J. D. [2006a]. *Hypersonic and High Temperature Gas Dynamics*, chapter 4, AIAA Education Series, pp. 107–111.

Anderson Jr., J. D. [2006b]. *Hypersonic and High Temperature Gas Dynamics*, chapter 6, AIAA Education Series, pp. 332–333.

Anderson Jr., J. D. [2006c]. *Hypersonic and High Temperature Gas Dynamics*, chapter 6, AIAA Education Series, p. 292.

Anderson Jr., J. D. [2006d]. *Hypersonic and High Temperature Gas Dynamics*, chapter 6, AIAA Education Series, p. 299.

Betts, J. T. [1998]. "Survey of Numerical Methods for Trajectory Optimization", *Journal of Guidance, Control, and Dynamics*, volume 21, Paper nr. 2, pp. 193–207.

Burden, R. L. and Faires, J. D. [2011]. *Initial-Value Problems for Ordinary Differential Equations*, chapter 5, Richard Stratton, pp. 259–355.

Castellini, F., Lavagna, M. R., Riccardi, A., and Buskens, C. [2010]. "Multidisciplinary Design Optimization Models and Algorithms for Space Launch Vehicles", *AIAA Multidisciplinary Analysis Optimization Conference*.

Dukeman, G. [2002]. "Profile-Following Entry Guidance Using Linear Quadratic Regulator Theory", *AIAA Guidance, Navigation, and Control Conference*. 2002-4457.

Dwight, R. P. [2013]. "AE2212 Applied Numerical Analysis: Course Reader", Technical report, Delft University of Technology.

ECSS Secretariat [2009]. "Software", in *Space Engineering*, European Cooperation for Space Standardization, ESA-ESTEC, Requirements & Standards Division, Noordwijk, The Netherlands.

European Space Agency [2016]. "Private Communication: SPHYNX Flight Mechanics", unpublished.

Gerard, Y. and Tumino, G. [2005]. "FLPP Re-Entry In-Flight Experimentation: The Intermediate Experimental Vehicle (IXV) Technical and Programmatic Objectives", *International Astronautical Congress*, , Paper nr. IAC-05-D2.P.04, doi:10.2514/6.IAC-06-D2.6.05.

Harpold, J. C. and Graves, C. A. [1979]. "Shuttle Entry Guidance", *American Astronautical Society*, volume 27, Paper nr. 3, pp. 239–268.

Joosten, K. B. [1985]. "Descent Guidance and Mission Planning for the Space Shuttle", in N. Chaffee (editor), *Space Shuttle Technical Conference*, Johnson Space Center, pp. 113–124.

Justus, C. G. and Johnson, D. L. [1999]. "The NASA/MSFC Global Reference Atmospheric Model - 1999 Version (GRAM-99)", Technical report, NASA. NASA/TM-1999-209630.

Kerr, M., Haya, R., Peñin, L. F., De Zaiacomo, G., Mostaza, D., and Marco, V. [2012]. "IXV Re-entry Guidance, Control & DRS Triggering: Algorithm Design and Assessment", *AIAA Guidance, Navigation, and Control Conference*. 2012-4841.

Klaedtke, R., Puerttmann, N., and Graf, E. [1999]. "X-38 European Partnership", *American Institute of Aeronautics & Astronautics*, , Paper nr. AIAA-99-4936, doi:10.2514/6.1999-4936.

Larson, W. J. and Wertz, J. R. [2005]. *Space Mission Analysis and Design*, chapter 16, Microcosm Press, p. 669.

Montenbruck, O. and Gill, E. [2001a]. *Satellite Orbits: Models, Methods and Applications*, chapter 5, Springer, pp. 185–189.

Montenbruck, O. and Gill, E. [2001b]. *Satellite Orbits: Models, Methods and Applications*, chapter 4, Springer, pp. 117–156.

Mooij, E. [1995]. "The HORUS-2B Reference Vehicle", Technical report, Delft University of Technology.

Mooij, E. [1997]. *The Motion of a Vehicle in a Planetary Atmosphere*, Delft University Press.

Mooij, E. [1998]. "Aerospace-Plane Flight Dynamics: Analysis of Guidance and Control Concepts", Ph.D. thesis, Delft University of Technology.

Mooij, E. [2014]. "AE4870B Re-entry Systems: Draft Lecture Notes", Technical report, Delft University of Technology.

Mooij, E. [2017]. "Re-entry Guidance for Path-Constraint Tracking", *AIAA Guidance, Navigation, and Control Conference*, volume 2014-4142.

Muratore, J. P. E. [2009]. "Emergency Systems", in G. E. Musgrave, A. M. Larsen, and T. Sgobba (editors), *Safety Design for Space Operations*, chapter 7, Butterworth-Heinemann, pp. 225–266.

Naftel, J. C. and Talay, T. A. [1993]. "Ascent Abort Capability for the HL-20", *Journal of Spacecraft and Rockets*, volume 30, Paper nr. 5, pp. 628–634.

NASA and USAF and NOAA [1976]. "US Standard Atmosphere 1976", Technical report, NASA, USAF and NOAA. NASA-TM-X-74335.

NIMA [2000]. "World Geodetic System 1984: Its Definition and Relationships with Local Geodetic Systems", Technical report, NIMA.

Sagliano, M., Mooij, E., and Theil, S. [2016]. "Onboard Trajectory Generation for Entry Vehicles via Adaptive Multivariate Pseudospectral Interpolation", *AIAA Guidance, Navigation, and Control Conference*. 2016-2115.

Sagliano, M., Samaan, M., Theil, S., and Mooij, E. [2014]. "SHEFEX-3 Optimal Feedback Entry Guidance", *AIAA SPACE 2014 Conference and Exposition*. 2014-4208.

Saraf, A., Leavitt, J. A., Chen, D. T., and Mease, K. D. [2004]. "Design and Evaluation of an Acceleration Guidance Algorithm for Entry", *Journal of Spacecraft and Rockets*, volume 41, Paper nr. 6, pp. 986–996.

Stone, H. W. and Piland, W. M. [1993]. "21st Century Space Transportation System Design Approach: HL-20 Personnel Launch System", *Journal of Spacecraft and Rockets*, volume 30, Paper nr. 5, pp. 521–528.

Tapley, D. T., Schutz, B. E., and Born, G. H. [2004a]. *Statistical Orbital Determination*, chapter 2, Elsevier Academic Press, pp. 29–32.

Tapley, D. T., Schutz, B. E., and Born, G. H. [2004b]. *Statistical Orbital Determination*, chapter 2, Elsevier Academic Press, pp. 77–81.

Tapley, D. T., Schutz, B. E., and Born, G. H. [2004c]. *Statistical Orbital Determination*, chapter 2, Elsevier Academic Press, pp. 56–61.

Wakker, K. F. [2015]. *Fundamentals of Astrodynamics*, Delft University of Technology.

Wang, M., Tang, Q., and Pei, T. [2006]. "Fast Optimization of Constrained Reentry Trajectory", *International Astronautical Congress*. IAC-06-C1.4.7.

Webb, K. D. and Lu, P. [2016]. "Entry Guidance by Onboard Trajectory Planning and Tracking", *AIAA Atmospheric Flight Mechanics Conference*. 2016-027.

Wie, B. [2008]. *Space Vehicle Dynamics and Control*, chapter 5, AIAA Education Series, pp. 323–347.

Xue, S. and Lu, P. [2010]. "Constrained Predictor-Corrector Entry Guidance", *Journal of Guidance, Control, and Dynamics*, volume 33, Paper nr. 4, pp. 1273–1281.

Yabar Valles, C. [2016]. "Branching Trajectories for PRIDE - Requirements Document", Technical report, European Space Agency.

Zaccagnino, E., Malucchi, G., Marco, V., Drocco, A., Dussy, S., and Preaud, J. P. [2011]. "Intermediate eXperimental Vehicle (IXV), the ESA Re-entry Demonstrator", *AIAA Guidance, Navigation, and Control Conference*. 2011-6340.

Zuojun, S. and Lu, P. [2004]. "Dynamic Lateral Entry Guidance Logic", *AIAA Guidance, Navigation, and Control Conference and Exhibit*, , Paper nr. 2004-4773.

A List of Constants

The goal of this appendix is to list all the physical constants used throughout this thesis. Such constants are listed in Table A.1.

Table A.1: List of physical constants.

Variable	Description	Value	Units
f	Earth's flattening parameter	3.32761797	-
J_2	J_2 term of the gravity field	0.001082627	-
N_A	Avogadro's number	$6.0221409 \cdot 10^{23}$	particles/mol
R^*	universal gas constant	8.31432	J/mol/K
R_E	equatorial radius	$6.378136 \cdot 10^{6}$	m
γ	ratio of specific heats	1.4	-
σ	Stefan-Boltzmann constant	$5.6704 \cdot 10^{-8}$	$W/m^2/K^4$
μ_E	Earth's gravitational parameter	$3.98600435436 \cdot 10^{14}$	m^3/s^2
ω_E	Earth's rotational speed	$7.2921150 \cdot 10^{-5}$	rad/s

B US76 Standard Atmosphere

In this appendix, supplementary information used to define the US76 Standard Atmosphere model is given. Table B.1 provides multiple parameters used to compute the temperature profiles in the upper layers. Figure B.1, shows a sketch with the altitude limits and thermal profiles of each layer in the model.

Table B.1: Parameters for the upper layers of the US76 atmosphere model (NASA and USAF and NOAA, 1976)

Parameter	Value	Unit
T_c	263.1905	K
A	-76.3232	K
h_8	91	km
b	19.9429	km
T_9	240	K
L_9	12	K/km
h_9	110	km
$Tref$	1000	K
T_{10}	360	K
λ	0.01875	km^{-1}
h_{10}	120	km

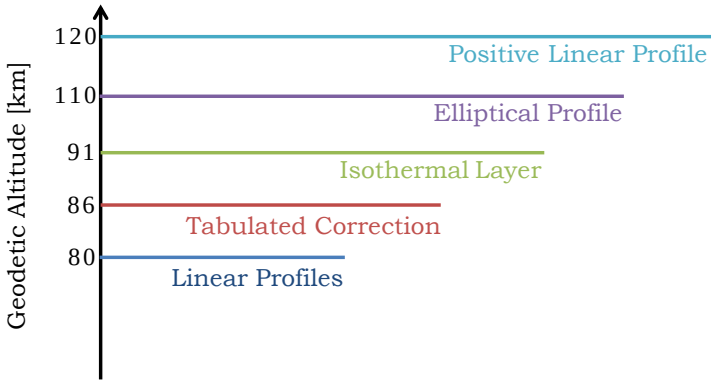

Figure B.1: Definition of the various temperature layers in the US76 standard atmosphere.