Mission Planner for Heating-Optimal Re-Entry Trajectories with Extended Range Capability

Zita Papp

Mission Planner for Heating-Optimal Re-Entry Trajectories with Extended Range Capability

by

Zita Papp

in partial fulfillment of the requirements for the degree of

Master of Science

in Aerospace Engineering

at the Delft University of Technology,

to be defended publicly on Tuesday December 16, 2014 at 10:00 AM.

Supervisor:	Dr. ir. E. Mooij	
Thesis committee:	Prof. dr. ir. P. N. A. M. Visser,	TU Delft
	ir. K. Cowan,	TU Delft

Acknowledgments

I hereby take this opportunity to thank my wonderful thesis supervisor, Erwin Mooij, for his support, great ideas, and incredible patience. During the past year I have learned a lot, not only about my thesis topic but about the attitude and approach needed to accomplish such a project – and not to mention a lot of $C++$ syntax. I consider these useful lessons for life.

I would also like to thank the wonderful 9th floor for the great atmosphere (pun intended) and lovely work environment, not to mention the help I have received from numerous people. I will definitely miss the "gezelligheid" of the student rooms, though I look forward to not having to fight over the last remaining monitor.

Finally, I would like to extend a thank-you to my wonderful parents Anna and Zoltán who have supported me throughout my studies, and just spent a weekend reading and commenting on a 150-page document without complaint. Also my awesome boyfriend Derek for his unending patience and great motivational speeches exactly at the times when I needed them. I could never have have done it without their support, and I will be always thankful.

Abstract

Atmospheric entry is a process defined by extremes, from the great velocity at the point of entry to the high deceleration loads and aerodynamic heating in the lower layers of the atmosphere. In addition to the vehicle maintaining structural integrity, an equally important aspect of the re-entry mission is safety. This safety can only be guaranteed if the vehicle is controllable throughout the re-entry process. Mission design plays a role in adhering to operational and safety requirements in the form of trajectory planning: the development of a re-entry trajectory wherein the vehicle remains within its operational constraints and is controllable throughout. The focus of this thesis lies on the design-time trajectory development part of mission design, i.e., the development and analysis of feasible trajectories under certain requirements to be flown for a specific mission. The research goal of this thesis is formulated as follows: to what extent can optimal re-entry trajectories be developed in the design-time phase of mission development for a winged entry vehicle that provide a maximum-range capability under the objective of minimizing heat loads and adhering to operational constraints? To answer this question, a mission design tool is developed in four successive stages: 1) Development of the re-entry simulator, 2) Design of the guidance algorithm, 3) Development of the mission planner, and 4) Mission planner testing.

The capacity to quickly and reliably simulate re-entry trajectories is paramount to a mission planner. For this purpose, a simulator was developed with the specific goal of later integration with the mission planner. Steering is achieved by modulating the vehicle's attitude in terms of its angle of attack and bank angle. The guidance profile is based on the specification of attitude commands at specific points in the trajectory related to the instantaneous energy of the vehicle. The course of an entire trajectory can be specified by its guidance profile. The purpose of the mission planner is to develop guidance profiles for trajectories that keep the vehicle within its operational constraints, minimize the heat load, and provide the largest possible range under these conditions. The objectives of minimum heat load and maximum range are conflicting. Trajectories with minimum heat load requirements are generally short in duration with smooth heat flux profiles where the heat flux is maintained close to its constraint value. The total heat load is minimized by keeping the duration of the re-entry as short as possible. This, however, is in direct opposition to the objective of maximizing range, where keeping the vehicle aloft for as long as possible is beneficial. The mission planner develops trajectories by specifying the individual parameters in this guidance profile by performing multiobjective optimization to determine the combinations of parameters that result in optimal trajectories in terms of the mission objectives.

The mission planner consistently provides a set of optimal trajectories over a diverse range of objective values and under the provision that operational constraints are met. Optimal trajectories are determined based on these conflicting objectives over a range of objective values, wherein the relative priorities of the objectives are varied. In all cases is the mission planner able to provide trajectories with an extended-range capability, even when minimizing the heat load is considered the main priority.

Acronyms

ARF	Aerodynamic Reference Frame
BFRF	Body-Fixed Reference Frame
c.o.g.	center of gravity
c.o.m.	center of mass
DE	Differential Evolution
DOF	degree of freedom
EA	Evolutionary Algorithm
EI	entry interface
ESA	European Space Agency
GNC	Guidance, Navigation and Control
HPMP	Horus Preliminary Mission Planner
IPRF	Inertial Planetocentric Reference Frame
LEO	low-Earth orbit
MOAD	Multiobjective Optimization Algorithm
MOEA	Multiobjective Optimization Algorithms
MOEA/D	Multiobjective Evolutionary Algorithm based on Decomposition
MOP	Multiobjective Optimization Problem
NED	North-East-Down
NUIN	non-uniform independent node control
ODE	ordinary differential equation
PaGMO	Parallel Global Multiobjective Optimizer
PRNG	Pseudorandom Number Generator
RK4	fourth-order Runge-Kutta
RKF45	Runge-Kutta-Fehlberg
RNG	Random Number Generator

RPRF	Rotating Planetocentric Reference Frame
TPS	Thermal Protection System
TRF	Trajectory Reference Frame
TUDAT	TU Delft Astrodynamics Toolbox
US62	United States Standard Atmosphere 1962
US76	United States Standard Atmosphere 1976
VRF	Vertical Reference Frame

List of Symbols

$\mathbf{a}$	acceleration vector (m s^{-2})
a	acceleration (m s^{-2})
a	speed of sound (m s^{1})
b_{ref}	aerodynamic reference length (m)
c_{ref}	aerodynamic reference length (m)
C_D	aerodynamic drag coefficient (-)
C_L	aerodynamic lift coefficient (-)
C_l	aerodynamic roll moment coefficient (-)
C_m	aerodynamic pitch moment coefficient (-)
C_n	aerodynamic yaw moment coefficient (-)
C_S	aerodynamic side-force coefficient (-)
D	aerodynamic drag force (N)
$\hat{E}$	total normalized specific energy (-)
E	total specific energy (J)
$\mathbf{F}$	force vector (N)
F	force (N)
$\mathbf{F_A}$	aerodynamic force vector (N)
$\mathcal{F}_\mathbf{A}$	Aerodynamic Reference Frame
$\mathcal{F}_\mathbf{B}$	Body-Fixed Reference Frame
$\mathbf{F_G}$	gravitational force vector (N)
$\mathcal{F}_\mathbf{I}$	Inertial Planetocentric Reference Frame
$\mathcal{F}_\mathbf{R}$	Rotating Planetocentric Reference Frame
$\mathcal{F}_\mathbf{T}$	Trajectory Reference Frame
$\mathcal{F}_\mathbf{V}$	Vertical Reference Frame
G	universal gravity constant (m^{3}kg^{-1}s^{-2})
$\mathbf{g}$	gravitational acceleration vector (m s^{-2})
g	gravitational acceleration (ms^{-2})
g_0	gravitational acceleration at sea level (ms^{-2})
h	(geopotential) altitude (m)
k	Boltzmann constant; $k = 1.380622 \cdot 10^{-22}$ (N m K^{-1})
L	aerodynamic roll moment (Nm)
L	aerodynamic lift force (N)
L_K	kinetic temperature gradient (K km^{-1})
L_M	molecular scale temperature gradient (K km^{-1})
M	Mach number (-)
M	aerodynamic pitch moment (Nm)
m	mass (kg)
$\mathbf{M_A}$	aerodynamic moment vector (Nm)

M	Molecular weight (kg mole^{-1})
M_0	Mean molecular weight at sea level (kg mole^{-1})
N	aerodynamic yaw moment (Nm)
N_A	Avogadro constant; $N_A = 6.022169 \cdot 10^{26}$ (kmol^{-1})
n_g	g-load (-)
p	pressure (Nm^{-2})
Q	total heat (J m^{-2})
$\bar{q}$	dynamic pressure (N m^{-2})
$\dot{q}_c$	heat flux rate kW/m^2/s
q_c	heat flux density (W/m^2)
R	Radial distance (m)
$\mathbf{r}$	position vector (m)
R_E	Earth equatorial radius (m)
R^*	universal gas constant (J mole^{-1} K^{-1})
R	gas constant for air (J mole^{-1}K^{-1})
R_N	nose radius (m)
S_{ref}	aerodynamic reference area (m^2)
S	aerodynamic side force (N)
T	(kinetic) temperature (K)
t	time (s)
T_M	molecular temperature (K)
U	gravitational potential (m^2 s^{-2})
$\mathbf{V}$	Velocity vector (m s^{-1})
V	Velocity (m s^{-1})
v_D	velocity in the down direction in the North-East-Down (NED) coordinate system (Cartesian representation) (m/s)
v_δ	velocity in lateral direction ($\mathcal{F}_{\mathbf{V}}$-frame) (m s^{-1})
v_E	velocity in the East direction in the NED coordinate system (Cartesian representation) (m/s)
V_G	groundspeed (m s^{-1})
v_N	velocity in the North direction in the NED coordinate system (Cartesian representation) (m/s)
v_R	velocity in radial direction ($\mathcal{F}_{\mathbf{V}}$-frame) (m s^{-1})
v_τ	velocity in longitudinal direction ($\mathcal{F}_{\mathbf{V}}$-frame) (m s^{-1})
X_{dr}	downrange (m)
z	geometric altitude (m)
ε	emissivity (-)
σ_B	Boltzmann's constant $5.667 \cdot 10^{-8}$(W/m^{2}K^4)

Contents

Chapter 1

Introduction

The Earth's atmosphere is continuously bombarded by objects of various sizes every single day; however, very few worth mentioning actually end up reaching the surface. The ones that do are charred fragments of their former selves, and are likely to cause damage due to their extremely high velocity. On February 15th 2013, a meteor exploded in a bright flash above the Chelyabinsk Oblast in Russia; its dispersed fragments and shockwave damaged over 7,000 buildings and injured over 1,500 people (Popova et al., 2013). Entering the atmosphere at a speed of 19 km/s, this meteor measuring about 20 m in diameter possessed enough kinetic energy to rival over twenty "Little Boy" atomic bombs (Yeomans and Chodas, 2013). For any space mission wherein a return to Earth is required, avoiding a similar fate is far from a given. Naturally a re-entry vehicle is dissimilar to a ten-thousand ton rock in many ways, but the kinetic energy it possesses at entry does not lie: over 30 MJ of kinetic energy per kilogram of vehicle makes the Space Shuttle the energy equivalent of over 650 tons of TNT. Allowing this amount of energy to be dissipated by the atmosphere alone would be catastrophic.

Returning to Earth in one identifiable piece is, however, not the only concern during a re-entry mission. Just as for any other part of a space mission, safety is paramount; not only for possible onboard personnel, but also for people on the ground that may be affected. The vehicle must resist the wear and tear of its return journey so its (often human) cargo is protected, and land in a predefined safe landing location in such a way that it does not cause damage to itself or the surrounding environment. It has to follow a re-entry trajectory wherein its design limits are not breached and the safety of people on the surface is guaranteed. Achieving this comes down to both vehicle design (e.g., structural integrity, heat shields) and mission design.

Atmospheric entry is a process defined by extremes. From its great velocity at the point of entry to the staggering deceleration loads and aerodynamic heating in the lower layers of the atmosphere, a re-entry vehicle has a lot to contend with on its way home. And just survival is not enough: in addition to the vehicle (and possible crew) remaining in one identifiable piece throughout the trajectory, an equally important aspect of the re-entry mission is the safety of anyone or anything on the ground (or in the air) that may be affected by it. This safety can only be guaranteed if the vehicle is controllable throughout the re-entry process, allowing it to be steered towards a predefined landing location were it is to arrive at a specified time.

Re-entry trajectories can take three general forms depending on the properties of the vehicle and the conditions at entry: ballistic, gliding, and skipping. Ballistic entry trajectories correspond to vehicles with minimal capacity to generate lift such as capsules; this method of re-entry is the shortest and incurs the highest loads. Gliding entry is reserved for lifting vehicles such as the Space Shuttle, and is characterized by a long range and duration with much lower incurred loads. Skipping entry is unique in the sense that both low-lift and high-lift have the capacity to perform it. Low-lift vehicles require a high entry velocity associated with e.g., Lunar return, while lifting vehicles are capable of inducing skips even from low-Earth orbit. The method offers the benefit of extending the re-entry range of the vehicle, in addition to a large part of the vehicle's deceleration taking place in the upper layers of the atmosphere where aerodynamic loading is less severe.

Mission design plays a role in adhering to operational and safety requirements in the form of trajectory planning: the development of a re-entry trajectory wherein the vehicle remains within its operational constraints and is controllable throughout. The distinction is made between design-time trajectory planning, i.e., the development and analysis of feasible trajectories under certain requirements to be flown for a specific mission, and on-the-fly trajectory planning which involves the real-time computation of guidance commands based on the instantaneous state of the vehicle and a nominal reference trajectory. The focus of this thesis lies on the design-time trajectory development part of mission design. More specifically, the research goal of this thesis work is formulated as follows:

To what extent can optimal re-entry trajectories be developed in the design-time phase of mission development for a winged entry vehicle that provide a maximum-range capability under the objective of minimizing heat loads and adhering to operational constraints?

To answer this question, a mission design tool is developed in four successive stages:

1. Development of the re-entry simulator
2. Design of the guidance algorithm
3. Development of the mission planner
4. Mission planner testing

The capacity to quickly and reliably simulate re-entry trajectories is paramount to a mission planner. For this purpose, a simulator was developed with the specific goal of later integration with the mission planner. The re-entry simulator uses the HORUS reference vehicle model from Mooij (1995). The HORUS reference vehicle is a Space Shuttle-like winged (i.e., lifting) vehicle capable of both gliding and skipping flight. Detailed and verified aerodynamic information is available for the vehicle model from Mooij (1995), as well as simulation data for a reference trajectory developed by Mooij (1998); this data was used to verify the re-entry simulator. Throughout the course of the simulator's design, considerable effort was extended to ensure the reliability of the both the simulator as a whole, and of its individual components. To this end, many larger verification steps were performed on the whole simulator based on its configuration at the time, first with very simple models and with each iteration increasing in complexity. The end result is a simulator capable of simulating the trajectory and all relevant parameters of the HORUS-2B reference vehicle by implementing a "three-and-a-half" degree-of-freedom flight-dynamic model with the inclusion of the pitch trim condition.

The vehicle performs an unpowered re-entry, meaning all the forces and moments acting on the vehicle throughout re-entry are a direct consequence of its instantaneous environment. The forces and moments acting on the vehicle are either gravitational or aerodynamic in origin; these forces are computed according to the gravitational field and atmospheric models used. The re-entry simulator makes use of a central gravity field with an additional J_2-term to model the latitudinal gravitational perturbations. The instantaneous atmospheric properties throughout the re-entry trajectory are computed using an analytical United States Standard Atmosphere 1976 (US76) atmospheric model. Finally, the Earth is taken to have a spherical shape.

Steering is achieved by modulating the vehicle's attitude in terms of its angle of attack and bank angle. A change in attitude results in a change in the direction and magnitude of the resultant aerodynamic force acting on the vehicle, causing the vehicle to alter its trajectory. The vehicle's attitude throughout the course of the trajectory is defined by a guidance profile. The guidance profile is defined in terms of a number of control nodes, at which the guided attitude is related to an independent parameter that indicates the vehicle's position along its trajectory. This independent parameter is the normalized total specific energy of the vehicle, which has value 1 at the entry point and 0 when the vehicle has come to a standstill on the Earth surface. The benefit of using this parameter is that not only are its limit values independent of the trajectory, but also that it is monotonically decreasing throughout the course of re-entry; this allows for an unambiguous definition of the guidance profile.

The simulator computes the vehicle's trajectory by integration of the initial state, which is defined by its inertial position and velocity components. The state derivative of the vehicle is computed based on the summation of the gravitational and aerodynamic forces it incurs as a result of its current state; the commanded attitude angles determine the definition of the aerodynamic force vector in the inertial frame. In addition to the state variables, the vehicle's horizontal velocity as well as the heat flux are integrated to obtain the total downrange and heat load at the end of the trajectory.

The purpose of the mission planner is to develop trajectories that:

- Keep the vehicle within its operational constraints
- Minimize the heat load
- Provide the largest possible range

The objectives of minimum heat load and maximum range are conflicting. Trajectories with minimum heat load requirements are generally short in duration with smooth heat flux profiles where the heat flux is maintained close to its constraint value. The total heat load is minimized by keeping the duration of the re-entry as short as possible. This, however, is in direct opposition to the objective of maximizing range, where keeping the vehicle aloft for as long as possible is beneficial. For extended-range trajectories, the altitude profile may contain a number of lofts or skips of the vehicle; this is beneficial for the total downrange, but results in an irregular heat flux profile. Large temperature gradients can cause significant thermal stresses, which are detrimental to the vehicle's Thermal Protection System (TPS).

The course of an entire trajectory can be specified by its guidance profile. The mission planner develops trajectories by specifying the individual parameters in this guidance profile. The output must be a qualitatively good trajectory with regard to both the mission's objectives and constraints. The mission objectives are the total downrange X_{dr} (to maximize) and the total heat load Q (to minimize), whereas the constraints are related to the heat flux q_c and $g$load. The quality of a certain trajectory in terms of these parameters is defined quantitatively by the value of the objective function, of which the values should be minimized. The objective function takes into account not only X_{dr}, Q and the constraint parameters, but also the shape of the heat flux profile in terms of the number and magnitude of any fluctuations.

The final output of the mission planner consists of a set of optimal trajectories with varying characteristics between minimum heat load performance and maximum range performance. Any trajectory within this set can be selected depending on the mission specific priorities in terms of heat load and total downrange.

This text gives an overview of the theory behind the developed mission planner, the design process, as well as an assessment of the mission planner's performance. Chapter 2 presents the theory behind the re-entry problem, as well as the models used in the re-entry simulator. The flight dynamic model as applied in the mission planner is discussed in Chapter 3. The most prominent numerical methods in use by the simulator and mission planner are explained in Chapter 4. The development and final design of the guidance system is discussed in Chapter 5, followed by the presentation of the simulator as a whole in Chapter 6. The methodologies used by the mission planner to obtain optimal trajectories, as well as the mission planner design process are discussed in Chapter 7. The evaluation of the mission planner's qualities and performance is given in Chapter 8. This text is concluded with Chapter 9 wherein the overall conclusions of this thesis work are presented, as well as future improvement opportunities.

Chapter 2

Re-Entry Missions Defined

A re-entry mission is defined in terms of requirements stemming from the trajectory and the mission's objectives, as well as constraints based on the re-entry vehicle's and possible crew's capacity (Ely, 1966). The totality of conditions that need to be taken into account in the design of a re-entry mission is often referred to as the re-entry problem; the aspects of this problem are discussed in Section 2.1. Section 2.2 defines in more technical detail the re-entry environment, as well as the design choices made in modeling this environment.

2.1 The Re-Entry Problem

The term "problem" may carry a negative connotation in the colloquial sense, but in engineering it simply means "something to solve". Such is also the case for designing a re-entry mission, with its many different aspects and considerations. Atmospheric entry combines the great speeds inherent to spaceflight with the more down-to-Earth considerations of, e.g., aerodynamics, making it an action characterized by extremes. The aerodynamic loads and heating a vehicle is subjected to during its return journey are unmatched by anything on Earth. In the face of these harsh conditions, the vehicle and its trajectory must de designed in such a way that its own safety – as well as that of people on the ground – is not compromised, in addition to meeting the mission's overall goals. This safety consideration encompasses not only the structural integrity of the vehicle, but also its ability to follow a prescribed trajectory in a sufficiently accurate manner – even when confronted with unpredictable circumstances. This thesis focuses largely on the re-entry trajectory, therefore the re-entry problem will mainly be discussed in that context. The trajectory of a re-entry vehicle is defined in terms of its constraints; from the entry point to landing, these must continuously be taken into account.

2.1.1 Vehicles

Re-entry vehicles come in many shapes and sizes, but can generally be divided into two categories: capsules (e.g., the Dragon re-entry capsule) and winged vehicles (e.g., the Space Shuttle). Their defining characteristic, the lift-to-drag ratio L/D, is related to both size and shape as it defines the amount of aerodynamic lift the vehicle is capable of producing relative to the aerodynamic drag it incurs. Capsules have low ($L/D < 1$) lift-to-drag ratios, whereas winged vehicles have high ones ($L/D > 1$); an L/D greater than zero indicates that the vehicle is capable of producing lift. The L/D of a vehicle is directly related to its response to aerodynamic loads; a vehicle can be steered by altering its aerodynamic profile – and thus its L/D – by modulating its attitude with respect to the incoming airflow. The L/D of a vehicle gives an indication of its capability both in terms of possible range, as well as controllability. A vehicle can utilize aerodynamic loading to exhibit control over its course by adjusting its attitude, and the more lift it can produce, the more control it has. Therefore winged vehicles with high L/D-ratios are marked by a

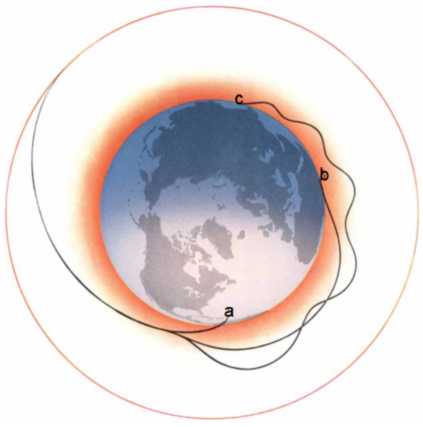


Figure 2.1: Re-entry trajectories (Becker, 1961).

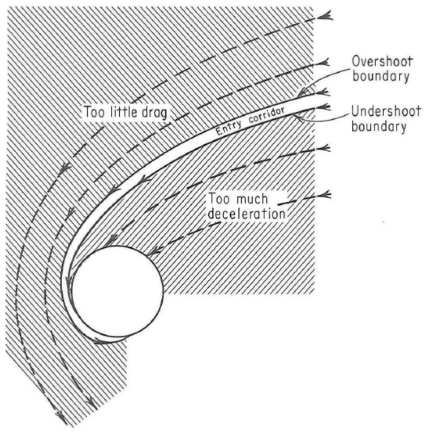


Figure 2.2: The re-entry corridor (Chapman, 1959).

higher degree of controllability than low-lift vehicles, allowing for a greater freedom in the specification of its trajectory.

2.1.2 Trajectories

The path a vehicle can follow through the atmosphere is determined largely by its aerodynamic attributes, as well as its velocity and flight path angle at entry interface (EI) (referred to as the entry angle). The flight path angle is denoted by γ and represents the angle between the vehicle's flight path and the local horizontal plane. Three characteristic re-entry trajectory types can be distinguished:

Ballistic entry is the simplest and most drastic of the three trajectory types (see trajectory **a** in Figure 2.1). The vehicle enters the atmosphere with high velocity at a steep entry angle, which remains (near-) constant throughout the vehicle's short journey to the ground. Due to the absence of lift forces, ballistic entry is performed by vehicles with a low (but nonzero) lift-to-drag ratio such as capsules; capsules are semi-ballistic and are capable of producing some amount of lift – this is necessary for aerodynamic controllability.. Due to the steepness of the trajectory, the vehicle reaches the denser layers of the atmosphere at high velocity, where it will experience large g-loads and a high heating rate.

Gliding entry is characterized by an almost constant shallow entry angle and can only be performed by vehicles with a high lift-to-drag ratio, such as winged vehicles (see trajectory **b** in Figure 2.1). The aerodynamic characteristics of the vehicle allow it to generate a lift force when the atmosphere is thick enough, keeping it aloft for a much longer time. As a consequence, the vehicle reaches the denser layers of the atmosphere at a greatly reduced speed and experiences much smaller g-loads and heating than the ballistic case.

Skipping entry is an entry mechanism wherein the vehicle's altitude does not monotonically decrease with time (see trajectory **c** in Figure 2.1). Instead, the vehicle performs one or more "skips", which may or may not take it out of the Earth's atmosphere again; if the skips are performed within the atmosphere, they are sometimes referred to as "lofts" instead. This skipping motion can be achieved without an added propulsive force depending on the vehicle's aerodynamic properties and velocity, by both low-L/D and high-L/D vehicles alike. Low-L/D vehicles can achieve a skip if their velocity is high enough, e.g., after Lunar return (Brunner and Lu, 2008). Winged vehicles are capable of inducing skip even after entering from low-Earth orbit (LEO). While it seems uncomfortable, performing one or more skips allows the vehicle to greatly increase its re-entry range, in addition to segmenting its deceleration and heating phases into smaller parts. During its exo-atmospheric phase, the vehicle describes a simple ballistic trajectory where it experiences no aerodynamic deceleration (i.e., no net deceleration) or heating.

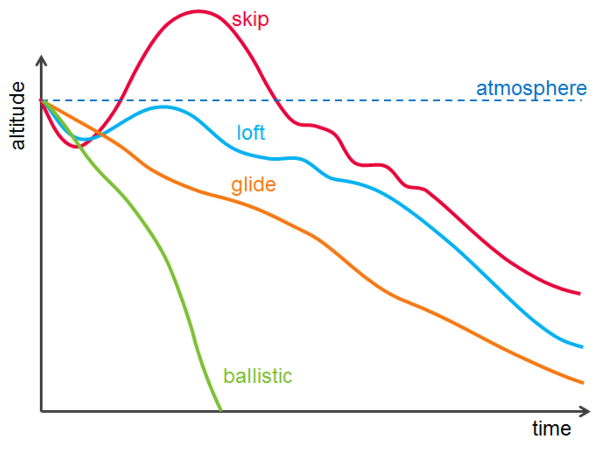


Figure 2.3: Schematic representation of the general altitude progression of skipping, lofting and gliding trajectories (not to scale).

A schematic representation of the different altitude profiles of these trajectories is shown in Figure 2.3. A distinction is made between the loft and skip trajectories, though these technically both belong to the skipping category. The difference lies in whether the vehicle performs an exoatmospheric skip, or remains within the atmosphere (i.e., lofting). In comparison to the glide trajectory, it can be seen that both types of skip trajectory have a positive effect on the vehicle's attainable range; the larger the skip, the better. Lifting vehicles are thus capable of performing both gliding and skipping entry when returning from LEO, so for the purpose of maximizing range, skipping entry may be a possibility.

2.1.3 Constraints

The re-entry trajectory that is flown is carefully designed around the mission's requirements and constraints. These constraints stem from the structural integrity requirements of both the vehicle and its payload (and possible crew), as well as safety considerations concerning people on the ground. If a vehicle performs a beautiful low-g re-entry with minimal heat load – but nevertheless lands inside a shopping mall – no-one will be patting the chief engineer on the back. The success of the mission is based on four basic considerations: aerodynamic capture within the atmosphere, aerodynamic load factor, aerodynamic heating, and control of the landing point (Graves and Harpold, 1970).

2.1.3.1 Re-Entry Corridor

All constraints imposed on the vehicle can be translated to trajectory constraints, which leads to the concept of a re-entry corridor. The reentry corridor is defined as "the set of space trajectories for which aerodynamic capture within the atmosphere of the Earth can be achieved and for which re-entry trajectory control can be accomplished without violating either flight-crew or vehicle stress limits" (Chapman, 1959). These limits are defined in terms of g-loads (i.e., deceleration loads) and aerodynamic heat loads.

Entering the atmosphere at an excessively shallow angle – termed overshoot – leads to loss of control over the trajectory wherein the vehicle is deflected back out of the atmosphere, possibly never to return to Earth again. Successful aerodynamic capture entails staying below this overshoot boundary. Conversely, entering the atmosphere too steeply – termed undershoot – causes the vehicle to incur decelerations and heat loads of unacceptable magnitudes. The overshoot and undershoot boundaries define the absolute outer limits of the possible re-entry corridor (see Figure 2.2). Remaining within these bounds is far from a guaranteed successful re-entry, but rather the avoidance of a guaranteed catastrophe. Within these absolute bounds, operational constraints as well as the specific intended destination of the vehicle greatly reduce the effective width of the re-entry corridor.

2.1.3.2 G-Loads

Before entering the atmosphere, an unpowered vehicle's motion is governed by gravitational forces only (Loh, 1968). Upon re-entry, the motion becomes affected by aerodynamic forces as well, specifically the lift force L and drag force D[1]. The lift force acts normal to the vehicle's path in the upwards direction, causing it to stay aloft longer, whereas the drag force acts counter to the vehicle's direction of motion, causing it to decelerate. The resultant of these forces translates to an acceleration proportional to the vehicle's mass (as per Newton's 2nd Law). The g-load experienced by the vehicle is simply the magnitude of this resultant acceleration normalized by the gravitational acceleration at sea-level g_0:

$$n_g = \tfrac{1}{m g_0}\sqrt{L^2 + D^2} \tag{2.1}$$

2.1.3.3 Aerodynamic Heating

The aerodynamic heat loads that can be incurred by the vehicle are subject to constraints in two different manners: the total heat input and the maximum (local) heating rate (Eggers et al., 1957). A re-entry vehicle enters the atmosphere at high speed, and thus possesses a large amount of kinetic energy (and a much smaller amount of potential energy). This energy is converted to heat due to friction occurring between the vehicle's skin and the surrounding air, causing the vehicle to decelerate. The total heat input to the vehicle is simply the integral of the heating rate taken over the entire trajectory (Bloom and Miele, 1962). The heating rate is dependent on the hypersonic aerodynamic properties of the specific vehicle and is quite complicated to calculate. A useful approximation as developed by Chapman (1960) is:

$$q_c = c^* {R_N}^n \left(\frac{\rho}{\rho_0}\right)^{1-n} \left(\frac{V}{V_c}\right)^m \tag{2.2}$$

where R_N is the nose radius of the vehicle, ρ is the atmospheric density (the subscript zero indicates its value at sea-level), V is the vehicle's velocity and V_c is the circular velocity defined to be 7905.4 m/s. The exact definitions of the constants $c^* = 1.1097 \cdot 10^8$, $n = 0.5$ and $m = 3.15$ is beyond the scope of this text, and it is sufficient to state that they represent the airflow characteristics around the vehicle.

2.1.4 Mission Heritage

2.2 Environment Modeling

The environment in which an (atmospheric) entry takes place, is defining for the mission. A successful entry on Mars is completely different from one on Earth, from start to finish. For example, Mars has a very thin atmosphere compared to the Earth's, and deceleration and heat loads incurred by the vehicle are much lower. A vehicle's motion and its capacity to meet predefined constraints is defined by the loads it is subjected to. These loads, in turn, are defined by the environment in which the entry occurs. In the context of re-entry, the environment encompasses three separate aspects of the body on which the vehicle is to land: its gravitational field, its atmosphere, and its shape. The average celestial body is far from a perfect sphere in a vacuum, and the Earth is no exception. Its characteristics cannot be described completely by any number of equations; to successfully simulate a re-entry trajectory, the environmental properties must be approximated in a sufficiently accurate way using environmental models.

2.2.1 Gravitational Field

A re-entry vehicle is subject to an external force that draws it towards the Earth: the gravitational force. Newton's Law of Gravitation states that two point masses which are separated by a vector distance $\mathbf{r}$

[1]The third aerodynamic force component, the side force S, can be neglected in terms of its contribution to the g-load.

exert an attractive force $\mathbf{F_G}$ on each other defined as

$$\mathbf{F_G} = \frac{GMm}{R^2}\hat{\mathbf{r}} \tag{2.3}$$

where R is the modulus of the vector distance $\mathbf{r}$, and $\hat{\mathbf{r}}$ is the normalized vector distance. G is the universal gravity constant ($G = 6.67259 \cdot 10^{-11}$ m^{3}s^{-2} (Standish, 1995)), and M denotes the mass of the central body; their product GM is defined as the gravitational parameter μ of the central body. The gravitational potential of the central body may be written as

$$U = -\frac{\mu}{R} \tag{2.4}$$

Equation 2.4 is commonly referred to as the central field model, and is the simplest approximation of a body's gravitational potential. The fraction μ/R is called the central field term.

2.2.1.1 Zonal harmonics

The point-mass approximation applied in Newton's Law of Gravitation is, however, not valid when calculating the gravitational acceleration due to the Earth as experienced by a re-entry vehicle. When the second body (in this case, the vehicle) is in such close range of the central body, other factors that may locally affect the gravitational field of the central body need to be taken into account. For Earth, the largest cause of deviation from the central field model is its inhomogeneous mass distribution. The deviation from the central field potential caused by this inhomogeneous mass distribution can be modeled as a correction term $U_c(R, \tau, \theta)$:

$$U = -\frac{\mu}{R} + U_c(R, \tau, \theta) \tag{2.5}$$

where τ is the longitude, and θ is the co-latitude, ranging from 0° for latitude $\delta = 90°$ to 180° for $\delta = -90°$.

For Earth, U_c takes the form of a series of Legendre polynomials and a spherical harmonic function. Equation 2.5 can be written as (Vallado, 2001):

$$U = -\frac{\mu}{R}\left\{1 - \sum_{\ell=2}^{\infty}\left[\left(\frac{R_E}{R}\right)^{\ell} J_\ell P_\ell^0(\cos\theta) + \sum_{m=1}^{\ell}(C_{\ell m}\cos m\tau + S_{\ell m}\sin m\tau)P_\ell^m(\cos\theta)\right]\right\} \tag{2.6}$$

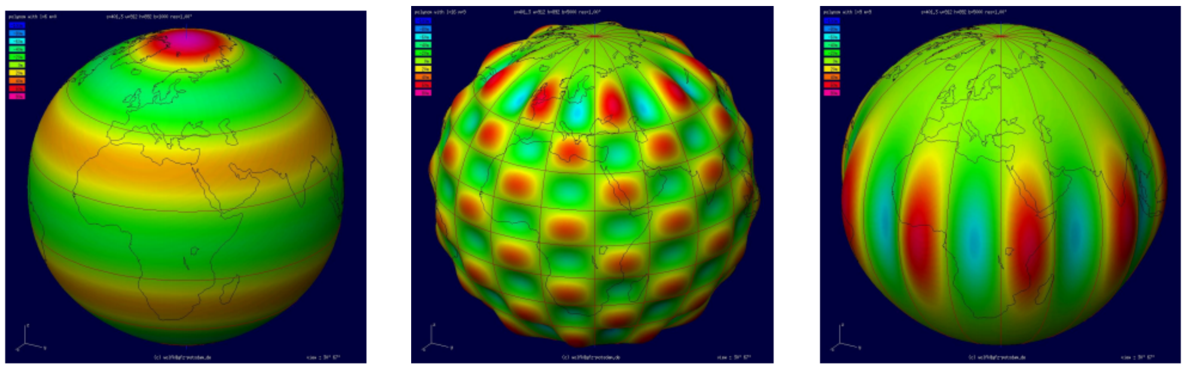
When a body of finite size is observed from an infinite distance, it looks like a point mass. This makes sense, because as $R \to \infty$, $U_c(R, \tau, \theta) \to 0$.

The indices ℓ and m represent the degree and order of the function, respectively. The zero-th degree solution simply reduces Equation 2.6 to the central field model in Equation 2.4, making the term J_0 equal to unity. The first degree solution corresponds to taking the origin of the system at the center of mass (c.o.m.) of the central body. The coefficients J_1, $C_{1,0}$, $C_{1,1}$, $S_{1,0}$ and $S_{1,1}$ are all null in this case (Vallado, 2001). This is the reason the summation indices for ℓ and m do not start at zero as one would expect.

The terms J_ℓ ($m = 0$) are zonal harmonic coefficients, and only define the latitudinal distribution of mass. $C_{\ell m}$ and $S_{\ell m}$ are spherical harmonic coefficients which represent sectoral harmonics when $\ell = m$ and tesseral harmonics when $\ell \neq m \neq 0$. Sectoral harmonics define the longitudinal distribution of mass, whereas tesseral harmonics divide the Earth into latitudinal and longitudinal patches. See Figure 2.4 for an illustration of zonal, sectoral and tesseral harmonics for a number of degrees and orders. The constants J_ℓ, $C_{\ell m}$ and $S_{\ell m}$ are empirically determined using precise measurements of the Earth's gravitational field (e.g., by satellites such as GRACE (Foerste et al., 2005)). Their values are constantly being updated as measurement techniques improve.

The terms $P_\ell^m(\cos\theta)$ are (associated) Legendre polynomials. A Legendre polynomial $P_\ell(x)$ is an ℓ^{th} degree polynomial that may be expressed as

$$P_\ell(x) = \frac{1}{2^\ell \ell!}\frac{d^\ell}{dx^\ell}\left[\left(x^2 - 1\right)^\ell\right] \tag{2.7}$$



(a) Zonal ($\ell = 6, m = 0$) (b) Sectoral ($\ell = 16, m = 9$) (c) Tesseral ($\ell = 9, m = 9$)

Figure 2.4: Zonal, sectoral and tesseral harmonics (Lambeck, 1990).

The associated Legendre polynomials $P_\ell^m(x)$ are obtained from $P_\ell(x)$ as

$$P_\ell^m(x) = (-1)^m \left(1 - x^2\right)^{\frac{m}{2}} \frac{d^m}{dx^m} \left(P_\ell(x)\right) \tag{2.8}$$

The expressions for $P_\ell^m(\cos\theta)$ for $0 \leq \ell \leq 4$ and $0 \leq m \leq 4$ can be found in, for instance, Stacey and Davis (2008).

2.2.1.2 Model accuracy

It is convenient to express the gravitational acceleration vector as consisting of a radial component g_R, a longitudinal component g_τ and a lateral component g_δ:

$$\mathbf{g} = \left(g_R, g_\tau, g_\delta\right)^T \tag{2.9}$$

The required accuracy of the gravitational model naturally depends on its application. For the purpose of re-entry mission planning, the zero-th order central field model is not sufficiently accurate; in the thin, upper layers of the atmosphere, the gravitational force dominates the motion of the vehicle. Guidance and navigation systems require a more accurate representation of the gravitational field (Mooij, 2013).

For a rotating Earth, the dependence of the gravitational potential on the longitude τ may be neglected in the , i.e., the Earth may be assumed to be an ellipsoid with symmetry about its polar axis. In this case, only the zonal terms contribute to the potential ($m = 0$) and the gravitational acceleration has no component out of the meridian plane (i.e., $g_\tau = 0$). Equation 2.6 reduces to

$$U = -\frac{\mu}{R} \left\{ 1 - \sum_{\ell=2}^{\infty} \left(\frac{R_E}{R}\right)^\ell J_\ell P_\ell^0(\cos\theta) \right\} \tag{2.10}$$

and the gravitational acceleration vector $\mathbf{g_V}$ – the subscript V indicating the vertical reference frame, see Section 3.1.1 – becomes

$$\mathbf{g_V} = \left(g_\delta, 0, g_R\right)^T \tag{2.11}$$

where g_R is the radial and g_δ the latitudinal component of $\mathbf{g}$. Additionally, if the equator is taken as a plane of symmetry, $J_n = 0$ for all odd values of n.

By far the largest effect on the Earth's gravitational field is due to J_2, i.e., the inhomogeneity in mass distribution caused by the Earth's equatorial bulge. The next largest term, J_3 (which accounts for the Earth's slight pear shape) is three orders of magnitude smaller. For re-entry simulations it is sufficient to neglect the longitudinal dependency and only include J_n-terms up to a few orders, though generally J_2 is deemed sufficient. For this thesis work, the longitudinal dependence as well as all J_n terms above J_2 are neglected in the gravitational model. In that case, the gravitational acceleration vector **g** becomes

$$\mathbf{g} = \begin{bmatrix} g_R \\ 0 \\ g_\delta \end{bmatrix} = \frac{\mu}{R^2} \begin{bmatrix} 1 - \frac{3}{2} J_2 \left(\frac{R_E}{R}\right)^2 \left(3\sin^2\delta - 1\right) \\ 0 \\ -3\left(\frac{R_E}{R}\right)^2 J_2 \sin\delta\cos\delta \end{bmatrix} \tag{2.12}$$

2.2.2 Atmosphere

In the presence of an atmosphere, a re-entry vehicle experiences two critical aspects in its mission: aerodynamic forces and aerodynamic heating. For the Earth, the atmosphere becomes the defining factor for the last leg (approximately below 40 km) of a re-entry mission. Its presence allows for the vehicle's trajectory to be controlled by adjusting its aerodynamic profile, as well as the deceleration of the vehicle to safe landing speeds by converting kinetic energy to heat due to friction (Mooij, 1994). However, aerodynamic decelerations in excess of $100g$ can easily occur in uncontrolled re-entries (Regan, 1993); for reference, the g-load limit for human spaceflight is around $3g$. Additionally, the vehicle still possesses such large amounts of kinetic energy in the lower layers of the atmosphere, that the heat loads incurred due to aerodynamic friction could easily incinerate it in absence of both thermal protection and an adequate guidance scheme. This navigation system needs to predict the aerodynamic forces on the vehicle with sufficient accuracy to perform a safe re-entry. Uncertainties and errors in doing so may cause catastrophic damage.

To determine the aerodynamic forces acting on a re-entry vehicle, an atmospheric model must be used. The Earth's atmosphere is dynamic; its defining properties such as temperature and density, as well the particle densities of its constituents, are constantly changing – primarily due to Solar radiation. Developing a complete model that encompasses all these changes and accurately represents the atmospheric conditions at all locations at all times is impossible (Vinh, 1981). Even the most complex and detailed atmospheric model will always be just that: an approximation. However, many aspects that make the Earth's atmosphere so complex are not relevant in the frame of aerodynamic load calculations on a re-entry vehicle (Mooij, 1997). In fact, the dominating atmospheric factors for aerodynamic accelerations and heating are fairly simple: the atmospheric density ρ and temperature T. The accurate simulation of the aerodynamic loads and heating incurred by a vehicle during the re-entry stage thus depends mainly on being able to reliably and conveniently model the atmospheric density (Marec, 1979). Many atmospheric models have been developed (and are being developed) for the Earth, with varying complexity. The distinction is made between standard and reference atmospheres, where accounting temporal variation is the most important divisor. Reference atmospheres were not considered for this thesis work due being unnecessarily detailed. A standard atmosphere is a theoretical distribution with altitude of the main physical properties of the atmosphere (e.g., density, pressure, temperature) that has been established by international agreement. By design, there is no temporal or spatial (other than altitude) element involved, as the model is intended to be representative of year-round, mid-latitude conditions.

In this thesis work, the popular US76 was used to model the atmospheric properties. The US76 is the most commonly used standard atmosphere in aviation and aerospace due to its simplicity and sufficiently accurate (for most applications) representation of the atmosphere. It is a revised version of an earlier model, the United States Standard Atmosphere 1962 (US62). The US76 is an analytic model of atmospheric property variations with altitude, and is based on the interpolation of known atmospheric values within predefined layers. It can be applied to geometric altitudes from -5 km to up to 1000 km. The type of interpolation depends on the specific layer. The course of the main atmospheric properties density, pressure, temperature and speed of sound with altitude is shown in Figure 2.5.

The US76 operates on the basis of a number of assumptions:

- The air behaves as a perfect gas; the pressure p, temperature T, and density ρ at any point in the atmosphere are related by the equation of state, also known as the perfect gas law:

  $$p = \frac{\rho R^* T}{M} \tag{2.13}$$

  where M is the molecular weight and $R^* = 8.3144621$ J/kg/mole is the universal gas constant.

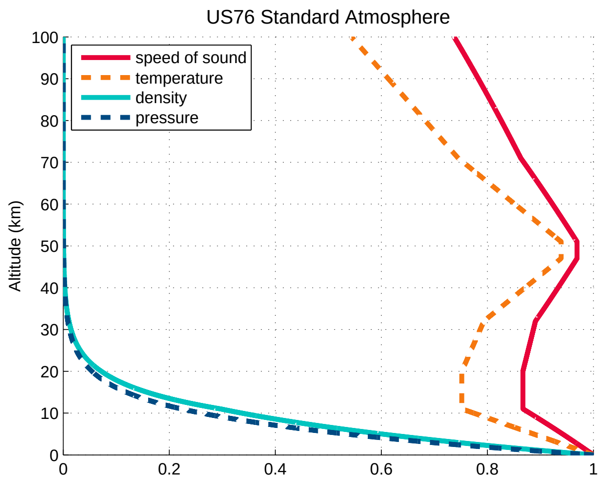


Figure 2.5: The US76 plot of geometric altitude against the normalized density, pressure, temperature and speed of sound.

- The atmosphere is in hydrostatic equilibrium. The hydrostatic equation is

$$dp = -g\rho dz \tag{2.14}$$

- The atmosphere is homogeneous up to an altitude of 80 km.

The US76 is based on the division of the Earth's atmosphere into a number of layers, or 'strata', with similar atmospheric properties. Up to a geometric altitude of $z = 86$ km (which is equal to a geopotential altitude of $h = 84.8520$ km), the atmosphere is split up into strata based on the geopotential altitude; upwards of $z = 86$ km, the strata are defined based on the geometric altitude. This discrepancy occurs because US76 is actually based on an earlier standard atmosphere, namely the US62, which used h as its altitude variable. The relation between z and h is given by

$$g_0 dz = g dh \tag{2.15}$$

which can be approximated by

$$z = \int_0^h \frac{g}{g_0} dh \approx \frac{R_0 h}{R_0 + h} \tag{2.16}$$

where $R_0 = 6356.766$ km, the Earth's radius at the latitude where $g_0 = 9.80665$ m/s^2. For the sake of clarity, in this section geopotential kilometers shall be indicated as km$'$ to distinguish them from geometric kilometers.

The most important atmospheric properties at a certain altitude that may be calculated using US76 are the temperature T, pressure, p and density ρ. Before these computations are explained, the mean molecular weight M must be defined.

2.2.2.1 Mean molecular weight

The mean molecular weight M of air is the average of the contributions of all its constituents, and can be approximated from the molecular weights of He, N_2, O_2 and Ar:

$$M = f_{He} M_{He} + f_{N_2} M_{N_2} + f_{O_2} M_{O_2} + f_{Ar} M_{Ar} \tag{2.17}$$

The f-terms represent the fractional concentrations of the individual constituents of air. Up to an altitude of 80 km, the atmosphere is assumed to be completely homogeneous; the fractional concentrations retain their constant sea-level values, and M is considered constant at $M_0 = 28.9644 \cdot 10^{-3}$ kg/mol. For geometric altitudes between 80 km and 86 km, M is linearly interpolated using tabulated values of the

Table 2.1: Reference levels of the temperature-height profile from the surface to 86 km. (NASA, 1976)

i	h **km′**	L_M **K/km′**	$T_M(h)$ **K**
0	0.0	-6.5	linear
1	11.0	0.0	linear
2	20.0	+1.0	linear
3	32.0	+2.8	linear
4	47.0	0.0	linear
5	51.0	-2.8	linear
6	71.0	-2.0	linear
7	84.8520	N/A	N/A
i	z **km**	L_K **K/km**	$T(z)$ **K**
7	86	0.0	linear
8	91		elliptical
9	110	12.0	linear
10	120		exponential
11	500		
12	1000		

ratio M/M_0, defined at intervals of 0.5 km, ranging from 1.0 at 80 km to 0.999579 at 86 km. These values are defined such that the boundary conditions at both 80 km and 86 km are met.

$$M_i = M_0 \left(\frac{M_i}{M_0}\right)_i \tag{2.18}$$

Above 86 km, the computation of M is more complex as the assumption of atmospheric homogeneity does not hold anymore; the number density values of the individual constituents vary differently with altitude. For this thesis work, the individual number densities is not of interest however; for the precise calculations the reader is referred to NASA (1976). For practical purposes, tabulated values of M between altitudes of 86 km and 120 km can be used to interpolate the mean molecular weight.

$$M = M_i + (M_{i+1} - M_i)\left(\frac{z - z_i}{z_{i+1} - z_i}\right) \tag{2.19}$$

Above altitudes of 120 km, M can for all practical purposes be considered zero.

2.2.2.2 Temperature

The temperature-altitude profile boundaries and values up to 86 km are indicated in the first part of Table 2.1. The use of h instead of z in the model definition at these altitudes makes it necessary to define the altitude variations of T and M in terms of h. For this purpose, the molecular scale temperature T_M is introduced.

$$T_M = T\frac{M_0}{M} \tag{2.20}$$

The molecular temperature T_M at a certain geopotential altitude h is simply calculated using the molecular scale temperature gradient L_M:

$$T_M = T_{M,i} + L_{M,i}(h - h_i) \tag{2.21}$$

The conversion to kinetic temperature T follows from Equation 2.20 and the value of the mean molecular weight M.

Above 86 km, the model is based on the geometric altitude z. The temperature-altitude profile is based on four successive functions; these are defined in such a way that the first derivative of T with respect to z is continuous over the entire altitude interval of 86 km to 1000 km. Each function is defined over a specific altitude region; these regions begin at the base altitudes shown in the second part of Table 2.1. The functions represent four separate atmospheric conditions.

Layer 1: 86 - 91 km is isothermal and the gradient dT/dz is equal to zero. The temperature throughout the layer is simply given by

$$T = T_7 = 186.8673\text{K}$$

Layer 2: 91 - 110 km has a temperature-altitude function dT/dz in the form of an ellipse:

$$T = T_c + A\sqrt{1 - \left(\frac{z - z_8}{a}\right)^2} \tag{2.22}$$

where $T_c = 263.1905$ K, $A = -76.3232$ K and $a = -19.9429$ km. The values of the constants T_c, A and a were obtained from fitting the ellipse equation to the values of $T_8 = T_7 = 186.8673$ K and $L_{K8} = L_{K7} = 0.0$ K/km at $z_8 = 91$ km, and $T_9 = 240.0$ K and $L_{K9} = 12.0$ K/km at $z_9 = 110$ km. The derivation can be found in NASA (1976).

Layer 3: 110 - 120 km has a temperature-altitude function that increases exponentially toward an asymptote with increasing z.

$$T = T_9 + L_{K9}(z - z_9)$$

Layer 4: 120 - 100 km contains strata 10 through 12 (see Table 2.1). For the entire layer, $T(z)$ has the exponential form

$$T = T_\infty - (T_\infty - T_{10})\exp(-\lambda\xi)$$

where

$$\lambda = \frac{L_{K9}}{T_\infty - T_{10}} = 0.01875 \ \left(\text{km}^{-1}\right)$$
$$\xi = \xi(z) = \frac{(z - z_{10})(R_E + z_{10})}{R_E + z} \ \text{km}$$

T_∞ is defined to be 1000 K.

2.2.2.3 Pressure

The method of pressure computation depends on the altitude. From the surface up to a geometric altitude of 86 km, the argument used is again the geopotential altitude h; for altitudes above 86 km, the argument is z. $p(h)$ can be determined by integrating another form of the hydrostatic equation (Equation 2.14):

$$d\ln p = \frac{dp}{p} = -\frac{gM}{R^*T}dz \tag{2.23}$$

The result of this integration takes two forms, depending on the value of the molecular scale temperature gradient L_M (see Table 2.1).

$$L_{M,i} = 0: \qquad p = p_i \exp\left[-\frac{g_0 M_0 (h - h_i)}{R^* T_{M,i}}\right] \tag{2.24a}$$

$$L_{M,i} \neq 0: \qquad p = p_i \left[\frac{T_{M,i}}{T_{M,i} + L_{M,i}(h - h_i)}\right]^{\frac{g_0 M_0}{R^* L_{M,i}}} \tag{2.24b}$$

The reference value of p for $i = 0$ is $p_0 = 101325.0$ N/m^2, the atmospheric pressure at sea level. The values of p_1 through p_6 can be obtained using Equations 2.24a and 2.24b.

Above altitudes of 86 km, the pressure computation is more complex as the assumption of atmospheric homogeneity does not hold anymore. The argument is the geometric altitude z, and thus the kinetic temperature T instead of the molecular temperature is used. The expression for $p(z)$ relates the total pressure p to the sum of all partial pressures due to the individual main atmospheric constituents.

$$p = \sum p_j = \sum n_j kT = \frac{\sum n_j R^* T}{N_A} \tag{2.25}$$

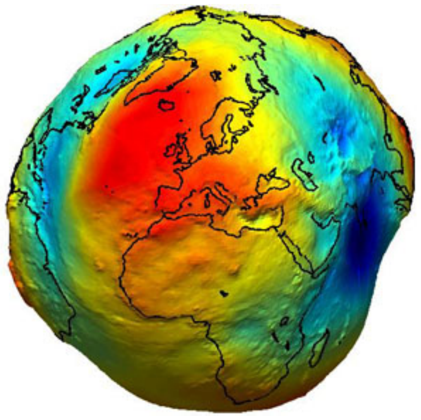


Figure 2.6: The (exaggerated) geoid as observed by GOCE, with gravitational undulations (ESA, 2009).

where $k = 1.380622 \cdot 10^{-22}$ Nm/K is the Boltzmann constant, $N_A = 6.022169 \cdot 10^{26}$ kmol^{-1} is Avogadro's constant and $\sum n_j$ represents the sum of the number densities of the individual gas species at that specific altitude. The calculations to obtain the number densities n_j are complex and beyond the scope of this text. The logarithms of tabulated values of the total number density $N = \sum n_j$ can be interpolated linearly to obtain N at a specific altitude:

$$N = N_i \exp\left[(z - z_i) \ln\left(\frac{\frac{N_{i+1}}{N_i}}{z_{i+1} - z_i}\right)\right] \tag{2.26}$$

The calculation of p then becomes straightforward from Equation 2.25.

2.2.2.4 Density

The air density ρ follows from the obtained values of mean molecular weight M (Equation 2.17), temperature T (Section 2.2.2.2), and pressure p (2.25) :

$$\rho = \frac{MpT}{R^*} \tag{2.27}$$

2.2.2.5 Speed of sound

The local speed of sound is then defined by

$$a = \sqrt{\gamma \frac{R^* T}{M_0}} = \sqrt{\gamma R T} \tag{2.28}$$

where $\gamma = 1.4$ is the ratio of specific heats, and $R = 8.314510$ (J mole^{-1} K^{-1}) is the molar gas constant in air.

2.2.3 Earth Shape

A planetary body is characterized by its shape, size, and rotational rate. The Earth is not a perfect sphere: it bulges at the equator, is slightly pear-shaped, and actually consists mainly of smaller bumps and indentations (not to mention mountainous regions and oceanic trenches) (O'Keefe et al., 1959). This subject was touched upon in Section 2.2.1, where the Earth's inhomogeneous mass distribution was discussed in the context of its gravitational field. An exaggerated representation of the geoid[2] is shown in Figure 2.6.

[2]The equipotential surface that mean sea level follows (Li and Götze, 2001)

Within this thesis, the Earth is modeled as a rotating sphere. The re-entry simulator has the capacity to model the Earth as an ellipsoid as well, however the simulation data available from (Mooij, 1998) was obtained using the spherical-Earth approximation. To allow for easier comparison of results to this reference trajectory, this spherical-Earth approximation was maintained.

Chapter 3

Flight Dynamics

The flight dynamic model as applied in the design of the re-entry simulator is discussed in this chapter. Section 3.1 starts with a general definition of the reference frames used in this work, followed by transformations necessary to switch between these frames. The vehicle state may be described by various definitions of the state variables; the ones used throughout this thesis as well as their relation to each other are discussed in Section 3.2. Section 3.3 gives an overview of the external forces and moments acting on the vehicle as a result of its environment. A small detour follows in Section 3.4, wherein the nominal Horus-2B vehicle model used throughout this thesis is presented. This chapter is concluded by a presentation of the equations of motion as applied in this thesis in Section 3.5.

3.1 Reference Frames

The following has been obtained from Mooij (1994). The frames discussed in this chapter are all right-handed and Cartesian.

3.1.1 Reference Frame Definitions

Inertial Planetocentric Reference Frame

The Inertial Planetocentric Reference Frame (IPRF) $\mathcal{F}_{\mathbf{I}}$ $(X_I Y_I Z_I)$ has its origin in the c.o.m. of the central body around which the vehicle is moving. The OX_IY_I-plane coincides with the equatorial plane of the central body. The Z_I-axis points North and is coincident with the rotational axis of the central body. The direction of the X_I-axis is determined by the reference meridian, and is defined by the vernal equinox at *J2000*. The Y_I-axis completes the right-handed system.

Rotating Planetocentric Reference Frame

The Rotating Planetocentric Reference Frame (RPRF) $\mathcal{F}_{\mathbf{R}}$ $(X_R Y_R Z_R)$ is fixed to the central body and coincides with $\mathcal{F}_{\mathbf{I}}$ at *J2000*. The Z_R-axis points north, the X_R-axis intersects the equator at zero longitude and the Y_R-axis completes the right-handed system.

Body-Fixed Reference Frame

The Body-Fixed Reference Frame (BFRF) $\mathcal{F}_{\mathbf{B}}$ $(X_B Y_B Z_B)$ has its origin at the vehicle's reference point. Generally this reference point is chosen to be the c.o.m. of the vehicle. If the gravity field is constant then this point coincides with the center of gravity (c.o.g.) of the vehicle. The reference frame remains fixed to the vehicle. The Z_B axis lies in the plane of symmetry of the vehicle and is positive in downward direction (for "normal" flight). The X_B-axis also lies in the plane of symmetry and is positive in forward

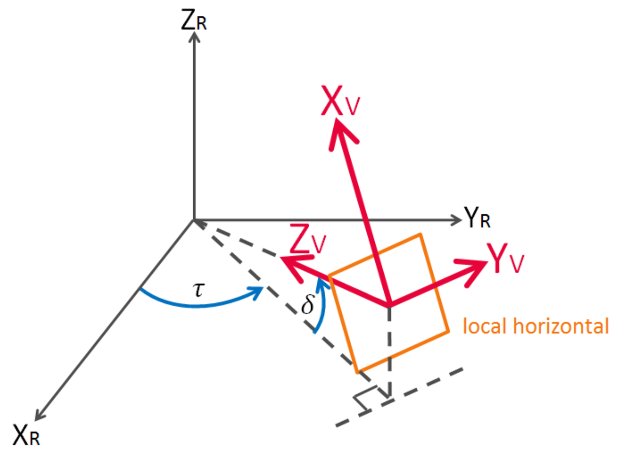


Figure 3.1: Definition of the vertical frame axis system with respect to the rotating planetocentric frame.

direction. Finally, the Y_B-axis points to the right, perpendicular to the symmetry plane, and completes the right-handed system.

Vertical Reference Frame

The Vertical Reference Frame (VRF) $\mathcal{F}_\mathbf{V}$ $(X_V Y_V Z_V)$ has the same origin as $\mathcal{F}_\mathbf{B}$. The Z_V-axis points towards the c.o.m. of the central body, along the radial component of the gravitational acceleration $\mathbf{g}$. The X_V-axis lies in a meridian plane, perpendicular to Z_V, and points North. The Y_V-axis completes the right-handed system. The $X_V Y_V$-plane is referred to as the local horizontal plane. N.B.: strictly speaking, this is only true when the central body is true sphere. A small error is introduced when an elliptical shape is assumed. Figure 3.2(a) depicts the V-frame axis system with respect to the R-frame.

Trajectory Reference Frame

The Trajectory Reference Frame (TRF) $\mathcal{F}_\mathbf{T}$ $(X_T Y_T Z_T)$ can be defined based on the groundspeed or the airspeed (subscripts TG and TA, respectively), the choice of which affects the definition of the X-axis. The groundspeed-based X_{TG} points along the velocity vector relative to the R-frame, whereas the airspeed-based X_{TA} is defined along the velocity vector relative to the atmosphere. These axes are equal when no wind is present. The $\mathcal{F}_\mathbf{T}$ has the same origin as $\mathcal{F}_\mathbf{B}$. Z_T points downwards in the vertical plane, and Y_T completes right-handed system.

Aerodynamic Reference Frame

The groundspeed-airspeed distinction is also made in the definition of the Aerodynamic Reference Frame (ARF) $\mathcal{F}_\mathbf{A}$ $(X_A Y_A Z_A)$; the subscripts AG and AA, respectively, are used to make this distinction. In the groundspeed-based case, X_{AG} is defined along the velocity vector relative to the R-frame, making it collinear with X_{TG}, whereas the airspeed-based X_{AA} points along the velocity vector relative to the atmosphere, which makes it collinear with X_{TA}. Again, these X-axes are equal in the absence of wind. The Z_A-axis is collinear with the aerodynamic lift force, but opposite in direction. The Y_A-axis completes the right-handed system.

Note that when the vehicle is not banking, the $\mathcal{F}_\mathbf{A}$- and $\mathcal{F}_\mathbf{T}$-frames are coincident.

3.1.2 Transformations

The transformation from one (right-handed Cartesian) frame to another can be performed via unit axis-rotations, directional cosine matrices and Quaternions. Any rotation can be described as a combination of a number of unit axis-rotations, in which positive rotations are defined according to the right-hand rule. A sequence of unit axis-rotations can be represented as a single transformation matrix by multiplying the individual unit rotation matrices in sequential order. It is important to note that the unit transformation matrices are all orthonormal (and therefore so are any products between them), meaning a unit transformation matrix's inverse is simply its transpose.

The transformation matrices necessary for the conversion between the reference frames defined in the previous section are listed in Appendix A.1. For the remainder of this text, the influence of wind is neglected, making the groundspeed-based and airspeed-based variants of the trajectory and aerodynamic frames equal; these reference frames will simply be designated with the subscripts T and A, respectively.

3.2 State Variables

The state variables of a vehicle give a representation of its instantaneous state with respect to a certain reference frame. All relevant state variables are discussed in this chapter. Generally, Cartesian components are preferred for numerical computations as the equations are in a simpler form and can be numerically integrated faster. However, the benefit of using spherical components is that they are much more intuitive. Often, the vehicle state will be discussed in terms of its spherical components, even though Cartesian components are used for the actual computations.

3.2.1 State Variable Definitions

3.2.1.1 Position and Velocity

Position and velocity state variables can be expressed in a number of different notations; in this thesis, preference is given to Cartesian and spherical components. Other representations such as Kepler elements are useful when discussing a re-entry vehicle's state before entry.

Cartesian components

Position and velocity may be expressed with respect to any of the previously mentioned reference frames. The most practical ones for the purpose of studying re-entry motion are the inertial reference frame $\mathcal{F}_\mathbf{I}$, the rotating reference frame $\mathcal{F}_\mathbf{R}$ and the vertical reference frame $\mathcal{F}_\mathbf{V}$.

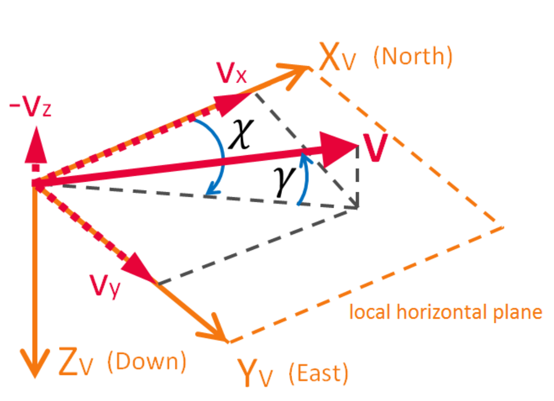
For the position, the same $\mathbf{p}(x,\ y,\ z)$ notation is used for all reference frames, with the frame being referenced denoted as a subscript (I, R and V, respectively). The velocity components are denoted differently depending on the choice of reference frame: $\mathbf{V_I}(\dot{x},\ \dot{y},\ \dot{z})$, $\mathbf{V_R}(u,\ v,\ w)$, and $\mathbf{V_V}(v_\delta,\ v_\tau,\ v_R)$. Note that v_R is defined as positive in negative Z_V direction, i.e., upwards.

Expressing the velocity in $\mathcal{F}_\mathbf{V}$ is intuitive, as it allows the velocity to be interpreted as consisting of a radial component v_R, a longitudinal component v_τ, and a lateral component v_δ. A useful variation on this representation stems from aviation: the NED notation. The NED coordinate system is the same as in $\mathcal{F}_\mathbf{V}$: $X_{NED}Y_{NED}Z_{NED} = X_V Y_V Z_V$. The components of the position are denoted $\mathbf{p}(x,\ y,\ z)$, whereas the velocity components are written as $\mathbf{V}_{NED}(v_N, v_E, v_D)$. Note that in contrast to the radial component of $\mathbf{V_V}$, v_D is defined as positive downwards, i.e., in the direction of $Z_{NED} = Z_V$, so $v_D = -v_R$.

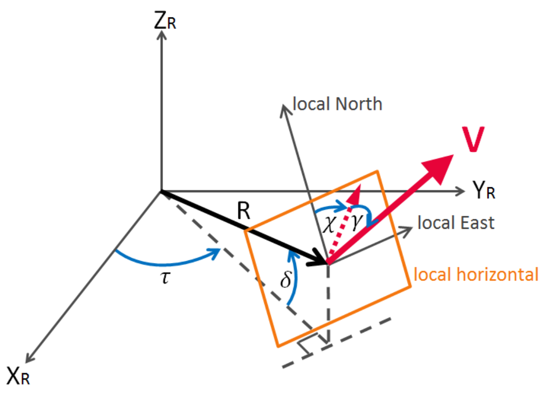
Figure 3.2(a) shows the vertical velocity components with respect to the local horizontal plane; the orientation of this plane relative to the R-frame was shown in Figure 3.1.

Spherical components

The spherical position and velocity are defined with respect to $\mathcal{F}_\mathbf{R}$; both consist of a magnitude and two angle components. The position is denoted $\mathbf{p}(R, \tau, \delta)$, and the velocity as $\mathbf{V}(V, \gamma, \chi)$.



(a) Vertical



(b) Spherical

Figure 3.2: Position and velocity components.

R denotes the radial distance between the c.o.m. of the central body (i.e., the reference frame origin) and the c.o.m. of the vehicle. The longitude τ ($-180° \leq \tau < 180°$) is the angular distance towards the East from the Greenwich meridian; it increases towards the East. The latitude δ ($-90° \leq \delta \leq 90°$) is the angular distance from the equator to the North; it is positive in the Northern hemisphere and negative in the Southern hemisphere.

V is the modulus of the velocity vector $\mathbf{V}$, i.e., the relative velocity. The flight-path angle γ ($-90° \leq \gamma \leq 90°$) is the angle between the local horizontal plane and $\mathbf{V}$; it is negative when the vehicle is descending (i.e., $\mathbf{V}$ lies below the local horizontal) and positive when the vehicle is ascending (i.e., $\mathbf{V}$ lies above the local horizontal). Finally, χ ($-180° \leq \chi < 180°$) defines the angle between the local North and the projection of $\mathbf{V}$ to the local horizontal plane; it is positive when the vehicle is moving towards the East (and its value is 90° when the vehicle is moving towards the East parallel to the equator). All the spherical velocity components are defined with respect to the velocity of the vehicle in the rotating frame. Figure 3.2(b) depicts the spherical position and velocity components with respect to $\mathcal{F}_{\mathbf{R}}$.

3.2.1.2 Attitude

The attitude of a vehicle describes the orientation of its body-fixed reference frame $\mathcal{F}_{\mathbf{B}}$ with respect to another reference frame. In this thesis, the vehicle's attitude is described using aerodynamic angles. The use of Quaternions was deemed unnecessary in the absence of extremely large rotations.

The aerodynamic angles are the angle of attack α, the angle of sideslip β and the bank angle σ. The angle of attack represents the angle between the vehicle's velocity vector and its body axis in the vertical plane (Z_BY_B), i.e., the up-down orientation of its nose with respect to the trajectory. It is defined on the range $-180° \leq \alpha < 180°$, where a positive angle of attack represents a nose-up attitude. The angle of sideslip β is the angle between the vehicle's velocity vector and its body axis in the horizontal plane (X_BY_B), i.e., the left-right orientation of its nose with respect to the trajectory. It is defined on the range $-180° \leq \beta < 180°$, where a positive sideslip angle represents a nose-left attitude. The bank angle σ is the inclination of the vehicle about its longitudinal axis (X_B). It is defined on the range $-180° \leq \beta < 180°$, where a positive bank angle means an inclination to the right. When the vehicle is not banking, $Y_A = Y_T$. The aerodynamic angles are illustrated in Figure 3.3.

3.2.2 Transformations

Conversions between Cartesian and spherical components both in terms of position and velocity are given in Appendix A.2.

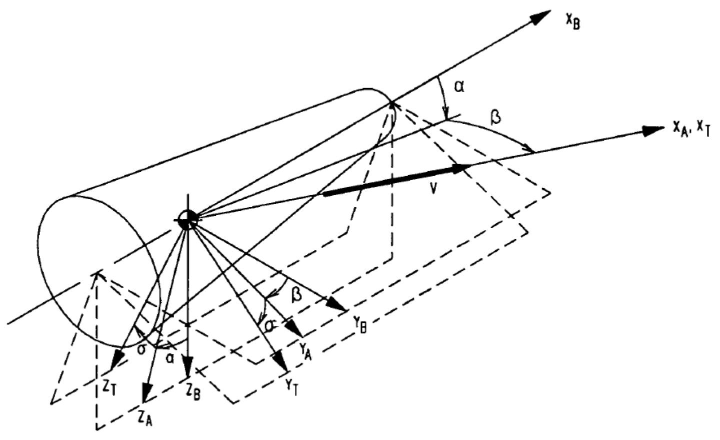


Figure 3.3: Definition of the attitude angles α, β, and σ. The A- and T-reference frames are groundspeed-based; subscripts have been omitted for clarity. Adapted from (Mooij, 1994).

3.3 External Forces and Moments

The re-entry vehicle is subject to a number of external forces and moments. In this thesis work, the vehicle is considered unpowered; therefore, only aerodynamic and gravitational forces and moments will be taken into account.

3.3.1 Aerodynamics

The aerodynamic forces and moments are dependent on the atmospheric density ρ and the airspeed variables V, α, β and σ. Defining quantities representing the aerodynamic conditions for the vehicle are the Mach number M and the dynamic pressure $\bar{q}$:

$$M = \frac{V}{a} \tag{3.1}$$

$$\bar{q} = \tfrac{1}{2}\rho V^2 \tag{3.2}$$

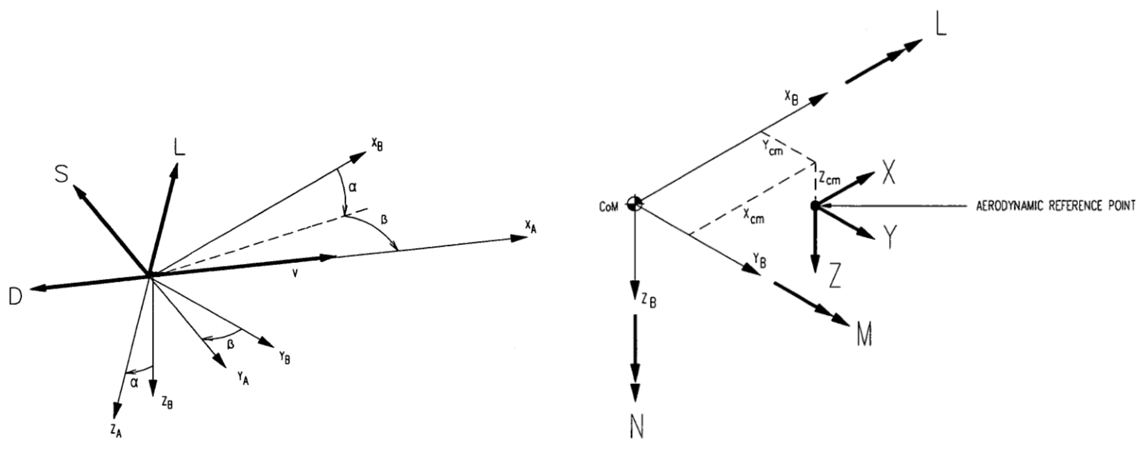
where a is the local speed of sound as defined in Equation 2.28.

3.3.1.1 Forces

The aerodynamic forces are defined in the aerodynamic reference frame $\mathcal{F}_{\mathbf{A}}$:

$$\mathbf{F}_{\mathbf{A},\mathbf{A}} = -\begin{bmatrix} D \\ S \\ L \end{bmatrix} = -\begin{bmatrix} C_D\bar{q}S_{ref} \\ C_S\bar{q}S_{ref} \\ C_L\bar{q}S_{ref} \end{bmatrix} \tag{3.3}$$

where D, S and L represent the aerodynamic drag, side, and lift force, respectively. These components are defined in opposite direction to the axes of the $\mathcal{F}_{\mathbf{A}}$-frame, hence the need for the minus signs. The orientation of the aerodynamic force components with respect to the A- and B-frames is shown in Figure 3.4(a). The coefficients C_D, C_S and C_L are the aerodynamic force coefficients – dimensionless quantities which represent the unique aerodynamic characteristics of the re-entry vehicle. They vary with Mach number and the aerodynamic angles. These coefficients cannot be computed analytically, but must be obtained empirically. Models and tabulated values with varying complexity for the relation between these coefficients, the Mach number and attitude angles exist for a multitude of (reference) re-entry vehicles; these can be applied in the simulation of the aerodynamic forces.



(a) Forces (b) Moments

Figure 3.4: Definition of the aerodynamic forces and moments (Mooij, 1994).

3.3.1.2 Moments

In contrast to the aerodynamic forces, the aerodynamic moments are expressed in the B-frame:

$$\mathbf{M_{A_M,B}} = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} C_l \bar{q} S_{ref} b_{ref} \\ C_m \bar{q} S_{ref} c_{ref} \\ C_n \bar{q} S_{ref} b_{ref} \end{bmatrix} \tag{3.4}$$

where L is the roll moment (rotation about the X_B-axis), M is the pitching moment (rotation about the Y_B-axis) and N is the yawing moment (rotation about the Z_B-axis). The coefficients C_l, C_m and C_n are aerodynamic moment coefficients, again uniquely defined for a specific re-entry vehicle and dependent on the Mach number and the aerodynamic angles. For winged vehicles, the reference lengths c_{ref} and b_{ref} generally represent the wing span and mean aerodynamic lift chord, respectively.

Depending on where the aerodynamic forces act on the vehicle, they will likely contribute to the aerodynamic moments. The vehicle's aerodynamic reference point – or aerodynamic center – is defined as the point about which the aerodynamic moment is constant. In avionics, it is customary to take this point as the location of where the aerodynamic force is applied. A force acting on the vehicle's aerodynamic reference point causes a contribution to the total aerodynamic moment vector of

$$\mathbf{M_{A_F,B}} = \mathbf{r_{cm}} \times \mathbf{F_{A,B}} \tag{3.5}$$

where $\mathbf{r_{cm}}$ is the vector distance between the vehicle's aerodynamic reference point and its c.o.m. (see Figure 3.4(b)).

The total aerodynamic moment vector is simply given by the sum

$$\mathbf{M_A} = \mathbf{M_{A_M,B}} + \mathbf{M_{A_F,B}} \tag{3.6}$$

3.3.2 Gravity

3.3.2.1 Forces

The gravitational acceleration experienced by the vehicle due to the Earth's gravitational field was discussed in Section 2.2.1. For this thesis work, a central field plus J_2 gravitational model was used; the gravitational acceleration vector $\mathbf{g_V}$ belonging to this model was given by Equation 2.12 and is repeated here:

$$\mathbf{g_V} = \begin{bmatrix} g_R \\ 0 \\ g_\delta \end{bmatrix} = \frac{\mu}{R^2} \begin{bmatrix} 1 - \frac{3}{2} J_2 \left(\frac{R_E}{R}\right)^2 \left(3\sin^2\delta - 1\right) \\ 0 \\ -3\left(\frac{R_E}{R}\right)^2 J_2 \sin\delta\cos\delta \end{bmatrix}$$

When the equations of motion are defined in spherical coordinates (in the R-frame), the components $\mathbf{g}_R$ and $\mathbf{g}_\delta$ are already in the right form. However, more often than not a representation in Cartesian coordinates in the I-frame is preferred to limit computation time. In Cartesian coordinates, $\mathbf{g}$ in the R-frame is given by (Mooij, 1997):

$$\mathbf{g_R} = \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} = -\frac{\mu}{R^3} \begin{bmatrix} x\left\{1 + \frac{3}{2}J_2\left(\frac{R_E}{R}\right)^2\left(1 - 5\frac{z^2}{R^2}\right)\right\} \\ y\left\{1 + \frac{3}{2}J_2\left(\frac{R_E}{R}\right)^2\left(1 - 5\frac{z^2}{R^2}\right)\right\} \\ z\left\{1 + \frac{3}{2}J_2\left(\frac{R_E}{R}\right)^2\left(3 - 5\frac{z^2}{R^2}\right)\right\} \end{bmatrix} \tag{3.7}$$

where R is the radial distance from the vehicle to the c.o.m. of the central body

$$R = \sqrt{x_R^2 + y_R^2 + z_R^2} \tag{3.8}$$

The gravitational force experienced by the re-entry vehicle in the I-frame is thus given by

$$\mathbf{F_{G,I}} = \mathbf{C_{I,R}}\mathbf{F_{G,R}} = m\mathbf{C_{I,R}}\mathbf{g_R} \tag{3.9}$$

where m is the vehicle's mass.

3.3.2.2 Moments

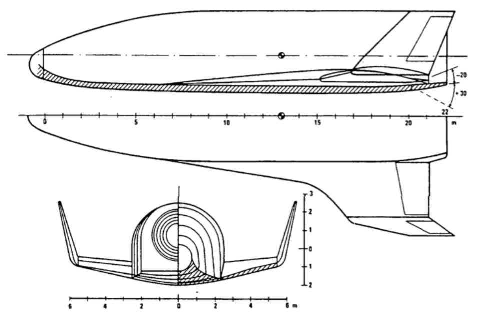
Gravitational moments may arise due to differences in the gravity gradient experienced by the vehicle; these moments are however negligible compared to the aerodynamic moments and are thus not taken into account.

3.4 Vehicle Model

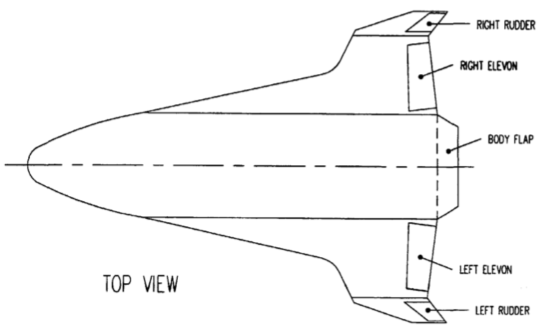
Once the flight environment models have been defined, the dynamics of the re-entry vehicle under influence of this environment can be determined. For this purpose, the vehicle itself must be modeled in such a way that the model sufficiently accurately represents the vehicle's flight characteristics under the influence of the environment within the flight interval of interest.

For this thesis work, the choice was made to simulate the vehicle's trajectory using a "3 + 1-degree of freedom (DOF)" flight dynamic model instead of the full 6-DOF representation. The 6-DOF model takes both translational and rotational motion into account, whereas 3-DOF only involves translational motion. Defining the vehicle's attitude angles directly without simulation of the necessary body-axis rotations gives the same amount of information about the trajectory profile. In addition to simulating the translational motion, consideration is paid to keeping the vehicle in the trimmed flight condition – i.e., longitudinally stable – by modulating the body flap deflection angle.

The reference vehicle used for this thesis work is the HORUS-2B (see Figure 3.5(a)); it is a Space Shuttle-like unpowered winged re-entry vehicle which was designed as a reusable second stage to the Ariane-5. The choice for the HORUS reference vehicle is twofold: its winged design allows for it to perform skipping entry even when returning from LEO, and there is a large amount of aerodynamic data available thanks to studies conducted by MBB (1988) and Mooij (1995). The data used for this thesis was exclusively obtained from Mooij (1995). In addition, the availability of simulation data of the HORUS-2B's nominal re-entry profile from Mooij (1998) was greatly beneficial for verification. The HORUS's nominal mission involves atmospheric re-entry at an altitude of 122 km; the re-entry phase ends when the vehicle is within 80 km of the runway. As HORUS is unpowered, steering occurs by modulating its angle of attack α and bank angle σ; the angle of sideslip β is considered a disturbance and is set to zero.



(a) HORUS-2B shape



(b) HORUS-2B control surfaces

Figure 3.5: HORUS-2B vehicle geometry (Mooij, 1995).

Table 3.1: HORUS-2B vehicle dimensions (Mooij, 1995).

Parameter		**Value**
Total length	m	25.0
Wing span	m	13.0
Total height	m	4.5
Exposed wing area	m^2	43
Projected fuselage area	m^2	101
Total plan view area	m^2	151
Reference area	m^2	110
Nose radius	m	0.8
c.o.m. x_{cm}	m	13
Landing mass	kg	26,029

3.4.1 Geometry

The shape of the HORUS resembles that of the Space Shuttle; see Figure 3.5(a). Some defining dimensions of HORUS are shown in Table 3.1. Lengths are measured in the B-frame, of which the origin is located 1 m in front of the vehicle's nose.

3.4.2 Control surfaces

As control surfaces, the HORUS-2B has two rudders, two elevons, and one body flap. The locations of these control surfaces are shown in Figure 3.5(b). Each type of control surface effects the main contribution to the aerodynamic moments about a particular axis. The rudders induce a yawing moment N about the Z_B axis, the ailerons induce a rolling moment L about the X_B-axis, and finally the body flap induces a pitching moment M about the Y_B-axis. Naturally, deflections of the control surfaces also contribute to some small degree to moments about the other axes well. In the "3+1-DOF" flight dynamic model, only longitudinal (pitching) moments are taken into account; the largest contributor to these moments by far is the body flap, and therefore the effects of the rudders and elevons are neglected; elevons become very important at low Mach numbers when trimmed flight cannot be maintained using the body flap – however, these velocity regimes are not treated in this thesis. As the yawing and rolling motion are neglected in the flight dynamic model, the (minor) contribution of the body flap to these motions is not relevant.

3.4.3 Aerodynamics

The aerodynamic database for the HORUS obtained from Mooij (1995) consists of tabulated values of the vehicle's base aerodynamic coefficients as a function of both the angle of attack α and the Mach number M, and the contributions to the aerodynamic coefficients due to the control surfaces as a function of α, M, and the control surface deflection angle. These values are interpolated using a bilinear and Lagrange linear interpolator to obtain the aerodynamic coefficients at a particular flight condition. The tabulated values of the relevant aerodynamic coefficients (the drag coefficient C_D, the lift coefficient C_L and the pitch moment coefficient C_m) for the clean configuration of the vehicle (indicated with subscript '0'), as well as the increments to C_D, C_L and C_m due to the body flap deflection (ΔC_{D_b}, ΔC_{L_b} and ΔC_{m_b}, respectively) can be found in Appendix B.

The database is based on a number of assumptions and simplifications (the legitimacy of these is assumed to be verified):

- Aeroelastic effects are not included.
- The influence of the Reynolds number on the skin-friction drag is included in the drag coefficient as averaged skin friction drag along a standard trajectory down to an altitude of 20 km. At lower altitudes, a drag decrement parameter needs to be included in the computation of the total drag coefficient.
- The flaps produce no interference effect.

As the vehicle trajectory is simulated to approximately an altitude of 25 km, the altitude-dependent drag decrement is not taken into account.

The aerodynamic force coefficients of HORUS as applied in this thesis work are thus given by

$$C_D = C_{D_0} + \Delta C_{D_b} \tag{3.10a}$$
$$C_S = 0 \tag{3.10b}$$
$$C_L = C_{L_0} + \Delta C_{L_b} \tag{3.10c}$$

And the aerodynamic moment coefficients:

$$C_l = 0 \tag{3.11a}$$
$$C_m = C_{m_0} + \Delta C_{m_b} \tag{3.11b}$$
$$C_n = 0 \tag{3.11c}$$

3.5 Equations of Motion

In the re-entry simulator developed for this thesis, the equations of motion were implemented in Cartesian coordinates in the I-frame. All relevant computed components are transformed to the I-frame before numerical integration.

3.5.1 Translational Motion

3.5.1.1 Cartesian components

The translational motion of a rigid body of constant mass m with respect to the inertial planetocentric frame is defined by

$$\frac{d\mathbf{V_I}}{dt} = \tfrac{1}{m}\mathbf{F_I} \tag{3.12a}$$
$$\frac{d\mathbf{r_I}}{dt} = \mathbf{V_I} \tag{3.12b}$$

where $\mathbf{r_I}$ is the distance to the I-frame's origin, $\mathbf{F_I}$ is the total of external forces acting on the body, and $\mathbf{V_I}$ is the body's inertial velocity. Equation 3.12a is called the dynamic equation; it describes the motion of a body under the influence of external forces in the form of an acceleration vector. Equation 3.12b is the kinematic equation, which describes the body's corresponding change in position, i.e., velocity. The Horus re-entry vehicle can be considered as a rigid body; additionally, the vehicle is unpowered, meaning it can also be assumed to have constant mass (as it uses no propellant). Therefore Equations 3.12 can be used to describe the motion of Horus in the I-frame. The Cartesian position and velocity vectors are denoted

$$\mathbf{r_I} = (x_I, y_I, z_I)^T \tag{3.13a}$$

$$\mathbf{V_I} = (\dot{x}_I, \dot{y}_I, \dot{z}_I)^T \tag{3.13b}$$

As the re-entry vehicle is unpowered, it experiences only the influence of aerodynamic and gravitational accelerations; these are defined in Sections 3.3.1-3.3.2. The resultant external force acting on the body is thus simply the sum of the aerodynamic and gravitational forces:

$$\mathbf{F_I} = \mathbf{F_{A,I}} + \mathbf{F_{G,I}} \tag{3.14}$$

The dynamic and kinematic equations are thus given by

$$\frac{d\mathbf{V_I}}{dt} = (\ddot{x}_I, \ddot{y}_I, \ddot{z}_I)^T = \tfrac{1}{m}\left(\mathbf{F_{A,I}} + \mathbf{F_{G,I}}\right) \tag{3.15a}$$

$$\frac{d\mathbf{r_I}}{dt} = (\dot{x}_I, \dot{y}_I, \dot{z}_I)^T \tag{3.15b}$$

The aerodynamic force $\mathbf{F_{A,A}}$ is computed in the (airspeed-based) aerodynamic frame, whereas the gravitational force $\mathbf{F_{G,V}}$ is computed in the vertical frame. Both first need to be transformed to the I-frame to compute $\mathbf{F_I}$.

$$\mathbf{F_{A,I}} = \mathbf{C_{I,A}}\mathbf{F_{A,A}} \tag{3.16a}$$

$$\mathbf{F_{G,I}} = \mathbf{C_{I,V}}\mathbf{F_{G,V}} \tag{3.16b}$$

Sometimes the motion of the vehicle may need to be defined with respect to the R-frame. For example, the aerodynamic force $\mathbf{F_{A,A}}$ is defined with respect to a moving atmosphere, and is dependent on airspeed-based quantities such as the angle of attack and Mach number. The groundspeed (which is equal to airspeed in the absence of wind) is defined with respect to a rotating frame, augmenting the inertial velocity $\mathbf{V_I}$ with a term due to the Earth's rotation. The velocity vector transformation from the I-frame to the R-frame is given in Equation A.11, and is repeated here:

$$\mathbf{V_R} = \mathbf{C_{R,I}}\left(\mathbf{V_I} - \omega_{\mathbf{cb}} \times \mathbf{r_I}\right) \tag{3.17}$$

With this definition of $\mathbf{V_R}$, the dynamic and kinematic equations in the R-frame are simply

$$\frac{d\mathbf{V_R}}{dt} = (\dot{u}, \dot{v}, \dot{w})^T = \tfrac{1}{m}\left(\mathbf{F_{A,R}} + \mathbf{F_{G,R}}\right) \tag{3.18a}$$

$$\frac{d\mathbf{r_R}}{dt} = (u, v, w)^T \tag{3.18b}$$

3.5.1.2 Spherical components

The definition of the equations of motion in spherical components is somewhat easier to interpret in terms of the physical quantities of the individual components; it highlights individual dependencies between the state variables. However, they have the large disadvantage of being much more complex and time-consuming to numerically integrate; therefore for numerical implementations the Cartesian representation is highly preferable. They will nevertheless be discussed here for the purpose of providing the reader with some physical insight into the vehicle's equations of motion.

The state variables in spherical components were discussed in Section 3.2.1.1. To recap: the position is defined by a radial distance R, longitude τ and latitude δ, and the (groundspeed-based) velocity by its magnitude V_G, flight-path angle γ_G and heading χ_G. The dynamic and kinematic equations are defined individually for each state variable. Taking into account previously defined assumptions (no side force S, no longitudinal gravitational acceleration component g_τ), the dynamic equations are given by:

$$\frac{dV}{dt} = \frac{F_V}{m} + {\omega_{cb}}^2 R\cos\delta\left(\sin\gamma\cos\delta - \cos\gamma\sin\delta\cos\chi\right) \tag{3.19a}$$

$$\frac{1}{\dot{V}}\frac{d\gamma}{dt} = \frac{F_\gamma}{m} + 2\omega_{cb} V\cos\delta\sin\chi + \frac{V^2}{R}\cos\gamma + {\omega_{cb}}^2 R\cos\delta\left(\cos\gamma\cos\delta + \sin\gamma\sin\delta\cos\chi\right) \tag{3.19b}$$

$$V\cos\gamma\frac{d\chi}{dt} = \frac{F_\chi}{m} + 2\omega_{cb} V\left(\sin\delta\cos\gamma - \cos\delta\sin\gamma\cos\chi\right) + \frac{V^2}{R}\cos^2\gamma\tan\delta\sin\chi + \ldots \tag{3.19c}$$
$$\qquad {\omega_{cb}}^2 R\cos\delta\sin\delta\sin\chi$$

where the subscript G is dropped for the airspeed variables in favor of readability. The terms including the rotational rate of the Earth ($2\omega_{cb}$ and ${\omega_{cb}}^2$) represent the Coriolis acceleration and transport acceleration, respectively, and account for the accelerations incurred due to the vehicle moving in a rotating frame. The V^2/R-terms represent the influence of the Earth's curvature (Mooij, 2013). The force magnitudes F_V, F_γ and F_χ are defined as:

$$F_V = -D - mg_r\sin\gamma - mg_\delta\cos\gamma\cos\chi \tag{3.20a}$$
$$F_\gamma = L - mg_r\cos\gamma - mg_r\cos\gamma + mg_\delta\sin\gamma\cos\chi \tag{3.20b}$$
$$F_\chi = -L\sin\sigma + mg_\delta\sin\chi \tag{3.20c}$$
$$\tag{3.20d}$$

The kinematic equations are somewhat more straightforward, and are given by

$$\frac{dR}{dt} = V\sin\gamma \tag{3.21a}$$
$$\frac{d\tau}{dt} = \frac{V\sin\chi\cos\gamma}{R\cos\delta} \tag{3.21b}$$
$$\frac{d\delta}{dt} = \frac{V\cos\chi\cos\gamma}{R} \tag{3.21c}$$

The spherical equations of motion were used in the verification process of the simulator.

3.5.2 Rotational Motion

As was mentioned in Section 3.4, a "3+1-DOF" flight dynamic model is used in this thesis. The first three degrees of freedom obviously represent the translational components of the motion that were discussed in the previous section. The remaining degree of freedom occurs in the form of a rotation in the longitudinal direction of the vehicle (i.e., pitch), which is exclusively used to keep the vehicle in a trimmed flight condition. Therefore the formal definition of the rotational equations of motion (as derived in Mooij (1994)) is not relevant in this case, and a pseudo-rotational motion equation is defined to account for maintaining trimmed flight.

To achieve trimmed flight, there must exist a moment equilibrium in the longitudinal direction, i.e., the total pitch moment must be null. The total pitch moment C_m for the Horus vehicle model is given in Equation 3.11b as

$$C_m = C_{m_0} + \Delta C_{m_b}$$

where

$$C_{m_0} = f(\alpha_C, M) \tag{3.22a}$$
$$\Delta C_{m_b} = f(\delta_b, \alpha_C, M) \tag{3.22b}$$

where the subscript C indicates that the angle α_C is commanded by the guidance system. These relationships are not defined analytically; values of C_{m_0} and ΔC_{m_b} given α, M and – in the case of ΔC_{m_b}, also δ_b – can be obtained from tables (see Appendix B) using interpolation methods.

From the equilibrium condition $C_m = 0$ follows that for trimmed flight

$$\Delta C_{m_b} = -C_{m_0} \tag{3.23}$$

The only unknown in this inequality is the required body flap deflection angle that makes the equilibrium condition hold. This angle can be determined by inverse-interpolating the δ_b values for which ΔC_{m_b} is defined to obtain the deflection angle for which $\Delta C_{m_b} = -C_{m_0}$ holds.

Chapter 4

Numerical Methods

This chapter presents the most prominent numerical methods implemented in the development of the re-entry simulator as well as the mission planner. Section 4.1 gives an overview of numerical integration methods. Both polynomial and spline interpolation methods are discussed in Section 4.2. Data search methods are applied quite often due to many models existing in table form; the methods used in this thesis work are outlined in Section 4.3. This chapter is concluded with a short explanation of the theory behind Monte Carlo simulations in Section 4.5.

4.1 Integration

An ordinary differential equation (ODE) of order n is defined as an equation containing a function of one independent variable $\mathbf{x}(t)$ and its derivatives $\mathbf{x}^{(i)}(t)$ with $i = 0, 1, \ldots, n$ (Press et al., 1989):

$$\mathbf{x}^{(n)} = f\left(t, \mathbf{x}', \mathbf{x}'', \ldots, \mathbf{x}^{(n-1)}\right) \tag{4.1}$$

An ODE is said to be linear when it can be written as a linear combination of the derivatives $\mathbf{x}^{(i)}(t)$:

$$\mathbf{x}^{(n)} = \sum_{i=0}^{n-1} a_i(t)\mathbf{x}^{(i)} + r(t) \tag{4.2}$$

where $a_i(t)$ and $r(t)$ are continuous functions in t. Linear differential equations have exact, closed-form solutions. However, most practical problems (such as equations of motion) are nonlinear and approximate solutions must be obtained numerically.

A generic problem defined by ODEs can be rewritten in the form of a set of n coupled first-order differential equations for the functions $\mathbf{x}_i$ with $(i = 1, 2, \ldots, n)$:

$$\mathbf{x}'_i = f_i\left(t, x_1, x_2, \ldots, x_n\right) \tag{4.3}$$

which allows for all computations to be performed in a single integration step – by including all parameters in a single array – instead of successively (Noomen, 2011). Integration of an ODE (or set of ODEs) is the process of solving the initial value problem, wherein all the values of x_i are defined at some initial point t_0 and must be computed at some final point t_f. The initial value problem cannot be solved analytically if the set of ODEs is nonlinear – which it is, for most practical purposes. Defining an ODE in terms of finite steps $\Delta\mathbf{x}$ and Δt gives an algebraic formula for the change in the function value when $\mathbf{x}$ is incremented by one stepsize Δt. Taking the limit of $\Delta t \to 0$ allows for an approximation of of the solution. Literal implementation of this procedure is the well-known Euler's method:

$$\mathbf{x}(t + \Delta t) \approx \mathbf{x}(t) + \Delta\mathbf{x} \quad \text{where} \quad \Delta\mathbf{x} = \dot{\mathbf{x}}(t)\Delta t \tag{4.4}$$

wherein small increments $\Delta\mathbf{x}$ are added to $\mathbf{x}$ in the form of derivatives multiplied by stepsizes Δt. While Euler's method is not recommended for any practical use, it gives a good indication of the underlying idea used in practical numerical integration methods (Press et al., 1989). Aspects to keep in mind when choosing a numerical integration method are (Noomen, 2011):

Truncation error: the error that is made when performing a single integration step (i.e., local error).

Ease of changing stepsize: the stepsize affects the accuracy of the integration result, and should be externally adjustable.

Speed: efficiency is defined as the number of evaluations for a defined interval, for a given accuracy. A tradeoff between accuracy and speed must be done, with priorities depending on the application.

Stability: also known as robustness; it gives a measure of how sensitive the end results are to (minor) changes in the initial conditions.

Error accumulation: the total uncertainty of the problem given all error sources. When a sequence of calculations subject to rounding error is made, errors may accumulate, sometimes dominating the calculation.

The most commonly used numerical integration methods are the Runge-Kutta methods. These methods propagate a solution over an interval by combining the information from several Euler-style steps, and then using the information obtained to match a Taylor series expansion up to some higher order (Press et al., 1989). Runge-Kutta methods have the advantage of being relatively computationally simple; they virtually always succeed and are the go-to method when moderate accuracy is required. The choice to use Runge-Kutta methods was based on the availability of a verified Runge-Kutta integrator within the TU Delft Astrodynamics Toolbox (TUDAT) repository (Doornbos, 2014).

4.1.1 Fourth-Order Runge-Kutta

The fourth-order Runge-Kutta (RK4) method – also known as the classical Runge-Kutta – is considered the workhorse for solving many engineering problems (Press et al., 1989). The process is simple: four different approximations are made to describe the changes of the parameters during one integration step. These values are then averaged in a weighted fashion to obtain the solution (Noomen, 2011). It is a single-step technique, i.e., only the value of x_i is used to obtain x_{i+1} and previous values of x do not factor into the solution. The RK4 method is defined by the following formulas:

$$x_{n+1} = x_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + O(\Delta t^5) \tag{4.5}$$

where

$$k_1 = f\left(t_n, x_n\right)\Delta t \tag{4.6}$$
$$k_2 = f\left(t_n + \tfrac{\Delta t}{2}, x_n + \tfrac{k_1}{2}\right)\Delta t \tag{4.7}$$
$$k_3 = f\left(t_n + \tfrac{\Delta t}{2}, x_n + \tfrac{k_2}{2}\right)\Delta t \tag{4.8}$$
$$k_4 = f\left(t_n + \Delta t, x_n + k_3\right)\Delta t \tag{4.9}$$
$$\tag{4.10}$$

The four increments k are given by the product of the stepsize Δt and an estimated slope specified by the value of the function f at different locations. The weighted average of these increments is Δx for that step. The term $O(\Delta t^5)$ represents the truncation error, which is of the order Δt^5. The RK4 method requires four evaluations of the right-hand side per step Δt. For Runge-Kutta formulas of orders M higher than four, more than M function evaluations (though never more than $M + 2$) are required; therefore the RK4 is the most "cost-efficient" of the Runge-Kutta family, explaining its popularity (Press et al., 1989). The stepsize Δt is fixed, which means the method is not adaptive to the shape of the derivative function.

4.1.2 Runge-Kutta-Fehlberg

The benefit of an adaptive-stepsize algorithm is that it does take the shape of the derivative function into account, and can thus take smaller steps when necessary (increasing accuracy), but also larger steps when possible (increasing speed). The algorithm determines the value of Δt such that the truncation error is kept below a predefined value for each step; in order to do this, the truncation error must be estimated with each step.

Adaptive-stepsize Runge-Kutta algorithms are based on the embedded Runge-Kutta formulas originally developed by Erwin Fehlberg. He discovered a fifth-order Runge-Kutta method requiring six function evaluations where another combination of the six functions results in a fourth-order method; this method is commonly referred to as the Runge-Kutta-Fehlberg (RKF45) method (Press et al., 1989). At each step, it determines whether the proper Δt is being used by comparing two different approximations of the solution – obtained using the fifth-order method and the embedded fourth-order formula. The difference is used as an estimate of the truncation error. If the two approximations are sufficiently close, the stepsize is deemed correct. If the two values do not agree to a specified accuracy, Δt is reduced; if the answers agree to more significant digits than necessary, Δt is increased Mathews and Fink (2004).

The following intermediate values are defined for each RKF45 step:

$$\begin{aligned}
k_1 &= hf(t_n, x_n) \\
k_2 &= hf(t_n + a_2 h, x_n + b21k_1) \\
k_3 &= hf(t_n + a_3 h, x_n + b_{31}k_1 + b_{32}k_2) \\
k_4 &= hf(t_n + a_4 h, x_n + b_{41}k_1 + b_{42}k_2 + b_{43}k_3) \\
k_5 &= hf(t_n + a_5 h, x_n + b_{51}k_1 + b_{52}k_2 + b_{53}k_3 + b_{54}k_4) \\
k_6 &= hf(t_n + a_6 h, x_n + b_{61}k_1 + b_{62}k_2 + b_{63}k_3 + b_{64}k_4 + b_{65}k_5)
\end{aligned} \tag{4.11}$$

The fifth-order solution is given by

$$x_{n+i} = x_n + c_1k_1 + c_2k_2 + c_3k_3 + c_4k_4 + c_5k_5 + c_6k_6 + O(\Delta t^6) \tag{4.12}$$

and the embedded fourth-order formula is

$$x^*_{n+i} = x_n + c^*_1k_1 + c^*_2k_2 + c^*_3k_3 + c^*_4k_4 + c^*_5k_5 + O(\Delta t^5) \tag{4.13}$$

The constants a, b, c and c^* have several definitions. The original values as developed by Fehlberg are given in Table 4.1.

The truncation error estimate is defined as:

$$\varepsilon \equiv x_{n+1} - x^*_{n+1} = \sum_{i=1}^{6} (c_i - c^*_i)\, k_i \tag{4.14}$$

which scales with Δt^5. If a certain step Δt produces an error ε, the step Δt^* that would have given the desired accuracy ε^* can be estimated:

$$\Delta t^* = \Delta t \left| \frac{\varepsilon^*}{\varepsilon} \right|^{\frac{1}{5}} \tag{4.15}$$

The amount with which to increase/decrease Δt to achieve the desired accuracy naturally follows.

4.2 Interpolation

4.2.1 Polynomial Interpolation

Lagrange or polynomial interpolation involves fitting a curve defined by an n-th degree polynomial to $n+1$ points. It has the drawback of being non-local, i.e, changing the value of one data point affects the shape of the entire curve. It is an adequate method when low-degree polynomials are needed; increasing the size of the dataset results in highly irregular curves which is generally unwanted.

Table 4.1: RKF45 parameters as defined by Fehlberg (Fehlberg, 1968).

i	a_i	b_{ij}					c_i	c_i^*
1							$\frac{16}{135}$	$\frac{25}{216}$
2	$\frac{1}{4}$	$\frac{1}{4}$					0	0
3	$\frac{3}{8}$	$\frac{9}{32}$	$\frac{9}{40}$				$\frac{6656}{12825}$	$\frac{1408}{2565}$
4	$\frac{12}{13}$	$\frac{1932}{2197}$	$-\frac{7200}{2197}$	$\frac{7296}{2197}$			$\frac{28561}{56430}$	$\frac{2197}{4104}$
5	1	$\frac{439}{216}$	-8	$\frac{3680}{513}$	$-\frac{845}{4104}$		$-\frac{9}{50}$	$-\frac{1}{5}$
6	$\frac{1}{2}$	$-\frac{8}{27}$	2	$-\frac{3544}{2565}$	$\frac{1859}{4104}$	$-\frac{11}{40}$	$\frac{2}{55}$	0
$j =$		1	2	3	4	5		

The generalized form of the interpolating polynomial of degree $n-1$ through the points $y_i = f(x_i)$ with $i = 1, \dots, n$ is given by

$$y = \sum_{j=1}^{n} \left(y_j \prod_{\substack{k=1 \\ k \neq j}}^{n} \frac{x_k - x}{x_k - x_j} \right) = y_1 \frac{(x_2 - x)\dots(x_n - x)}{(x_2 - x_1)\dots(x_n - x_1)} + \dots + y_n \frac{(x_1 - x)\dots(x_{n-1} - x)}{(x_1 - x_n)\dots(x_{n-1} - x_1)} \tag{4.16}$$

Linear interpolation

The commonly used linear interpolation formula is simply Equation 4.16 with $n = 2$. Given a tabulated function $y_i = f(x_i)$ with $i = 1, \dots, n$, linear interpolation in an interval $[x_j, x_{j+1}]$ gives

$$y = Ay_j + By_{j+1} \tag{4.17}$$

with

$$A \equiv \frac{x_{j+1} - x}{x_{j+1} - x_j} \tag{4.18a}$$

$$B \equiv \frac{x - x_j}{x_{j+1} - x_j} \tag{4.18b}$$

Bilinear interpolation

Bilinear interpolation is the application of linear interpolation within a two-dimensional data matrix defined by the tabulated function $z_{i,j} = f(x_i, y_j)$. A point $P = (x, y)$ is located within the interval $[(x_i, x_{i+1}), (y_j, y_{j+1})]$ of which the values of z at the four corners $Q_{11} = (x_i, y_j)$, $Q_{12} = (x_i, y_{j+1})$, $Q_{21} = (x_{i+1}, y_j)$, and $Q_{22} = (x_{i+1}, y_{j+1})$ are known. To determine $f(P)$, two consecutive interpolations must be performed: one in the x-direction and one in the y-direction. The results of the first interpolation are the function values at the points $R_1 = (x, y_j)$ and $R_2 = (x, y_{j+1})$:

$$f(R_1) \approx f(Q_{11}) \cdot \frac{x_{i+1} - x}{x_{i+1} - x_i} + f(Q_{21}) \cdot \frac{x - x_i}{x_{i+1} - x_i} \tag{4.19a}$$

$$f(R_2) \approx f(Q_{12}) \cdot \frac{x_{i+1} - x}{x_{i+1} - x_i} + f(Q_{22}) \cdot \frac{x - x_i}{x_{i+1} - x_i} \tag{4.19b}$$

What remains is performing a linear interpolation in the y-direction to obtain $z = f(x, y)$:

$$z \approx f(R_1) \cdot \frac{y_{j+1} - y}{y_{j+1} - y_j} + f(R_2) \cdot \frac{y - y_j}{y_{j+1} - y_j} \tag{4.20}$$

Bilinear interpolation is used in this work to compute the aerodynamic coefficients of HORUS from tabulated values.

4.2.2 Spline Interpolation

A spline is defined as a piecewise polynomial. Interpolation using splines involves fitting a multitude of low-degree polynomials through data points to create a curve fit. The degree of continuity C_n of a spline refers to how well all the different polynomials fit together. C_0-continuity means that the splines are continuous in position (i.e., they originate/end in the same point), C_1 that they are tangent and C_2 that they are continuous in curvature. The degree to which continuity is required depends on the application. Often, C_0 and C_1 continuity is sufficient.

Cubic splines

Cubic spline polynomials are the most common; they are the lowest order polynomials that can be used to interpolate between two points. The goal of cubic spline interpolation is to obtain an interpolation formula that is smooth in the first derivative, and continuous in the second derivative, both within an interval and at its boundaries (Press et al., 1989).

Given a tabulated function $y_i = f(x_i)$ and its tabulated second derivatives $y_i'' = f''(x_i)$, the interpolating cubic spline over the interval (x_j, x_{j+1}) is given by

$$y = Ay_j + By_{j+1} + Cy_j'' + Dy_{j+1}'' \tag{4.21}$$

where A and B were defined in Equations 4.18 and

$$C \equiv \tfrac{1}{6}(A^3 - A)(x_{j+1} - x_j)^2 \tag{4.22a}$$

$$D \equiv \tfrac{1}{6}(B^3 - B)(x_{j+1} - x_j)^2 \tag{4.22b}$$

Hermite splines

A (cubic) Hermite spline is a spline consisting of third-degree polynomials specified by both their values and first derivatives at the end points of their respective domain intervals. The Hermite interpolating polynomial for a unit interval $(0, 1)$ with starting point p_0 and endpoint p_1 with tangents m_0 and m_1, respectively, is given by

$$p(t) = h_{00}(t)p_0 + h_{10}(t)m_0 + h_{01}(t)p_1 + h_{11}(t)m_1 \tag{4.23}$$

where $t \in [0, 1]$ and the basis functions $h(t)$ are defined as

$$\begin{aligned} h_{00}(t) &= 2t^3 - 3t^2 + 1 \\ h_{10}(t) &= t^3 - 2t^2 + 1 \\ h_{01}(t) &= -2t^3 + 3t^2 \\ h_{11}(t) &= t^3 - t^2 \end{aligned}$$

Interpolation on an arbitrary interval (x_j, x_{j+1}) then occurs by mapping this interval to the unit interval through a change of variable:

$$p(x) = h_{00}(t)p_j + h_{10}(t)(x_{j+1} - x_j)m_j + h_{01}(t)p_{j+1} + h_{11}(t)(x_{j+1} - x_j)m_{j+1} \tag{4.24}$$

with

$$t = \frac{x - x_j}{x_{j+1} - x_j}$$

All that remains is choosing the tangents m. If the tangents for intervals sharing endpoints are equal, the resulting curve is globally continuously differentiable. Several different methods of choosing the tangents exist; a cardinal spline is obtained when the tangents are chosen using

$$m_j = (1 - c)\frac{p_{j+1} - p_{j-1}}{t_{j+1} - t_{j-1}} \tag{4.25}$$

where $0 \leq c \leq 1$ is the tension parameter.

Hermite spline interpolation is used in this work to compute the attitude commands between control nodes (see Section 5.3.3).

4.3 Data Search

4.3.1 Binary Search

A binary search/ bisection algorithm finds the position of a specified input value (i.e., the search key) within an array sorted by key value (either ascending or descending) (Cormen et al., 2009). The algorithm works on a sorted table by comparing the search key value with the key value of the middle element of the table, eliminating the half of the table in which the key cannot lie, and then repeating the procedure iteratively. If the keys match, then a matching element has been found and its index, or position, is returned. This method will find the right place in about $\log_2 n$ tries (Press et al., 1989).

4.3.2 Search with Correlated Values

In the event that a large table needs to be searched many times, it is inefficient to perform a binary search with each consecutive search. Preceding a binary search with a ”hunting” algorithm increases efficiency. The routine starts with a guessed position in the table, after which it hunts either up or down in increments of 1, then 2, then 4, etc., until the search key is bracketed. Then it uses bisection to search the bracketed interval. At worst, this routine is a factor 2 slower than a simple binary search, and at best a factor $\log_2 n$ faster (Press et al., 1989).

4.3.3 Nearest Neighbor Search

The nearest neighbor search problem is an optimization problem for finding closest (or most similar) points. Formally, the problem is defined as follows: given a set S of points in a space M and a query point $q \in M$, find the closest point in S to q. Closeness is generally defined in terms of a dissimilarity function: the less similar the objects, the larger the function values.

The simplest nearest neighbor search method to perform a linear search by computing the distance from the search key to every other key value, and keeping track of the closest match found so far, commonly referred to as the naive approach. In this thesis work, the nearest neighbor search was conducted by applying a binary search algorithm.

Data search is applied in this work in the computation of the aerodynamic coefficients of HORUS from tabulated values.

4.4 Pseudo-Random Number Generation

A Pseudorandom Number Generator (PRNG) is an algorithm that generates a sequence of numbers that closely resembles the properties of a truly random sequence. The precise philosophy and qualitative discussion of PRNGs is beyond the scope of this text. However, an essential property of a PRNG is its *seed*, a set of initial values upon which the sequence of pseudorandom numbers it generates is based. Within this work, a PRNG from the $C++$-based Boost library called mt19937 (Watanabe, 2010) is used.

4.5 Monte Carlo Simulation

Monte Carlo is, simply said, the approach of using random numbers to compute something which is not random (Robert and Casella, 2004). Monte Carlo simulations involve running the same simulation a large number of times with randomly determined inputs to obtain the probabilistic distribution of or other information about a problem. Note the distinction between Monte Carlo methods and simulations: simulation refers to producing random variables with a certain distribution just to look at the simulation results, whereas Monte Carlo methods are a tool of quantitative statistical analysis; these are beyond the scope of this text. Performing a Monte Carlo simulation is especially useful for problems with many degrees of freedom and nonlinearities, of which the distribution of the outcome cannot be estimated analytically. The re-entry problem is such a problem, as the outcome is dependent on so many factors that a reliable estimate of the realm of possibilities for the results is difficult to determine.

The implementation of a Monte Carlo simulation has at its core a PRNG designed to generate a sequence of random numbers with a uniform distribution; this PRNG is used to produce the random inputs for the simulations.

Chapter 5

Guidance and Control Design

For a re-entry mission to be successful, a vehicle has to follow a trajectory within its entry corridor and reach a prespecified landing location at a certain time. The geometry of the entry corridor is determined by vehicle and crew/cargo constraints in the form of load and heat limits, as well as the controllability of the vehicle (see Section 2.1.2). Additional limitations may be imposed based on, e.g., safety considerations for the surrounding area. To maintain a trajectory within this corridor and to end up in the correct place at the correct time, the vehicle's motion must be controlled in some fashion. The system responsible for this is the vehicle's Guidance, Navigation and Control (GNC) system. The GNC system in general is discussed in Section 5.1. The guidance system in the context of the mission planner is discussed in more detail in Section 5.2. Section 5.3 does the same for the control system.

5.1 GNC Systems

As its name suggests, a GNC system consists of three separate modules. These modules work in tandem to keep the vehicle on course by issuing steering commands. These commands depend on the vehicle's current state compared to its intended (reference) trajectory, as well as mission-specific considerations such as loading and trajectory constraints. A simplified illustration of the workings of a GNC system is shown in Figure 5.1. The on-board mission planner determines the state the vehicle should be in based on its current state in relation to a reference trajectory (Tigges et al., 2006).

Based on the difference between the reference state and the vehicle's current state, the guidance logic module determines the resultant force and moment the vehicle needs to be subjected to achieve that state. The forces and moments acting on the vehicle can be adjusted by modulating guidance parameters, i.e., the vehicle's aerodynamic profile is adjusted by modifying its attitude and – in the case of powered vehicles – an additional thrust moment can be effected. The guidance system issues a command in the form of these parameters to the control module.

The control module consists of two separate parts: the abstract control algorithm (i.e., software) and the physical actuators of the vehicle (i.e., hardware). The control algorithm translates the commands from the guidance system into commands that can be issued to the actuators, e.g., relating an attitude command to the control surface deflections necessary to achieve that attitude. These commands are computed based on vehicle-specific models that define its response to actuator settings. For example, the HORUS reference vehicle's aerodynamic response to the body flap deflection angle δ_b under certain α and M conditions is tabulated in terms of increments to the aerodynamic coefficients – see Tables B.5 through B.22. The actuators perform the control commands issued to them, the vehicle experiences a change in forces and moments and as a result of this continues (hopefully) towards the reference state.

The navigation system again consists of both a hardware and software component: sensors and a state estimator algorithm, respectively. Sensors are used to measure a number of quantities (e.g., dynamic pressure) that the state estimator algorithm can use to compute the vehicle's current state. The

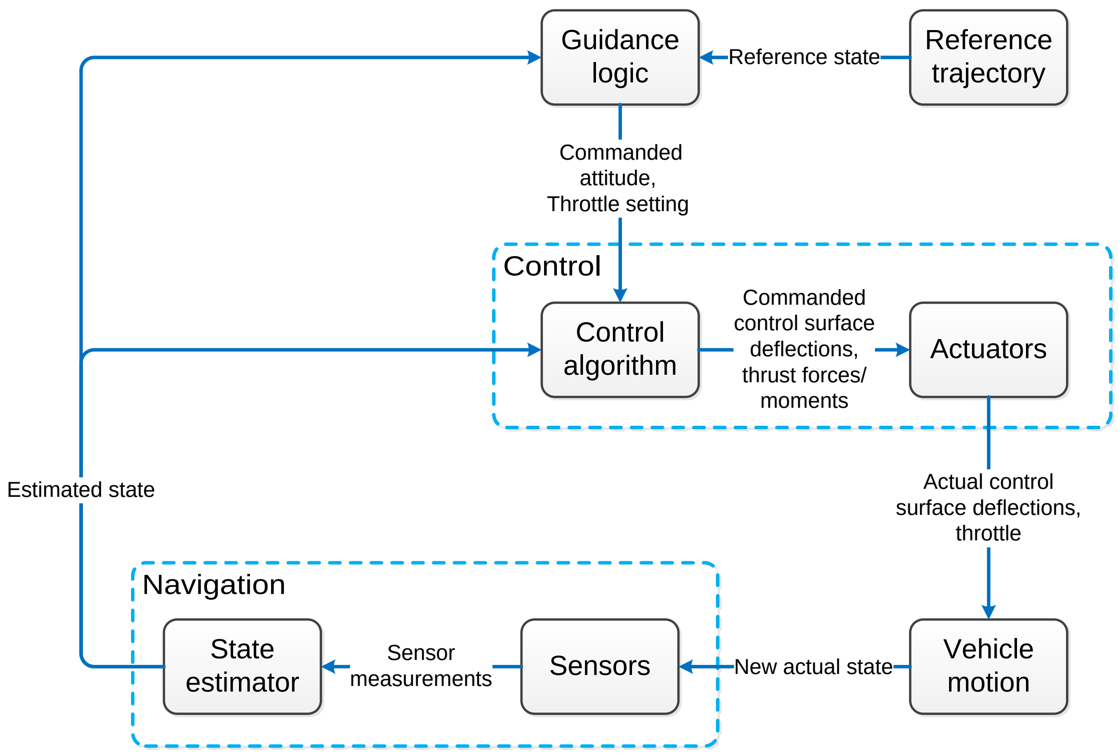


Figure 5.1: Schematic GNC architecture.

translation of sensor measurement data to information about the vehicle's state relies on theoretical and/or empirical models.

The GNC system is susceptible to errors at every interface:

- Computation of guidance commands (guidance algorithm)
- Translation of guidance commands to control commands (control algorithm)
- Performing the control commands (actuators)
- Sensor measurements (sensors)
- Interpretation of sensor measurements (state estimator)

These errors have the risk of propagating throughout the trajectory, as the GNC system is feedback-based. The extent to which it can mitigate these errors is called the robustness of the system.

In this thesis work, these errors are not modeled, i.e., the GNC system is idealized. Per module, this means:

Guidance The guidance algorithm computes a guidance command in the form of the attitude angles α_C and σ_C (the subscript C indicating "commanded") that accurately reflects the vehicle's required response in its current state with respect to the guidance profile. This command is relayed without error to the control module.

Control The control algorithm translates α_C and σ_C to the corresponding necessary control surface deflections without error. The actuators perform the control commands perfectly, i.e., α_C and σ_C as specified by the guidance module is achieved.

Navigation The sensors perform perfect measurements of the parameters required for the computation of the estimated state. The state estimator interprets these measurements without error and thus provides the guidance module with the vehicle's actual current state.

The schematic of this idealized GNC system as applied in this work is shown in Figure 5.2. The control

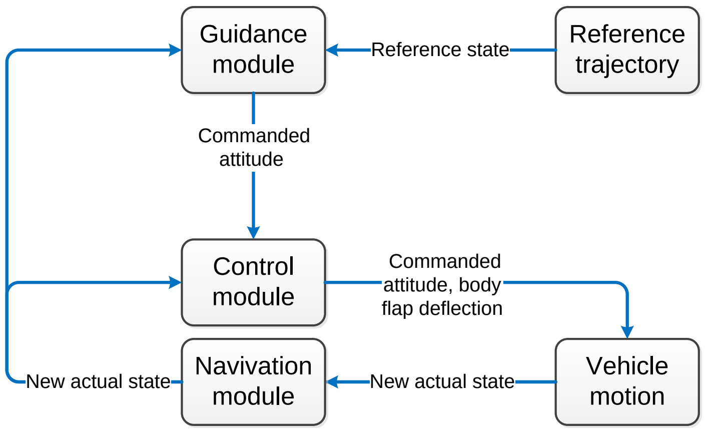


Figure 5.2: Schematic idealized GNC architecture.

system issuing guidance commands does not adhere to the definition previously given; however it is shown as such in the Figure to indicate the one-on-one correspondence of the control module's

The goal of this thesis work was to design a design-time mission planner for a winged (reference) vehicle that is capable of developing optimal re-entry trajectories, wherein optimal is defined depending on user requirements. The preliminary aspect of the planner indicates that these optimal trajectories will be developed beforehand, e.g., to assess requirements feasibility, and not continuously during the actual re-entry itself. This has a large impact on the extent of the detail to which individual systems are modeled. The design choices regarding the entirety of the simulator are discussed in Section 6.1. In this section, the discussion will be limited to the GNC system; specifically, the navigation system is considered to be ideal and is not modeled, and will therefore not be discussed. The guidance and control module is of special importance due to its direct link with the (global) planning process. It was designed in such a way that it would seamlessly integrate with an – at the time – undefined algorithm, or at the very least would require little adjustment. The specific architecture of the module is discussed in the next chapter.

5.2 Guidance System

5.2.1 Guidance Variables

A re-entry vehicle's trajectory can be controlled by changing the magnitude and direction of the resultant forces and moments acting on the vehicle (Brunner and Lu, 2010). In practice – for unpowered vehicles – this comes down to altering the aerodynamic forces and moments incurred by the vehicle, as the gravitational force is not really subject to change. Aerodynamic forces and moments are discussed in detail in Section 3.3.1. The orientation of the individual force components – and therefore the magnitude and direction of the aerodynamic force vector $\mathbf{F_A} = [-D, -S, -L]^T$ – is defined in terms of the vehicle's aerodynamic characteristics, and the aerodynamic attitude angles: the angle of attack α, the angle of sideslip β and the bank angle σ. The aerodynamic properties of the HORUS-2B reference vehicle model used in this thesis are discussed in Section 3.4.

The simplifications applied to the (aerodynamic) vehicle model as well as the flight dynamic model are directly relevant in defining the guidance variables. The vehicle is assumed to produce no propulsive force, so thrust moments cannot be induced. The sideslip angle β is considered to be a disturbance and is set to zero; it only functions as a guidance variable in the sense that it maintains this value throughout re-entry. In practice, this leads to the guidance variables α and σ. Re-entry guidance has a long history of modulating α and σ exclusively to define a vehicle's trajectory. In fact, even for complex vehicles such as the Space Shuttle the guidance algorithm relied on α and σ as guidance variables, with the addition of a lateral logic module for the purpose of minimizing the crossrange error (NASA, 1980).

The angle of attack α has a direct effect on the vehicle's trajectory profile (Harpold and Graves, 1979). In terms of the aerodynamic force, α defines the direction of the drag force D – and thus also the lift force L – acting on the vehicle and together with the Mach number determines the magnitude of the aerodynamic coefficients C_D, C_L and C_m, as well as the pitch moment increment ΔC_{mb} induced by the body flap (Cox, 1973). The bank angle σ defines the orientation of the lift force L with respect to the vehicle's velocity vector, and thus has a prominent effect on the re-entry range (Bairstow, 2006). The magnitude of σ determines the total re-entry range, and its direction (i.e., positive or negative value of σ) affects the heading angle χ. The angle of attack α as a guidance variable has the advantage that the (smaller) effect on the vehicle's course as a result of modulation occurs faster than for bank angle changes, which affect the course relatively slowly (Harpold and Graves, 1979).

5.2.2 Constraints

The guidance variables are subject to constraints; these are partly dependent on the vehicle, and partly design choices. It was already mentioned that the angle of sideslip β is considered null. The remaining aerodynamic angle constraints for the HORUS-2B reference vehicle are:

$$\begin{aligned} 0^\circ \leq \alpha \leq 45^\circ \\ -90^\circ \leq \sigma \leq 90^\circ \end{aligned} \tag{5.1}$$

In the following, only one-sided lateral motion will be considered to limit the optimization search space; this will not make a difference in terms of aerodynamic loading and will only affect the vehicle's crossrange. Therefore the bank angle limits are redefined as

$$0^\circ \leq \sigma \leq 90^\circ \tag{5.2}$$

5.3 Control System

The control system is tasked with translating the guidance commands issued by the guidance logic to direct commands to the vehicle's actuators.

5.3.1 Control Variables

In practice, the HORUS-2B's control surfaces are used to produce the aerodynamic moments to obtain the commanded attitude, and the control system would command the necessary deflection. However as was explained in Section 3.4, the flight dynamics of HORUS are modeled as a "3+1-DOF" system, wherein only translational motion with an additional trim condition is taken into account. As a result, roll and yaw motion are not considered in the context of flight dynamics and rudder and elevon deflections are not modeled. Pitch motion is only incorporated for the purpose of nullifying the pitch moment to keep the vehicle aerodynamically trimmed. Therefore the rotational equations of motion are not incorporated into the flight dynamic model (as explained in Section 3.5.2).

This leaves the control system in charge of determining the body flap deflection angle $\delta_{b,trim}$ that results in an equal and opposite contribution ΔC_{mb} to the pitch moment C_m currently experienced by the vehicle as a result of aerodynamic forces. The deflection angle $\delta_{b,trim}$ is calculated according to the method described in Section 3.5.2; interpolation is performed using a bilinear interpolator (see Section 4.2.1). The control surface deflections required in order to achieve the guided attitude $(\alpha, \sigma)_g$ are not calculated by the control system. Instead, the control system is assumed to be ideal in the sense that it perfectly couples $(\alpha, \sigma)_g$ to the vehicle's actuators. The attitude angles (α, σ) can therefore also be considered control variables. This distinction is only semantic in nature, as they are the exact same as the attitude angles commanded by the guidance system. They will be referred to as guidance variables for the remainder of this report. The final control variable is the body flap deflection angle $\delta_{b,trim}$.

5.3.2 Constraints

The only additional control variable produced by the control system is the body flap deflection angle δ_b. Its limits are given by

$$-20^\circ \leq \delta_b \leq 30^\circ \tag{5.3}$$

The limits to the attitude angles α and σ have already been defined in the previous section.

5.3.3 Node Control

Control of the HORUS re-entry vehicle's trajectory is achieved using node control (Mooij and Hänninen (2009), Dijkstra (2012), Dijkstra et al. (2013)). A node is a point along the vehicle's trajectory where guidance commands are specified; its position is given in terms of an independent variable. At a node $\mathcal{N}$, a number of guidance variable values are defined; generally, these are the commanded attitude angles:

$$\mathcal{N}_i = \{v_i,\ \alpha_{C,i},\ \sigma_{C,i}\} \quad \text{with } i = 1, \ldots, n \tag{5.4}$$

where n is the number of nodes, v_i is the independent variable value at node i, and $\alpha_{C,i}$ and $\sigma_{C,i}$ are the commanded angle of attack and bank angle at node i, respectively. The guidance matrix $\mathbf{\Gamma}$ containing the locations and commands at these nodes defines the entirety of the trajectory guidance profile:

$$\mathbf{\Gamma} = \begin{bmatrix} \mathcal{N}_1 \\ \mathcal{N}_2 \\ \vdots \\ \mathcal{N}_{n-1} \\ \mathcal{N}_n \end{bmatrix} = \begin{bmatrix} v_1 & \alpha_{C,1} & \sigma_{C,1} \\ v_2 & \alpha_{C,2} & \sigma_{C,2} \\ \vdots & \vdots & \vdots \\ v_{n-1} & \alpha_{C,n-1} & \sigma_{C,n-1} \\ v_n & \alpha_{C,n} & \sigma_{C,n} \end{bmatrix} \tag{5.5}$$

Throughout the trajectory, attitude commands are issued based on some algorithm that relates the vehicle's state in terms of the independent variable to the defined attitude commands at the control nodes. A convenient approach is one wherein the guidance system computes attitude commands by interpolating the values of the two surrounding nodes with respect to the current independent variable value.

Independent variable

The guidance logic couples the guidance variables to a measurable independent quantity. Using time may seem like a logical solution initially, but in guidance context it is actually quite a meaningless quantity, e.g., "200 seconds after entry" can mean virtually anything and gives no practical information about the situation. More importantly, optimization of the guidance profile would then require information about the re-entry duration corresponding to a specific profile before it is simulated. This problem can be avoided by defining the guidance profile in terms of an independent quantity of which the initial and final values are already known for any re-entry trajectory – the vehicle's total specific energy E, i.e., the sum of the vehicle's potential and kinetic energy per unit mass:

$$E = gh + \tfrac{1}{2}V^2 \tag{5.6}$$

E is a meaningful quantity as it relates to both the vehicle's position and velocity, as well as having the added benefit of being monotonically decreasing over the vehicle's trajectory, allowing for an unambiguous definition of the guidance profile. The magnitude of E at entry is easily computed from the vehicle's state, and can simply be considered zero at the end of the trajectory – corresponding to a standstill on the Earth surface. In this thesis work, the values of E that define the guidance profile are normalized with respect to the value of E at entry – i.e., its highest value – mapping the node locations to the interval of the normalized total specific energy $\hat{E} \in [0, 1]$:

$$\hat{E} = \frac{E}{E_{\text{entry}}} \tag{5.7}$$

NUIN

The positions of the individual nodes can be determined using various techniques; a number of methods were investigated by Dijkstra (2012) in the context of atmospheric re-entry. His conclusion was that non-uniform independent node control (NUIN), on top of being easiest to implement, also performed the best in establishing the guidance logic. In this method, the positions of the nodes and the corresponding control angles are determined at random, and two nodes have fixed positions at the beginning and end of the trajectory, respectively. NUIN has the advantage of being relatively computationally "cheap" when combined with an optimization process. The conclusions drawn by Dijkstra (2012) led to the choice of using NUIN as the node definition method in this work. A small difference between this work and the approach used by Dijkstra (2012) is that in addition to fixing the positions of the initial and final control nodes, their values are fixed as well. The reasons for this are:

- It greatly decreases the dimension of the optimization problem: for a guidance profile defined by n nodes, the number of variables to optimize becomes $3(n-2)$ instead of $3n$.
- The vehicle's attitude at $\hat{E}=0$ – i.e., at standstill – can reasonably be assumed to be neutral, with $\alpha_c = \sigma_c = 0°$.
- The vehicle's initial attitude at $\hat{E}=1$ can reasonably be assumed to be predefined along with the initial state due to the general specifications of the mission. Were this not the case, the vehicle's actual attitude at entry (assuming it is reasonable) is not defining for the rest of the trajectory per se, as the influence of aerodynamic loads is much less prominent at such high altitudes.

Guidance profile

The guidance matrix $\mathbf{\Gamma}$ in this thesis work thus has the general form:

$$\mathbf{\Gamma} = \begin{bmatrix} \mathcal{N}_s \\ \mathcal{N}_1 \\ \vdots \\ \mathcal{N}_{n-2} \\ \mathcal{N}_f \end{bmatrix} = \begin{bmatrix} \hat{E}_s & \alpha_{C,s} & \sigma_{C,s} \\ \hat{E}_1 & \alpha_{C,1} & \sigma_{C,1} \\ \vdots & \vdots & \vdots \\ \hat{E}_{n-2} & \alpha_{C,n-2} & \sigma_{C,n-2} \\ \hat{E}_f & \alpha_{C,f} & \sigma_{C,f} \end{bmatrix} \tag{5.8}$$

Nodes $\mathcal{N}_i$ with $i = 1, ..., n-2$ represent the mutable control nodes in a profile defined by a total of n nodes. The locations and values of the control nodes $\mathcal{N}_s$ and $\mathcal{N}_f$ are fixed:

$$\mathcal{N}_s = \{\hat{E}_s,\ \alpha_{C,s},\ \sigma_{C,s}\} = \{1,\ \alpha_0,\ \sigma_0\} \tag{5.9}$$

$$\mathcal{N}_f = \{\hat{E}_f,\ \alpha_{C,f},\ \sigma_{C,f}\} = \{0,\ 0°,\ 0°\} \tag{5.10}$$

where α_0 and σ_0 represent the angle of attack and bank angle of the vehicle at entry, respectively. Two additional constraints are imposed on the interval wherein the mutable control nodes $\mathcal{N}_i$ may be placed:

$$\hat{E}_{\min} = 0.1 \leq \hat{E}_i \leq \hat{E}_{\max} = 0.98 \tag{5.11}$$

The values of $\hat{E}_{\min} = 0.1$ and $\hat{E}_{\max} = 0.98$ are based on the consideration of not allowing nodes to be located too close to the limit values of $\hat{E}$. The upper limit of 0.98 was chosen to prevent guidance commands from being issued too early on in the trajectory, where they have little to no effect, whereas the lower limit of 0.1 is taken to avoid control nodes occurring after the simulation cuts off at an altitude of 20 km. At such low altitudes, a different guidance logic is needed to steer the vehicle to its final landing point. Between $\hat{E}=1$ and $\hat{E}=0.98$, the vehicle's initial attitude is maintained.

Figure 5.3 shows the generalized definition of the guidance profile in terms of control nodes as used in this thesis. The figure should be read from left-to-right, as $\hat{E}=1$ corresponds to the entry point and $\hat{E}=0$ to standstill. Inter-node guidance commands are determined by interpolating between the two nodes currently "surrounding" the vehicle. This interpolation is performed using Hermite splines (see Section 4.2.2), which results in a guidance profile wherein no discontinuities in either magnitude or first derivative occur. While the values of the attitude angles α_C and σ_C *at* a specific control nodes are constrained according to the vehicle's capabilities, their interpolated values *between* control nodes may still exceed the defined limits. Therefore the guidance algorithm operates on the basis that if the interpolated value

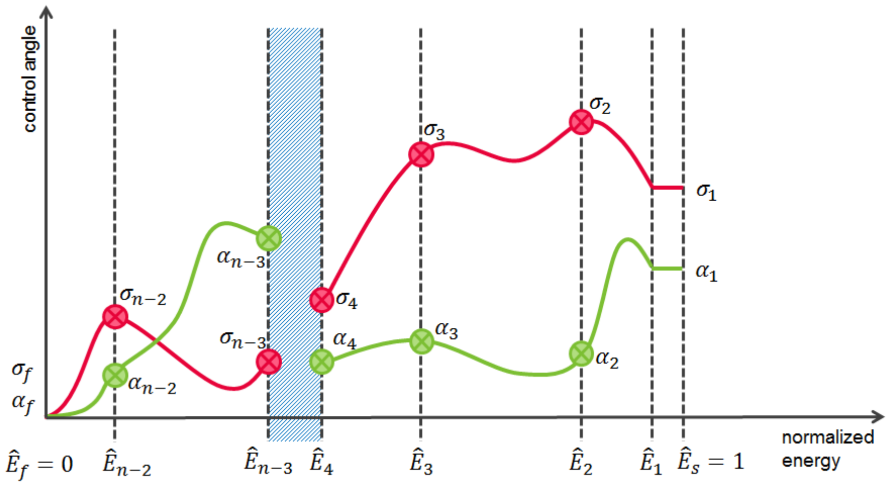


Figure 5.3: Geometry of the guidance profile as defined by nodes.

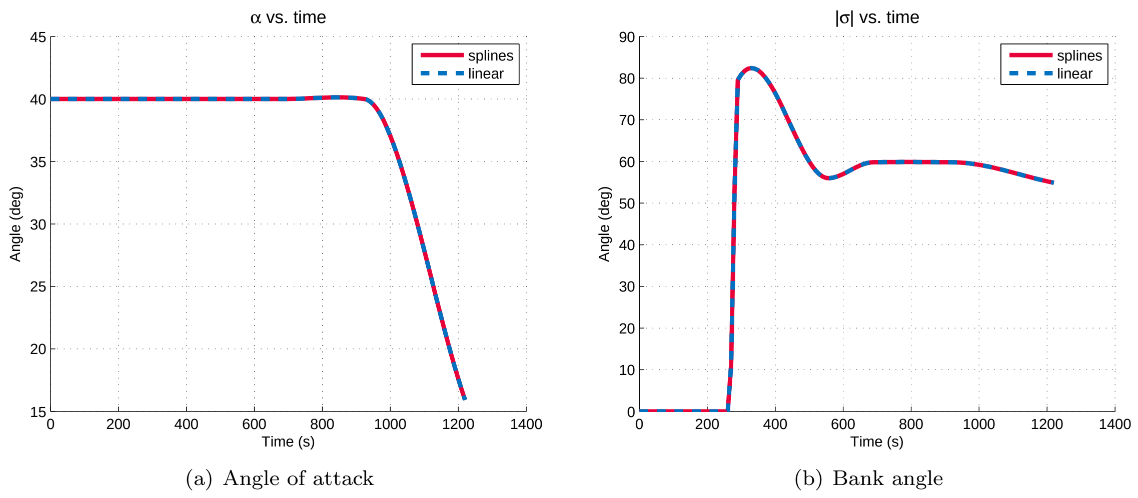


Figure 5.4: Nominal HORUS guidance profile vs. cubic spline interpolated guidance profile

of an attitude angle would exceed its lower or upper boundary, it returns a commanded attitude angle equal to this limit value instead.

Nominal reference profile

The nominal HORUS reference guidance profile as defined in Mooij (1998) is specified by seven time-dependent nodes. Its guidance matrix $\mathbf{\Gamma}_{\text{HORUS, nom}}$ is given by:

$$\mathbf{\Gamma}_{\text{HORUS, nom}} = \begin{bmatrix} t_1 & \alpha_1 & \sigma_1 \\ t_2 & \alpha_2 & \sigma_2 \\ t_3 & \alpha_3 & \sigma_3 \\ t_4 & \alpha_4 & \sigma_4 \\ t_5 & \alpha_5 & \sigma_5 \\ t_6 & \alpha_6 & \sigma_6 \\ t_7 & \alpha_7 & \sigma_7 \end{bmatrix} = \begin{bmatrix} 0 & 40^\circ & 0^\circ \\ 264 & 40^\circ & 0^\circ \\ 290 & 40^\circ & 79.6^\circ \\ 554 & 40^\circ & 56.0^\circ \\ 686 & 40^\circ & 59.8^\circ \\ 924 & 40^\circ & 59.8^\circ \\ 1319 & 11.5^\circ & 54^\circ \end{bmatrix} \tag{5.12}$$

The nominal reference profile is computed using linear interpolation; the difference with Hermite spline interpolation of the same profile is shown in Figure 5.4.

Chapter 6

Re-Entry Simulator Design

The foundation of a preliminary re-entry mission planner is its capability of simulating a vehicle's trajectory through the atmosphere given a certain set of circumstances. The specific design of the simulator depends on the requirements of the mission planner in terms of e.g., accuracy and speed. These requirements define the extent of the detail included in individual models and the complexity of the numerical methods used. The design choices made in the development of the re-entry simulator are discussed in Section 6.1. The state-space model relating to the problem is presented in Section 6.2. The software architecture of the simulator in its final form is shown in Section 6.3. Finally, the reliability of the simulator is demonstrated through verification results in Section 6.4.

6.1 Design Choices

The goal of this thesis work was to develop a design-time mission planner for a winged re-entry vehicle, with the purpose of designing an optimal trajectory. Optimal in this context first and foremost means maximizing range while minimizing the heat load, in addition to keeping the vehicle within its design limits. Due to the amount of factors exhibiting influence on the vehicle – to greater or lesser extent – the solution space of this optimization has a large dimensionality. The independent variables included in the simulation thus need to be limited; this must be done in such a way that the mission planner may still produce sufficiently adequate and reliable results given its specific purpose. In practice, this limitation of independent variables is included in the choices of the individual models used.

Throughout this text, a number of design choices with respect to the simulator were mentioned. They are summarized here for clarity.

Environment The gravitational field is central with an added $J2$-term. The atmospheric model used is the US76. The Earth shape is modeled as a spheroid.

Flight dynamics There is no wind. The vehicle produces no propulsive force and has a constant mass. The sideslip angle β is considered a disturbance and is set to zero. As a result, the side force S acting on the vehicle becomes null. The flight dynamic model is "three-and-a-half-DOF" as it accounts for the trim condition in addition to the translational motion; rotational motion is neglected.

Vehicle The Horus-2B reference vehicle is used. Its aerodynamic data was obtained from Mooij (1995). Only the body-flap deflection is taken into account; all aerodynamic moment coefficients not relating to the pitch moment are neglected, as well as the pitch moment contributions of other control-surfaces.

6.2 State Space Model

Reducing the re-entry problem to its essence: during its course through the atmosphere the vehicle is subject to forces defined by its environment, which in turn define its motion. To simulate the vehicle's trajectory, the equations of motion are integrated numerically. For this purpose, they need to be written in the form of a set of coupled first-order differential equations (see Section 4.1), i.e., as a state-space model:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t), t) \qquad \text{with} \quad \mathbf{x}(0) = \mathbf{x}_0 \tag{6.1}$$

Here, $\mathbf{x}$ is the state vector and $\mathbf{u}$ is the input (or control) vector. The dynamic and kinematic equations for the translational motion of the vehicle with respect to the I-frame were defined in Equation 3.12 as:

$$\begin{aligned} \frac{d\mathbf{V_I}}{dt} &= \tfrac{1}{m}\mathbf{F_I} \\ \frac{d\mathbf{r_I}}{dt} &= \mathbf{V_I} \end{aligned}$$

Rewritten in state-space form this gives

$$\mathbf{x} = [\mathbf{V_I}, \mathbf{r_I}]^T = [\dot{x}_I, \dot{y}_I, \dot{z}_I, x_I, y_I, z_I]^T \tag{6.2a}$$

$$\dot{\mathbf{x}} = [\mathbf{a_I}, \mathbf{V_I}]^T = [\ddot{x}_I, \ddot{y}_I, \ddot{z}_I, \dot{x}_I, \dot{y}_I, \dot{z}_I]^T \tag{6.2b}$$

$$\mathbf{u} = \mathbf{F_I} \tag{6.2c}$$

with

$$\mathbf{a_I} = [\ddot{x}_I, \ddot{y}_I, \ddot{z}_I]^T = \left[\frac{F_{x,I}}{m}, \frac{F_{y,I}}{m}, \frac{F_{z,I}}{m}\right]^T \tag{6.3}$$

As was discussed in Section 3.5.2, the dynamic model of the vehicle is a "three-and-a-half-DOF" configuration wherein the trim condition is taken into account in addition to the translational motion. The body-flap deflection angle is modulated to nullify the total pitch moment and other aerodynamic moments are neglected, meaning the equations of rotational motion need not be represented in the state-space.

6.3 Software Architecture

The simulator is developed in $C++$. Architecturally, it is divided into a number of code blocks that individually contribute to the calculation of the state-space. The vehicle is defined at a certain time by its state, attitude and control-surface (i.e, body-flap) deflection. The state is propagated using a numerical integration scheme, whereas the attitude and control-surface deflection are a consequence of the guidance algorithm.

The vehicle position serves as input to the environment module, which determines the instantaneous environment of the vehicle. The temperature T serves as input for the guidance algorithm (where it will be used in conjunction with the vehicle's velocity V to determine the Mach number M), and the gravitational acceleration $\mathbf{g}$ and density ρ are required to calculate the external forces acting on the vehicle at that time. The environment module makes use of an analytical model in conjunction with externally defined tabulated values characterizing the US76 atmosphere to obtain $\mathbf{g}$, ρ and T. The vehicle's attitude angles α, β and σ are used to determine the aerodynamic forces in conjunction with the aerodynamic force coefficients as given by the vehicle model for a certain α and δ_b; these are calculated using external aerodynamic property tables. Finally, α is implemented in the guidance algorithm module to calculate the new body-flap deflection angle δ_b required to maintain trim. Following the guidance module, the aerodynamic angles as well as the body-flap deflection settings of the vehicle are updated to their new commanded values. After calculating the external forces in the correct reference frame, the state derivative is calculated and the state is propagated using a RKF45 integration method.

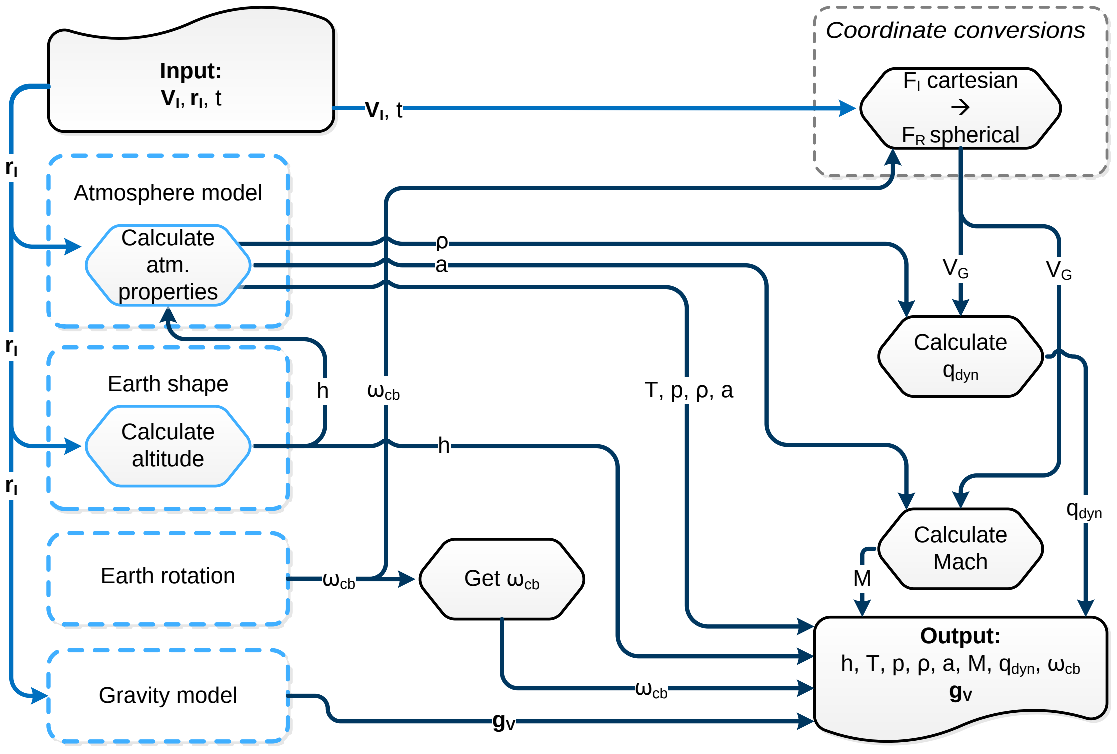


Figure 6.1: Flowchart of the environment module.

6.3.1 Environment

A schematic overview of the environment module is shown in Figure 6.1. The environment class is defined by four attributes: the atmosphere model, the Earth shape, the Earth rotational speed and the gravitational field. The model used for each attribute is specified at initialization of the environment class. Three of these attributes are defined as virtual base classes consisting of a number of derived classes which represent the available models. Environment attribute objects are instantiated upon initialization of the environment, where the requested models are specified. The environment class can be considered a "container" of sorts for these properties; any future call to a method or member of an environmental attribute by an external class occurs through the environment class and not directly.

Atmosphere model The atmosphere model is a virtual base class wherein three derived classes are defined; each derived class represents a different atmospheric model. The atmospheric properties temperature T, pressure p, density ρ, and speed of sound a are computed with the altitude h as input according to the atmospheric model assigned to the environment object:

- Analytical US76: see Section 2.2.2.
- Tabulated US76: cubic spline interpolation (Section 4.2.2) is used to interpolate between tabulated values of T, p, ρ, and a with h as the independent variable.
- Exponential: p and ρ vary exponentially with altitude ($p = p_0 \exp(-\frac{h}{h_0})$ and $\rho = \rho_0 \exp(-\frac{h}{h_0})$), T is obtained via the ideal gas law given in Equation 2.13 and a follows from Equation 2.28.

The preferred atmosphere model is the analytical US76, which is used to calculate atmospheric properties at a given altitude given tabulated values of the defining properties per atmospheric layer (see Section 2.2). However the simulator is also equipped to calculate atmospheric properties from a tabulated US76 atmosphere using cubic spline interpolation (Section 4.2.2) or a simple exponential atmosphere model. The exponential atmosphere model is however too inaccurate for practical use and is only included for verification purposes.

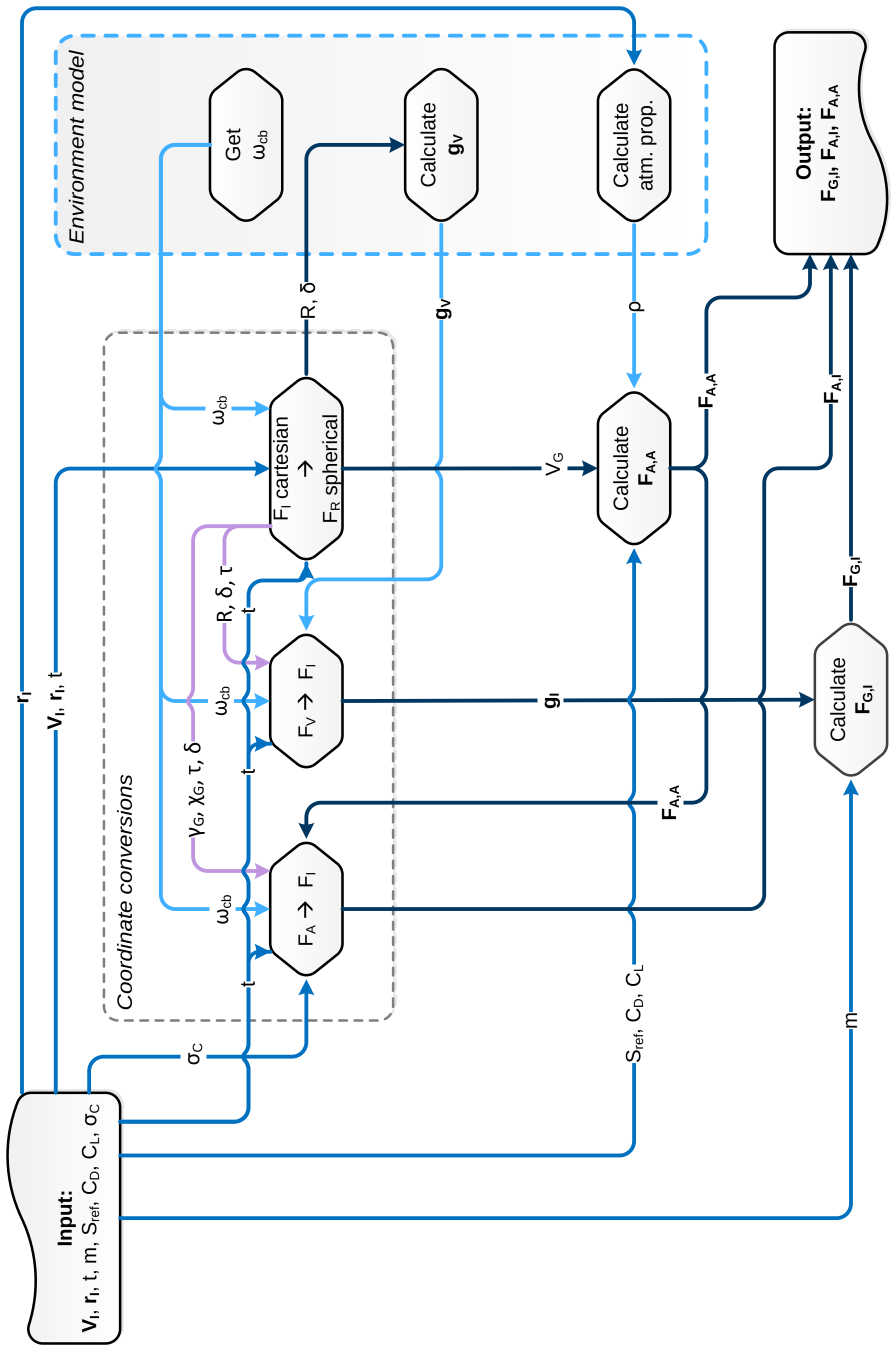


Figure 6.2: Flowchart of the external forces module.

Earth shape The Earth shape is a virtual base class consisting of a derived class representing a spheroid Earth. It contains a single method tasked with returning the vehicle's altitude h above the surface given its position vector $\mathbf{r_I}$; for a spherical Earth the equation is simply $h = R - R_E$.

Earth rotation The Earth rotation is an environmental property that is defined as either rotating or non-rotating at initialization of the environment class; the value of the rotational rate ω_{cb} is returned by the environment class.

Gravitational model The gravitational model is a virtual base class consisting of derived classes representing the central field model and a spherical harmonic model including the J_2-term; it consists of a single method that returns the gravitational acceleration vector in the V-frame $\mathbf{g_V}$ given the vehicle's radial distance R and latitude δ. The computation is performed using Equation 2.12 for the spherical harmonic model and is simply $\mathbf{g_V} = [0, 0, \mu/R^2]^T$ for the central field model.

The environment class contains methods to compute the dynamic pressure $\bar{q}$ and Mach number M given the vehicle's state $\mathbf{x} = [\mathbf{V_I}, \mathbf{r_I}]^T$. The dynamic pressure is obtained by using $\mathbf{r_I}$ to obtain ρ from the atmospheric model and $\mathbf{x}$ and t to compute the groundspeed by converting $[\mathbf{V_I}, \mathbf{r_I}]^T$ to spherical coordinates; $\bar{q}$ follows from Equation 3.2. The Mach number is computed using Equation 3.1 with this same groundspeed and the local speed of sound a obtained from the atmospheric model given an input of $\mathbf{r_I}$.

6.3.2 External Forces

A schematic overview of the external forces class is shown in Figure 6.2. This class is initialized with a pointer to an existing environment object. The computation of the external forces relies on the environmental properties local gravitational acceleration $\mathbf{g_V}$ and local air density ρ to compute the gravitational force vector $\mathbf{F_{G,I}}$ and aerodynamic force vector $\mathbf{F_{A,A}}$, respectively. In addition, coordinate frame transformations involving the I-frame require information about the Earth's rotation, which is also a property of the environment object.

The flowchart may seem complex at first but that is mainly due to the large number of coordinate frame transformations that are needed; essentially it only consists of two methods (and a half): the calculations of $\mathbf{F_{G,I}}$ and $\mathbf{F_{A,A}}$, and the conversion of $\mathbf{F_{A,A}}$ to $\mathbf{F_{A,I}}$. The gravitational force $\mathbf{F_{G,I}}$ is computed by transforming the input position vector $\mathbf{r_I}$ to spherical R-coordinates to obtain the vehicle's radial distance R and latitude δ which are necessary for the computation of $\mathbf{g_V}$. A transformation to the I frame is performed to obtain $\mathbf{g_I}$, the multiplication of which with the vehicle's mass m gives $\mathbf{F_{G,I}}$. The aerodynamic force vector $\mathbf{F_{A,A}}$ is computed by transforming the input velocity vector $V_\mathbf{I}$ to spherical R-coordinates to obtain the groundspeed V_G. After also computing the local atmospheric density ρ with $\mathbf{r_I}$ as input and given aerodynamic properties of the vehicle (coefficients C_D and C_L as well as the reference area S_{ref}), $\mathbf{F_{A,A}}$ is computed according to Equation 3.3. A transformation to the I-frame is performed to obtain the final output variable $\mathbf{F_{A,I}}$.

6.3.3 Vehicle Model

The vehicle model class is basically a container of information about the vehicle properties. At initialization, tabulated values of the vehicle's aerodynamic coefficients as a function of angle of attack α, Mach number M, and – depending on the coefficient – also the body-flap deflection δ_b are imported from external data files and stored in matrices. Vectors containing the values of the independent variables α, M and δ_b at which the tabulated aerodynamic coefficients occur are also imported, and the vehicle's mass m and aerodynamic reference area S_{ref} are defined.

The class computes the aerodynamic coefficients (C_D, C_S, C_L) of Horus given an input of α, M and δ_b:

$$C_D = C_{D0}(\alpha, M) + \Delta C_{Db}(\alpha, M, \delta_b) \tag{6.4}$$

$$C_S = 0 \tag{6.5}$$

$$C_D = C_{L0}(\alpha, M) + \Delta C_{Lb}(\alpha, M, \delta_b) \tag{6.6}$$

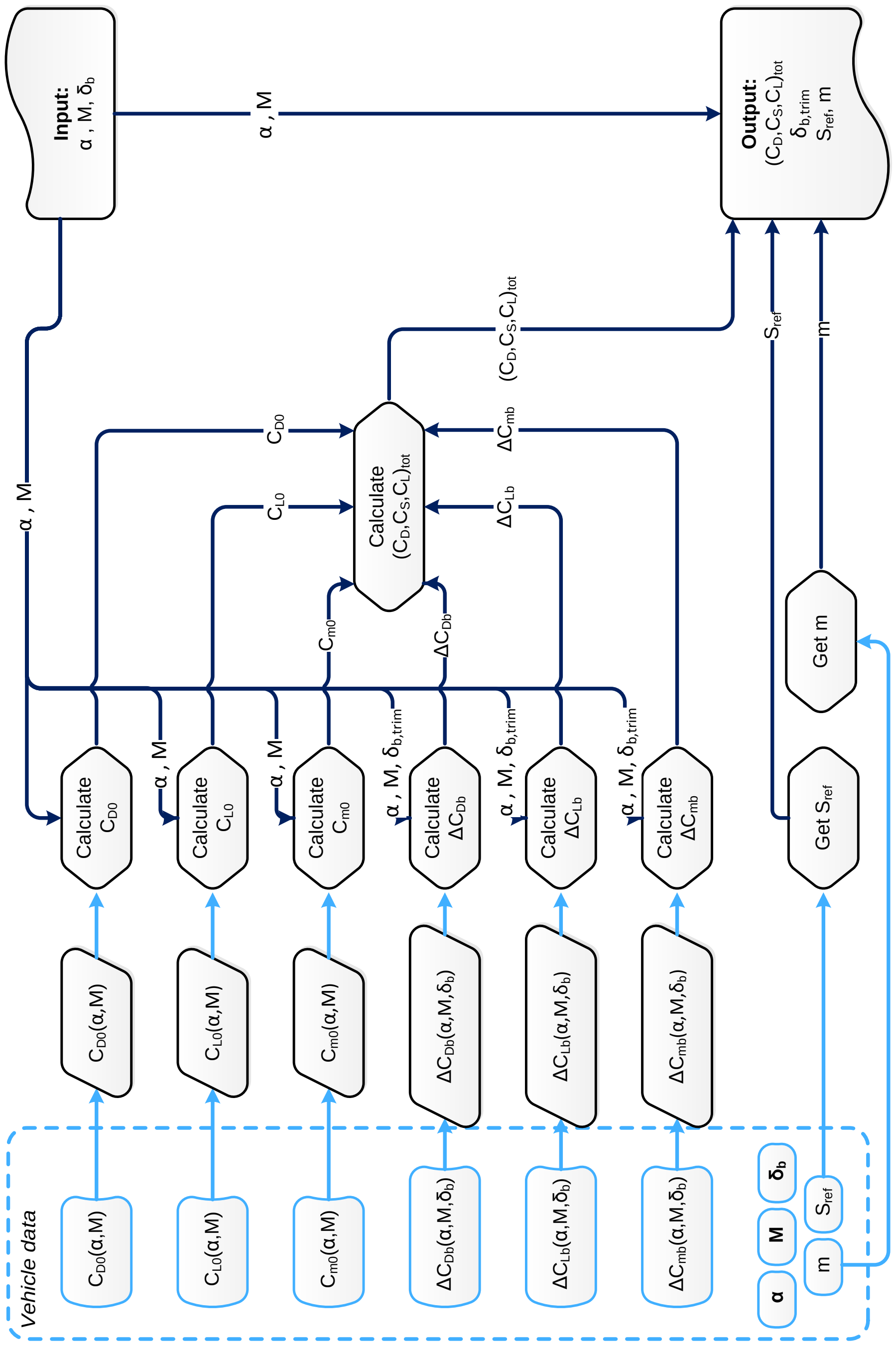


Figure 6.3: Flowchart of the vehicle model module.

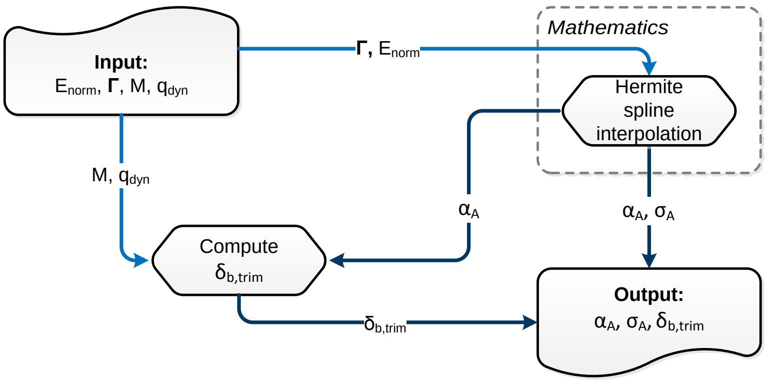


Figure 6.4: Flowchart of the guidance module.

Base coefficients The base coefficients C_{D0} and C_{L0} are both functions of α and M. Their data matrices are given in Tables B.2 and B.3, respectively. The values corresponding to a given input are computed using bilinear interpolation as described in Section 4.2.1.

Increments The body-flap causes an increment in the lift and drag coefficients: ΔC_{Db} and ΔC_{Lb}, respectively. In addition to being dependent on the values of α and M, the magnitude of the increments is also a function of the deflection angle δ_b; the tables relating ΔC_{Db} and ΔC_{Lb} to α and M for an array of body-flap deflection angles are given in Tables B.5-B.10 and Tables B.11-B.16, respectively. Because ΔC_{Db} and ΔC_{Lb} are dependent on three variables, two separate interpolations on the data must be performed. First, a bilinear interpolation with the given values of α and M is performed on all the data matrices, each representing a particular body-flap deflection angle δ_{bi} with $i = 1, \ldots, n$. An n-length array of the corresponding $(\Delta C_{Db})_i$ and $(\Delta C_{Lb})_i$-values is generated; these arrays are interpolated linearly with independent variable δ_b to obtain the increment values for the given α, M, and δ_b.

Finally, the class also returns the vehicle's mass m and aerodynamic reference area S_{ref}.

6.3.4 Guidance/Control Module

A schematic overview of the guidance and control module is shown in Figure 6.4. The guidance and control class is initialized with a vehicle model object that contains all vehicle aerodynamic properties and methods to compute the aerodynamic coefficients and the body-flap deflection angle δ_b. The guidance and control module is given a prespecified $(n-2) \times 3$ guidance matrix $\mathbf{\Gamma} = [\hat{\mathbf{E}}, \boldsymbol{\alpha}, \boldsymbol{\sigma}]$:

$$\mathbf{\Gamma} = \begin{bmatrix} \hat{E}_1 & \alpha_0 & \sigma_0 \\ \hat{E}_2 & \alpha_1 & \sigma_1 \\ \vdots & \vdots & \vdots \\ \hat{E}_{n-1} & \alpha_{n-1} & \sigma_{n-1} \end{bmatrix} \tag{6.7}$$

where the guidance variables α_C and σ_C are defined at each node position $\hat{E}$. This matrix is used by the guidance module to calculate the guidance and control commands at any instance given an input of the normalized total specific energy $\hat{E}$. The computation of the attitude angles α_C and σ_C is therefore fairly straightforward. Given the input $\hat{E}$ and the guidance matrix $\mathbf{\Gamma}$, a Hermite spline interpolation is performed on both $\boldsymbol{\alpha}$ and $\boldsymbol{\sigma}$ with the independent data vector $\hat{\mathbf{E}}$. To this purpose, the Hermite-spline interpolation method from the mathematics module is used; this method is based on the procedure outlined in Section 4.2.2.

The class also contains a method to compute the body-flap deflection angle required for trim. To achieve trim stability, the body-flap most produce a moment increment equal in magnitude but in opposite direction to the base pitch moment coefficient $C_{m0}(\alpha, M)$. Therefore $\delta_{b,trim}$ is the body-flap deflection angle for which $\Delta C_{mb} = -C_{m0}$. First, given α and M, C_{m0} is computed by performing bilinear interpolation of Table B.4. The increments $\Delta C_{mb}(\alpha, M, \delta_b)$ are defined in the same manner as ΔC_{Db} and ΔC_{Lb}; see Tables B.17-B.22. Therefore an n-length array of $(\Delta C_{mb})_i$-values is constructed, wherein each $(\Delta C_{mb})_i$ corresponds to a specific body-flap deflection angle δ_{bi}. This array is then interpolated linearly with $-C_{m0}$ as the independent variable to obtain $\delta_{b,trim}$. Trimmed flight is not performed at very high altitudes; until the dynamic pressure $\bar{q}$ experienced by the vehicle reaches 100 N/m^2, δ_b is constant at 15° (Mooij, 1998). In addition, once δ_b passes the threshold of 20° during the course of the trajectory, this angle of deflection is maintained until the end of the trajectory as the vehicle becomes untrimmable.

6.3.5 Entry Vehicle

The re-entry vehicle at any given time is defined by its state vector $\mathbf{x}$, attitude vector $\boldsymbol{\psi}$ and body-flap deflection angle δ_b:

$$\mathbf{x} = [\mathbf{V_I}, \mathbf{r_I}]^T \tag{6.8a}$$

$$\boldsymbol{\psi} = [\alpha_A, \beta_A, \sigma_A]^T \tag{6.8b}$$

$$\delta_b = f(\alpha, M) \tag{6.8c}$$

These characteristics are stored in the Entry Vehicle class and are continuously updated with each integration step (the state $\mathbf{x}$) or guidance command $(\boldsymbol{\psi}, \delta_b)$ (the attitude and body-flap deflection angle).

At initialization, the entry vehicle class is given pointers to existing environment, external forces, guidance algorithm, and vehicle model objects, along with an initial time, state and attitude and the guidance matrix $\mathbf{\Gamma}$ as given by Equation 6.7. It is essentially the class that contains all the information and references to values and methods that the simulator requires to successfully simulate the trajectory. At the center lies the method that computes the vehicle's state derivative given the current time and vehicle state. This state derivative is returned to the simulator where it will be integrated using an RKF45-algorithm (see Section 4.1.2). In the next iteration, this new state along with the updated time serves as input for the state derivative computation.

State derivatives are calculated in this method not only for $\mathbf{x} = [\mathbf{V_I}, \mathbf{r_I}]^T$, but also a number of other parameters whose integral value will become relevant during the optimization process. Therefore a distinction is made between the vehicle state $\mathbf{x}$ and the "appended state" $\mathbf{z}$:

$$\mathbf{z} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ x \\ y \\ z \\ Q \\ \int n_g \\ X_{dr} \end{bmatrix}, \qquad \dot{\mathbf{z}} = \begin{bmatrix} a_{I,x} \\ a_{I,y} \\ a_{I,z} \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ q_c \\ n_g \\ V_G \cos\gamma_G \end{bmatrix} \tag{6.9}$$

where Q is the total heat supplied to the vehicle per unit area and X_{dr} represents the vehicle's downrange. The g-load n_g and heat flux density q_c are found using Equations 2.1 and 2.2, respectively.

The method is given an input of the vehicle's current appended state $\mathbf{z}$ and time t, and returns the state derivative $\dot{\mathbf{z}}$. A flowchart of the method's architecture is shown in Figure 6.5. Throughout the simulation, the vehicle's state vector and the current time are stored as class variables in the entry vehicle class. When the state derivative computation method is called, these variables are updated with the input $\mathbf{x_I}$ and t. Before the components of the state derivative vector $\dot{\mathbf{x}}$ can be computed, the guidance algorithm determines the guidance commands. The vehicle's attitude $\boldsymbol{\psi}$ and body-flap deflection angle δ_b are also class variables, and their values are updated after each call to the guidance algorithm. Knowing the vehicle's new attitude, the aerodynamic force vector $\mathbf{F_{A,I}}$ and gravitational force vector

$\mathbf{F}_{\mathbf{G},\mathbf{I}}$ are computed to form the acceleration vector $\mathbf{a}_\mathbf{I} = (\mathbf{F}_{\mathbf{A},\mathbf{I}} + \mathbf{F}_{\mathbf{G},\mathbf{I}})/m$. The components of $\mathbf{a}_\mathbf{I}$ are the derivative values of $\dot{\mathbf{r}}_\mathbf{I}$.

6.3.6 Problem Properties

The problem properties class is a storage container that gets passed to the simulator containing all necessary initialization inputs; it is required to make the integration of the simulator with the optimization process more convenient. It is initialized with[1]:

- Initial time t_0
- Integration stepsize Δt
- End time t_{end}
- Initial state $(\mathbf{r}_\mathbf{I})_0$
- Initial attitude $\boldsymbol{\psi}_0$
- Initial body-flap deflection $\delta_{b,0}$
- Earth rotation: yes/no
- Type of gravitational field model
- Type of Earth shape model
- Type of atmosphere model

Variables are stored as class variables, and the following objects are initialized:

- Environment
- External forces
- Entry vehicle model
- Guidance algorithm

Pointers to these objects are also stored as class variables.

6.3.7 Simulation

The simulation class is basically an implementation of the state derivative computation; the flowchart shown in Figure 6.5 gives a good general overview of the global simulator architecture without unnecessary details.

The simulation class is initialized with a problem properties object. The method performing the simulation takes an additional input of the guidance vector $\boldsymbol{\Gamma}$ to initialize the entry vehicle. Initial values of all relevant output variables are computed, and the appended state $\mathbf{z}$ is integrated using RKF45. The output variables should be stored at equal intervals; however, RKF45 is variable-step. To solve this, an extra loop is implemented that performs a complete variable-stepsize integration between each interval and stores the final output only, resulting in evenly spaced output data. A flowchart of this integration process can be seen in Figure 6.6. The output of the simulator method is a data file containing the following variables of interest between t_0 and t_{end} at intervals of Δt:

$$\left[t, x_I, y_I, z_I, \dot{x}_I, \dot{y}_I, \dot{z}_I, h, \tau, \delta, V, \gamma, \chi, \alpha_\mathrm{C}, \sigma_\mathrm{C}, n_g, q_c, \delta_{b,\mathrm{trim}}, M, \hat{E}, \textstyle\int n_g, Q, X, \bar{q}\right]$$

6.4 Simulator Verification

To conclude that the simulator performs correctly, it is subjected to a verification process, wherein proof is collected to demonstrate the simulator's reliability. Verification is not a linear procedure, as errors will undoubtedly be discovered and code will need to be rewritten. Therefore, it is tantamount to not only test whether the software works according to plan when it is completely assembled; finding individual errors

[1]More initialization variables are required but their relevancy only becomes apparent with regard to the optimization process.

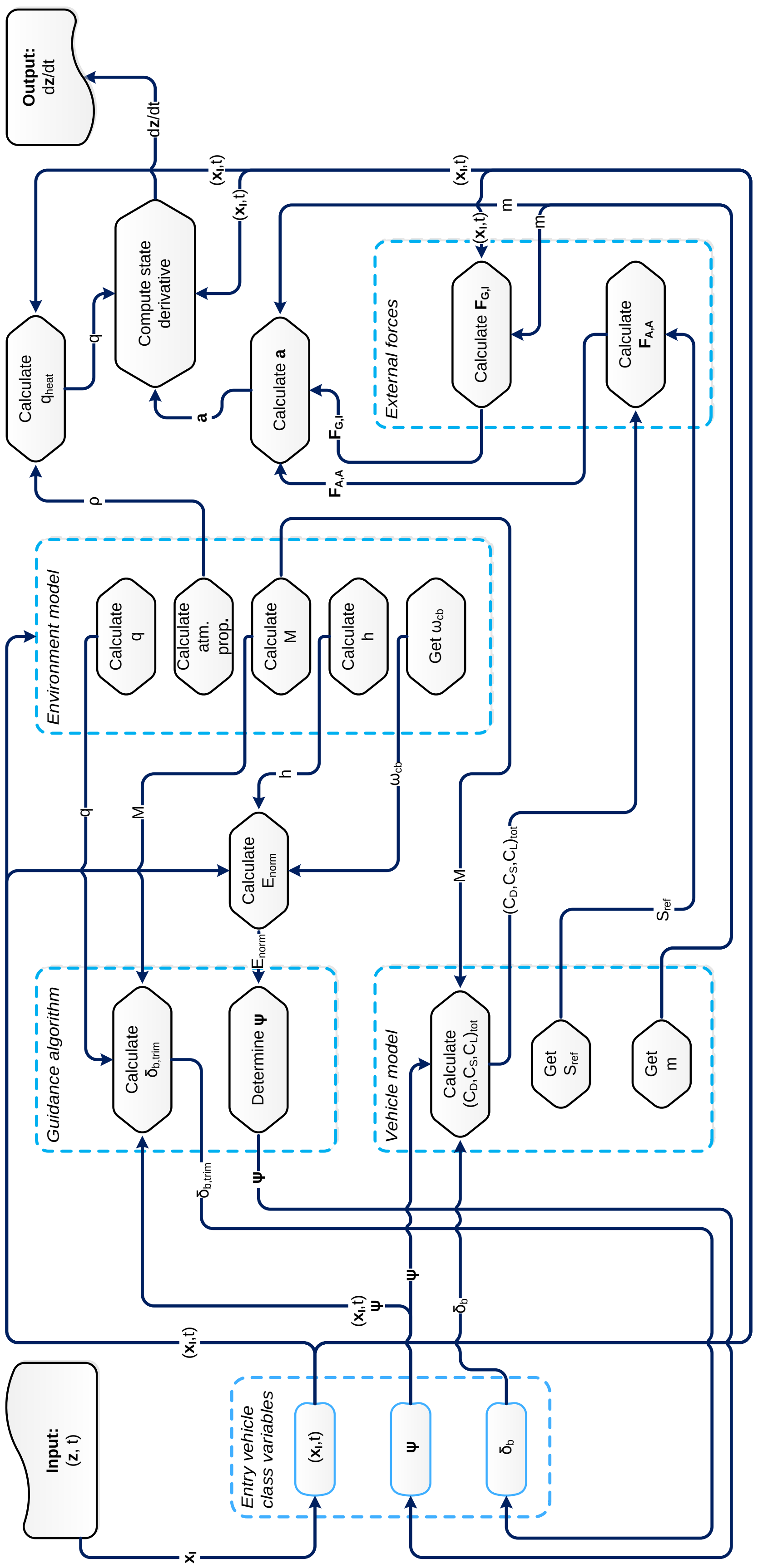


Figure 6.5: Flowchart of the state derivative computation.

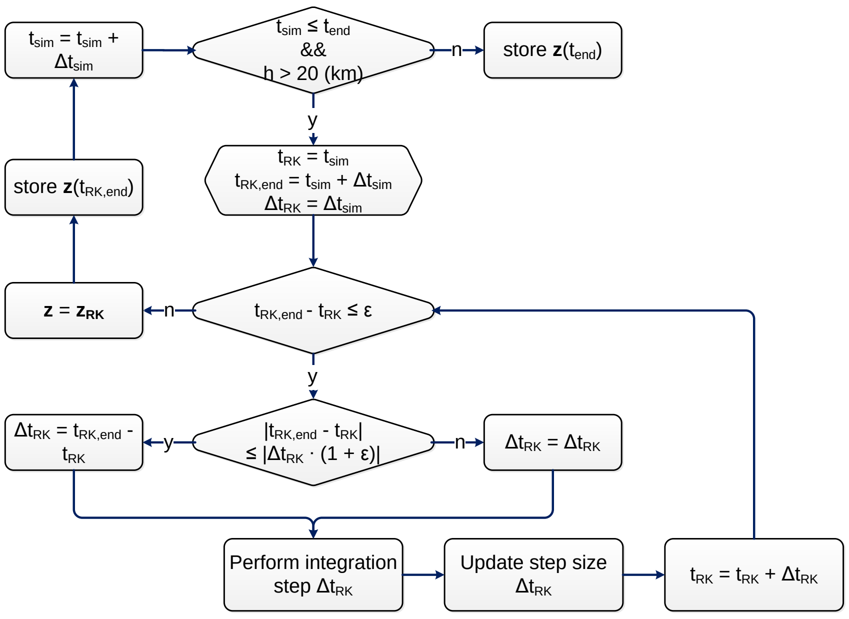

$t_{sim} = t_{sim} + \Delta t_{sim}$
$t_{sim} \leq t_{end}$ && h > 20 (km)
n
store $\mathbf{z}(t_{end})$
y
$t_{RK} = t_{sim}$
$t_{RK,end} = t_{sim} + \Delta t_{sim}$
$\Delta t_{RK} = \Delta t_{sim}$
store $\mathbf{z}(t_{RK,end})$
$\mathbf{z} = \mathbf{z_{RK}}$
n
$t_{RK,end} - t_{RK} \leq \varepsilon$
y
$\Delta t_{RK} = t_{RK,end} - t_{RK}$
y
$|t_{RK,end} - t_{RK}| \leq |\Delta t_{RK} \cdot (1 + \varepsilon)|$
n
$\Delta t_{RK} = \Delta t_{RK}$
Perform integration step Δt_{RK}
Update step size Δt_{RK}
$t_{RK} = t_{RK} + \Delta t_{RK}$

Figure 6.6: Flowchart of the integration algorithm.

on such a scale is nearly impossible. The process is therefore divided into two segments: unit verification and system-level verification. It is stressed that these two segments are not performed sequentially; often the best-hidden errors come to light during software verification, after which the module containing the error must be fixed – and verified again to ascertain that no new errors have been included.

6.4.1 Module Verification

The Horus re-entry simulator can be divided into a number of separate modules; most of these were discussed in Section 6.3:

- Mathematics
  - Interpolators: results compared to manual computations, as well as computations performed built-in Matlab interpolators.
- Reference Frames
  - Transformation matrices: matrices for single transformations were created with easy inputs (e.g., $\omega_{cb}t = 90°$) of which the results could be manually checked. For multiple transformations, equivalent matrix multiplications were performed in Matlab and compared.
  - Reference frame conversions: Sanity checks using sketches by hand. Matrix multiplications were performed in Matlab.
  - State vector conversions: Sanity checks using sketches by hand. As these computations apply the already-verified reference frame conversions no further verification was necessary.
- Utilities
  - Find minimum/ maximum: applied to sort arrays and compared to arrays sorted in Matlab.
  - Calculate mean: compared to computation in Matlab.
  - Sort ascending/descending: application of verified minimum/ maximum determination; compared to arrays sorted in Matlab.
- Environment
  - Atmosphere models
    * Analytical US76 and tabulated US76: a table of atmospheric properties at an array of altitudes was computed and compared to the tables provided by NASA (1980); these were

found to match.

- Gravity models
  - Central field with J_2: the gravitational acceleration vector was computed in both the V-frame (Equation 2.12) and I-frame (Equation 3.7) and frame conversions were applied both ways. In addition, the gravitational acceleration calculation from the Tudat software repository was compared to the values obtained for $\mathbf{g_I}$.
- Earth shape models
  - Ellipsoid Earth: sanity check with known properties of the Earth.

- External Forces
  - Force computations: sanity check.
- Guidance/Control
  - Determining commanded attitude: straightforward after verification of interpolation methods.
- Vehicle Model
  - Computation of aerodynamic coefficients: reproduction of Tables B.2 - B.22
  - Body flap deflection for trim computation: reverse-reproduction of Tables B.17 - B.22 (i.e, computation of known values of δ_b given $\Delta C_{m,b}$, α and M).
- Entry Vehicle
  - Calculation q_c: sanity check.
  - Calculation n_g: sanity check.
  - State derivative computation: integration was performed on a problem defined by a definite integral (i.e., sphere of 1 kg falling towards the Earth) and compared. In addition, the state derivative computation using simplified circumstances was implemented in Matlab using the spherical translational equations of motion given in Equations 3.19 and 3.21 and integrated using the built-in ode45 integrator.

6.4.2 Simulator Verification

Once all modules were deemed verified (rightfully or unrightfully so – in which case possible errors would become apparent during global verification), the simulator itself was tested (and tested...). The first step in this testing process was already mentioned in the previous section, namely the implementation of the spherical translational equations of motion given in Equations 3.19 and 3.21.

The simplified model applies the following simplifications to the Horus re-entry simulator:

- The vehicle model has constant aerodynamic coefficients (C_D, C_S, C_L).
- The vehicle remains at a constant attitude.
- The trim condition is not taken into account.
- No guidance is applied; the vehicle's motion is only subject to its environment.
- The environment is defined by an exponential atmosphere, a central gravity field and a rotating, spherical Earth.

Trajectories were simulated using the simplified model for the nominal entry condition by integrating the spherical equations of motion using the built-in ode45 solver (the RKF45 equivalent) in Matlab. The simplified model was then used in the Horus re-entry simulator with the same initial conditions and integrator tolerances. The trajectories were found to match, leading to the conclusions that the integrator could be considered verified, and that the basic parts of the simulator seemed to be working soundly.

Finally, the simulator in its totality needed to be subjected to some testing to prove its validity. Lacking real-life validation data, output data from a more sophisticated re-entry simulator developed by E. Mooij in FORTRAN was used as reference. As this simulator used a very different GNC-system, the attitude angles found in the reference output data were implemented directly in the Horus re-entry simulator by expanding the guidance algorithm base class with a time-dependent guidance class. This methodology of this class is equivalent to that of $\hat{E}$-dependent one detailed in Section 6.3.4 with the exception of the independent variable being t. The guidance vector $\mathbf{\Gamma}$ consists of the attitude angles found in the reference data at the designated times. Using time-dependent guidance in this way has the added benefit of automatically allowing for a second verification of the $\hat{E}$-dependent guidance algorithm.

The reference data was obtained using the following specifications:

- US76
- Spherical, rotating Earth
- Central gravity field with J_2
- Initial state:

$$h = 122 \text{ (km)}, \quad \tau = -106.7^\circ, \quad \delta = -22.3^\circ, \quad V_{G,0} = 7.4355 \text{ (km/s)}, \quad \gamma_0 = -1.43^\circ, \quad \chi = 70.75^\circ$$

- Initial attitude:

$$\alpha_{A,0} = 40^\circ, \quad \beta_{A,0} = 0^\circ, \quad \sigma_{A,0} = 0^\circ$$

The Horus re-entry simulator was run with the same specifications, barring a few differences:

- Rotational motion of the vehicle was not taken into account; this has no effect as the vehicle's attitude throughout the trajectory is copied from the reference data and is applied to the vehicle using time-dependent attitude guidance.
- The elevator deflection δ_e was not taken into account; this has minimal effect as the elevator is only used at the very end of the reference trajectory after δ_b has reached its maximum and trim fight is not possible.
- The altitude-dependent drag increment $\Delta C_{D,h}$ was neglected during computation of the aerodynamic coefficients; this has no effect as $\Delta C_{D,h}$ becomes relevant under an altitude of 20 (km).

The results are shown in Figures 6.7 through 6.12.

The conclusion can be drawn that the Horus re-entry simulator functions correctly. The differences between the verification and simulation data are so small that it can be faithfully said that the simulator is verified. The largest discrepancy between the verification and simulation data can be seen in Figure 6.11 representing the g-load versus time. This difference is still well within acceptable bounds, and can be attributed to the simulation being the solution to an initial value problem: very small errors at the beginning tend to add up and cause larger deviations. This conclusion is reinforced by deviations only becoming apparent later on in the trajectory. This deviation could be eliminated by applying very mindor changes to α and σ. Due to the deviations being minimal and explainable, the simulator can be considered verified.

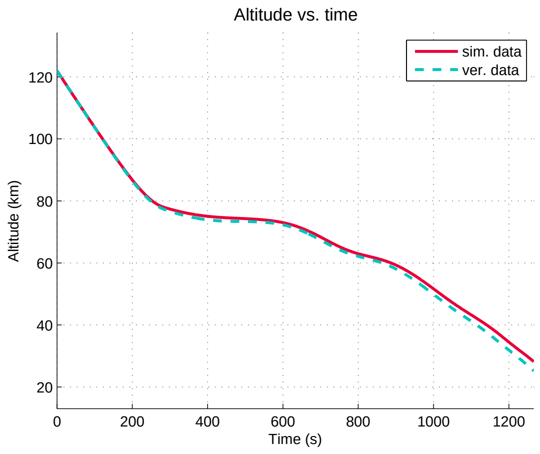


Figure 6.7: Altitude vs. time

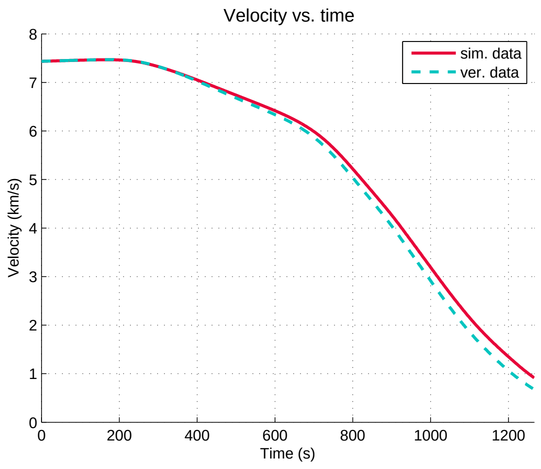


Figure 6.8: Velocity vs. time

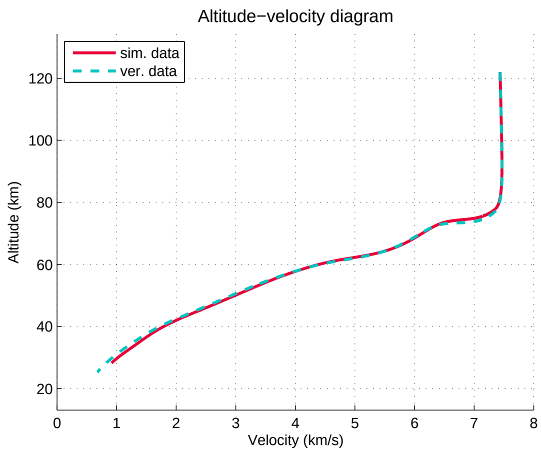


Figure 6.9: h-V diagram

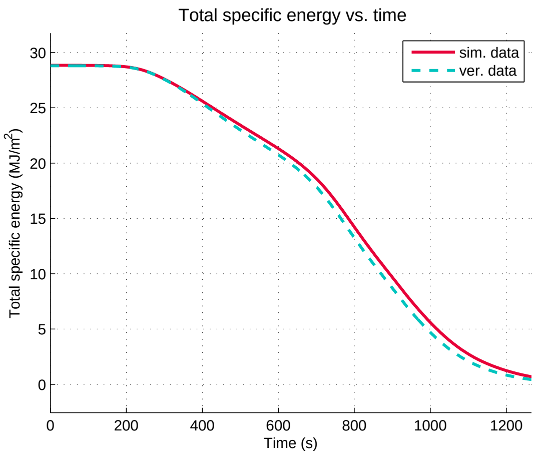


Figure 6.10: Total specific energy vs. time

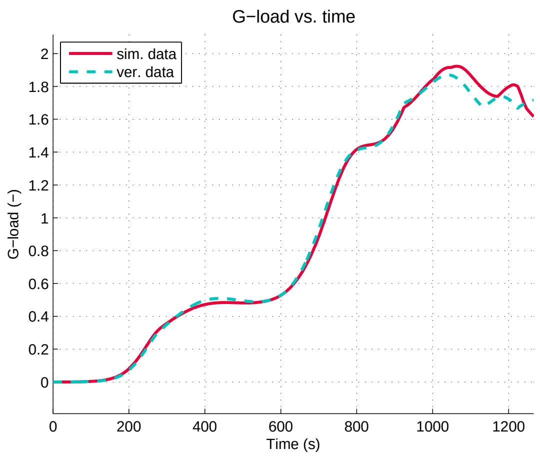


Figure 6.11: g-load vs. time

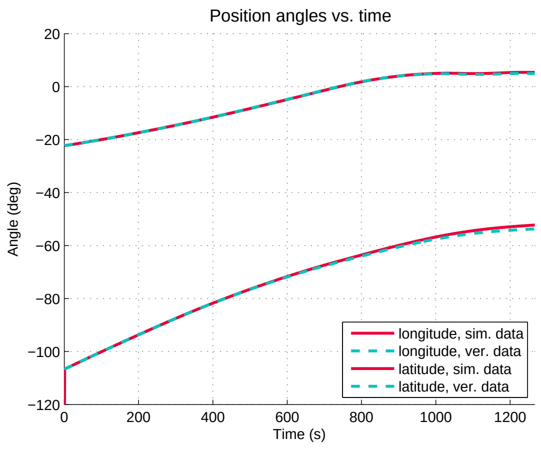


Figure 6.12: Position angles vs. time

Chapter 7

Trajectory Optimization

The complete approach to developing an optimal re-entry trajectory in terms of the mission objectives and constraints is discussed in this chapter. Section 7.1 gives an overview of the methodology applied in multiobjective optimization; Section7.1.1 gives a general explanation of multiobjective optimization, Section 7.1.2 provides an explanation of the Multiobjective Evolutionary Algorithm based on Decomposition (MOEA/D) multiobjective optimization algorithm, and Section 7.1.3 details external software used in the development of the mission planner. Section 7.2 discusses the optimization problem as it relates to the mission planner in terms of its decision variables (Section 7.2.1)), objectives (Section 7.2.2), constraints (Section 7.2.3) and the objective function (Section 7.2.4); Section 7.2.5 then gives an indication of the shape of the problem's objective space.

7.1 Methodology

7.1.1 Multiobjective Optimization

In simple terms, an optimization problem entails finding the best solution from all feasible solutions; what defines "best" is based on the optimization criteria.

7.1.1.1 Definition

Multiobjective optimization is defined as the problem of finding a set of decision variables in the form of a decision vector $\mathbf{x}$ that optimizes the objective function $\mathbf{f}(\mathbf{x})$ (Coello, 1999). A Multiobjective Optimization Problem (MOP) generally has a global optimum and a multitude of local optima; optimization methods concerned with finding global optima as opposed to local optima are called global optimization methods. Mathematically, a MOP is stated as follows:

$$\begin{aligned} \text{find: } & \mathbf{x} \in \Omega \\ \text{to minimize: } & \mathbf{f}(\mathbf{x}) \\ \text{subject to: } & \mathbf{g}(\mathbf{x}) \geq \mathbf{c_{ineq}} \\ & \mathbf{h}(\mathbf{x}) = \mathbf{c_{eq}} \end{aligned} \tag{7.1}$$

or:

$$\min\{\mathbf{f}(\mathbf{x}) | \mathbf{x} \in \Omega\} \tag{7.2}$$

$$\text{with: } \mathbf{g}(\mathbf{x}) \geq \mathbf{c_{ineq}} \tag{7.3}$$

$$\mathbf{h}(\mathbf{x}) = \mathbf{c_{eq}} \tag{7.4}$$

where $\mathbf{f}(\mathbf{x})$ is a set of objective functions $f_i(\mathbf{x})$ with $i = 1, \ldots, m$. The elements $f_i(\mathbf{x})$ define the performance criteria of a point $\mathbf{x}$ in mathematical terms. Ω represents the decision (or solution) space of the MOP, which is the set of all possible points $\mathbf{x}$ of the problem that satisfy the problem's constraints. The objective space $F \in \mathbb{R}^m$ is the set of objective function values corresponding to Ω, i.e., $\mathbf{f} : \Omega \to \mathbb{R}^m$ where m is the number of objectives of the problem. The attainable objective set is defined as the set $\{\mathbf{f}(\mathbf{x}) | \mathbf{x} \in \Omega\}$, i.e., the objective function values corresponding to points $\mathbf{x}$ within the solution space. In most practical problems there is no single solution possible that simultaneously optimizes every objective $f_i(\mathbf{x})$, meaning the objectives are in some degree of conflict with each other; such problems are called nontrivial. In a nontrivial problem, the concept of Pareto optimality is used to define tradeoff among the objectives (Zhang and Li, 2007).

7.1.1.2 Pareto optimality

Given two solutions $\mathbf{u} \in \Omega$ and $\mathbf{v} \in \Omega$ with objective function values $\mathbf{f}(\mathbf{u}) \in F$ and $\mathbf{f}(\mathbf{v}) \in F$, $\mathbf{u}$ is said to dominate $\mathbf{v}$ if and only if $f_i(\mathbf{u}) \leq f_i(\mathbf{v})$ for every $i \in 1, \ldots, m$ and $f_j(\mathbf{u}) < f_j(\mathbf{v})$ for at least one index $j \in 1, \ldots, m$, i.e., the decision vector $\mathbf{u}$ has better objective function values than $\mathbf{v}$ with respect to at least one objective, and no worse values with respect to all the other objectives. In other words, any improvement in a Pareto optimal point in one objective must lead to deterioration in at least one other objective (Coello, 1999).

A solution is considered Pareto optimal if there exists no other solution in the solution set that it is dominated by (Fonseca and Fleming, 1995). The set of all Pareto optimal points is called the Pareto set and the set of all the Pareto optimal objective vectors is the Pareto front (Zhang and Li, 2007). Figure 7.1 illustrates these concepts for a two-dimensional problem, and indicates the best (dominating) and worst (dominated) solutions with respect to the Pareto front; indifferent solutions are neither dominating nor dominated.

Very often, since the objectives $f_i(\mathbf{x})$ contradict each other, no point in Ω maximizes all of them simultaneously, and a tradeoff has to be performed based on Pareto optimality. For many practical problems, an approximation of the Pareto front is needed by a human decision maker choose a final solution. Most MOPs have many or even infinite Pareto optimal vectors, making it (nearly) impossible to obtain a complete Pareto front. It is more workable to determine a much smaller number of Pareto optimal vectors, which are evenly distributed along the Pareto front to get a good idea of the shape of the Pareto front without drowning in unnecessary data; this is the objective of many multiobjective optimization algorithms.

7.1.1.3 Evolutionary algorithms

A branch of global optimization algorithms especially capable of solving MOPs is that of evolutionary algorithms. These algorithms are capable of searching for multiple solutions in parallel – which allows for a Pareto set to be determined after a single run – as opposed to performing the separate runs in series (Zitzler and Thiele, 1998). In addition, relatively little knowledge about the problem being solved is necessary for successful implementation (Zhang and Li, 2007).

An Evolutionary Algorithm (EA) is based on the biological evolution process: it uses natural selection as the search engine to solve problems (Abraham et al., 2005). The terminology is defined accordingly. A population is a set of individuals (or candidate solutions) that all carry a unique defining set of chromosomes. These individuals are evolved over a number of generations to obtain an optimal solution. The chromosomes determine the fitness of the individual and thus its probability of reproducing. During reproduction, the chromosomal properties of two individuals are combined to produce a new individual, allowing for chromosomes resulting in good fitness values to be passed on to the next generation (Dijkstra, 2012). The general schematic of an evolutionary algorithm is shown in Figure 7.2.

7.1.1.4 Discussion

The merits of Multiobjective Optimization Algorithms (MOEA) are discussed in the previous section. These algorithms are generally Pareto dominance-based; such algorithms generally approximate the

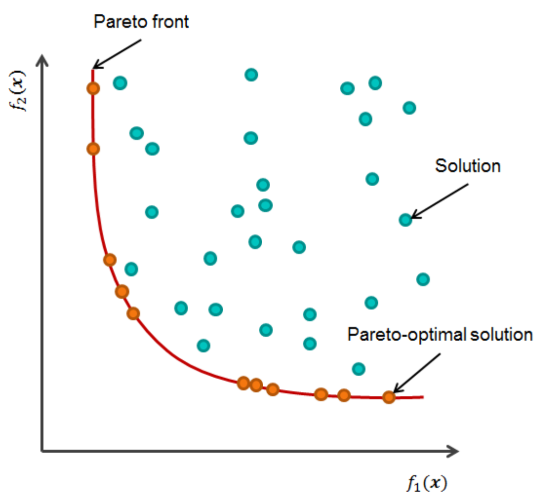


Figure 7.1: Pareto front of a two-dimensional optimization problem.

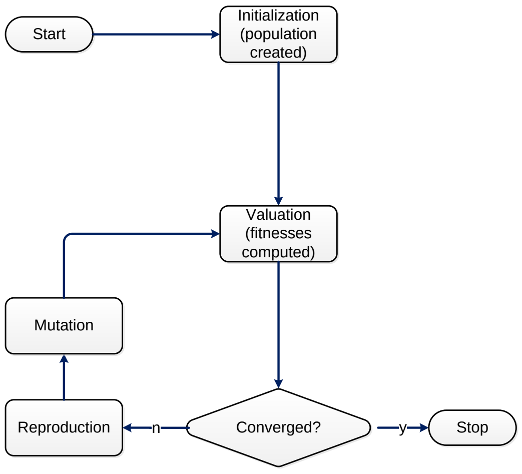

Start
Initialization (population created)
Valuation (fitnesses computed)
Mutation
Reproduction
n
Converged?
y
Stop

Figure 7.2: Flowchart of a general evolutionary algorithm.

Pareto front well when the MOP has two or three objectives, but degrade in performance as the number of objectives increases, because almost all the solutions are nondominated by each other (Qi et al., 2014). A MOEA not based on Pareto dominance is the MOEA/D, which offers several advantages over Pareto dominance-based algorithms in terms of e.g., computational efficiency (Qi et al., 2014).

In terms of the re-entry problem, MOEA/D offers the parallel computation benefit along with its capacity of handling more than three objectives and is a logical choice. Its performance was compared to Pareto dominance-based algorithms VEGA and NSGA-II from the Parallel Global Multiobjective Optimizer (PaGMO) software toolbox (Biscani, 2014) and decidedly produced the best results.

7.1.2 MOEA/D

MOEA/D is a relatively recently developed algorithm by Zhang and Li (2007). It is based on the decomposition of a MOP into a number of scalar (i.e., single-objective) optimization subproblems and optimizing them simultaneously. This methodology is based on the fact that a Pareto optimal solution to a MOP can be defined as the optimal solution of a scalar optimization problem in which the objective is an aggregation of all the objectives of the MOP (Zhang and Li, 2007). Each subproblem is optimized by only using information from a number of its neighboring subproblems, resulting in a relatively low computational complexity compared to other EAs.

7.1.2.1 Decomposition

The approach of combining objectives into a single function is called aggregating functions (Coello, 1999). Several methods can be found in literature, such as the weighted sum approach, the Tchebycheff approach and boundary intersection; a discussion of their relative merits is beyond the scope of this text. In this work, the Tchebycheff approach was used to construct the aggregation functions as this was the standard approach specified for the MOEA/D algorithm in PaGMO, and therefore the discussion will be limited to its specific methodology.

A weight vector $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_m]$ is defined, of which the components $\lambda_i \geq 0$ add up to 1. In the Tchebycheff approach, a number of scalar optimization problems are solved in the form (Zuiani and Vasile, 2013):

$$\min_{\mathbf{x}\in\Omega}\{g(\mathbf{f}(\mathbf{x})|\boldsymbol{\lambda}, \mathbf{z}^*)\} = \min_{\mathbf{x}\in\Omega}\left(\max_{i=1,\ldots,m}\{\lambda_i|f_i(\mathbf{x}) - z_i^*\}\right) \tag{7.5}$$

where $\mathbf{z}^* = [z_1^*, \ldots, z_m^*]$ is the reference objective vector whose components are given by

$$z_i^* = \min\{f_i(\mathbf{x}) | \mathbf{x} \in \Omega\} \quad \text{with} \quad i = 1, \ldots, m \tag{7.6}$$

For each Pareto optimal point $\mathbf{x}^*$ there exists a weight vector $\boldsymbol{\lambda}$ such that $\mathbf{x}^*$ is the optimal solution of Equation 7.5; per definition, each of these optimal solutions is a Pareto optimal solution of Equation 7.2 as well (Zhang and Li, 2009). By solving a number of problems in this form with different weight vectors, different Pareto optimal solutions to the MOP can be obtained.

7.1.2.2 Algorithm

The MOEA/D algorithm takes a set of evenly spread weight vectors $\boldsymbol{\lambda}^1, \ldots, \boldsymbol{\lambda}^N$ and a reference point $\mathbf{z}^*$, and decomposes the MOP given into N scalar optimization subproblems using the Tchebycheff approach (Zhang and Li, 2007).

The objective function of the j^{th} subproblem is given by

$$g(\mathbf{f}(\mathbf{x}) | \boldsymbol{\lambda}^j, \mathbf{z}^*) = \max_{i=1,\ldots,m} \{\boldsymbol{\lambda}_i^j | f_i(\mathbf{x}) - z_i^*\} \tag{7.7}$$

where $\boldsymbol{\lambda}^j = [\lambda_1^j, \ldots, \lambda_m^j]$. The algorithm minimizes all these N objective functions simultaneously in a single run. If $\boldsymbol{\lambda}^j$ and $\boldsymbol{\lambda}^i$ are close to each other, it follows that the optimal solution of $g(\mathbf{f}(\mathbf{x})|\boldsymbol{\lambda}^j, z^*)$ is close to that of $g(\mathbf{f}(\mathbf{x})|\boldsymbol{\lambda}^i, z^*)$. MOEA/D makes use of this property by using information about the objective functions' weight vectors close to $\boldsymbol{\lambda}^i$ optimize $g(x|\boldsymbol{\lambda}^i, z^*)$ (Zhang and Li, 2007). The set of a number of closest weight vectors to a specific weight vector $\boldsymbol{\lambda}^j$ is called a neighborhood. The neighborhood of the i^{th} subproblem g_i consists of all the subproblems with the weight vectors from the neighborhood of $\boldsymbol{\lambda}^i$. Only the current solutions to its neighboring subproblems are used to optimize a subproblem in MOEA/D.

Every iteration of the algorithm wherein a new population is created is called a generation. The population is composed of the best solution found so far for each subproblem. Within each generation, the following parameters are maintained by the algorithm:

- The current population $\mathbf{P}$ consisting of N points $\{\mathbf{x}^1, \ldots, \mathbf{x}^N | \in \Omega\}$, where $\mathbf{x}^i$ is the current solution to the i^{th} subproblem g_i.
- The fitness vectors $\{\mathbf{F}^1, \ldots, \mathbf{F}^N | \in F\}$ where $\mathbf{F}^i = \mathbf{f}(\mathbf{x}^i)$ for each $i = 1, \ldots, N$.
- The ideal point vector $\mathbf{z} = [z_1, \ldots, z_m]$ containing the best value found so far z_j for each objective f_j with $i = 1, \ldots, m$.
- An external population archive denoted $\mathbf{PA}$ that stores all nondominated solutions found during the search.

The MOEA/D algorithm performs the following steps (Zhang and Li, 2007):

Step 1: Initialization

1.1 An empty population archive **PA** is initialized.

1.2 The T neighboring weight vectors are computed for each $\boldsymbol{\lambda}^i$ with $i = 1, \ldots, N$ by computing their individual Euclidean distances. The set $B(i) = \{i_1, \ldots, i_T\}$ is defined for each $\boldsymbol{\lambda}^i$, containing the indices i of these neighboring weight vectors.

1.3 An initial population $\mathbf{P}$ containing the individuals $\mathbf{x}^i$ with $i = 1, \ldots, N$ is initialized randomly. The fitness vectors $\mathbf{F}^i = \mathbf{f}(\mathbf{x}^i)$ are computed for each individual.

1.4 The ideal point vector $\mathbf{z}$ is computed for the initial population $\mathbf{P}$.

Step 2: Update

For $i = 1, \ldots, N$:

2.1 Two indices k and l are selected from the set $B(i)$, defining individuals $\mathbf{x}^{i,k}$ and $\mathbf{x}^{i,l}$ as mating partners.

2.2 The offspring $\mathbf{y}$ of $\mathbf{x}^{i,k}$ and $\mathbf{x}^{i,l}$ is produced and evaluated using a genetic Differential Evolution (DE) operator.

2.3 The ideal point vector $\mathbf{z}^i$ is updated: for each of its elements $j = 1, \ldots, m$, if $z^i_j < f_j(\mathbf{y})$, then $z_j = f_j(\mathbf{y})$.

2.4 The neighboring solutions are updated: for each index $k \in B(i)$, if $g(\mathbf{f}(\mathbf{y})|\lambda^k, \mathbf{z}) \leq g(\mathbf{f}(\mathbf{x})|\lambda^k, \mathbf{z})$, then $\mathbf{x}^k = \mathbf{y}$ and $\mathbf{F}^j = \mathbf{f}(\mathbf{y})$.

2.5 The population archive **PA** is updated:

- All vectors dominated by $\mathbf{f}(\mathbf{y})$ are removed.
- $\mathbf{f}(\mathbf{y})$ is added if it is not dominated by any of the vectors in **PA**.

Step 3: Stopping criteria

When the stopping criteria are satisfied, the algorithm stops and outputs **PA**. Otherwise, **Step 2** is repeated.

7.1.3 External Software

PaGMO is an open-source software platform developed at European Space Agency (ESA) with the purpose of solving high-dimensional global optimization problems (Biscani et al., 2010) (Izzo, 2014). The platform – at its core a C++ library – contains a large number of global and local optimization algorithms and is designed to be easily extensible due to its object-oriented architecture.

The capability of performing parallel computations is pretty much inherent in every laptop and PC with a multi-core processor, and is highly beneficial for the efficiency of running optimization processes. However, parallelizing algorithms is a difficult and time-consuming process, and often beyond the scope of an average engineer's capabilities and/or interests. PaGMO provides a generalized version of the island model – a coarse-grained approach to the parallelization of genetic algorithms – which is applicable to a multitude of optimization algorithms. This allows for the platform to be used without any inherent knowledge of parallelization (Biscani et al., 2010).

Optimization within the Horus Mission Planner was achieved by integrating the Horus Re-Entry Simulator with the PaGMO toolbox. To do this, the specific methodology of PaGMO base components has to be adhered to.

7.1.3.1 Problem

Any optimization problem implemented in PaGMO needs to derive from the PaGMO problem base class. This class represents a box-bounded, multiobjective, mixed-integer, constrained optimization problem defined by (Biscani, 2014):

- The global dimension n, i.e., the number of dimensions of the global search space.
- The dimension of the integral (or combinatorial) part of the problem ni.
- The lower and upper bounds of the global search space, denoted **lb** and **ub**, respectively.
- The total number of constraints nc.

- The number of inequality constraints *nic*.
- The constraints tolerance (vector) c_{tol}.

The implementation of the re-entry problem in PaGMO in the context of the mission planner is discussed in Section 7.2.

All derived classes of the problem base class must at least contain the objective function computation method, here denoted *objfun*; this method takes as input a decision vector denoted $\mathbf{x}$ and returns a vector of fitnesses, $\mathbf{f}$. The length of the decision vector $\mathbf{x}$ defines the total dimension of the problem $n + ni$, and the length of the fitness vector $\mathbf{f}$ defines the fitness dimension of the problem. The base class contains a number of other virtual methods that may be reimplemented in derived classes, including a constraint-computation function that returns a constraint vector $\mathbf{c}$ given an input of $\mathbf{x}$.

Section 5.3.3 explains the methodology behind steering the Horus re-entry vehicle. The objective of the optimization performed in the Horus Preliminary Mission Planner (HPMP) is to find an optimal trajectory by way of finding the guidance vector $\mathbf{\Gamma}$ that results in an optimal trajectory.

7.1.3.2 Population

An instance of the optimization problem is passed to the PaGMO population class; the created population is now associated with that specific problem. The PaGMO class contains a set of N individuals (candidate solutions) defined by $[\mathbf{x}^i, \mathbf{f}^i, \mathbf{c}^i, \mathbf{x_{best}}^i, \mathbf{f_{best}}^i, \mathbf{c_{best}}^i]$ with $i = 1, \ldots, N$, where the vectors with subscript "best" represent the best values obtained by the individual so far; these best values are analogous to the concept of the ideal point vector $\mathbf{z}$ mentioned in Section 7.1.2. Upon creation, the population is filled with random decision vectors, the components of which have boundaries defined by the boundary vectors **lb** and **ub**. These random decision vectors are created using a Random Number Generator (RNG); the seed used in this generator is either assigned by the population class itself, or can be supplied to the class as a user input.

7.1.3.3 Algorithm

All optimization algorithms implemented in PaGMO derive from the algorithm base class; the one thing they all have in common is the implementation of the virtual *evolve* method. This method is at the core of all optimization algorithms; it takes an instance of the population as input, and performs the optimization process according to the specific optimization algorithm it is called in reference to.

The algorithms applied in the HPMP are *moead* (Section 7.1.2) and *monte_carlo* (Section 4.5).

MOEA/D

The PaGMO *moead* class is, not surprisingly, the implementation of the MOEA/D optimization algorithm. The class is initialized with a number of variables defining the specific parameters of the MOEA/D algorithm (e.g., the crossover parameter, the method of weight generation etc.); these parameters are selected in such a way that the optimization algorithm offers a good performance for the problem at hand. An instance of the population is supplied to the *evolve* method, which performs the optimization for *gen* number of generations according to the method outlined in Section 7.1.2.

Monte Carlo

The PaGMO *monte_carlo* class is very simple; it takes an instance of the population and "evolves" it by randomly generating a new individual using a PRNG at each iteration and replacing the worst individual if it happens to be better. This class is used in this thesis to perform Monte Carlo simulations only; the *evolve* method is neglected.

7.2 Optimization Problem

The problem of finding an optimal trajectory or a number of optimal trajectories for atmospheric re-entry is a multiobjective optimization problem. In addition to developing a trajectory that keeps the vehicle within its operational and structural constraints, mission-specific goals and priorities defined by the user/ client must be taken into consideration as well. In the design-phase trajectory development considered in this thesis, these priorities are twofold:

1. Minimizing the heat load
2. Maximizing the (down)range

While achieving maximum range is the defining objective of the mission planner, minimizing the total heat load incurred by the vehicle during its trajectory is the objective of every re-entry mission. The mission planner will develop trajectories with maximal range capabilities under the provision that the total heat load is minimized. The optimization problem must therefore be defined in such a way that optimization occurs according to both these objectives with regard to their relative importance, as well as the adherence to operational constraints.

Consistent with the notation used in Section 7.1.1, the MOP for the preliminary design of the Horus trajectory can be written as:

$$\begin{aligned} \text{find: } & \mathbf{x} \in \Omega \\ \text{to minimize: } & \mathbf{f}(\mathbf{x}) = [-X_{dr}, Q] \\ \text{subject to: } & \mathbf{h}(\mathbf{x}) \leq [q_{c,\max}, n_{g,\max}, |\dot{\alpha}_{\max}|, |\dot{\sigma}_{\max}|] \end{aligned} \tag{7.8}$$

where X_{dr} is the downrange, Q the total heat load where $Q = \int q_c(t)dt$, and the inequality constraints are given by the maximum heat flux $q_{c,\max}$, the g-load limit $n_{g,\max}$, and the maximum rate of change of the attitude angles: $\dot{\alpha}_{\max}$ and $\dot{\sigma}_{\max}$. In determining the quality of a trajectory, consideration will also be paid to the longevity of possible constraint violations in the form of their integrated effects.

7.2.1 Decision Variables

The course of a re-entry trajectory is determined by the steering (i.e., guidance) commands to the vehicle. The methodology behind generating and interpreting guidance commands was explained Section 5.3.3: a number of nodes at which given attitude angles are defined are specified in terms of the normalized specific energy $\hat{E}$ of the vehicle at certain points throughout the trajectory. The guided attitude angles between these nodes are computed using Hermite spline interpolation each time the guidance system is called.

The guidance matrix $\mathbf{\Gamma}$ defining the locations $\hat{\mathbf{E}} = [\hat{E}_0, \ldots, \hat{E}_n]^T$, and attitude commands $\boldsymbol{\alpha}(\hat{\mathbf{E}}) = [\alpha_0, \ldots, \alpha_n]^T$ and $\boldsymbol{\sigma}(\hat{\mathbf{E}}) = [\sigma_0, \ldots, \sigma_n]^T$, with n being the number of mutable nodes in the trajectory is of the form

$$\mathbf{\Gamma} = \begin{bmatrix} \mathcal{N}_s \\ \mathcal{N}_1 \\ \vdots \\ \mathcal{N}_{n-1} \\ \mathcal{N}_f \end{bmatrix} = \begin{bmatrix} \hat{E}_s & \alpha_s & \sigma_s \\ \hat{E}_0 & \alpha_0 & \sigma_0 \\ \hat{E}_1 & \alpha_1 & \sigma_1 \\ \vdots & \vdots & \vdots \\ \hat{E}_{n-1} & \alpha_{n-1} & \sigma_{n-1} \\ \hat{E}_f & \alpha_f & \sigma_f \end{bmatrix} \tag{7.9}$$

Note that the initial and final nodes – $\mathcal{N}_s$ and $\mathcal{N}_f$, respectively – as well as the values of the commanded attitude angles α_C and σ_C are pre-specified and are thus not contained in the mutable node set; the reasoning behind this was explained in Section 5.3.3. The total number of nodes specifying the attitude guidance is thus $n + 2$.

The guidance profile of the vehicle is thus completely shaped by the guidance matrix $\mathbf{\Gamma}$. By specifying the guidance matrix, the course of the re-entry trajectory is defined as well. The relative shapes of α-guidance

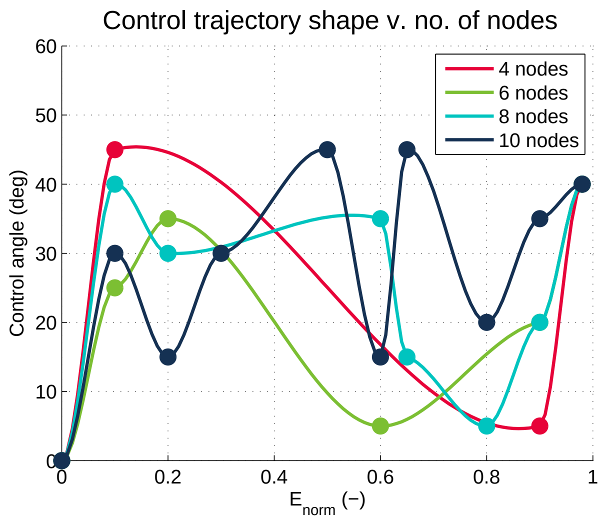


Figure 7.3: Guidance profile shapes defined by four, six, eight and ten nodes.

profiles defined by a certain number of nodes as computed by Hermite spline interpolation are shown in Figure 7.3. The guidance matrices corresponding to these profiles are:

$$\mathbf{\Gamma}_4 = \begin{bmatrix} \mathcal{N}_s \\ \mathcal{N}_1 \\ \mathcal{N}_2 \\ \mathcal{N}_f \end{bmatrix} = \begin{bmatrix} 0.0 & 0^\circ \\ 0.1 & 45^\circ \\ 0.9 & 5^\circ \\ 0.98 & 40^\circ \end{bmatrix}, \quad \mathbf{\Gamma}_6 = \begin{bmatrix} \mathcal{N}_s \\ \mathcal{N}_1 \\ \mathcal{N}_2 \\ \mathcal{N}_3 \\ \mathcal{N}_4 \\ \mathcal{N}_f \end{bmatrix} = \begin{bmatrix} 0.0 & 0^\circ \\ 0.1 & 25^\circ \\ 0.2 & 35^\circ \\ 0.6 & 5^\circ \\ 0.9 & 20^\circ \\ 0.98 & 40^\circ \end{bmatrix}$$

$$\mathbf{\Gamma}_8 = \begin{bmatrix} \mathcal{N}_s \\ \mathcal{N}_1 \\ \mathcal{N}_2 \\ \mathcal{N}_3 \\ \mathcal{N}_4 \\ \mathcal{N}_5 \\ \mathcal{N}_6 \\ \mathcal{N}_f \end{bmatrix} = \begin{bmatrix} 0.0 & 0^\circ \\ 0.1 & 40^\circ \\ 0.2 & 30^\circ \\ 0.6 & 35^\circ \\ 0.65 & 15^\circ \\ 0.8 & 5^\circ \\ 0.9 & 20^\circ \\ 0.98 & 40^\circ \end{bmatrix}, \quad \mathbf{\Gamma}_{10} = \begin{bmatrix} \mathcal{N}_s \\ \mathcal{N}_1 \\ \mathcal{N}_2 \\ \mathcal{N}_3 \\ \mathcal{N}_4 \\ \mathcal{N}_5 \\ \mathcal{N}_6 \\ \mathcal{N}_7 \\ \mathcal{N}_8 \\ \mathcal{N}_f \end{bmatrix} = \begin{bmatrix} 0.0 & 0^\circ \\ 0.1 & 30^\circ \\ 0.2 & 15^\circ \\ 0.3 & 30^\circ \\ 0.5 & 45^\circ \\ 0.6 & 15^\circ \\ 0.65 & 45^\circ \\ 0.8 & 20^\circ \\ 0.9 & 35^\circ \\ 0.98 & 40^\circ \end{bmatrix}$$

Figure 7.3 shows that the variety in guidance profile and thus trajectory shape greatly increases with the number of (mutable) nodes. In terms of locating an optimal trajectory, this is beneficial as it allows for greater guidance profile variety. However, the dimension of the solution space Ω_n of the optimization problem OP_n scales as:

$$\dim \Omega_n = 3(n-2) \tag{7.10}$$

In other words, for optimal solutions to be found for trajectories defined by more nodes, more iterations of the optimization algorithm are required. The tradeoff to be made is that generally it can be expected for more-node guidance profiles to lead to better trajectories due to the larger amount of controllability.

The task of the optimization algorithm in the larger context of the mission planner is to find the guidance profile defined by $\mathbf{\Gamma}$ that results in an optimal trajectory according to the optimization objectives. The dimension of the optimization problem is given by the number of decision variables in the problem definition, i.e., the length of the decision vector $\mathbf{x}$ belonging to an individual. The $3(n-2)$ decision variables of an optimization problem OP_n are the individual components of $\mathbf{\Gamma}$:

$$\mathbf{x} = \left[\hat{E}_1, \ldots, \hat{E}_{n-2},\ \alpha_1, \ldots, \alpha_{n-2},\ \sigma_1, \ldots, \sigma_{n-2}\right] \tag{7.11}$$

The timely success of the optimization algorithm is dependent on the number of decision variables; too many cause the solution space to become incredibly large, which drastically increases the difficulty of finding the global optimum. The preliminary trajectory planning method developed in this thesis investigates trajectories defined by four, six, eight and ten nodes; the corresponding optimization problems are denoted OP_4, OP_6, OP_8, and OP_{10}, and the dimensions of their solution spaces are:

$$\begin{aligned}\dim\Omega_4 &= 6\\ \dim\Omega_6 &= 12\\ \dim\Omega_8 &= 18\\ \dim\Omega_{10} &= 24\end{aligned} \tag{7.12}$$

The lower and upper boundary constraints **lb** and **ub** that define the intervals wherein the values of each decision variable must lie, are given by

$$\mathbf{lb} = [\hat{E}_{\min},\ldots,\hat{E}_{\min},\ \alpha_{\min},\ldots,\alpha_{\min},\ \sigma_{\min},\ldots,\sigma_{\min}] \tag{7.13a}$$

$$\mathbf{ub} = [\hat{E}_{\max},\ldots,\hat{E}_{\max},\ \alpha_{\max},\ldots,\alpha_{\max},\ \sigma_{\max},\ldots,\sigma_{\max}] \tag{7.13b}$$

Note that $\dim\mathbf{lb} = \dim\mathbf{ub} = \dim\mathbf{x} = 3(n-2)$.

7.2.2 Problem Objectives

The mission planner has two primary objectives: the maximization of the vehicle's downrange and the minimization of the total heat load incurred by the vehicle during the course of its trajectory. However, an optimal trajectory in terms of one objective is very different from a trajectory that is optimal in terms of the other; in fact, these objectives are conflicting.

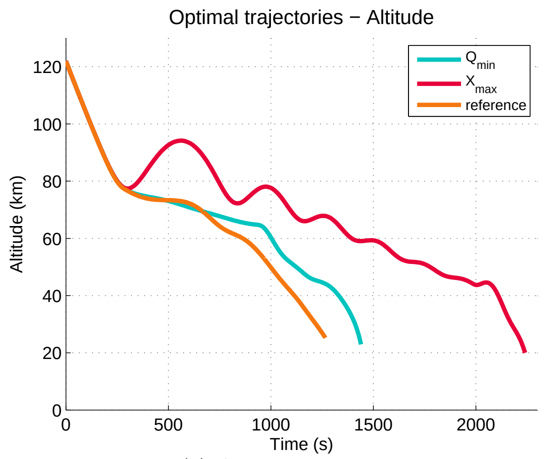
Minimum heat load A trajectory of which the only concern is to minimize the heat load will generally take the form of a gliding entry trajectory. The total heat load supplied to the vehicle is simply the integral of the heat flux over the duration of the entry; in such a trajectory the total heat load is minimized by performing the entry in as short a time as possible without exceeding the operational constraints. This results in a smooth heat flux profile where the heat flux magnitude is maintained right below its constraint value for the majority of the trajectory. Performing the entry in less time would cause the vehicle to exceed its heat flux constraints, whereas a slower entry would incur a higher total heat load due to its longer duration.

Maximum range When trying to achieve maximum range, the vehicle must stay in the air for as long as possible. To this purpose, the duration of the re-entry must be maximized without exceeding the operational constraints. It is therefore not necessary for the heat flux to remain close to its constraint value throughout the trajectory; this leads to the allowance for a more irregular heat flux profile and therefore also for a skip trajectory. A skip trajectory per definition has a fluctuating heat flux profile: going in the upwards direction during re-entry both slows down the vehicle and takes it to a less-dense atmosphere – both of which have the effect of decreasing the heat flux. After the vehicle has reached a peak and continues downwards, its velocity and the atmospheric density increase again, causing the heat flux to increase as well. When optimizing for maximum range, attention must be paid to the rate of change of the heat flux caused by these fluctuations, as well as the amount of heating cycles; a too-quick increase in q_c or too many heat flux peaks can cause damage to the vehicle's TPS.

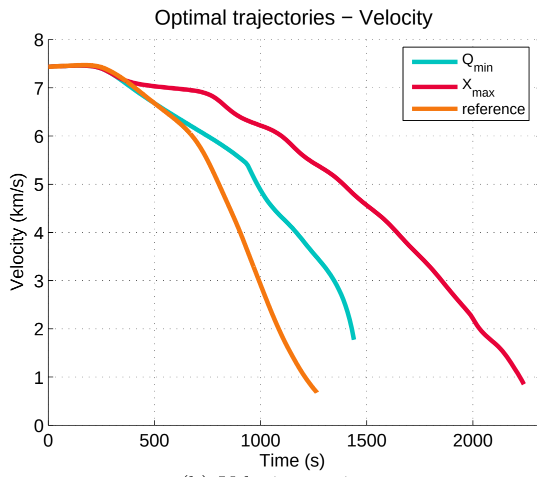
Figure 7.2.2 shows the difference between a heating-optimal and a range-optimal trajectory[1]. In addition, the nominal trajectory of Horus as defined in Mooij (1998) is shown in comparison[2]. The difference in

[1]These trajectories were computed using the MOEA/D algorithm with a population of 20 over 100 generations and a two-dimensional objective function with a parameter relating to either the range or the total heat load and a parameter relating to the rates of change of the attitude angles. A more detailed discussion of these parameters is provided in Sections 7.2.3 and 7.2.4.

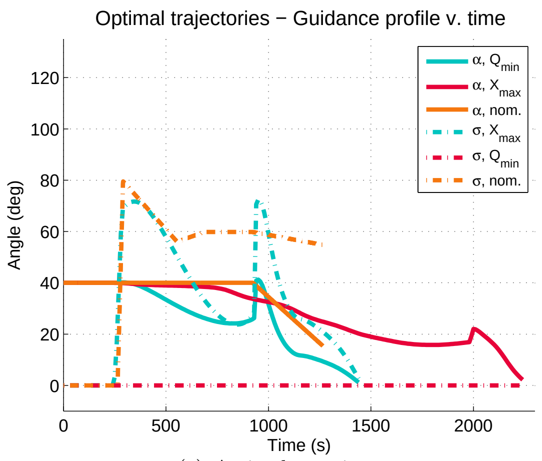
[2]Simulation data for the nominal HORUS trajectory was obtained from Mooij (2014). Data for the heat flux profile was not available; therefore it was computed by simulating the trajectory with the linearly interpolated reference guidance profile from 5.12. The resulting heat flux profile was verified with the heat flux profile plot from (Mooij, 1998)



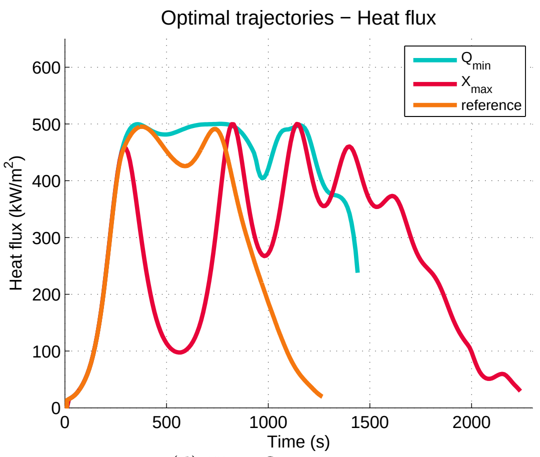
(a) Altitude v. time



(b) Velocity v. time



(c) Attitude v. time



(d) Heat flux v. time

altitude profiles enforces that it is indeed beneficial for the range to follow a skipping trajectory. The consequence of these skips in terms of the vehicle's velocity can be seen in Figure 7.4(b); the vehicle decelerates slower which allows it to stay aloft longer. The heat flux fluctuations corresponding to these skips become apparent in Figure 7.4(d). The extent to which these fluctuations are to be allowed depends on the maximum heat flux rate $\dot{q}_c$ the TPS of the vehicle can withstand. It is apparent that the nominal Horus reference trajectory was determined with minimizing the heat load Q as the primary objective.

The largest conflicts in determining an optimal trajectory that meets both the maximum range and minimum heat load objectives thus are:

- Minimum vs. maximum entry duration
- Smooth heat flux profile vs. (likely) fluctuating heat flux profile

Ideal would be to find a trajectory that provides a feasible compromise between the two. Section 7.1.1.2 discussed the concept of Pareto optimality. Given a certain Pareto set, the solution that represents the best compromise between these two conflicting objectives is the one that is closest to a certain ideal; this ideal is defined as the "utopia point". The location of the utopia point $\mathbf{p}^*$ is the point specified by the lowest f_1 and f_2 values in the Pareto set:

$$\mathbf{p}^* = (p_1^*,\ p_2^*)) = (\min(\mathbf{f}_1),\ \min(\mathbf{f}_2)) \tag{7.14}$$

where $\mathbf{f}_1$ and $\mathbf{f}_2$ are vectors containing the f_1 and f_2 values of all the points in the Pareto set.

Table 7.1: Horus operational constraints.

Constraint		Value	
Max. heat flux	c_{q_c}	500	kW/m^2
Max. g-load	c_{n_g}	3	(-)
Max. g-load rate	$c_{dn_g/dt}$	0.25	s^{-1}
Max. α-rate	$c_{\|d\alpha/dt\|}$	5	deg/s
Max. σ-rate	$c_{\|d\sigma/dt\|}$	15	deg/s
Max. T_w gradient	$c_{\|dT_w/dt\|}$	35	K/s

7.2.3 Constraint Handling

7.2.3.1 Optimization boundaries

The boundary constraints specified in Equation 7.13 define the magnitude of the entire solution space Ω. The limits of the decision variables $\hat{E}$, α_C and σ_C are specified in the **lb** and **ub** vectors:

$$\mathbf{lb} = \{\hat{E}_{\min}, \alpha_{\min}, \sigma_{\min}\} = \{0.1, 0^\circ, 0^\circ\}$$
$$\mathbf{ub} = \{\hat{E}_{\max}, \alpha_{\max}, \sigma_{\max}\} = \{0.98, 45^\circ, 90^\circ\}$$

The choice of the limit values $\hat{E}_{\min} = 0.1$ and $\hat{E}_{\max} = 0.98$ was discussed in Section 5.3.3. The limits to the attitude angles are based on the limits used in the Space Shuttle guidance profile (Harpold and Graves, 1979) and the nominal Horus guidance profile (Mooij, 1998). The lower limit to the bank angle, $\sigma_{\min} = 0^\circ$ is the only deviation from these profiles; in this work, σ_C is taken to represent an absolute value of the bank angle in the interval $[-90^\circ, 90^\circ]$. The relevance of the sign of σ_C is limited to the lateral component of the vehicle's motion; bank reversals (i.e., maintaining the bank angle magnitude but modulating its sign) are a common method of limiting the crossrange error during entry (Saraf et al., 2004). The direction of σ_C has no effect on the magnitude of the aerodynamic forces acting on the vehicle, nor on the final downrange X_{dr}, which is simple the integrated value of $V \cos\gamma$ over the re-entry time. It was decided that as the goal of the mission planner is to maximize downrange and not to target a specific location – and therefore there is no crossrange error to compute – to only optimize for the absolute value of σ_C as this leads to a smaller solution space, allowing the optimization algorithm to find an optimal solution more efficiently.

7.2.3.2 Operational constraints

Additional limitations are imposed on the problem in the form of operational constraints, which define the attainable subset of the total solution space Ω. The operational constraints are defined based on vehicle and crew capacities; for the re-entry problem considered in this thesis, these are given in Table 7.1. All operational constraints are in the form of inequality constraints.

The maximum heat flux constraint c_{q_c} refers to the maximum local heat flux at the nose of the vehicle (with radius $R_N = 0.8$ m^2, see Table 3.1) as computed by Equation 2.2:

$$q_c = c^* {R_N}^n \left(\frac{\rho}{\rho_0}\right)^{1-n} \left(\frac{V}{V_c}\right)^m$$

The value of $c_{q_c} = 500$ kW/m^2 is based on the nominally defined heat flux constraint of 530 kW/m^2 (Mooij, 1998). The g-load constraint of $n_{n_g} = 3$ is taken from the g-load constraint defined for the Space Shuttle entry (Harpold and Graves, 1979) and can be considered representative for the case when the vehicle has human occupants. The maximum g-load rate dn_g/dt was added as an extra measure to increase the comfort of the crew; its value is taken such that a change in n_g of $1g_0$ may occur during four seconds at its fastest. The rates of change of the attitude angles are taken from literature for the Horus reference vehicle (Mooij, 1998). The maximum rate of change of the wall temperature of the vehicle $c_{|dT/dt|}$ is taken to be 35 K/s (Mooij, 2014).

The mission planner utilizes the MOEA/D-algorithm provided by PaGMO (see Section 7.1.3). This algorithm does not offer a built-in constraint handling method. To incorporate the operational constraints into the problem, two options were considered: either write and add a constraint handling method to the existing MOEA/D-algorithm in PaGMO, or incorporate the constraints into the calculation of the objective function using penalty functions. The drawback of imposing hard constraints is that it effectively turns the solution space Ω into Swiss cheese, especially in a highly nonlinear problem such as atmospheric re-entry, thereby drastically hindering the search for a global optimum (Dijkstra, 2012). Therefore the choice was made to handle constraints by defining penalty functions based on a method developed by Feoktistov (2006).

Penalty functions are created for each operational constraint, relating the magnitudes of parameters to a penalty value p that will be added as an increment to the objective function. Because of the difference in units and magnitudes of the penalty parameters (e.g., $c_{q_c} = 500$ kW/m^2 and $c_{n_g} = 3$), their corresponding penalty functions must be defined in such a way that the resulting penalty values are comparable in magnitude. This is done by normalizing the parameter penalty computation with its corresponding constraint value. In general, the penalty parameters only have nonzero values when constraints are exceeded; the operational constraints are defined in such a way that sustained flight within these constraints is considered safe. Minimizing constrained parameters within these bounds is unnecessary and deteriorates the performance of the optimization algorithm with regard to its primary objectives. One exception is made for a component of the heat flux penalty; the reason for this is explained in the following section.

Heat flux

The heat flux is directly related to an objective parameter: it is the derivative of the heat load Q. In addition to its integral value over the whole trajectory, the heat flux profile of a trajectory is defining for its feasibility and optimality. The heat flux penalty function p_{q_c} is the sum of three components:

$$p_{q_c} = p_{q_c,1} + p_{q_c,2} + p_{q_c,3} \tag{7.15}$$

1) Maximum heat flux $q_{c,\max}$ should stay below its constraint value c_{q_c}. If the constraint is exceeded, the integral value of the heat flux excess over the duration of the constraint violation should be as low as possible.

$$p_{q_c,1} = \begin{cases} 0 & \text{if } q_{c,\max} \le c_{q_c} \\ \frac{q_{c,\max}}{c_{q_c}} + \Delta p_{q_c} & \text{if } q_{c,\max} > c_{q_c} \end{cases} \tag{7.16a}$$

$$\text{where} \quad \Delta p_{q_c}(t) = \frac{\sum_{i=1}^{n} \int_{t_{0,i}}^{t_{f,i}} (q_c(t) - c_{q_c})_i dt}{\int_{t_0}^{t_f} c_{q_c} dt} \tag{7.16b}$$

where n is the number of violations and the interval $t_{f,i} - t_{0,i}$ is the duration of each violation. The numerators are a summation of the areas under the curves of the parameters as a function of time minus the area under the constant constraint value for the same time periods, i.e., the areas representing the actual excess only. The denominator is the constraint value integrated over the entire time duration of the trajectory.

2) Maximum temperature gradient dT_w/dt should stay below its constraint value $c_{dT_w/dt}$. The wall temperature gradient is computed by applying thermal equilibrium between the aerodynamic heat flux q_c and the radiative heat flux at the wall q_{rad}:

$$q_c = q_{rad} = \varepsilon \sigma_B T_w^4 \tag{7.17}$$

where $\varepsilon = 0.8$ is the emissivity and $\sigma_B = 5.667 \cdot 10^{-8}$ W/m^2/K^4 is Boltzmann's constant. This gives for the relation between dT_w/dt and dq_c/dt:

$$\frac{dq_c}{dt} = 4\varepsilon\sigma_B T_w^3 \frac{dT_w}{dt} \tag{7.18}$$

Computing $T_{w,\max}$ is computed as the maximum temperature gradient corresponding to Taking the temperature and therefore the gradient value at the maximum allowed heat flux Therefore

$$c_{dq_c/dt} = 4c_{dT_w/dt} \sqrt[4]{\varepsilon\sigma_B c_{q_c}^3} \approx 20 \text{ kW/m}^2\text{/s} \tag{7.19}$$

The heat flux value in the equation was taken to be c_{q_c} to ensure that cdq_c/dt is not underestimated. It is preferred to have a trajectory with a relatively smooth heat flux profile, i.e., fluctuations are allowed but the heat gradient should not approach the vehicle's limits. Therefore to avoid unnecessarily fluctuating heat flux profiles, cdq_c/dt is set at 10 kW/m^2/s. The penalty function for the wall temperature gradient dT_w/dt is given by:

$$p_{q_c,2} = \begin{cases} 0 & \text{if } (dT_w/dt)_{\max} \leq c_{dT_w/dt} \\ \frac{(dT_w/dt)_{\max}}{c_{dT_w/dt}} & \text{if } (dT_w/dt)_{\max} > c_{dT_w/dt} \end{cases} \tag{7.20}$$

3) Fluctuations $p_{q_c,3}$ is the only penalty parameter that always has a nonzero value (with the exception of trajectories that have only one heat flux peak), not only when a certain limit is exceeded. The reason for this is simply that the amount of peaks is not subject to any predefined constraint, though optimally the combined effect of their magnitudes and frequency should be minimized.
A completely smooth heat flux profile is not required per se due to conflict with the range maximization objective (see Section 7.2.2), but a profile with too many fluctuations is considered disadvantageous to the vehicle's TPS, so it is taken into account in the penalty function to encourage the creation of trajectories with the most range with minimal fluctuations. The second component of this penalty function is the magnitude of the peaks in the heat flux profile – drastic fluctuations in the heat flux profile may damage the vehicle's TPS and should be avoided.
The number of peaks and valleys n_{pv} is determined by counting the number of sign changes in dq_c/dt over the trajectory. To avoid small irrelevant fluctuations from interfering with the computation, only sustained sign changes that last over 100 s are taken into account. The individual fluctuation magnitudes $|\Delta q|$ between the peaks and valleys in the profile (excluding the initial and final peaks) are summed, and divided by the number of fluctuations $n_f - 2$ between these peaks; the final penalty value is obtained by dividing the total by c_{q_c}.

$$p_{q_c,3} = \frac{\sum |\Delta q|/(n_{pv} - 2)}{c_{q_c}} \tag{7.21}$$

G-load

The g-load penalty p_{n_g}, consists of two components: one with regard to its maximum value, and one for the rate of change:

$$p_{n_g} = p_{n_g,1} + p_{n_g,2} \tag{7.22}$$

The first component $p_{n_g,1}$ is analogous to $p_{q_c,1}$:

$$p_{n_g,1} = \begin{cases} 0 & \text{if } n_{g,\max} \leq c_{n_g} \\ \frac{n_{g,\max}}{c_{n_g}} + \Delta p_{n_g} & \text{if } n_{g,\max} > c_{n_g} \end{cases} \tag{7.23a}$$

$$\text{where} \quad \Delta p_{n_g}(t) = \frac{\sum_{i=1}^{n} \int_{t_{0,i}}^{t_{f,i}} (n_g(t) - c_{n_g})_i dt}{\int_{t_0}^{t_f} c_{n_g} dt} \tag{7.23b}$$

The second component $p_{n_g,2}$ is given by:

$$p_{n_g,2} = \begin{cases} 0 & \text{if } \dot{n}_{g,\max} \leq c_{|dn_g/dt|} \\ \frac{\dot{n}_{g,\max}}{c_{|dn_g/dt|}} + \Delta p_{n_g} & \text{if } \dot{n}_{g,\max} > c_{|dn_g/dt|} \end{cases} \tag{7.24}$$

$$\text{where} \quad \dot{n}_{g_{\max}} = \max\left(\dot{n}_{g,\max}, |\dot{n}_{g,\min}|\right) \tag{7.25}$$

Attitude rates

The penalties for exceeding the attitude rate constraints $c_{|d\alpha/dt|}$ and $c_{|d\sigma/dt|}$ are dependent on the magnitude of the constraint violation only. They are given by:

$$p_{|d\alpha/dt|} = \begin{cases} 0 & \text{if } \dot{\alpha}_{\max} \leq c_{|d\alpha/dt|} \\ \frac{\dot{\alpha}_{\max}}{c_{|d\alpha/dt|}} & \text{if } \dot{\alpha}_{\max} > c_{|d\alpha/dt|} \end{cases} \quad \text{where } \dot{\alpha}_{\max} = \max\left(\dot{\alpha}_{\max}, |\dot{\alpha}_{\min}|\right) \tag{7.26a}$$

$$p_{|d\sigma/dt|} = \begin{cases} 0 & \text{if } \dot{\sigma}_{\max} \leq c_{|d\sigma/dt|} \\ \frac{\dot{\sigma}_{\max}}{c_{|d\sigma/dt|}} & \text{if } \dot{\sigma}_{\max} > c_{|d\sigma/dt|} \end{cases} \quad \text{where } \dot{\sigma}_{\max} = \max\left(\dot{\sigma}_{\max}, |\dot{\sigma}_{\min}|\right) \tag{7.26b}$$

Note that the constraints defining the maximum rates of change of the attitude angles are absolute values.

7.2.4 Objective Function

The objective function $\mathbf{f}(\mathbf{x})$ computes the fitness of an individual $\mathbf{x^i}$ with regard to the optimization objectives. The primary purpose of the mission planner is to find minimal-heat-load trajectories that provide an extended range, while adhering to boundary and operational constraints. Therefore the defining parameters in the objective function are the total downrange X_{dr} and the total heat load Q. The individual contributions of these parameters to the objective function are defined as "reward functions":

$$r_{X_{dr}} = -\frac{X_{dr}}{\pi R_E} \tag{7.27a}$$

$$r_Q = \frac{Q}{\int_{t_0}^{t_f} c_{q_c} dt} \tag{7.27b}$$

Similarly to the penalty functions, the objective values are normalized to make their magnitudes comparable as well. The downrange X_{dr} is divided by half the Earth's circumference, and the heat load Q by the integral of the heat flux constraint over the duration of the trajectory. Note that the objective is to minimize the values of the objective function; therefore the negative range $-X_{dr}$ is used in $r_{X_{dr}}$.

These form the basis of the objective function. The inclusion of the operational constraints in the objective function requires a balance between the importance of the specific constraint parameter in relation to the problem objectives. While algorithms such as MOEA/D are highly capable of solving complex multidimensional problems, making the objective space too large could make the algorithm not see the wood for the trees, as it were – greatly degrading its performance. Simply defining each operational constraint violation as a parameter to be minimized is not wise. The objective function should be defined in such a way that:

- The priorities in terms of what is expected of the solution are clear and are reasonably represented.
- The impact of the magnitudes of constrained parameters on the objective function is proportional to their relative importance to the overall solution.
- The above should be achieved with the lowest possible amount of objectives.

The defining parameters of the quality of a solution are of course the range X_{dr} and the heat load Q; these are defined as two separate objectives as values to be maximized and minimized, respectively. The penalties for the heat flux c_{q_c} and the g-load c_{n_g} are added to both these objective components to ensure that even if these primary objectives are weighted with respect to each other, the adherence to the constraints represented by these parameters is not neglected.

In addition the heat flux penalty p_{q_c} is included in both these objectives. The n_g-constraint is relatively easily adhered to during the trajectory as a consequence of range maximization, and is not considered an optimization objective on its own; therefore its contribution to the objective function remains in the form of its penalty function values. The penalties relating to the attitude angle constraints are defined as

Table 7.2: Penalty and objective function values for the HORUS reference trajectory.

		Penalty
Heat flux	p_{q_c}	1.65
	$p_{q_c,1}$	0.00
	$p_{q_c,2}$	1.38
	$p_{q_c,3}$	0.27
G-load	p_{n_g}	0.00
	$p_{n_g,1}$	0.00
	$p_{n_g,2}$	0.00
Alpha-rate	$p_{\lvert d\alpha/dt\rvert}$	0.00
Sigma-rate	$p_{\lvert d\sigma/dt\rvert}$	0.00
		Reward
Range	$r_{X_{dr}}$	-0.34
Heat load	r_Q	0.58
		Fitness
Objective	f_1	-0.07
	f_2	0.85
	f_3	0.00

a single objective; their importance relative to each other can be assumed to be the same. The objective function is given by:

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ f_3(\mathbf{x}) \end{bmatrix} \tag{7.28}$$

where

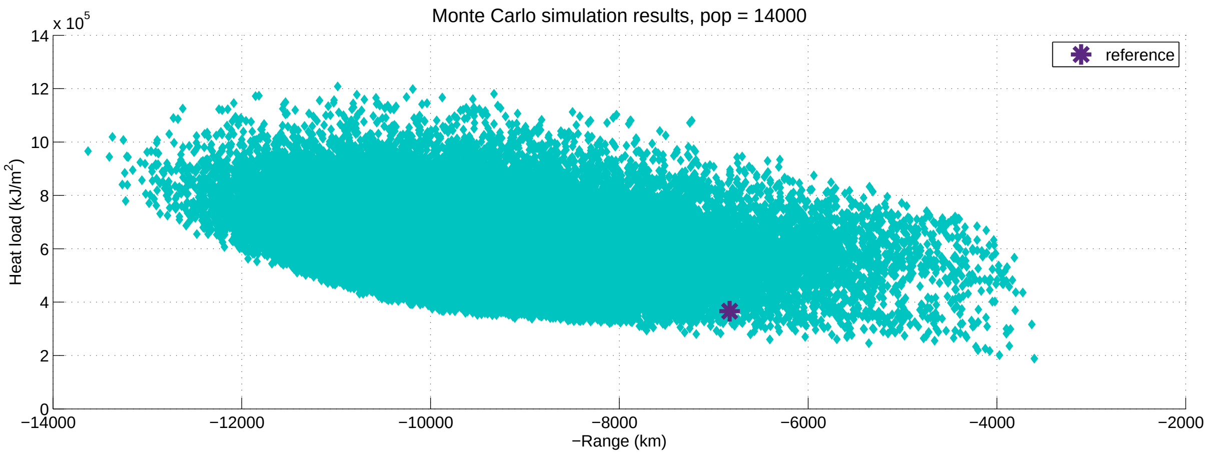
$$f_1 = r_{X_{dr}} + p_{q_c} + p_{n_g} \tag{7.29a}$$
$$f_2 = r_Q + p_{q_c} + p_{n_g} \tag{7.29b}$$
$$f_3 = p_{|d\alpha/dt|} + p_{|d\sigma/dt|} \tag{7.29c}$$

The penalty and objective values for the HORUS reference trajectory are shown in Table 7.2 to give an indication of the magnitudes of the different components of the objective function.

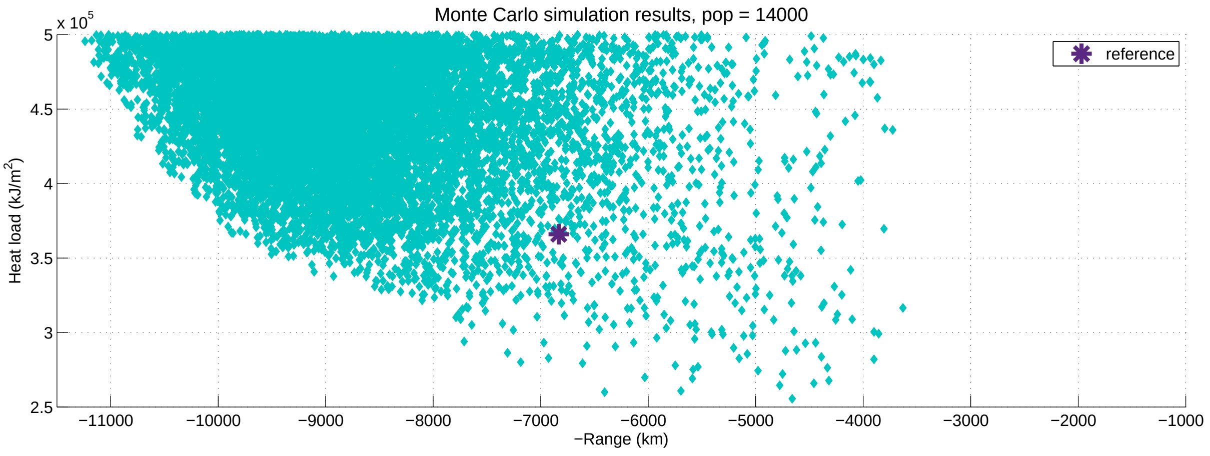
7.2.5 Objective Space

To obtain a general indication of the re-entry problem's objective space, a Monte Carlo simulation with a population of 14,000 individuals generated with a uniform distribution was performed for each control-node problem – i.e., OP$_4$, OP$_6$, OP$_8$ and OP$_{10}$ – and the solutions aggregated. The result is the scatter plot indicating the expected shape and dimensions of the solution space shown in Figures 7.4 and 7.5; the asterix indicates the objective values of the HORUS reference trajectory. A number of initial conclusions can be drawn:

- Improvements in terms of both downrange X_{dr} and total heat load Q are possible with respect to the reference trajectory. In addition, a lerge number of solutions exist which represent an inprovement in terms of one of these objectives. It should be noted that the Figures 7.4 and 7.5 give no indication of whether an individual solution meets the mission's objective constraints. It is likely that solutions with $Q_{\text{tot}} < Q_{\text{ref}}$ with a short range exceed the heat flux constraint, for instance.
- The objective space is orders of magnitude larger in every dimension than the attainable objective space, i.e., the region where the operational constraints are satisfied. This means that finding a global optimum is difficult as the algorithm is likely to converge on a local optimum. In terms of practicality, this need not be a problem as for preliminary mission design purposes close-to-optimal solutions may be satisfactory.



(e) Total

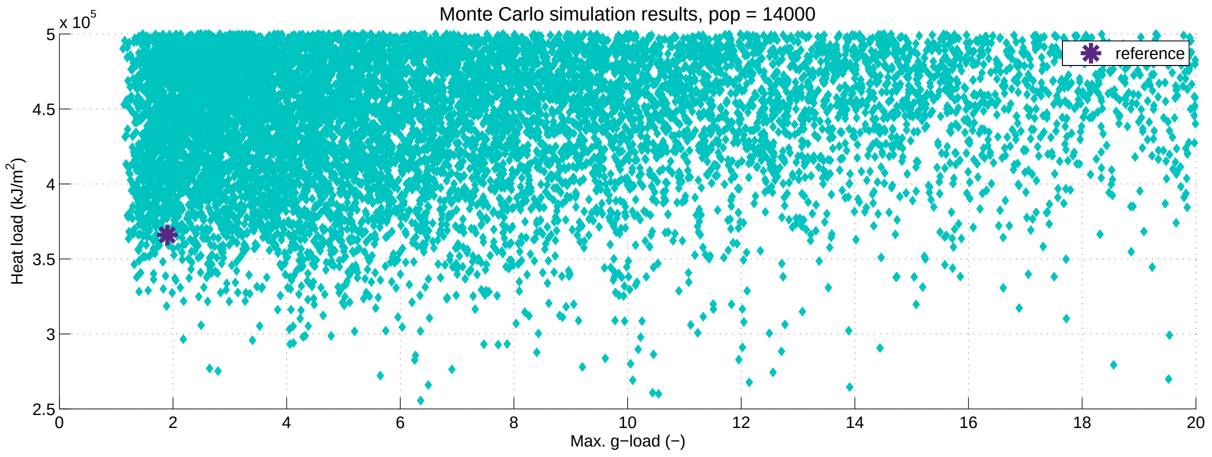


(f) Detail

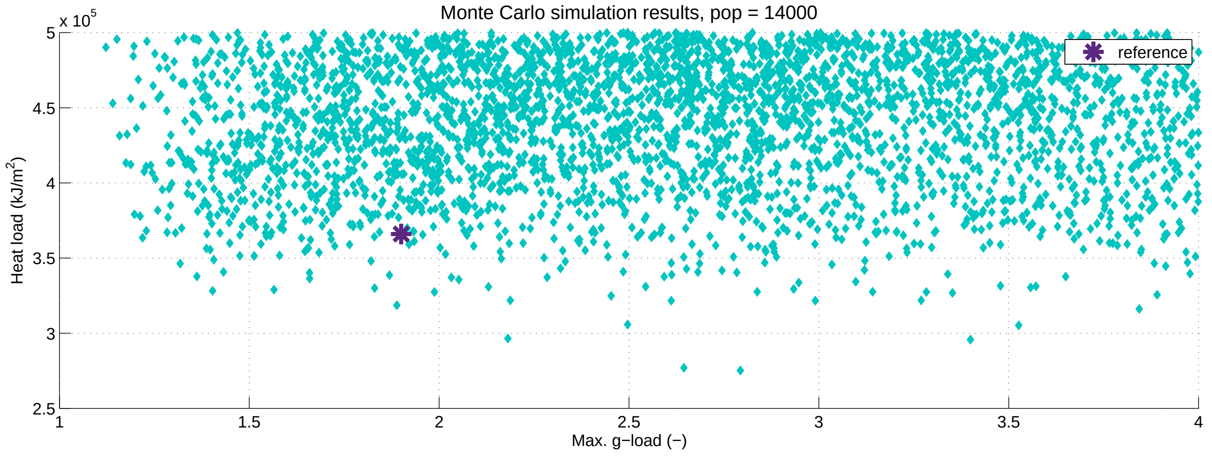
Figure 7.4: Monte Carlo simulation of the re-entry problem in terms of -range vs. maximum heat flux.

- The minimum attainable heat load approaches a limit of 3 kJ/m^2 (Figures 7.4(f) and 7.5(b)).
- The minimum $n_{g,\max}$ approaches a limit of 1 (Figure 7.5(b)).

Were it the case that all these attainable solutions were clustered around a global minimum, solving the re-entry MOP would be a simple task. However as it turns out, the set of solutions corresponding to the attainable objective space is widely distributed (see e.g., the different guidance profiles in Figures 8.3(a) and 8.3(a) which result in very similar-looking trajectories). The best approach for determining not only sufficiently good but near-optimal preliminary trajectories in such a solution space can be either one or both of two polar opposites: the highly capable and robust Multiobjective Optimization Algorithm (MOAD) found in the MOEA/D-algorithm, or a fully heuristic approach in the form of a large number of Monte Carlo simulations. In The following chapter, both approaches are evaluated for mission planning, as well as two multi-step methods consisting of one or both of these approaches.



(a) Total



(b) Detail

Figure 7.5: Monte Carlo simulation of the re-entry problem in terms of maximum g-load vs. maximum heat flux.

Chapter 8

Mission Planner Design and Testing

8.1 Design

The mission planner strives to determine a number of optimal trajectories in terms of the mission objectives minimal heat load and maximum range within its operational constraints. Ideally, these trajectories are somewhat varied in their performance with respect to these two primary objectives while meeting all mission requirements, i.e., they are all objectively "good" trajectories but are varied in their degree of optimality with regards to the primary objectives. In other words, the mission planner should determine trajectory solutions with a distributed Pareto set with regards to the objectives f_1 and f_2 as defined in Section 7.2.4 (Equation 7.29). This allows the user some agency in choosing the trajectory to meet their specific needs.

The design of the mission planner relies on the integration of the HORUS re-entry simulator (see Chapter 6) with the open-source optimization library PaGMO. The simulator was developed such that it is suitable for integration at the top level, allowing the mission planner to treat it as a "black box" that is fed decision variables and returns objective values when called. Optimization algorithms need to perform a certain number of evaluations of the objective function – which translates to running an entire trajectory simulation – per iteration, reinforcing both the necessity of efficient code and early convergence on a solution. Due to the highly irregular solution space, a concrete choice in terms of the approach to obtaining optimal trajectories was left up in the air thus far; no final design choices in this regard could be made without empirically determining the best option. Therefore four different approaches towards mission planning were developed and a tradeoff was performed to determine which approach gives the best performance according to a number of criteria; see Section 8.2.6.

Throughout the design process, the nominal mission defined for HORUS was used as the test case. The conditions at entry are given in Table 8.1.

Within the mission planner, the HORUS re-entry simulator is integrated with the PaGMO library to provide the foundation for the objective function evaluations. The PaGMO-framework was discussed in the general sense in Section 7.1.3.

The basis of any optimization problem implemented in PaGMO is its unique problem class, containing the method that performs the evaluation of the objective function for a specific individual. The optimization problems $\{\mathrm{OP}_4, \mathrm{OP}_6, \mathrm{OP}_8, \mathrm{OP}_{10}\}$ associated with the mission planner are implemented as derived classes of the PaGMO problem base class, and can be linked to both the Monte Carlo and MOEA/D algorithms provided by PaGMO.

The mission planner problems $\{\mathrm{OP}_4, \mathrm{OP}_6, \mathrm{OP}_8, \mathrm{OP}_{10}\}$ are initialized with:

- The PaGMO problem base class variables **lb**, **ub**, n, ni, nc, nic and c_{tol} (see Section 7.1.3.1).

Table 8.1: Nominal initial conditions.

Initial parameter		**Value**	
Vehicle state			
Altitude	h_0	122	km
Longitude	τ_0	-106.7	deg
Latitude	δ_0	-22.3	deg
Velocity	V_0	7.4355	(km/s)
Flightpath angle	γ_0	-1.43	deg
Heading angle	χ_0	70.75	deg
Vehicle attitude			
Angle of attack	α_0	40	deg
Angle of sideslip	β_0	0	deg
Bank angle	σ_0	0	deg

– **lb** and **ub**, both with $\dim n$, are defined according to Equation 7.13:

$$\begin{aligned}\mathbf{lb} &= [\hat{\mathbf{E}}_{\min}, \boldsymbol{\alpha}_{\min}, \boldsymbol{\sigma}_{\min}] \quad \text{with} \quad \{\hat{E}_{\min}, \alpha_{\min}, \sigma_{\min}\} = \{0.1, 0^\circ, 0^\circ\} \\ \mathbf{ub} &= [\hat{\mathbf{E}}_{\max}, \boldsymbol{\alpha}_{\max}, \boldsymbol{\sigma}_{\max}] \quad \text{with} \quad \{\hat{E}_{\max}, \alpha_{\max}, \sigma_{\max}\} = \{0.98, 45^\circ, 90^\circ\}\end{aligned}$$

– The global dimension of the problem $n = 3 \times (m - 2)$ where m is the number of mutable control nodes $\mathcal{N}_i(\hat{E}_i)$ with $i = 1, \ldots, m$ defines the length of the decision vector $\mathbf{x}$.

– The problems have no integral part and constraints are handled via penalty functions, so $ni = nc = nic = c_{tol} = 0$.

- A pointer to an instance of the problem properties (Section 6.3.6). In addition to the simulation variables, the problem properties class contains the operational constraint parameters as defined in Table 7.1.

The method to evaluate the objective function calls the simulator with a decision vector $\mathbf{x}$ and the pointer to the problem properties; the simulator returns the parameters necessary to implement Equation 7.29.

8.2 Approach

The mission planner is modular in nature, allowing for customization of the planning approach based on the user's needs. The software is developed with four built-in alternative methodologies, the choice of which can be user-specified. The motivation for this customizability is to allow for the comparison of the performance of vastly different approaches in the context of finding optimal re-entry trajectories for a problem defined by a highly irregular objective space. Due to the modularity of the code, methods can be combined and staggered in different ways, and additions can be easily implemented if the framework is followed.

As mentioned in the beginning of Section 8.1, different approaches with regard to determining the best solution were developed and implemented as it was not possible to predict with any certainty whether a heuristic or deterministic approach (or both) would be the most promising. Four methodologies were developed in two separate categories: single and dual approach.

Single approach The mission planner solutions are generated using one method only.

- **mp_MC** uses Monte Carlo simulations only. The method configuration is specified by one parameter, namely the population size n_p for the Monte Carlo simulations associated with each OP – therefore, during each run of **mp_MC**, $4n_\text{p}$ solutions are generated. The **mp_MC** method will form the baseline to which the other methods will be compared, as it is by far the easiest to implement; if the other methods provide no noticeable benefit to Monte Carlo simulations, the search space is likely highly nonlinear.

- **mp_MOEA/D** uses the MOEA/D optimization algorithm only. The configuration is specified by the population size n_{p} and number of generations (i.e., iterations) n_{g} of the MOEA/D algorithm. After each run, $4n_{\mathrm{p}}$ solutions are output by the method in the form of the aggregated final populations of each separate OP.

Dual approach Two approaches are used in tandem, wherein the solutions of the first specify the begin conditions of the second. The purpose of this approach is to perform a targeted search within the specific area in the solution space where a Pareto-optimal solution has been found, in case a better solution is located close-by. The Pareto set generated by the first method is used to define the targeted solution spaces wherein the second part of the method will continue the search:

1. From the Pareto set, a number n_{dv} of solutions closest to the utopia point of the set are selected.
2. The decision vectors belonging to these solutions are used to generate "targeted" solution spaces around these solutions by adding and subtracting some value from each decision vector component to form the boundary of the new solution. A user-specified vector of "deltas" with $\dim(n)$ defines the size of the new solution space for each variable:

$$\boldsymbol{\Delta}\mathbf{x} = \left[\Delta\hat{\mathbf{E}}, \Delta\boldsymbol{\alpha}, \Delta\boldsymbol{\sigma}\right] \tag{8.1}$$

So the new boundaries are:

$$lb_{i,new} = \begin{cases} x_i - \Delta x_i & \text{if} \quad x_i - \Delta x_i > lb_i \\ lb_i & \text{if} \quad x_i - \Delta x_i \leq lb_i \end{cases} \tag{8.2}$$

$$ub_{i,new} = \begin{cases} x_i + \Delta x_i & \text{if} \quad x_i + \Delta x_i < ub_i \\ ub_i & \text{if} \quad x_i + \Delta x_i \geq ub_i \end{cases} \tag{8.3}$$

with $i = 1, \ldots, n$.

A targeted search using Monte Carlo simulations (TMC) – specified by $n_{\mathrm{p,TMC}}$ – is then performed in each new targeted solution space to try to find better solutions than the ones provided by the first part.

- **mp_MOEA/D-TMC** uses the MOEA/D optimization algorithm to generate solutions within the first part. The Pareto set of the final aggregated population of size $4n_{\mathrm{p}}$ (i.e., the sum of the final populations obtained for each OP) is computed, and a number n_{dv} of solutions closest to the utopia point of that set are selected to create the new targeted solution spaces for the TMC.
- **mp_MC-TMC** Monte Carlo simulations are used in the first part as well, to generate the solutions that the TMC will continue with.

In each case, the mission planner will output solutions in three categories:

t_all the entire final population.

t_pareto the Pareto set of the final population **t_all**.

t_ideal a user-specified number k of Pareto-ideal (as defined in Section 7.2.2) solutions. The ideal solutions are computed by defining a Pareto utopia point with respect to f_1 and f_2 and finding the k closest points. The solutions corresponding to these points are returned.

8.2.1 Performance

Each approach is defined by a set of parameters that influence the quality of the obtained solutions:

- **mp_MC**:
  1. population size n_{p}
- **mp_MOEA/D**:
  1. population size n_{p}
  2. number of generations (i.e., iterations) n_{g}
- **mp_MOEA/D-TMC**:
  1. MOEA/D population size $n_{\mathrm{p,MOEA/D}}$

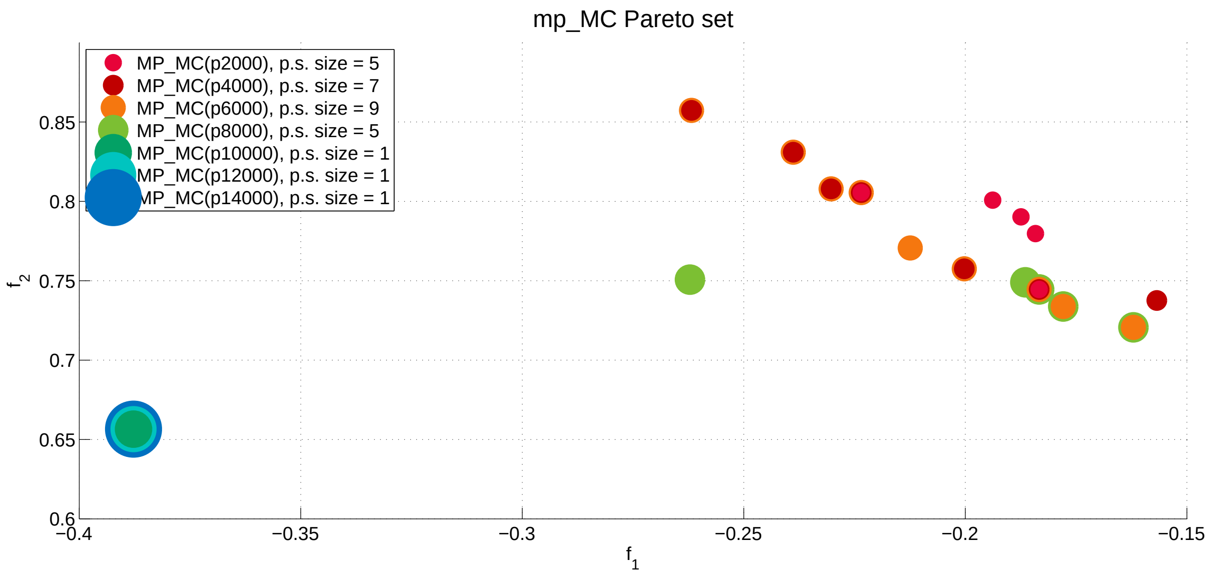


Figure 8.1: **mp_MC** Pareto set with increasing n_p.

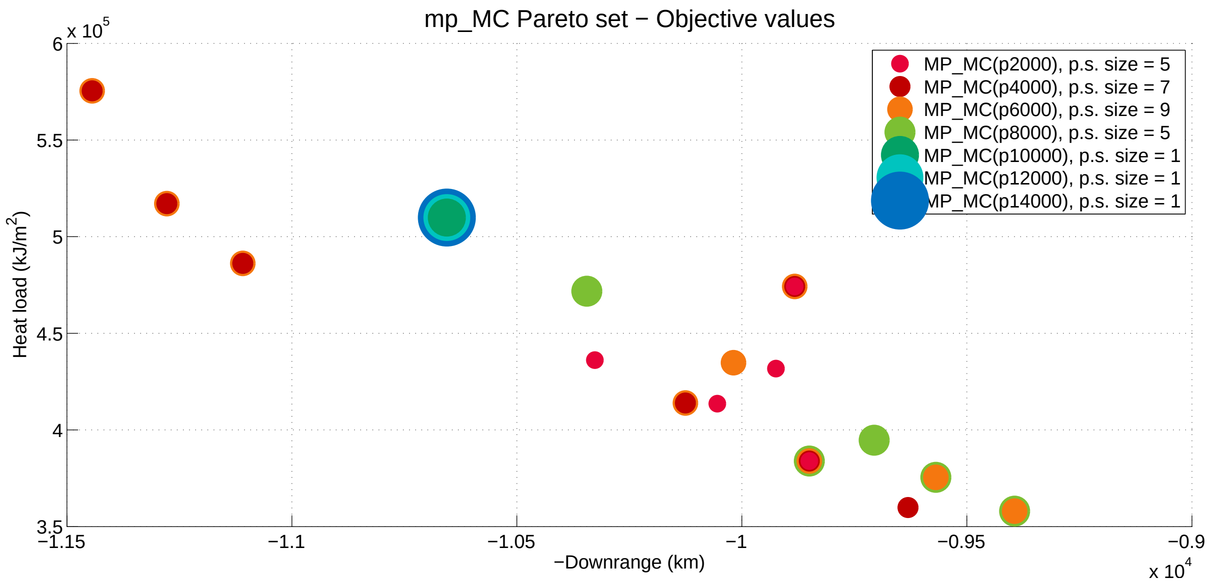


Figure 8.2: **mp_MC** Pareto set objective values with increasing n_p.

  2. number of generations (i.e., iterations) n_{g}
  3. number of decision vectors n_{dv}
  4. components of the delta-vector $\mathbf{\Delta x} = [\Delta\hat{\mathbf{E}}, \Delta\boldsymbol{\alpha}, \Delta\boldsymbol{\sigma}]$
  5. targeted Monte Carlo population size $n_{\mathrm{p,TMC}}$
- **mp_MC-TMC**:
  1. Monte Carlo population size $n_{\mathrm{p,MC}}$
  2. number of generations (i.e., iterations) n_{g}
  3. number of decision vectors n_{dv}
  4. components of the delta-vector $\mathbf{\Delta x} = [\Delta\hat{\mathbf{E}}, \Delta\boldsymbol{\alpha}, \Delta\boldsymbol{\sigma}]$
  5. targeted Monte Carlo population size $n_{\mathrm{p,TMC}}$

The parameters of influence on the performance of the dual methods are not all independent of each other; three factors can be defined that independently influence the performance of a dual method:

1. The relative allocation of function evaluations to the first and second part of the method defined by the ratio r:
$$r = \frac{n_{\mathrm{eval,1}}}{n_{\mathrm{eval,2}}} \tag{8.4}$$
2. The number of decision vectors n_{dv} provided by the first part from which the targeted solution spaces for the TMC's in the second part are generated
3. The size of the targeted solution spaces belonging to each decision vector, i.e., the components of $\mathbf{\Delta x}$

For a dual method to be considered useful, the results it generates should have better objective values than the starting points found by the first part. Factors independently influencing the performance of a dual method are the ratio r, the number of decision vectors n_{dv} and the delta-vector $\mathbf{\Delta x}$. The effect of each of these factors is evaluated separately for both dual methods:

1. Effect of r: configurations are defined with constant $n_{\mathrm{dv}} = 5$ and $\mathbf{\Delta} = [0.1,\ 5°,\ 10°]$ while r takes the values of $r = \{3,\ 1,\ \frac{1}{3}\}$.
2. Effect of $\mathbf{\Delta}$: configurations are defined with constant $r = 1$ and $n_{\mathrm{dv}} = 5$ while $\mathbf{\Delta}$ takes the values of:
$$\begin{aligned}\mathbf{\Delta} &= [0.01,\ 0.5°,\ 1°]\\ \mathbf{\Delta} &= [0.05,\ 2.5°,\ 5°]\\ \mathbf{\Delta} &= [0.15,\ 7.5°,\ 15°]\\ \mathbf{\Delta} &= [0.20,\ 10°,\ 20°]\end{aligned}$$
3. Effect of n_{dv}: configurations are defined with constant $r = 1$ and $\mathbf{\Delta} = [0.1,\ 5°,\ 10°]$ while n_{dv} takes the values of $r = \{3,\ 5,\ 7\}$.

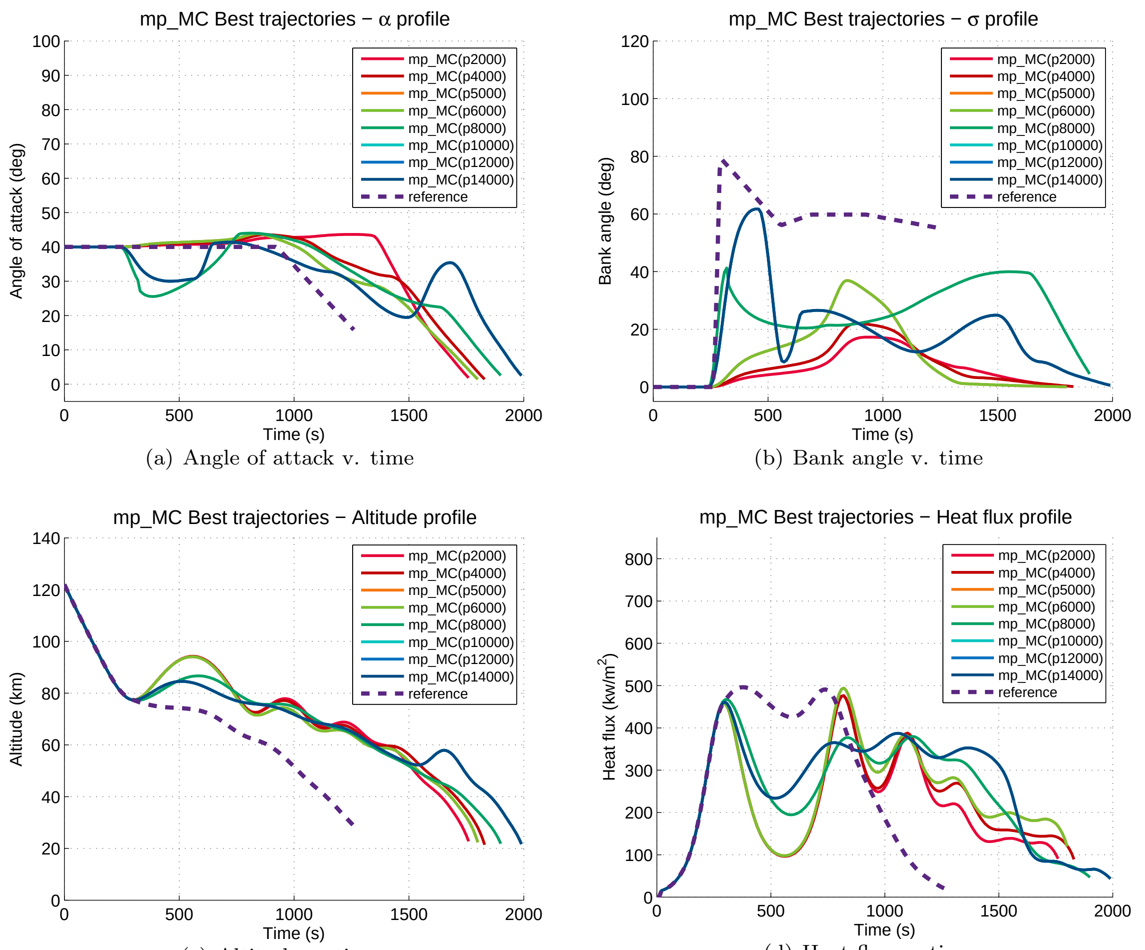
The parameters should be selected such that the method is capable of delivering its best performance. To this purpose, the influence of the parameter selection on each method was investigated. A baseline was established based on the number of evaluations n_{eval} performed during one run of the mission planner at $n_{\mathrm{eval}} = 40,000$, i.e., 10,000 evaluations per OP. An evaluation is defined as a single computation of the objective function, i.e., simulation of a trajectory. For the evaluation of these methods, the mission planner was initialized with the initial conditions of the HORUS reference trajectory as shown in Table 8.1.

8.2.2 mp_MC

Neglecting the influence of the random seed, the performance of **mp_MC** is solely dependent on the population size n_{p}, which is corresponds one-to-one to the number of performed evaluations. The total number of function evaluations performed during a run of **mp_MC** is thus:

$$n_{\mathrm{eval,MC}} = 4n_{\mathrm{p,MC}} \tag{8.5}$$

The configuration that corresponds to the baseline of $n_{\mathrm{eval}} = 40,000$ is **mp_MC**(p10,000) – the population size is defined per OP.



(a) Angle of attack v. time

(b) Bank angle v. time

(c) Altitude v. time

(d) Heat flux v. time

Figure 8.3: Best **mp_MC** trajectory profiles.

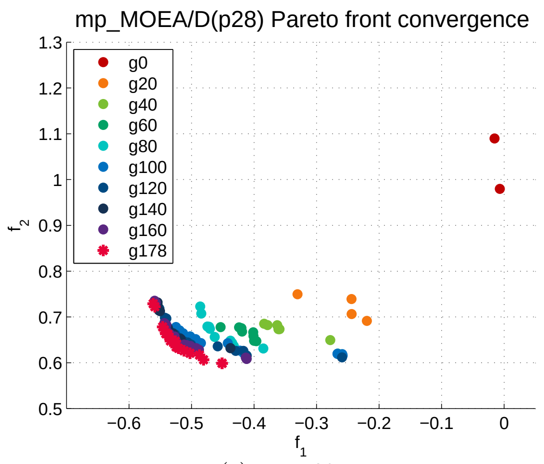
8.2.2.1 Pareto fitness

Figure 8.1 shows the Pareto set – in terms of $\{f_1, f_2\}$ as defined in Equation 7.29 – corresponding to **mp_MC**(p10,000), as well as the Pareto sets of smaller and larger Monte Carlo populations for comparison. The Monte Carlo simulations used to generate these populations were initialized with the same random seed[1] to eliminate its influence on the relative performances; this causes the overlap between the Pareto sets. The dots indicating the Pareto sets of each **mp_MC** configuration are different sizes to avoid solutions being hidden due to this overlap. In addition, Figure 8.2 shows the corresponding objective values of the individuals in the Pareto set; these do not correspond one-on-one with the objective fitness values, as the fitness values are computed based on also the adherence to operational constraints – see Equation 7.29.

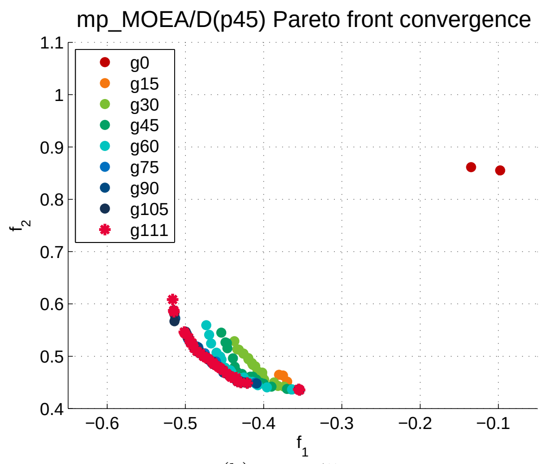
An increase in n_p has a positive effect on the objective fitnesses of the solutions in the population's Pareto set. A larger population generated at random[2] gives a larger probability of finding a solution with better objective values, so the initial improvements come as no surprise. However, no improvement to the Pareto set occurs from $n_\mathrm{p} = 10,000$ onwards through $n_\mathrm{p} = 14,000$. This is coupled to a sudden decrease in the size of the Pareto set to just a single point, which is likely the consequence of the Monte Carlo simulation finding an exceptionally good solution which pushes all the other points out of the Pareto set. Increasing the population size to 14,000 does not lead to the generation of any better solutions than this one solution, leading to the same Pareto set being maintained. Likely this point is close to a local

[1]1843629728

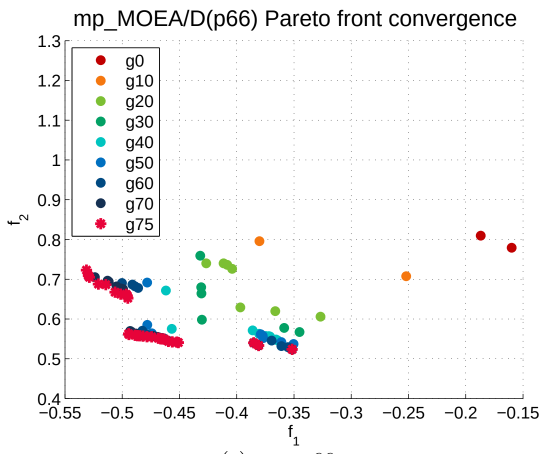
[2]N.B. With a uniform distribution, as no information is known about the distribution.



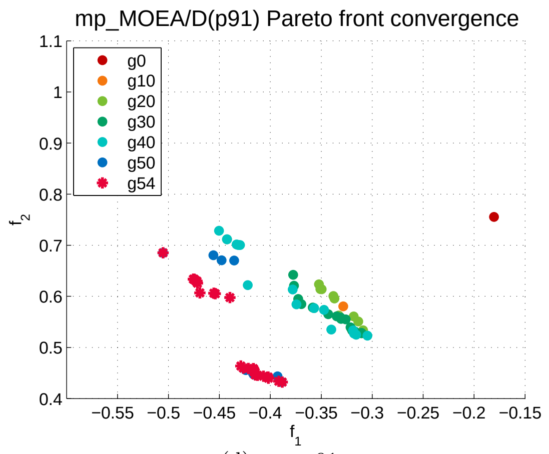
(a) $n_\mathrm{p} = 28$



(b) $n_\mathrm{p} = 45$



(c) $n_\mathrm{p} = 66$



(d) $n_\mathrm{p} = 91$

Figure 8.4: **mp_MOEA/D** Pareto convergence.

optimum, meaning a better solution is probably located very close by; as individuals are generated at random using a uniform distribution, the likelihood of a point being generated with a decision vector both very close to such a Pareto point and in the right direction is very small. Herein lies the benefit of applying an optimization algorithm to such a problem, which instead of heuristically determining the next point, bases it on the solutions obtained thus far. It should be noted that this occurrence and the population size at which it occurs is highly dependent on the random seed used to initialize the Monte Carlo simulations; e.g., the Pareto set belonging to $n_\mathrm{p} = 10,000$ with random seed 340397439 consists of five individuals.

8.2.2.2 Trajectory profiles

The trajectory profiles for a number of parameters corresponding to the best solution (in terms of its proximity to the Pareto utopia point) determined by **mp_MC** for each n_p are shown in Figure 8.3 in comparison to the HORUS reference trajectory from (Mooij, 1998). Note that some trajectory profiles are hidden as they are equal to the best profile within (an) other population(s). The solutions obtained by Monte Carlo simulations are adequate in terms of the operational constraints of the problem (see Figure 8.3(d)), but cannot be considered "good" re-entry trajectories in the context of the mission objectives – mainly due to the generally large heat flux fluctuations. In comparison to the reference trajectory, the largest differences occur in the total duration of the trajectory (and thus also the total downrange), as well as the shape of the heat flux profile. The HORUS reference trajectory is based on minimum

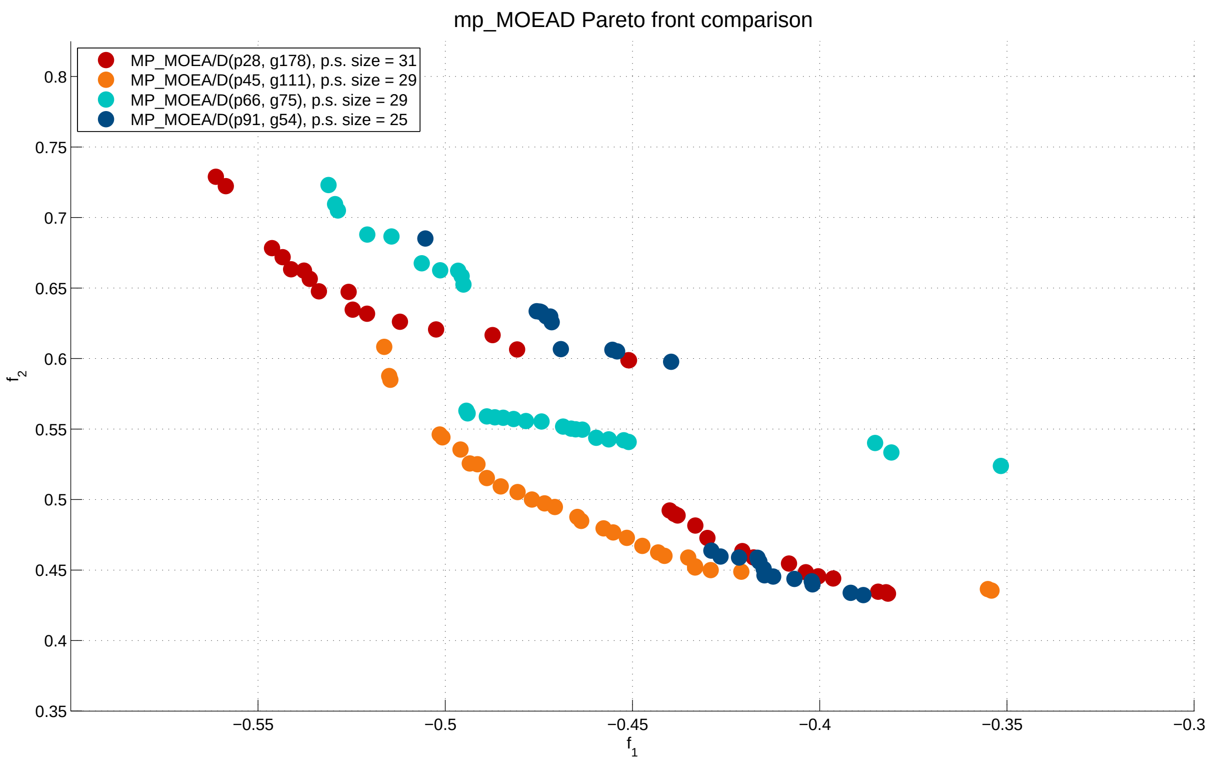


Figure 8.5: Pareto front comparison with different $\{n_\mathrm{p}, n_\mathrm{g}\}$'s for **mp_MOEA/D**.

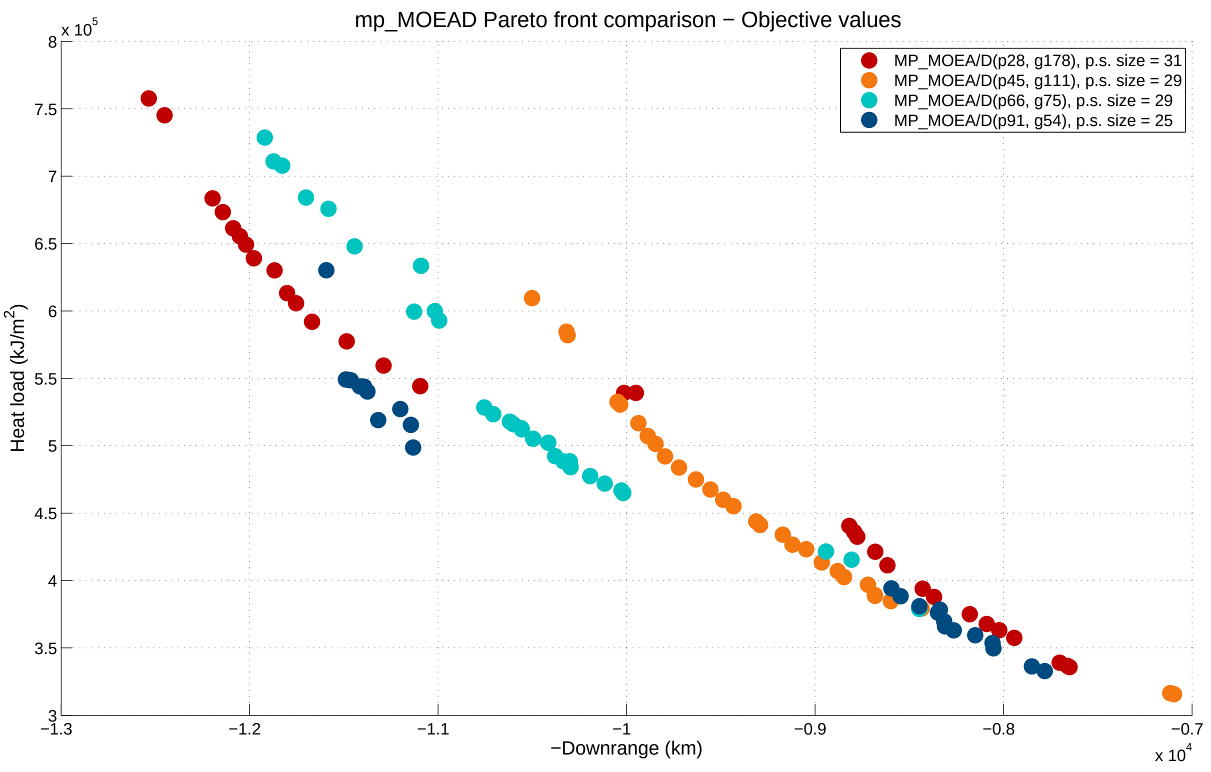


Figure 8.6: Pareto front objective values comparison with different $\{n_\mathrm{p}, n_\mathrm{g}\}$'s for **mp_MOEA/D**.

heat-load objective with no additional objective in terms of the total downrange – therefore the reference trajectory is similar to the minimum-heat-load trajectory discussed in Section 7.2.4: short in duration with q_c maintained close to c_{q_c} and very few fluctuations in the heat flux profile. The total heat load incurred during the **mp_MC**-generated trajectories is thus much larger, as can be seen from the areas under the respective profiles.

The solutions generated by **mp_MC** represent either merely seemingly adequate (e.g, **mp_MC**(p10,000)) or unacceptable (e.g, for **mp_MC**(p2,000) through **mp_MC**(p8,000)) trajectories. **mp_MC** in these configurations does not represent an adequate approach to re-entry mission planning, and likely will offer little improvement with increased population size. The dual method **mp_MC-TMC** was developed to determine whether a staggered heuristic approach would provide better results.

8.2.3 mp_MOEA/D

The factor that influences the performance of the single method **mp_MOEA/D** is the selection of the number of iterations n_g versus the population size n_p for a certain n_eval. It was determined that, given a population size of n_p, the MOEA/D-algorithm performs a number of function evaluations in the order of $\sim 2.5 n_\mathrm{p}$ per generation. Therefore the total number of function evaluations performed per run of **mp_MOEA/D** is taken to be – taking the creation of the initial population into account:

$$n_{\mathrm{eval,MOEA/D}} = 4 n_{\mathrm{p,MOEA/D}}(1 + 2n_\mathrm{g}) \tag{8.6}$$

Due to the weight generation method used in the MOEA/D-algorithm Zhang and Li (2009), the population size n_p is not freely selectable as it depends on the objective dimension of the problem. For $\dim(\mathbf{f}) = 3$, possible population sizes are

$$n_\mathrm{p} = \binom{H + \dim(\mathbf{f}) - 1}{\dim(\mathbf{f}) - 1} = \{\ldots,\ 15\ 28,\ 36,\ 45,\ 66,\ 78,\ 91, \ldots\} \tag{8.7}$$

where H is an integer value.

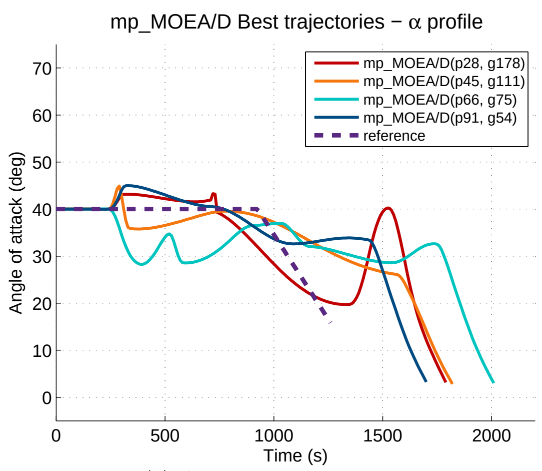
To provide an indication of the influence of the $n_\mathrm{p}/\ n_\mathrm{g}$-ratio for a given number of function evaluations on the performance of the method, four combinations were selected that each correspond to $n_\mathrm{eval} = 40,000$:

- **mp_MOEA/D**(p28, g178) where $n_\mathrm{p}/n_\mathrm{g} \approx 0.16$
- **mp_MOEA/D**(p45, g111) where $n_\mathrm{p}/n_\mathrm{g} \approx 0.41$
- **mp_MOEA/D**(p66, g75) where $n_\mathrm{p}/n_\mathrm{g} \approx 0.88$
- **mp_MOEA/D**(p91, g54) where $n_\mathrm{p}/n_\mathrm{g} \approx 1.69$

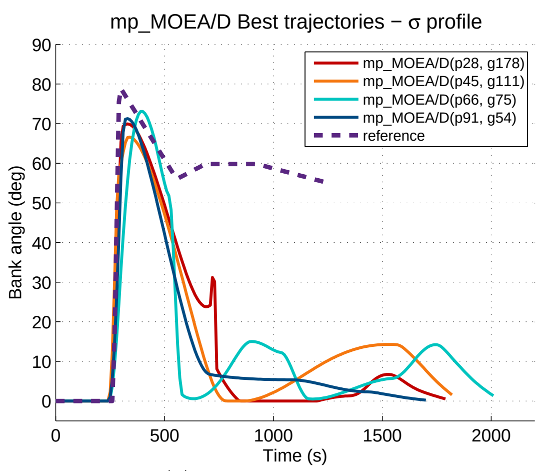
8.2.3.1 Pareto fitness

Figure 8.5 shows the Pareto sets of the selected **mp_MOEA/D**(n_p, n_g)'s at intermediate populations. Notably the Pareto sets seem to approach a certain limit with increasing iterations, i.e., they are converging. The rate of convergence is an indicator of the quality of an optimization algorithm, as a faster convergence typically corresponds to a greater efficiency. **mp_MOEA/D**(p45, g111) has the highest rate of convergence, and it can be seen from Figure 8.4(b) that from generation 75 onwards the change in the Pareto front is minimal. This indicates that the algorithm has converged on an optimum, i.e., has found a solution. The remaining **mp_MOEA/D**(n_p, n_g)'s are less converged at their final iteration, meaning additional iterations will likely yield slightly different (and objectively better) Pareto sets. Furthermore, the Pareto sets belonging to the larger populations of 66 and 91 are quite unevenly distributed, which is not preferable.

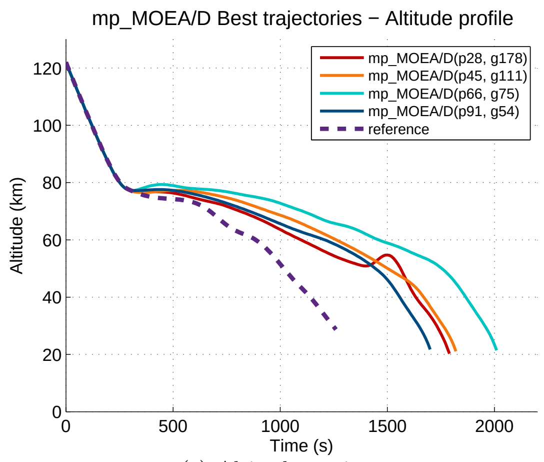
Figure 8.5 shows the Pareto sets of the final populations belonging to the **mp_MOEA/D**($n_\mathrm{p}, n_\mathrm{p}$) configurations side-by-side for comparison. In addition, Figure 8.6 shows the corresponding objective values of the individuals in the Pareto set; these do not correspond one-on-one with the objective fitness values, as the fitness values are computed based on also the adherence to operational constraints – see



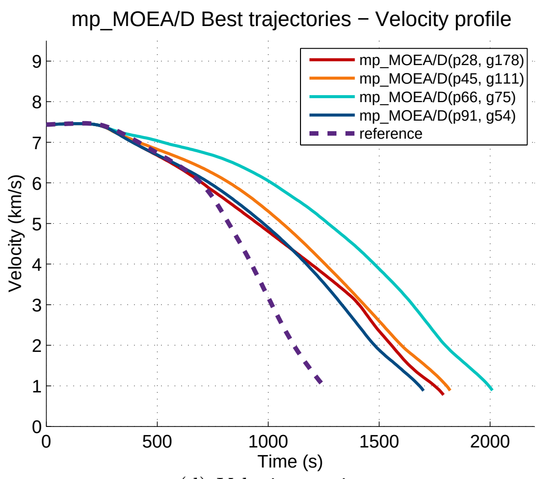
(a) Angle of attack v. time



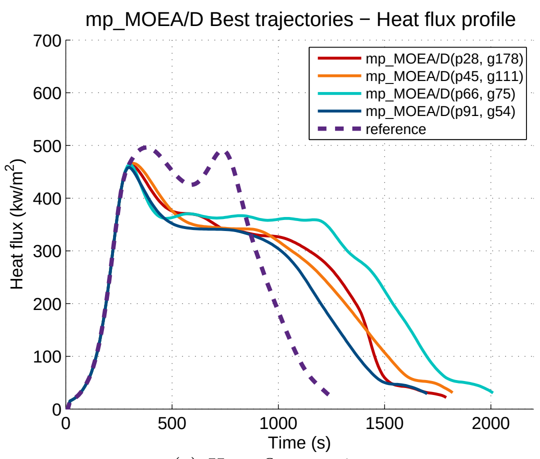
(b) Bank angle v. time



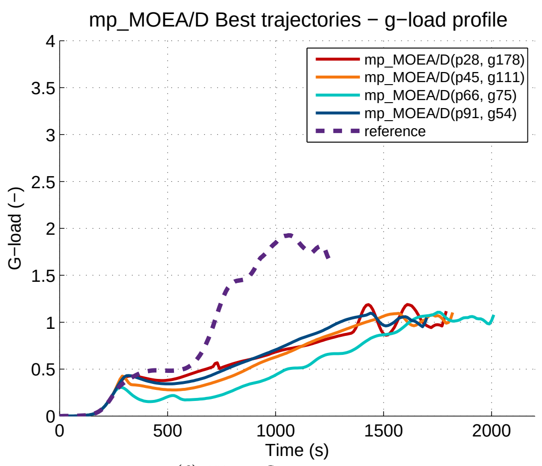
(c) Altitude v. time



(d) Velocity v. time



(e) Heat flux v. time



(f) Heat flux v. time

Figure 8.7: Best **mp_MOEA/D** trajectory profiles.

Equation 7.29. The **mp_MOEA/D**(p45, g111) configuration provides the best objective fitness values, whereas **mp_MOEA/D**(p28, g178) results in a slightly larger and Pareto set with a somewhat larger spread. The Pareto sets belonging to the larger populations are less consistent in shape and provide poorer objective values than the Pareto sets of the smaller populations. This is due to the algorithm not having converged on a solution within the allotted amount of iterations. From the convergence of the Pareto sets as well as the wuality of the results, it can be concluded that for the mission planner, an **mp_MOEA/D** configuration where $n_\mathrm{p}/n_\mathrm{g}$ is close to 0.5 is likely the most beneficial.

8.2.3.2 Trajectory profiles

The best trajectory profiles obtained by **mp_MOEA/D** for each $\{n_\mathrm{p}, n_\mathrm{p}\}$ are shown in Figure 8.7, as well as the HORUS reference trajectory from (Mooij, 1998). Again, these trajectories represent the solutions closest to the unweighted utopia point of each Pareto set. It can be seen that **mp_MOEA/D** provides decidedly better solutions than **mp_MC** for the same n_eval. The first and foremost deciding factor in the dominance of these trajectories over those obtained by **mp_MC** is the smoothness of the heat flux profile (Figure 8.7(e)), as fewer q_c fluctuations means less wear and tear on the vehicle's TPS. The preference for this smoothness was specified in the problem's objective function in the form of the penalty functions $p_{q_c,2}$ and $p_{q_c,3}$; see Equations 7.20 and 7.21, respectively. In addition to maintaining a relatively constant q_c throughout the trajectory, the magnitude of this constant q_c remains as low as possible to allow the vehicle to fly a longer range (Equation 7.27a) while keeping the total heat load to a minimum (Equation 7.27b).

Note the similarity in the guidance profiles of the reference trajectory and the trajectories produced by **mp_MOEA/D**. If the mission planner were given a specified range equal to that of the reference profile, the output trajectory would be almost the same as the reference trajectory. In its current configuration, the mission planner has a bias towards maximizing the downrange, which can be seen in the bank angle profile in Figure 8.7(b) – after an initial large bank angle, σ_C is kept low to minimize the deceleration and maintain altitude longer. If range maximization were not included in the objectives, a bank angle profile closer to that of the reference trajectory would be generated.

The large initial bank angle σ_C serves to keep the magnitude of the initial/maximum peak of q_c below or at c_{q_c}. The guidance profiles begin to deviate once the denser atmosphere is reached at an altitude of around 70 km. The reference trajectory, based on minimizing Q, maintains a near-constant bank angle of $\sigma_\mathrm{C} = 60°$ to achieve quick deceleration of the vehicle (while maintaining $n_g \leq c_{n_g}$) to minimize the duration of the re-entry; this can be seen in Figure 8.7(f). The **mp_MOEA/D** guidance profiles are generated with both minimal Q as well as maximum X_{dr} in mind; there is no benefit to minimizing the duration of the re-entry, as this would occur at the cost of the downrange. It is more beneficial in that case to maintain altitude and decelerate slowly, as is evidenced by Figure 8.7(c) and 8.7(d). The altitude profiles of the **mp_MOEA/D** trajectories all have the same general shape, wherein the vehicle is kept in the upper layers of the denser atmosphere (50 km - 80 km) for a longer period, and deceleration occurs slowly and gradually. The effect on the g-load profile in comparison to the more sudden deceleration in the reference trajectory can be seen in Figure 8.7(f). The velocity profiles in Figure 8.7(d) also show this more gradual deceleration. This slower deceleration at higher altitude is achieved by maintaining a low σ_C after the initial maximum-σ_C (Figure 8.7(b)).

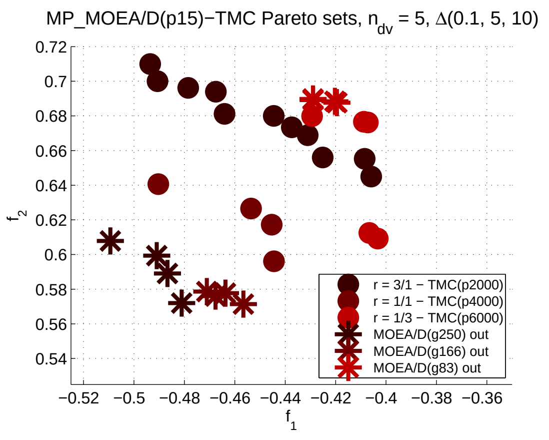
8.2.4 mp_MOEA/D-TMC

The total number of function evaluations performed during one run of **mp_MOEA/D-TMC** is:

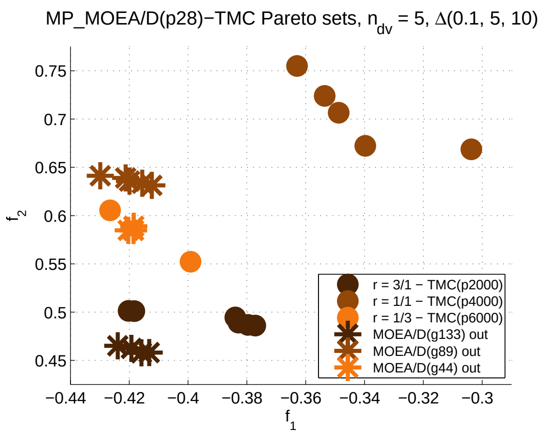
$$n_\mathrm{eval} = 4n_\mathrm{p,MOEA/D}(1 + 2n_\mathrm{g}) + n_\mathrm{dv}n_\mathrm{p,MC} \tag{8.8}$$

1. **Effect of r** For ratios $r = \{3,\ 1,\ \frac{1}{3}\}$, **mp_MOEA/D-TMC**-configurations are defined for MOEA/D populations of $n_\mathrm{p,MOEA/D} = \{15,\ 28,\ 45,\ 66\}$. The number of generations n_g corresponding to each population size $n_\mathrm{p,MOEA/D}$ is given by:

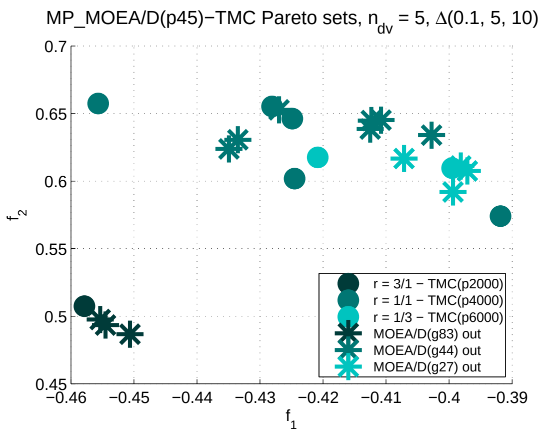
$$n_\mathrm{g} = \frac{1}{2}\left(\frac{r n_\mathrm{dv} n_\mathrm{p,MC}}{4n_\mathrm{p,MOEA/D}} - 1\right) \tag{8.9}$$



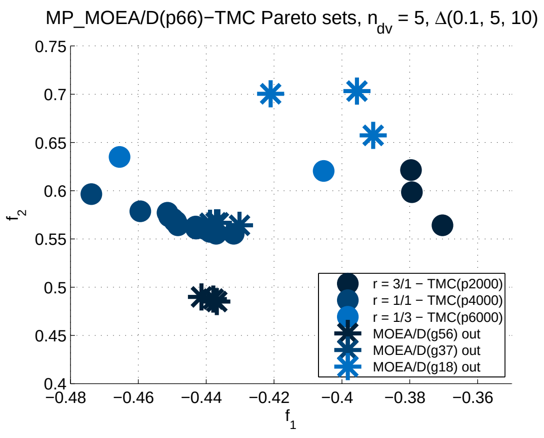
(a) **mp_MOEA/D(p15)-TMC**



(b) **mp_MOEA/D(p28)-TMC**



(c) **mp_MOEA/D(p45)-TMC**

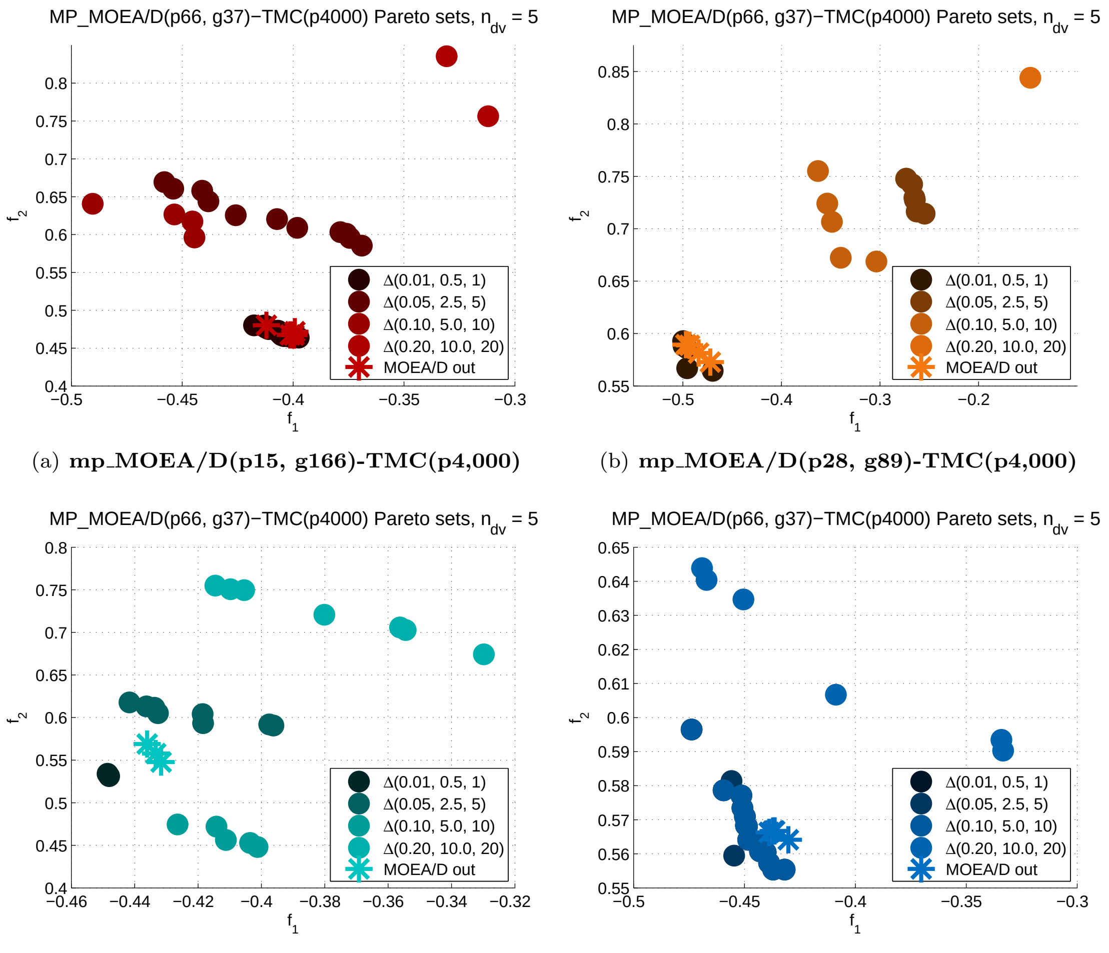


(d) **mp_MOEA/D(p66)-TMC**

Figure 8.8: Effects of r on the Pareto set of **mp_MOEA/D-TMC**.

The number of decision vectors and the delta-vector are kept constant at $n_{\mathrm{dv}} = 5$ and $\mathbf{\Delta} = [0.1,\ 5°,\ 10°]$, respectively. The results are shown in Figure 8.8 for the different **mp_MOEA/D-TMC**-configurations. The asterixes represent the starting values generated by the MOEA/D part of the method, i.e., the $n_{\mathrm{dv}} = 5$ closest solutions to the utopia point of the Pareto set. The dots represent the Pareto sets generated by the corresponding TMC part.
In none of the evaluated cases does consistent improvement upon the MOEA/D results occur due to the application of TMC. The overall trend seems to be that allocating more than half of the function evaluations towards the TMC part – i.e., $r < 1$ is beneficial for the achieved results, or at least not detrimental in terms of objective values. This occurs because sufficient random solutions need to be generated by the TMC part to find a solution better than the one generated by MOEA/D within the constrained solution space. If not enough function evaluations are allocated, only worse solutions will likely be found, leading to worse results than the starting points provided by MOEA/D. Therefore to benefit from supplementing the MOEA/D algorithm with targeted Monte Carlo simulations, the function evaluations need to be allocated such that the new constrained solution space based on the MOEA/D results and $\mathbf{\Delta}$ can be searched sufficiently. If this is not possible, benefit can likely be obtained from decreasing the size of $\mathbf{\Delta}$.

2. Effect of $\mathbf{\Delta}$ The resulting Pareto sets of configurations with varied delta-vectors and $n_{\mathrm{p,MOEA/D}} = \{15,\ 28,\ 45,\ 66\}$ with corresponding n_{g}'s are shown in 8.9. It can be observed that in none of the configurations does any (relevant) improvement occur; in fact, generally the TMC method results in a decrease in the solution quality with respect to the starting points. The **mp_MOEA/D-TMC** methods with the smallest delta-vector of $\mathbf{\Delta} = [0.01,\ 0.5°,\ 1°]$ consistently provide qualitatively



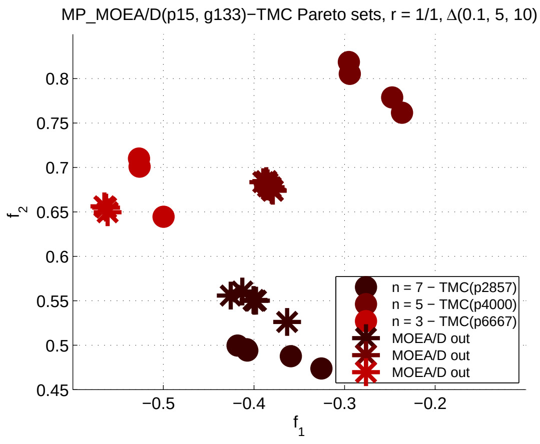
(a) **mp_MOEA/D(p15, g166)-TMC(p4,000)** (b) **mp_MOEA/D(p28, g89)-TMC(p4,000)**

(c) **mp_MOEA/D(p45, g55)-TMC(p4,000)** (d) **mp_MOEA/D(p66, g37)-TMC(p4,000)**

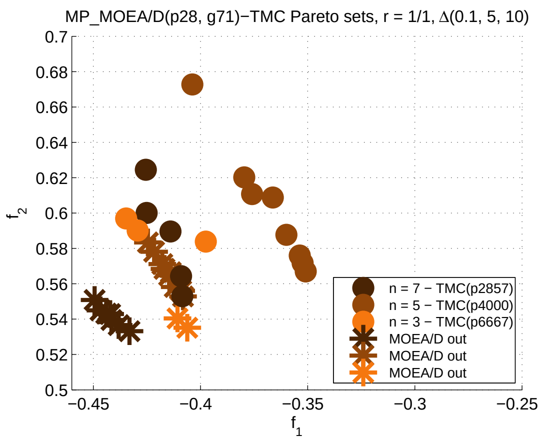
Figure 8.9: Effects of $\mathbf{\Delta}$ on the Pareto set of **mp_MOEA/D-TMC**.

equivalent or somewhat better solutions to the starting points. In general, an increase in the size of $\mathbf{\Delta}$ corresponds to a decrease in the quality of the solutions with respect to the starting points. It was already mentioned in the previous section that for the additional TMC part to have any positive effect, it needs to be defined in such a way that the targeted solution space can be sufficiently searched by the Monte Carlo method. Therefore both an increase in the population size of TMC, as well as a decrease of the size of the targeted solution space, has a beneficial effect on the quality of the TMC's results – bearing in mind that taking the solution space too small is disadvantageous due to the exclusion of too many possible solutions.

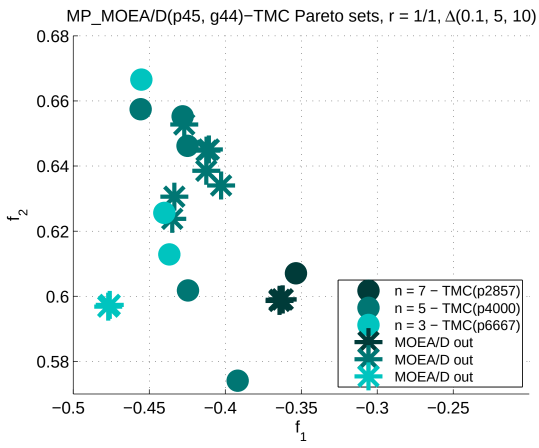
3. Effect of n_{dv} To observe the effect of changing the number of decision vectors n_{dv}, configurations are defined with $r = 1$ and $\mathbf{\Delta} = [0.1,\ 5°,\ 10°]$ for $n_{\text{p,MOEA/D}} = \{15,\ 28,\ 45,\ 66\}$. The Pareto sets of these configurations with corresponding n_{g}'s and $n_{\text{p,TMC}}$'s are shown in 8.10. Generally the solutions provided by the **mp_MOEA/D-TMC** method improve with decreasing n_{dv}, i.e., with increasing TMC-evaluations per solution space for a constant ratio r. This observation corresponds to the previous ones, where it was noted that for an adequate result to be achieved, sufficient Monte Carlo simulations needed to be performed per targeted solution space. The overall trend can be observed where decreasing n_{dv} (and thus increasing $n_{\text{p,TMC}}$) has a beneficial effect on the objective fitnesses of the Pareto set. More can be gained in terms of the quality of the solutions in the Pareto set by generating more random individuals over a smaller number of search spaces than with less individuals spread over a larger number of search spaces of the same size. The size of $n_{\text{p,TMC}}$ is therefore relatively more important for the performance of **mp_MOEA/D-TMC** than n_{dv}.



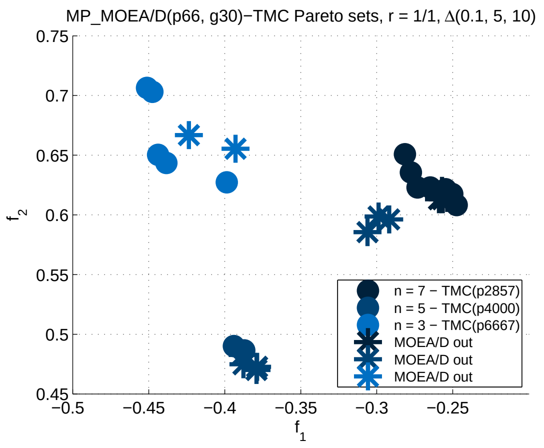
(a) **mp_MOEA/D(p15, g166)-TMC(p4,000)**



(b) **mp_MOEA/D(p28, g89)-TMC(p4,000)**



(c) **mp_MOEA/D(p45, g55)-TMC(p4,000)**



(d) **mp_MOEA/D(p66, g37)-TMC(p4,000)**

Figure 8.10: Effects of n_{dv} on the Pareto set of **mp_MOEA/D-TMC**.

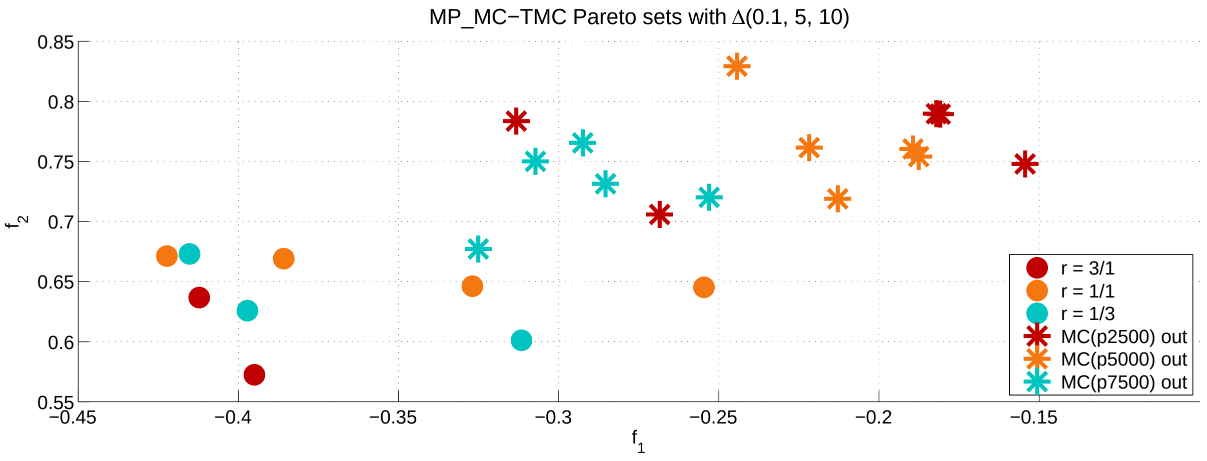
In none of the configurations evaluated thus far does the additional TMC actually provide consistently better solutions than the ones represented by the starting points. Overall, the Pareto sets obtained from **mp_MOEA/D-MC** are qualitatively less than those generated by **mp_MOEA/D-MC** for the same number of function evaluations, both in objective fitness and in magnitude.

8.2.5 mp_MC-TMC

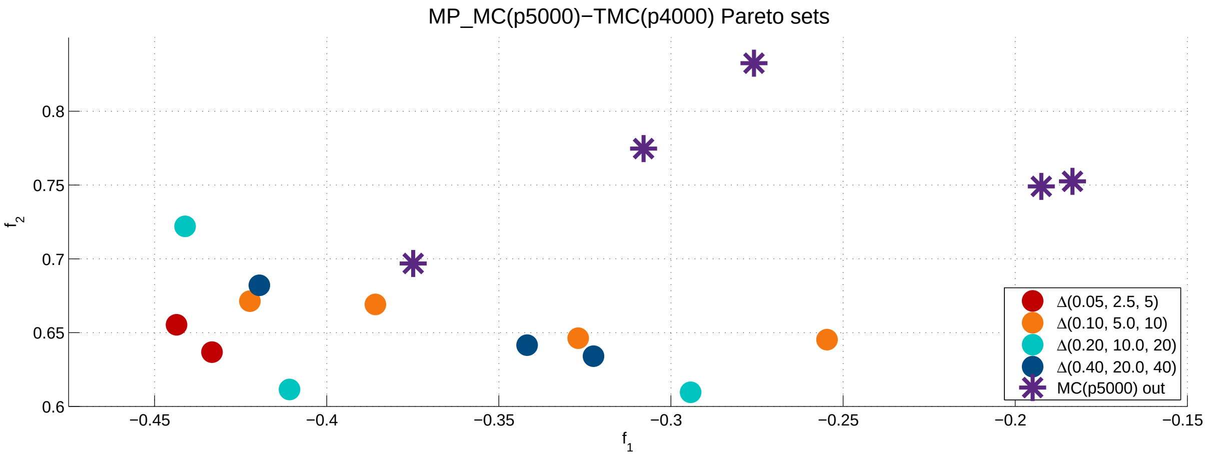
The performance of the **mp_MC-TMC**-method depends on the Monte Carlo population size $n_{\mathrm{p,MC}}$, the number of decision vectors n_{dv}, the components of the delta-vector $\mathbf{\Delta x} = [\Delta\hat{\mathbf{E}}, \Delta\boldsymbol{\alpha}, \Delta\boldsymbol{\sigma}]$, and the targeted Monte Carlo population size $n_{\mathrm{p,TMC}}$. The total number of function evaluations performed during one run of **mp_MC-TMC** is:

$$n_{\mathrm{eval}} = 4n_{\mathrm{p,MC}} + n_{\mathrm{dv}}n_{\mathrm{p,TMC}} \tag{8.10}$$

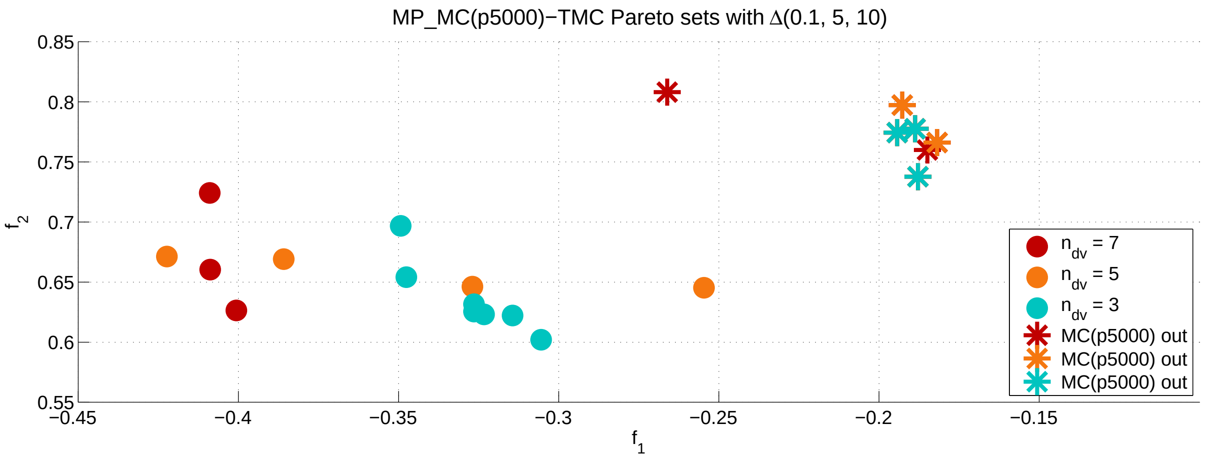
1. **Effect of r mp_MC($n_{\mathrm{p,MC}}$)-TMC($n_{\mathrm{p,TMC}}$)**-configurations are defined for the ratios $r = \{3,\ 1,\ \frac{1}{3}\}$, with constant $n_{\mathrm{dv}} = 5$ and $\mathbf{\Delta} = [0.1,\ 5°,\ 10°]$. The results of these r-configurations – i.e., **mp_MC(p2,500)-TMC(p6,000)**, **mp_MC(p5,000)-TMC(p4,000)**, and **mp_MC(p7,500)-TMC(p2,000)** – are shown in 8.8. The asterixes represent the fitness values of the decision vectors passed on to the TMC part of the method; these are the $n_{\mathrm{dv}} = 5$ closest solutions to the utopia point of the Pareto set generated by the preceding Monte Carlo simulations. The dots represent the Pareto sets generated by the corresponding TMC part.



(a) Changing r



(b) Changing $\boldsymbol{\Delta}$



(c) Changing n_{dv}

Figure 8.11: Effects on the Pareto set of changes in **mp_MC-TMC** parameters.

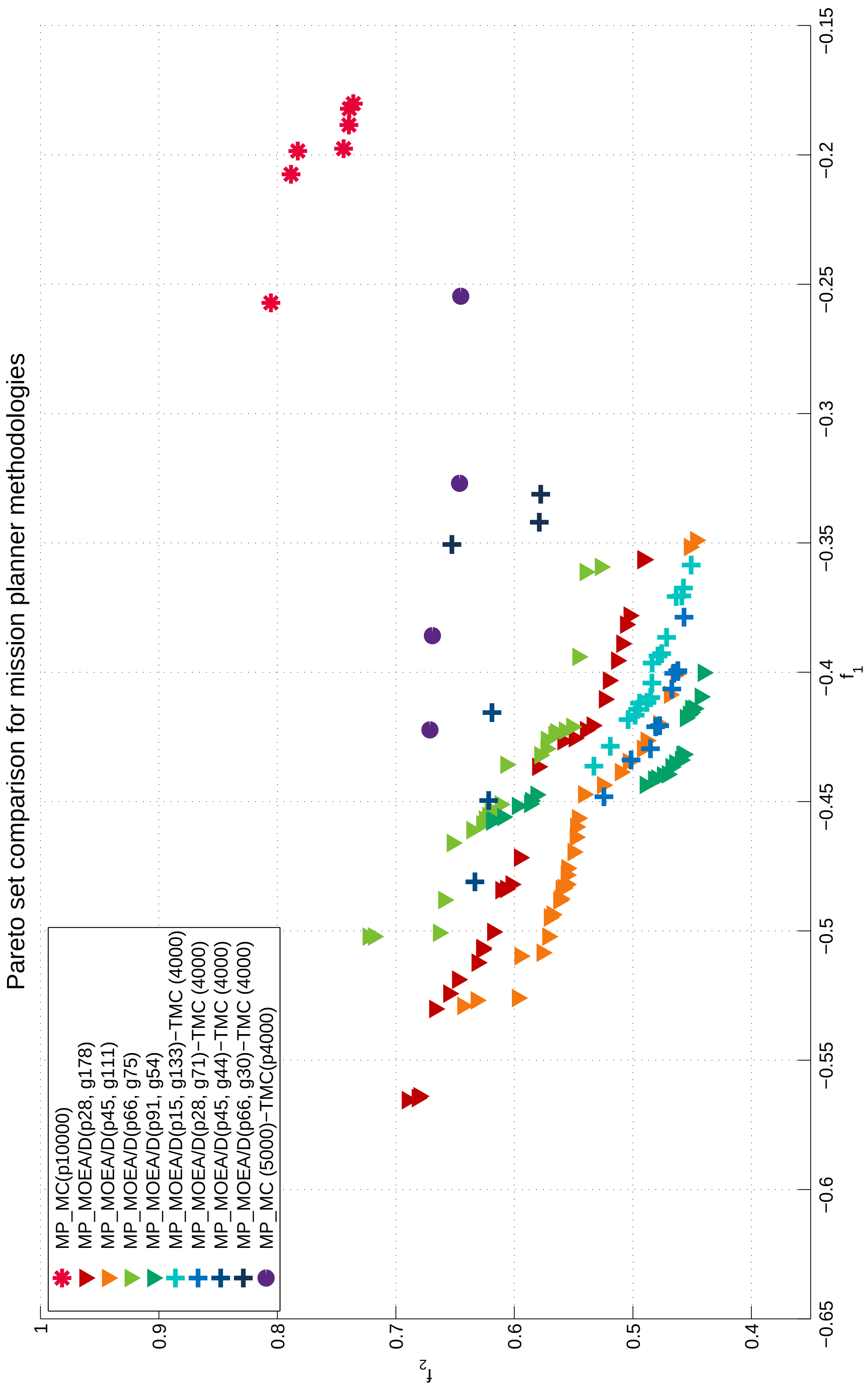


Figure 8.12: Comparison of methodology Pareto sets.

In all cases does the addition of the TMC part offer an improvement on the initial Monte Carlo results. The most improvement is achieved with higher r-ratios, i.e., more TMC evaluations, which corresponds to the observations made with regard to the effect of r on **MOEA/D-TMC**. Also consistent with the **MOEA/D-TMC** is the effect of the size of $\mathbf{\Delta}$: the smaller $\mathbf{\Delta}$ is for an equal number of TMC-evaluations, the greater the probability of finding a better solution than the one provided by the first part of the method.

2. **Effect of $\mathbf{\Delta}$** Three test cases in which $r = 1$ and $n_{\text{dv}} = 5$ were kept constant and $\mathbf{\Delta}$ was varied were investigated; the results are shown in Figure 8.11(b). The observations correspond to those made for the same test for **MOEA/D-TMC**: taking $\mathbf{\Delta}$ smaller gives better results as it increases the probability of finding a good solution within the targeted solution space. It should be noted that this effect does not continue indefinitely; taking $\mathbf{\Delta}$ too small is detrimental to the quality of the solution it does not leave enough opportunity for TMC to find a good solution is the solution space is too constrained.
3. **Effect of $\boldsymbol{n}_{\mathbf{dv}}$** To observe the effect of changing the number of decision vectors $n_{\text{dv}} = \{3,\ 5,\ 7\}$, configurations are defined with $r = 1$ and $\mathbf{\Delta} = [0.1,\ 5°,\ 10°]$ for $n_{\text{p,MC}} = 5,000$. The Pareto sets of these configurations with corresponding $n_{\text{p,TMC}}$'s are shown in 8.11(c). The results are fairly equivalent, though for the cases tested better results were achieved for a higher n_{dv}.

8.2.6 Tradeoff

The four methodologies **mp_MC**, **mp_MOEA/D**, **mp_MC-TMC** and **mp_MOEA/D-TMC** were evaluated on their relative merits to select a single one of them for further testing and development. The following selection criteria were taken into account:

Quality of results The output must represent feasible and smooth trajectories that adhere to both boundary and operational constraints. Sudden changes or jumps in control parameters or the vehicle state are not acceptable.

Number of results The number of feasible trajectories generated by a method is defined as the trajectories from the final Pareto set that adhere to both boundary and operational constraints. The idea is that the user can choose any trajectory from this set that meets their specific preferences regarding the mission; a large amount of possibilities to choose from is preferred.

Variety of results Preference is given to an output representing a wide array of solutions, i.e., a broad and evenly-spaced Pareto set.

Computational speed/ efficiency While the mission planner is developed as a preliminary mission planner and therefore almost-instantaneous answers are not of the essence, efficiency is always strived for.

Ease of implementation All else being equal, preference is given to the programmatically simpler algorithm.

Evaluated cases

To fairly evaluate the methodologies' performances with respect to each other, the basis of comparison is taken to be the result in terms of the Pareto set achieved after an equal number of function evaluations (i.e., calculations of the objective function). The baseline is defined at $n_{\text{eval}} = 40,000$. Table 8.2 shows the cases evaluated in the tradeoff. A number of computationally equivalent cases were chosen to evaluate in the tradeoff; these are given in Table 8.2. The dual-approach methods are all defined on the basis of $r = 1/1$, $n_{\text{dv}} = 5$, and $[\Delta\hat{E}, \Delta\alpha, \Delta\sigma]^T = [0.1, 5.0°, 10.0°]^T$.

Conclusions

The Pareto sets for these evaluated cases in terms of the primary objectives f_1 and f_2 (as defined in Equation 7.29) are shown in Figure 8.12. This figure indicates the presence of very clear and diverse Pareto fronts being obtainable in terms of f_1 (relating to the range) and f_2 (relating to the heat load). The third objective, f_3, which is related to the attitude rate constraints, was found to have a zero-value for all member of the Pareto sets determined by all four methodologies (f_3 is the summation of the

Table 8.2: Mission planner methodology tradeoff: evaluated cases.

Method	**Parameters**				
	$n_{\text{p,MC}}$	$n_{\text{p,TMC}}$	$n_{\text{p,MOEA/D}}$	n_{g}	n_{dv}
mp_MC	10,000				
mp_MOEA/D			15	133	
			28	71	
			45	44	
			66	30	
mp_MC-TMC	5,000	4,000			5
mp_MOEA/D-TMC		4,000	28	178	
		4,000	45	111	
		4,000	66	75	
		4,000	91	54	

attitude rate penalty functions, which only have a nonzero value when their respective constraints are exceeded.); the Pareto fronts in relation to f_3 are therefore not shown.

The methodologies' individual performances in terms of the tradeoff criteria are shown in Table 8.3:

1. Quality of results
2. Variety of results
3. Number of results
4. Computational speed/ efficiency
5. Ease of implementation

It becomes apparent that the methods that make use of the MOEA/D-algorithm provide better performance than the methods that rely on heuristics only. A surprising outcome is the degree to which **mp_MOEA/D** outperforms the other methods. It was expected that the staggered approach of **mp_-MOEA/D-TMC** would have the best performance due to combining the strength of the MOEA/D-algorithm with the heuristic Monte Carlo, allowing for the possibility of locating a slightly better optimum if the MOEA/D-algorithm happens to have converged on a local minimum; this was however not the case in comparison to **mp_MOEA/D**. In general, it can be concluded that for methods involving the MOEA/D-algorithm, function evaluations allocated to the MOEA/D-algorithm always give more "bang for the buck" over the range of n_{p}–n_{g} combinations considered; this can be extrapolated to be true for the general case, assuming the population size and the number of generations are chosen realistically. While **mp_MOEA/D-TMC** results in solutions with better objective values than **mp_MC**, the size of the resulting Pareto sets is unacceptably small when larger MOEA/D populations (with less generations) are used. While **mp_MOEA/D-TMC**(p15, g133) and **mp_MOEA/D-TMC**(p28, g71) provide a definite improvement to **mp_MC**, none of its outcomes manage to outperform the single-approach **mp_MOEA/D** in any of the categories, which defeats the purpose of the existence of the **mp_MOEA/D-TMC** method entirely.

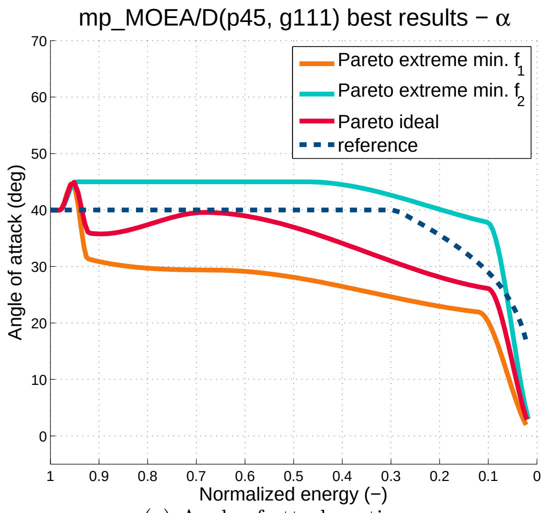
In general for the MOEA/D-algorithm, it seems that in this case the combinations of smaller population sizes with a greater amount of generations seem to result in more consistent Pareto sets. This is also the case for the staggered approach **mp_MOEA/D-TMC**. **mp_MOEA/D**(p45, g75) provides the most preferable outcome both in terms of Pareto set size, as well as variety and the quality of the results.

In the case of the second dual approach, **mp_MC-TMC**, some improvement in objective values can be achieved by staggering the search for solutions in the form of basing the solution space used in the second search on the results from the first. However, this comes at the cost of the size of the Pareto set. For the case evaluated, the resulting Pareto set is unacceptably small. However the actual benefit may be heavily dependent on the tuning of the defining parameters of the method in terms of allocation of population sizes and the number of separate solution spaces evaluated by the second part. Due to the much better performance provided by **mp_MOEA/D**, and therefore the minimal chance of approaching the same quality of results in any meaningful way, the choice was made to not evaluate the effects of tuning the **mp_MC-TMC**-parameters further.

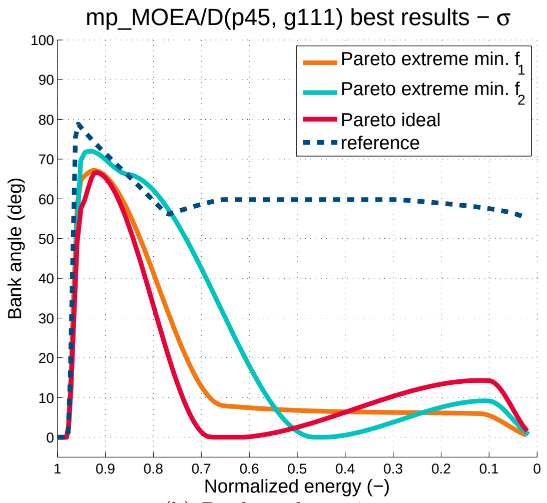
Table 8.3: Mission planner methodology tradeoff: scoring.

Method	**Tradeoff criterion**				
	1	**2**	**3**	**4**	**5**
mp_MC $n_\mathrm{p} = 10{,}000$	$--$	$-$	7	$+-$	$++$
mp_MOEA/D $n_\mathrm{p} = 21$, $n_\mathrm{g} = 178$	$+$	$++$	28	$++$	$-$
mp_MOEA/D $n_\mathrm{p} = 45$, $n_\mathrm{g} = 111$	$++$	$++$	31	$++$	$-$
mp_MOEA/D $n_\mathrm{p} = 66$, $n_\mathrm{g} = 75$	$+$	$+$	23	$++$	$-$
mp_MOEA/D $n_\mathrm{p} = 91$, $n_\mathrm{g} = 54$	$+$	$+$	24	$++$	$-$
mp_MC-TMC $n_{\mathrm{p},1} = 5{,}000$, $n_{\mathrm{p},2} = 4{,}000$	$-$	$+-$	4	$--$	$+$
mp_MOEA/D-TMC $n_{\mathrm{p},1} = 15$, $n_\mathrm{g} = 133$, $n_{\mathrm{p},2} = 4{,}000$	$+$	$-$	17	$+$	$-$
mp_MOEA/D-TMC $n_{\mathrm{p},1} = 28$, $n_\mathrm{g} = 71$, $n_{\mathrm{p},2} = 4{,}000$	$+$	$-$	10	$+$	$-$
mp_MOEA/D-TMC $n_{\mathrm{p},1} = 45$, $n_\mathrm{g} = 44$, $n_{\mathrm{p},2} = 4{,}000$	$+-$	$-$	3	$+$	$-$
mp_MOEA/D-TMC $n_{\mathrm{p},1} = 66$, $n_\mathrm{g} = 30$, $n_{\mathrm{p},2} = 4{,}000$	$-$	$-$	3	$+$	$-$

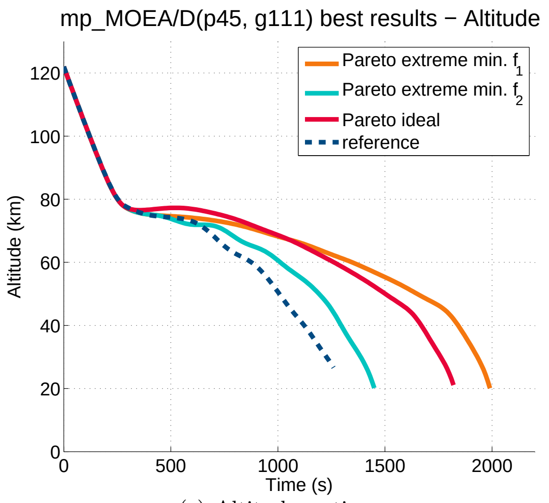
The choice for continuing with **mp_MOEA/D** is therefore self-evident; specifically, **mp_MOEA/D**(p45, g111) will be used in the remainder of this text to give an indication of the mission planner outputs. Note that the method's low "ease of implementation" score in Table 8.3 is based on the complexity of the algorithm itself; applying it after integration with the PaGMO-library is simple as PaGMO algorithms are standardized quite well.



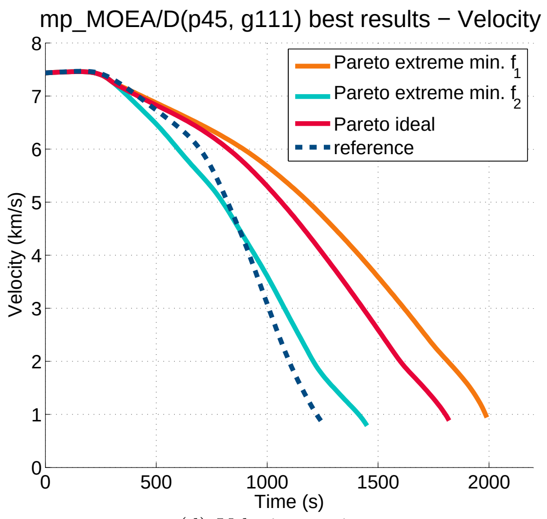
(a) Angle of attack v. time



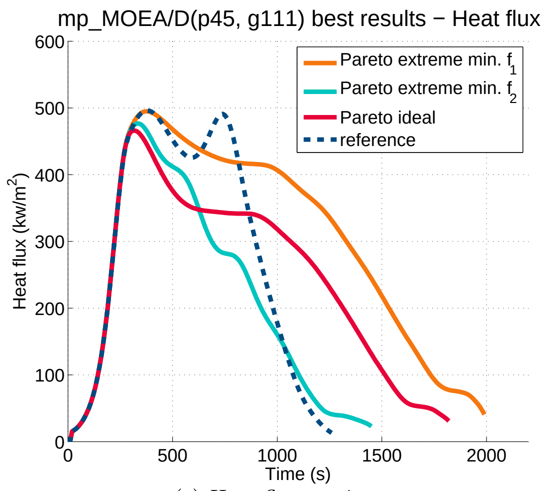
(b) Bank angle v. time



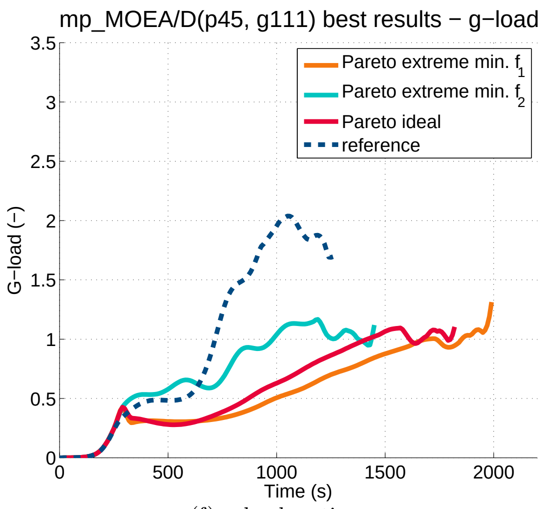
(c) Altitude v. time



(d) Velocity v. time



(e) Heat flux v. time



(f) g-load v. time

Figure 8.13: **mp_MOEA/D**(p45, g111) output trajectories – trajectory profiles.

Table 8.4: Trajectory guidance parameters for **mp_MOEA/D**(p45, g 111) output trajectories.

Trajectory	$\#\mathcal{N}$	Parameter	Values
$f_{1\,\mathrm{min}}$	6	$\hat{\boldsymbol{E}}$	[0.98, 0.95, 0.85, 0.47, 0.11, 0.10]
		$\boldsymbol{\alpha}$	[40°, 45°, 45°, 45°, 37°, 0°]
		$\boldsymbol{\sigma}$	[0°, 71°, 66°, 0°, 9°, 0°]
f^*	6	$\hat{\boldsymbol{E}}$	[0.98, 0.95, 0.92, 0.68, 0.11 0.10]
		$\boldsymbol{\alpha}$	[40°, 45°, 35°, 40°, 26°, 0°]
		$\boldsymbol{\sigma}$	[0°, 57°, 67°, 0°, 14°, 0°]
$f_{2\,\mathrm{min}}$	6	$\hat{\boldsymbol{E}}$	[0.98, 0.95, 0.92, 0.65, 0.12 0.10]
		$\boldsymbol{\alpha}$	[40°, 45°, 31°, 29°, 22°, 0°]
		$\boldsymbol{\sigma}$	[0°, 64°, 67°, 8°, 6°, 0°]
f^*	7	**t**	[0, 264, 290, 554 ,686, 924, 1319]
		$\boldsymbol{\alpha}$	[40°, 40°, 40°, 40°, 40°, 40°, 11.5°]
		$\boldsymbol{\sigma}$	[0°, 0°, 79.6°, 56.0°, 59.8°, 59.0°, 54°]

8.3 Output

Figures 8.13(a) through 8.13(f) give an indication of the output of **mp_MOEA/D**(p45, g111) in comparison to the reference trajectory. In addition the optimal trajectory f^* as determined by its proximity to the Pareto utopia point, the two Pareto extremes corresponding to $f_{1,\mathrm{min}}$ and $f_{2,\mathrm{min}}$ are plotted to indicate the breadth of the Pareto set. Again, it can be observed that the mission planner is range-dominated in its outputs; the guidance profiles shown in Figures 8.13(a) and 8.13(b) correspond to trajectories with the maximum-range objective (even the $f_{2,\mathrm{min}}$-trajectory). This is apparent from the shape of the bank angle guidance profile in comparison to the profile of the reference trajectory; the commanded bank angle in the trajectories determined by the mission planner is kept low after the initial peak to minimize deceleration, as opposed to the reference bank angle profile where a higher σ_C is maintain precisely to accomplish this deceleration. If the objective of the mission planner were specified in terms of a specific landing location with a range equal to that of the reference trajectory – while maintaining the minimum heat load objective – the guidance profile would have the same shape as the reference guidance profile and the trajectory generated by the mission planner would be near-equal.

It can be seen that the $f_{2,\mathrm{min}}$ trajectory – corresponding to the heat load objective – favors the general shape of the HORUS reference trajectory. This is to be expected as the reference trajectory is defined with a minimum heat load objective. The difference in the shape of the respective heat flux profiles belonging to the $f_{2,\mathrm{min}}$ and reference trajectories can be attributed to the penalty parameters $p_{q_c,2}$ and $p_{q_c,3}$ included in the mission planner's problem definition, due to which fluctuations in the heat flux profile are discouraged.

Table 8.4 indicates the guidance parameters of the three trajectories shown in the figures, as well as the guidance profile of the HORUS reference trajectory. All three output trajectories are solutions to the OP_6 problem, i.e., they are defined by six control nodes. In fact, the entire Pareto set as obtained by **mp_MOEA/D**(p45, g111) contains only six-node solutions. This does not mean that the best guidance profile is always defined by six nodes, however. The problem dimension scales with $3(n-2)$ where n is the number of nodes, meaning OP_6 has a problem dimension of 12. For the optimization algorithm to converge on an optimal solution for a problem defined by more nodes – i.e., with more dimensions – a larger number of iterations are required. Therefore the solutions for OP_8 and OP_{10} are qualitatively less than those of OP_6 for the specific **mp_MOEA/D**(p45, g111) configuration, and not in general; e.g., the Pareto set of **mp_MOEA/D**(p28, g178) consists of exclusively of eight-node and ten-node solutions.

The mission planner trajectories are compared to the HORUS reference trajectory based on the mission

Table 8.5: Trajectory results for **mp_MOEA/D**(p45, g 111) output trajectories.

	Objective						**Constraint**	
	X_{dr} km		X_{tot} km		Q kWs/m^2		$q_{c,\max}$ kW/m^2	$n_{g,\max}$ (-)
		Δ		Δ		Δ		
$f_{1\,\min}$	$10,313$	$+51\%$	$15,904$	$+44\%$	$5.82 \cdot 10^5$	$+60\%$	498	1.3
f^*	$9,291$	$+36\%$	$13,170$	$+19\%$	$4.41 \cdot 10^5$	$+21\%$	466	2.1
$f_{2\,\min}$	$7,115$	$+4\%$	$10,519$	-5%	$3.16 \cdot 10^5$	-14%	477	1.2
Reference	$6,832$	–	$11,053$	–	$3.66 \cdot 10^5$	–	496	1.9

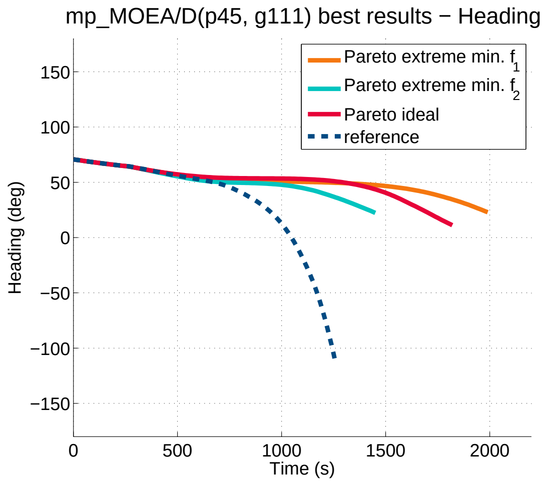


Figure 8.14: **mp_MOEA/D**(p45, g111) output trajectories – heading profiles.

objectives and constraint parameters defined in this work. Any qualitative assessments of the mission planner output with respect to the HORUS reference trajectory are only made with regard to these parameters, and disregard additional merits of the reference trajectory as these form no basis of comparison with the mission planner solutions. Therefore when a trajectory is deemed qualitatively better or worse than the reference trajectory, this statement only refers to the quality of the trajectory in terms of the parameters shown in Table 8.5.

Table 8.5 shows the results of the output trajectories in terms of the objectives range X and heat load Q, as well as the constraint parameters heat flux q_c and g-load n_g in comparison to the HORUS reference trajectory. The percentage difference between the selected output trajectories and the reference trajectory is shown for the objective parameters as well. It can be seen that the trajectories determined by the mission planner indeed provide an increase in downrange with respect to the reference trajectory. It should be noted that throughout the HORUS reference trajectory, bank angle modulation is applied to minimize the crossrange error; this is not done in the trajectories generated by the mission planner. This difference is apparent in the heading profiles of the respective trajectories, as can be seen in Figure 8.14. During the course of the reference trajectory, a "zig-zag" motion is performed as a consequence of the bank angle changing sign; due to this motion, the vehicle's total range X_{tot} is much larger than its downrange X_{dr} would suggest. The mission planner's objectives are defined in terms of maximizing the total downrange of the trajectory only, and the crossrange isn't taken into account. Bank angle modulation isn't performed in the reference trajectories (as the bank angle commands σ_{C} are absolute values), meaning this zig-zag motion does not occur either. For the range comparison between the mission output trajectories and the Horus reference trajectory to have any meaning, the total (absolute) ranges covered by each are shown as well; these are indicated as X_{tot} in Table 8.5.

The optimal trajectory, as well as the $f_{1,\min}$ trajectory, both provide an increase in total range X_{tot} with respect to the HORUS reference trajectory – 44% and 19%, respectively. However, this occurs at the cost of an increased total heat load, in the case of the $f_{1,\min}$ trajectory even 60%. Whether this increase is acceptable depends on the specific vehicle used for the mission, however it can be assumed that a 60% increase in the total heat load in comparison to the reference trajectory is too much for the trajectory to be considered feasible.

In terms of the total heat load Q, the $f_{2,\min}$ trajectory provides a 14% decrease in comparison to the reference trajectory, at the cost of 5% of the total range. This trajectory can be considered an improvement on the reference trajectory – in terms of the specific mission objectives defined in this work – as the benefit to be gained in terms of Q from flying this trajectory surpasses the (relative) cost in X. The division of the Pareto set in terms of whether a trajectory provides a relative benefit to the reference HORUS trajectory is likely situated at or around the solution closest to the utopia point. Depending on the mission objectives and constraints, a trajectory can be selected from the Pareto set from in between the solutions f^* and $f_{2,\min}$ that provides the X_{dr}-Q combination most attractive to the mission at hand. The trajectories in this set provide a percentage benefit in terms of one objective to the reference trajectory that is larger than or equal to the percentage cost in terms of the other objective[3].

The conclusion can be drawn that the mission planner is capable of generating trajectories of equivalent or higher (in terms of the mission objectives defined in this work) quality to the reference trajectory. Not all solutions in the Pareto set generated by the mission planner may be feasible, as evidenced by the large heat load incurred by trajectory f_1. The solutions situated in between the ideal point f^* and minimum f_2-point represent a better set of solutions from which a selection can be made based on the user's preference in terms of X_{dr}-Q.

8.4 Repeatability

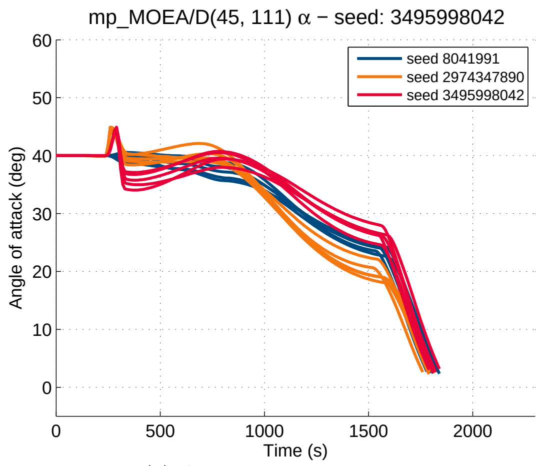
The MOEA/D-algorithm uses a PRNG both in the production of the initial population and the creation of new individuals by mutation. A PRNG is an algorithm that generates a sequence of numbers that closely resembles the properties of a truly random sequence. The precise philosophy and qualitative discussion of PRNGs is beyond the scope of this text. However, an essential property of a PRNG is its *seed*, a set of initial values upon which the sequence of pseudorandom numbers it generates is based. The PRNG used by the PaGMO library and thus also by the MOEA/D algorithm returns an unsigned integer in the $(0, 2^{32-1})$ range. It applies a PRNG from the $C++$-based Boost library called mt19937 (Watanabe, 2010).

The value of the seed directly affects the outcome of the optimization algorithm. For this purpose it is important to be aware of the specific seed used to obtain a result in order to be able to reproduce it. The value of a scientific result and the conclusions drawn from it is only as good as its reproducibility; otherwise it may have been a random stroke of luck or completely made up. The random seed used by MOEA/D is defined at initialization of the algorithm; it may be specified by the user or in absence of this specification, obtains one from the PaGMO library.

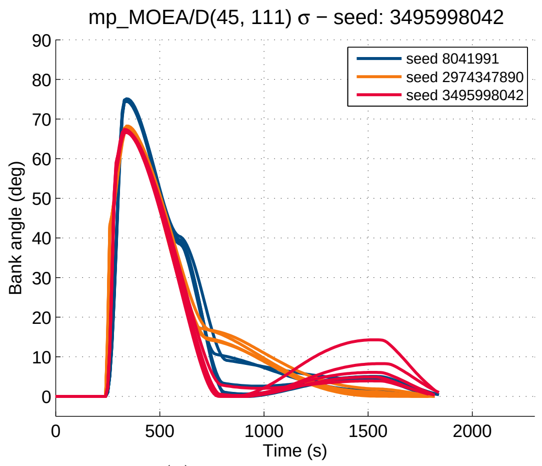
Specifying the same seed at initialization always leads to the same resulting trajectories. If no seed is provided, PaGMO initializes an algorithm with a different seed each time. During the course of the development of the mission planner a large multitude of runs and evaluations and different configurations were performed with very consistent results, demonstrating that the mission planner's reliability is independent of the value of the random seed. In this section, the trajectory profiles are shown for the obtained results in Figures 8.15 through 8.17 for the **mp_MOEA/D(p45, g111)** setup used in the previous section, initialized with three different random seeds: 1067772492, 2974347890, and 3495998042. The figures show the five best trajectories determined by the mission planner for each.

The influence of the random seed on the profiles of the optimization variables α_C and σ_C is the most noticeable, as evidenced by Figure 8.15. This difference is quite minimal, and the results of the mission planner are consistently similar independently of the random seed used to initialize the optimization.

[3]N.B. based on the total range X_{tot} of the trajectory, not the actual mission objective X_{dr}

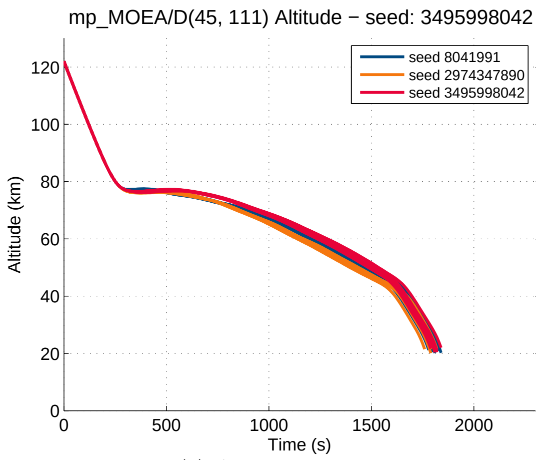


(a) Angle of attack v. time

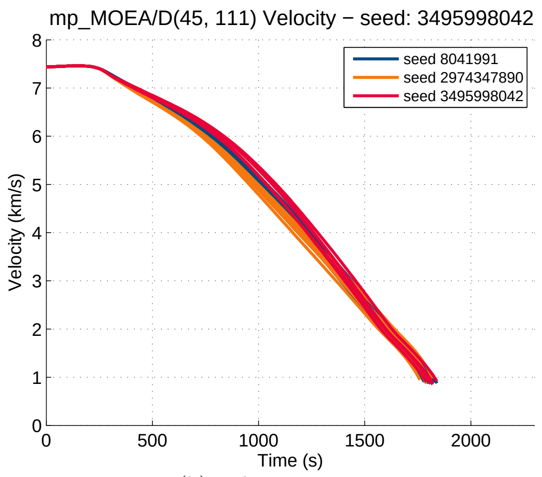


(b) Bank angle v. time

Figure 8.15: Best **mp_MOEA/D** trajectories for different random seeds – attitude profiles.

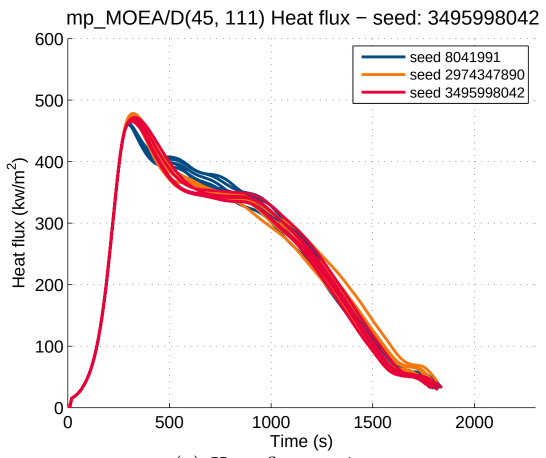


(a) Altitude v. time

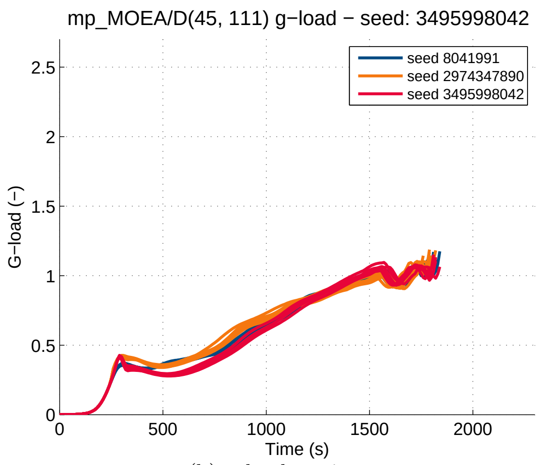


(b) Velocity v. time

Figure 8.16: Best **mp_MOEA/D** trajectories for different random seeds – altitude and velocity profiles.



(a) Heat flux v. time



(b) g-load v. time

Figure 8.17: Best **mp_MOEA/D** trajectories for different random seeds – heat flux and g-load profiles.

Table 8.6: Perturbed parameters used for sensitivity analysis.

			$-\Delta$	reference	$+\Delta$
Initial conditions					
Altitude	h_0	km	116	122	128
Velocity	V_0	km/s	7.0637	7.4355	7.8073
Flight path angle	γ_0	(deg)	-1.36	-1.43	-1.50
Initial attitude					
Angle of attack	α_0	(deg)	35	40	45
Bank angle	σ_0	(deg)	-	0	5, 10
Operational constraints					
Heat flux	$q_{c,\max}$	kW/m^2	450	500	550

8.5 Sensitivity Analysis

Sensitivity analysis is "the study of how the uncertainty in the output of a mathematical model or system (numerical or otherwise) can be apportioned to different sources of uncertainty in its inputs" (Saltelli and Ratto, 2008). A sensitivity analysis is also conducted on the mission planner using the same **mp_MOEA/D(p45, g11)** setup as before. Three main reasons for this analysis are:

- Ascertain the robustness of the mission planner
- Identify weaknesses of the mission planner
- Identify the influence of certain parameters on the course of the re-entry trajectory and the general re-entry problem

The nominal initial conditions as defined in Table 8.1 were perturbed parameter-for-parameter to identify specific relationships between inputs and outputs, as well as the specific causes of certain weaknesses of the mission planner that may be brought to light. In addition to perturbing the initial conditions, the main operational constraints c_{n_g} and c_{q_c} as defined in Table 7.1 were perturbed to observe the effect on the mission planner. A summary of the perturbed parameters is given in Table 8.6. The Δ perturbation magnitude for each parameter is taken to be 5% of the reference parameter value; as the bank angle is an absolute value, σ_0 is perturbed by $+0.05\sigma_0$ and $+0.10\sigma_0$ instead.

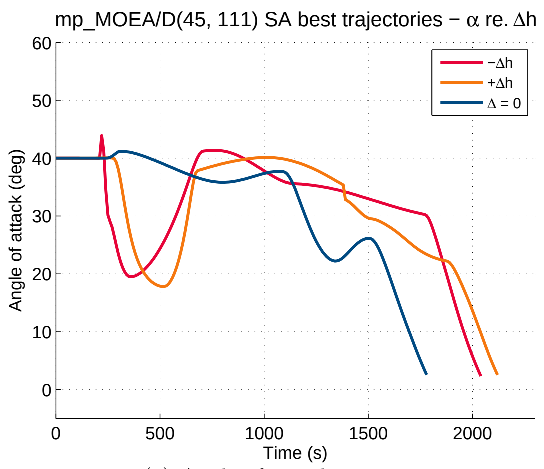
In the following section, the trajectory profiles of parameters are shown for each perturbed parameter set wherein the effect of the perturbed parameter is the most notable. The additional figures are included in Appendix C. While the g-load parameter n_g is considered defining for the mission, the g-load profiles are also delegated to the Appendix unless the c_{n_g} constraint is breached or a parameter perturbation has some other notable consequence.

8.5.1 Initial Conditions

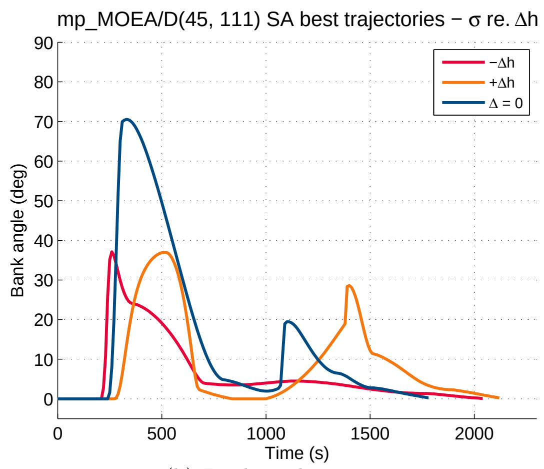
8.5.1.1 Initial states

The effects on the best solution determined by the mission planner caused by perturbation of the initial altitude h_0 by plus or minus $0.05h_0 = 6$ km are shown in Figure 8.18. The perturbation has no notable effect on the quality of the mission planner's output. Small lofts occur in the altitude progression of the trajectories corresponding to both perturbations, which lead to the slightly fluctuating heat flux profiles, as well as longer ranges. These lofts are the consequence of the smaller initial bank angle commands given in both the $+\Delta h$ and $-\Delta h$ trajectories.

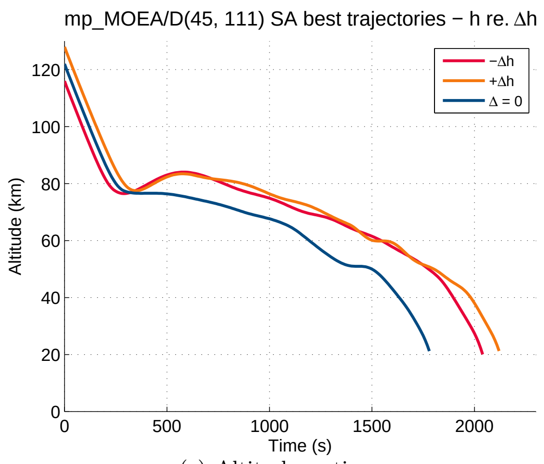
The effects of perturbing the initial velocity V_0 by plus or minus $0.05V_0 = 0.3718$ km/s are shown in Figures 8.19 and 8.20. The most notable consequence of this perturbation is the exoatmospheric skip that results from an increase of just 5% in the initial velocity V_0. The size of this skip can be mitigated by entering the atmosphere at a steeper angle γ_0; however in the case of the mission planner γ_0 is not an



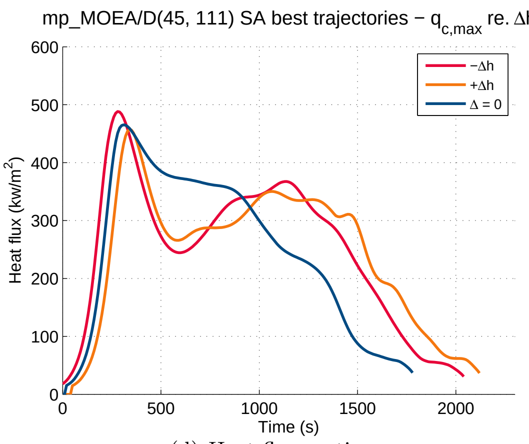
(a) Angle of attack v. time



(b) Bank angle v. time

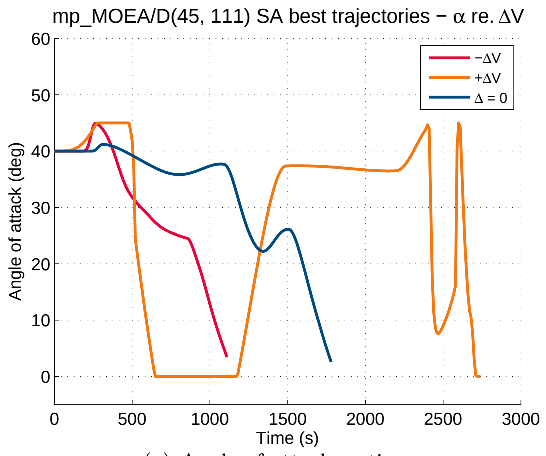


(c) Altitude v. time

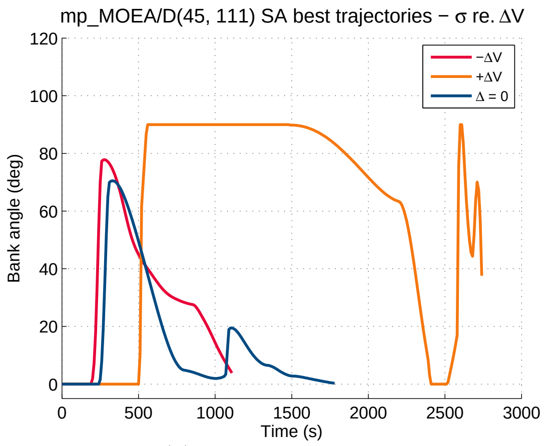


(d) Heat flux v. time

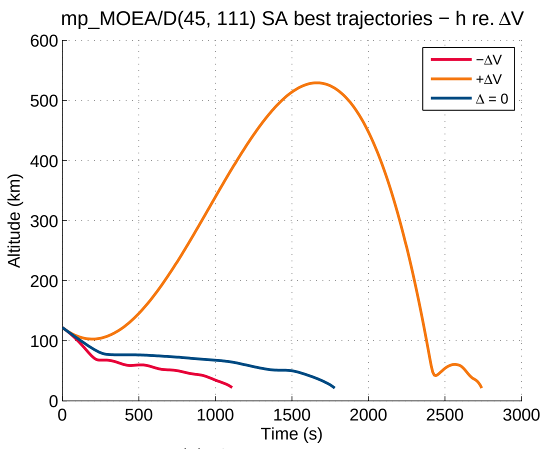
Figure 8.18: Sensitivity analysis – effect of h_0 perturbation.



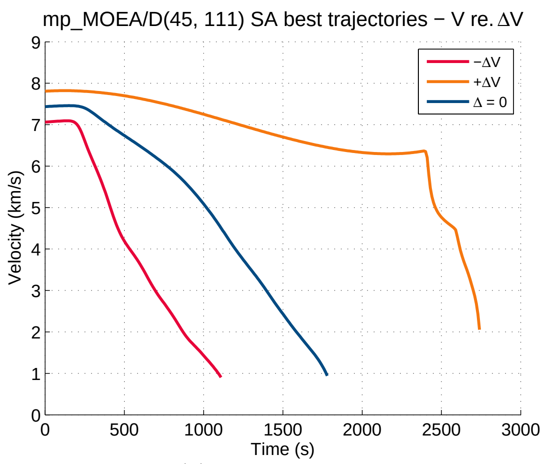
(a) Angle of attack v. time



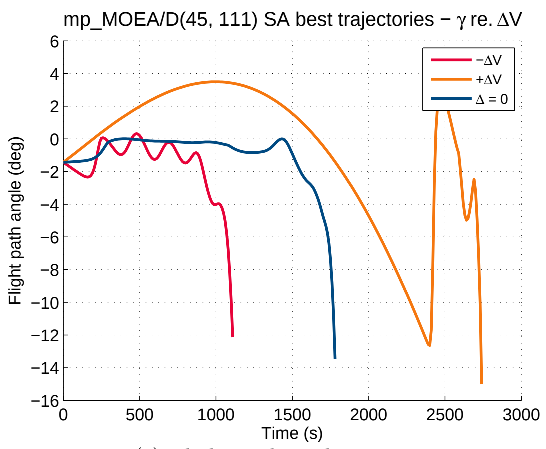
(b) Bank angle v. time



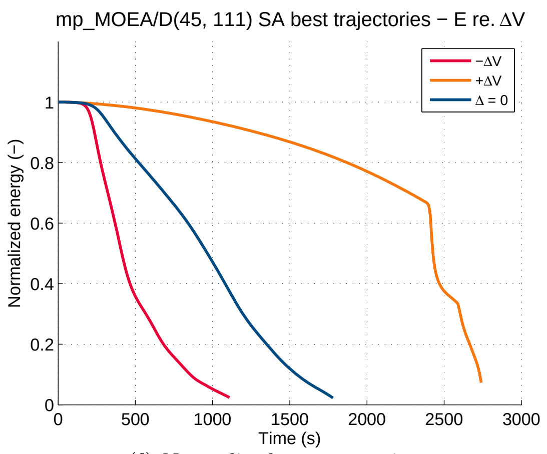
(c) Altitude v. time



(d) Velocity v. time

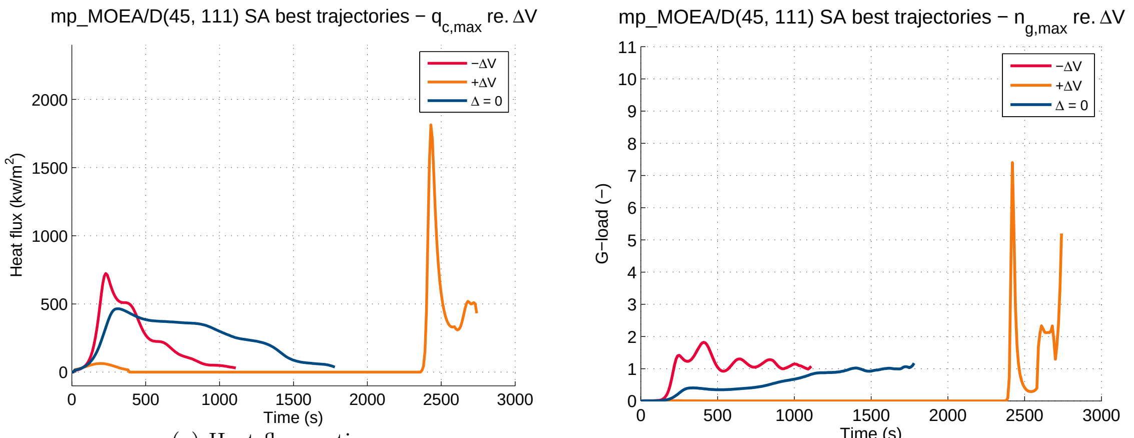


(e) Flight path angle v. time



(f) Normalized energy v. time

Figure 8.19: Sensitivity analysis – effect of V_0 perturbation.



(a) Heat flux v. time   (b) g-load v. time

Figure 8.20: Sensitivity analysis – effect of V_0 perturbation cont'd.

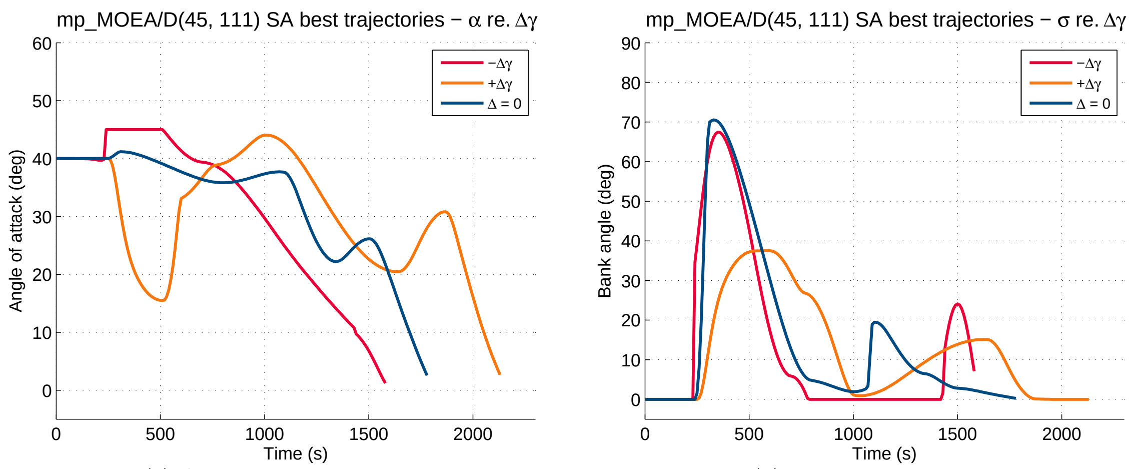
optimization variable. The mission planner has problems with finding a good trajectory in this case. The likely cause is that the mission planner is not specifically equipped to handle the exoatmospheric phase of the trajectory, wherein guidance commands are ineffective due to the lack of atmosphere. Figure 8.19(e) shows that the vehicle re-enters the atmosphere at an extremely steep angle $\gamma \approx 12°$, which the mission planner cannot recover from. The guidance profile during the first atmospheric phase during such an entry must be specified such that it its exit conditions lead to suitable entry conditions at the second entry, as the vehicle cannot be controlled during the ballistic phase. The optimization objectives do not take this possibility into account, causing the guidance system to generate futile guidance commands during the ballistic phase as well, not leaving enough control nodes after the second entry to allow for a suitable guidance profile. Likely the optimization algorithm would eventually converge on a fairly good solution, however a large number of iterations would probably be necessary before this happens. An possibility for improvement of the mission planner is to explicitly incorporate the capability of handling exoatmospheric skips. In general, it can be stated that the mission planner in its current configuration is sensitive to increases in the initial velocity.

The effects of perturbing the initial flight path angle γ_0 by plus or minus $0.05\gamma_0$ are shown in Figure 8.21. While the shape of the re-entry trajectory is generally quite sensitive to γ_0, the mission planner deals very well with this perturbation in finding a good guidance profile (Figures 8.22 and 8.23) to keep the vehicle within operational constraints while maintaining a smooth heat flux profile (Figure 8.21(d)) and maximizing range. It can be stated that the mission planner is not sensitive to small perturbations in the initial flight path angle.

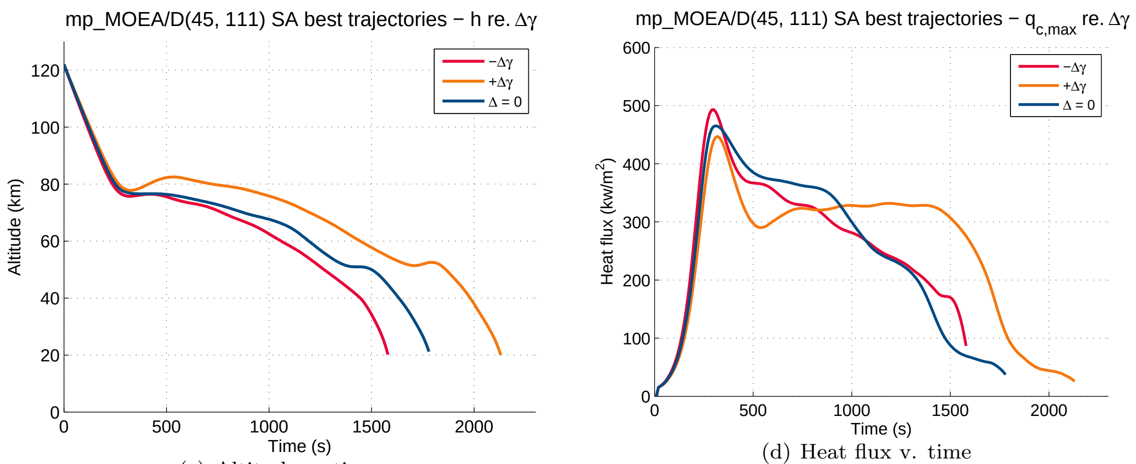
8.5.1.2 Initial attitude

In addition to the initial state, the initial attitude angles α_0 and σ_0 were also perturbed; α_0 by plus or minus 2.5° which corresponds to a perturbation of approximately 5% of its reference value, and σ_0 with plus 5° or 10° – as σ_C is an absolute value. The effects of these perturbations can be seen in Figure 8.22. The decrease in α_0 results in a tendency toward a loft in the trajectory, which is responsible for larger heat flux fluctuation and a longer range and thus a larger total heat load. The mission planner is capable of handling the α_0-perturbation and generating feasible trajectories, though it should be noted that the attitude guidance profiles are somewhat erratic in the case of the lower α_0.

The effects of perturbing the initial bank angle σ_0 are shown in Figures 8.23. The expectation was that increasing σ_0 would result in a tendency to loft, which is indeed observable for the larger perturbation of $\Delta\sigma_0 = 10°$. The consequence of this loft on the heat flux profile is similar (though much less extreme) to the effect observed in 8.20(a); altitude fluctuations lead to fluctuations in the heat flux profile. Likely a large-enough σ_0 would eventually lead to an exoatmospheric skip. In general, the mission planner is



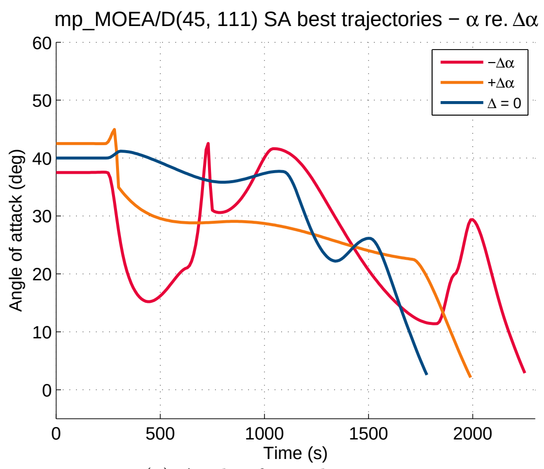
(a) Angle of attack v. time (b) Bank angle v. time



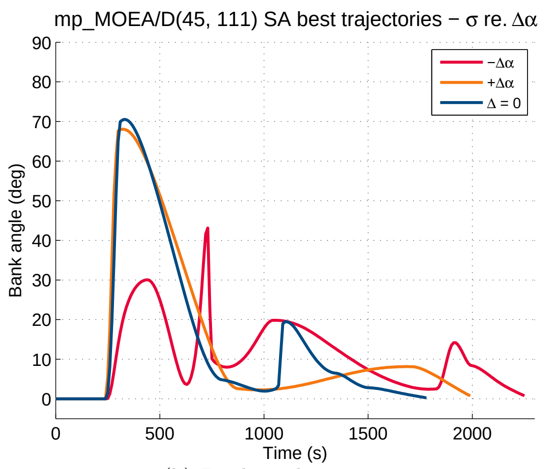
(c) Altitude v. time (d) Heat flux v. time

Figure 8.21: Sensitivity analysis – effect of γ_0 perturbation.

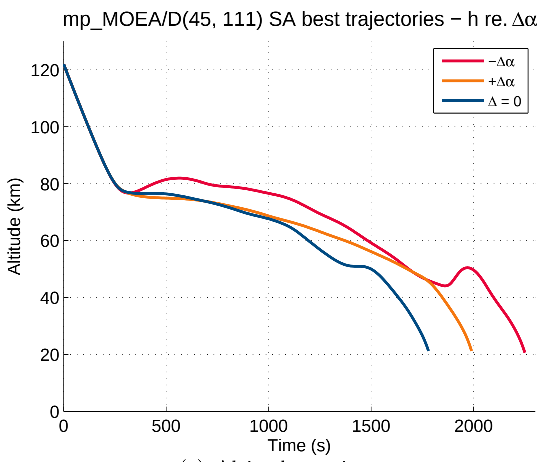
capable of handing the σ_0 perturbations.



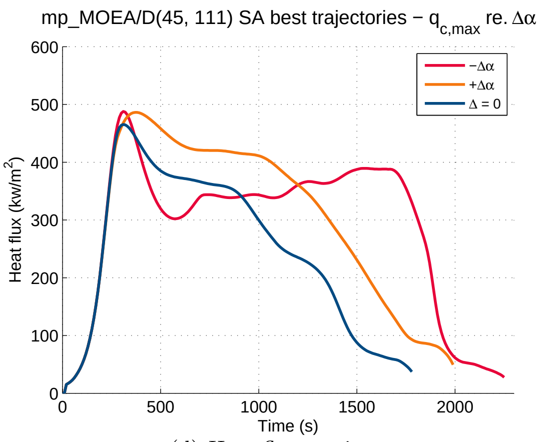
(a) Angle of attack v. time



(b) Bank angle v. time

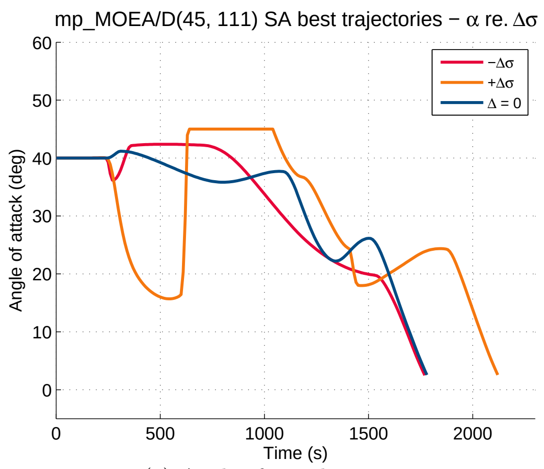


(c) Altitude v. time

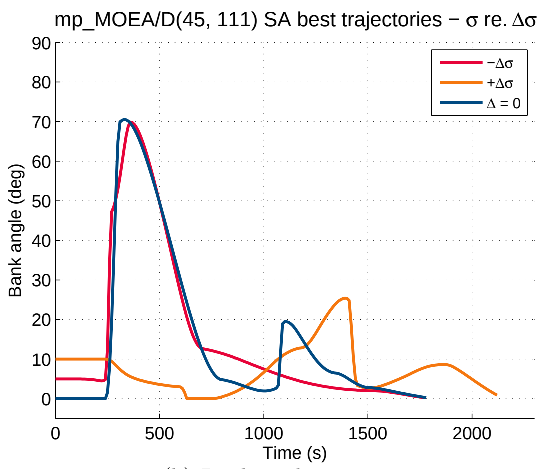


(d) Heat flux v. time

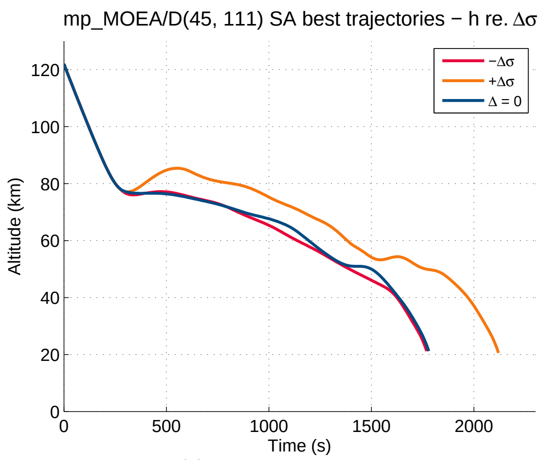
Figure 8.22: Sensitivity analysis – effect of α_0 perturbation.



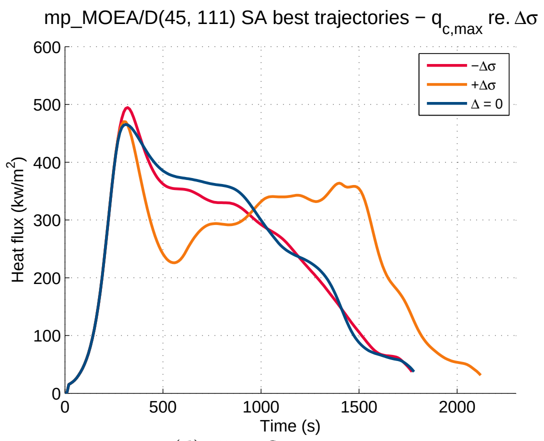
(a) Angle of attack v. time



(b) Bank angle v. time

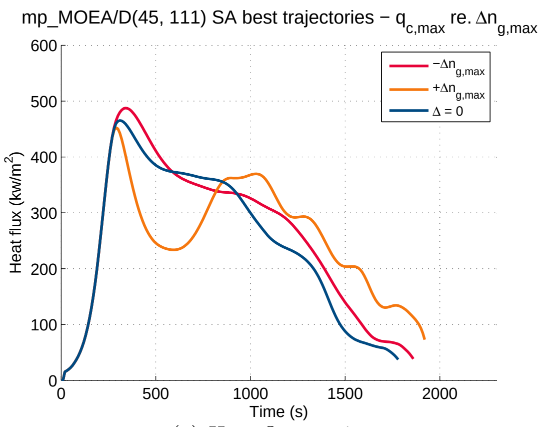


(c) Altitude v. time

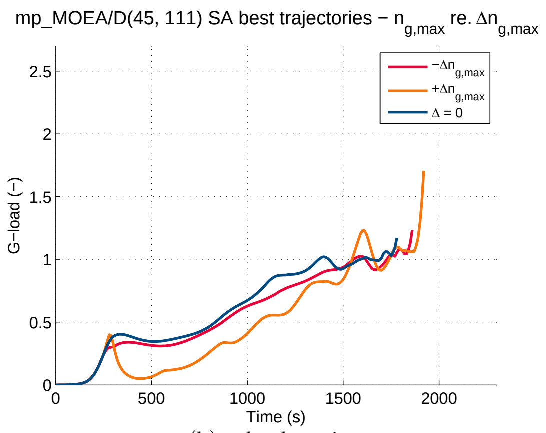


(d) Heat flux v. time

Figure 8.23: Sensitivity analysis – effect of σ_0 perturbation.

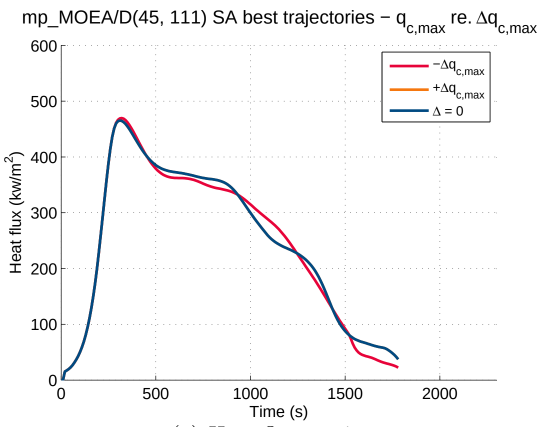


(a) Heat flux v. time

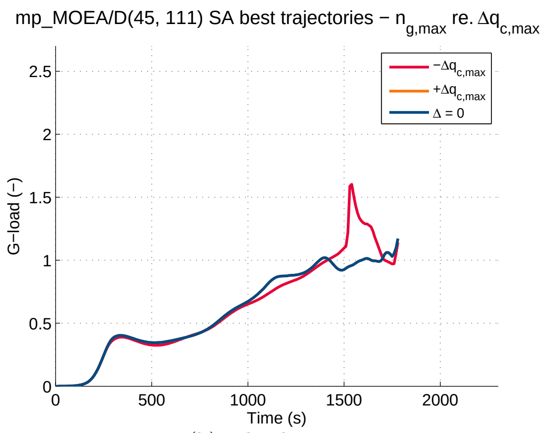


(b) g-load v. time

Figure 8.24: Sensitivity analysis – effect of c_{n_g} perturbation.



(a) Heat flux v. time



(b) g-load v. time

Figure 8.25: Sensitivity analysis – effect of c_{q_c} perturbation.

Chapter 9

Conclusions and Recommendations

The focus of this thesis is the development of a tool for the purpose of the design-time production of optimal trajectories given a specific set of requirements.

The main question of this thesis work is formulated as:

To what extent can optimal re-entry trajectories be developed in the design-phase of mission development for a winged entry vehicle that provide a maximum-range capability under the objective of minimizing the heat load and adhering to operational constraints?

The approach to answering this question consisted of four identifiable separate segments:

1. Development of the re-entry simulator
2. Design of the guidance algorithm
3. Development of the mission planner
4. Mission planner testing

Before anything else, the mission planner relies on its capability of accurately and efficiently simulating re-entry trajectories. The re-entry simulator uses the HORUS reference vehicle model from Mooij (1995). The HORUS reference vehicle is a Space Shuttle-like winged (i.e., lifting) vehicle capable of both gliding and skipping flight. Detailed and verified aerodynamic information is available for the vehicle model from Mooij (1995), as well as simulation data for a reference trajectory developed by Mooij (1998); this data was used to verify the re-entry simulator.

The vehicle performs an unpowered re-entry, meaning all the forces and moments acting on the vehicle throughout re-entry are a direct consequence of its instantaneous environment. The forces and moments acting on the vehicle are either gravitational or aerodynamic in origin; these forces are computed according to the gravitational field and atmospheric models used. The re-entry simulator makes use of a central gravity field with an additional J_2-term to model the latitudinal gravitational perturbations. The instantaneous atmospheric properties throughout the re-entry trajectory are computed using an analytical US76 atmospheric model. Finally, the Earth is taken to have a spherical shape.

The HORUS Re-Entry Simulator was developed with the specific goal of top-level integration with the mission planner in mind. During its development, steps were taken continuously to verify the reliability of both individual code modules as well as the overall output produced by the simulator. These global verification steps were applied from the ground up throughout the entire course of the simulator's development, first relying on highly simplified models representing the Apollo capsule, and with each iteration adding a level of complexity. The final result is a simulator capable of simulating the trajectory of HORUS with "three-and-a-half" degrees of freedom, i.e., the translational motion with the addition of keeping the vehicle in an aerodynamically trimmed state.

The vehicle is steered by modulating its attitude in terms of its angle of attack and bank angle. A change in attitude results in a change in the direction and magnitude of the resultant aerodynamic force acting on the vehicle, causing the vehicle to alter its trajectory. The vehicle's attitude throughout the course of the trajectory is defined by a guidance profile that relates the commanded angle of attack α_{C} and bank angle σ_{C} to an independent parameter that indicates the vehicle's position along its trajectory. This independent parameter is the normalized total specific energy of the vehicle $\hat{E}$, which has value 1 at the entry point and 0 when the vehicle has come to a standstill on the Earth surface. The benefit of $\hat{E}$ is that not only are its limit values independent of the trajectory, but also that it is monotonically decreasing throughout the course of re-entry; this allows for an unambiguous definition of the guidance profile. The guidance profile is defined in terms of a number of control nodes, at which the guidance angles α_{C} and σ_{C} are specifically related to a value of $\hat{E}$. Between these control nodes, the values of the attitude angles are obtained by Hermite spline interpolation of the attitude values at the two surrounding nodes. The guidance system also computes the corresponding body flap deflection angle δ_b required for the vehicle to maintain trim stability for the commanded α_{C}. The vehicle's control system is assumed to be ideal, i.e., guidance commands are carried out by the vehicle exactly as specified by the guidance system.

The simulator computes the vehicle's trajectory by integration of the initial state, which is defined by its inertial position and velocity components. The state derivative of the vehicle is computed based on the summation of the gravitational and aerodynamic forces it incurs as a result of its current state; the commanded attitude angles α_{C} and σ_{C} determine the definition of the aerodynamic force vector in the inertial frame. In addition to the state variables, the vehicle's horizontal velocity as well as the heat flux are integrated to obtain the total downrange and heat load at the end of the trajectory.

The purpose of the mission planner is to develop trajectories that:

- Keep the vehicle within its operational constraints
- Minimize the heat load
- Provide the largest possible range

The objectives of minimum heat load and maximum range are conflicting. Trajectories with minimum heat load requirements are generally short in duration with smooth heat flux profiles where the heat flux is maintained close to its constraint value. The total heat load is minimized by keeping the duration of the re-entry as short as possible. This, however, is in direct opposition to the objective of maximizing range, where keeping the vehicle aloft for as long as possible is beneficial. For extended-range trajectories, the altitude profile may contain a number of lofts or skips of the vehicle; this is beneficial for the total downrange, but results in an irregular heat flux profile. Large temperature gradients can cause significant thermal stresses, which are detrimental to the vehicle;'s TPS

The course of an entire trajectory can be specified by its guidance profile. The mission planner develops trajectories by specifying the parameters $\hat{E}$, α_{C}, and σ_{C} in this guidance profile. Guidance profiles specified by four, six, eight and ten nodes are generated (denoted OP_4, OP_6, OP_8 and OP_{10}, respectively), of which the first and last node are predefined. The output must be a qualitatively good trajectory with regard to both the mission's objectives and constraints. The mission objectives are the total downrange X_{dr} (to maximize) and the total heat load Q (to minimize), whereas the constraints are related to the heat flux q_c and $g$load, as well as the rates of change of the attitude angles. The quality of a certain trajectory in terms of these parameters is defined quantitatively by the value of the objective function, of which the values should be minimized. The objective function takes into account not only X_{dr}, Q and the constraint parameters, but also the shape of the heat flux profile in terms of the number and magnitude of any fluctuations.

The final output of the mission planner consists of a set of candidate trajectories, its Pareto set, and an optimal trajectory based on its objective values' proximity to the Pareto utopia point. This set of candidate trajectories can be generated by four separate approaches, two of which use only a single method, and two of which apply methods in tandem to obtain a set of solutions. The dual methods are based on the consideration that any solution found by either a heuristic (i.e., Monte Carlo) or deterministic (i.e., an optimization algorithm) method may represent a local optimum, and a better optimum may be located close by.

- **mp_MC** uses Monte Carlo simulations to generate solutions for each OP in tandem. The separate populations are aggregated to determine the final output.

- **mp_MOEA/D** uses the MOEA/D optimization algorithm only for each OP in tandem. The separate populations are aggregated to determine the final output.
- **mp_MOEA/D-TMC** uses the MOEA/D optimization algorithm to generate a number of candidate solutions. Targeted solution spaces are defined around each candidate solution, within which a targeted Monte Carlo simulation is performed to identify any better points.
- **mp_MC-TMC** uses Monte Carlo simulations to generate a number of candidate solutions. Targeted solution spaces are defined around each candidate solution, within which a targeted Monte Carlo simulation is performed to identify any better points.

Each approach is defined by a number of parameters; the influence of these parameters on the respective outcomes was evaluated to determine the best configuration for each separate method. The outputs of the methods were evaluated in a tradeoff with respect to each other, from which was concluded that **mp_MOEA/D** outperforms the other methods by a large margin. Therefore the decision was made to continue the development of the mission planner with the **mp_MOEA/D** method.

The mission planner in its final configuration was evaluated in three gradations:

- The quality of results with respect to the reference trajectory
- The repeatability of obtained results and sensitivity to random seeds
- Its sensitivity to perturbations in the initial condition

Quality of results The trajectories obtained by the mission planner given the reference initial condition were evaluated for the configuration **mp-MOEA/D(p45, g111)**. The optimal trajectory as well the two extremes of the Pareto set were shown and compared to the HORUS reference trajectory to indicate the breadth of possible solutions. From the progression of the defining trajectory parameters throughout these trajectories can be concluded that the results produced by the mission planner in this configuration are qualitatively good and meet expectations with regard to both objective performance as well as adherence to constraints. The **mp_MOEA/D** outputs were further evaluated in comparison to the HORUS reference trajectory:

- Minimum heat load: The trajectory corresponding to the Pareto extreme with the lowest heat load objective is similar to the reference trajectory. It has a slightly longer duration and a lower total heat load due to a differently-shaped heat flux profile. The reference trajectory's heat flux profile has two small peaks, whereas the minimum-heat-load trajectory determined by the mission planner has one initial peak of the same magnitude, after which the heat flux consistently decreases throughout the remainder of the trajectory. The definition of the mission planner's objective function takes into account the number and magnitude of fluctuations in the heat flux profile as parameters to minimize, causing the preference for smooth heat flux profiles. This minimum-heat-load trajectory results in a 14% decrease in the total heat load in comparison to the reference trajectory at the cost of 5% of the total range flown. Taking only the optimization objective of downrange into account, the minimum heat load trajectory generated by the mission planner actually provides a 5% increase to the reference downrange.
- Optimal: The optimal trajectory as determined by the mission planner has a significantly longer duration than the reference trajectory. The consequence of this longer duration is a 20% increase in the total range flown with respect to the reference trajectory. This increase in duration however comes at the cost of a 20% increase in the total heat load. Whether this is acceptable depends on the vehicle specifications and the mission requirements.
- Maximum range: The maximum range trajectory determined by the mission planner provides a 45% increase in the total range flown at the cost of an increase of 60% in the total heat load. If such a range is required, the TPS of the vehicle needs to be designed to withstand this increased heat load.

It seems that the Pareto set is split at the optimal point in terms of whether a solution provides a benefit over the reference trajectory (i.e., percentage benefit to one objective is greater than the percentage cost to the other) or not. The mission planner outputs corresponding to this part of the Pareto set provide a relative improvement on the reference trajectory in terms of either one or the other mission objective (i.e., minimum heat or maximum range). The half of the Pareto set corresponding to better range-objective values are characterized by an increased heat load due to the duration of the re-entry. Should such a range be required, the increased heat load needs to be taken into account in the design of the vehicle's TPS.

In general, it can be concluded that the mission planner is capable of generating a range of trajectories with good downrange and heat load properties; the best trajectory within this range depends on the specific priorities of the mission. In comparison to the reference trajectory, the mission planner results offer an improvement either in terms of a lower heat load or an increase in total range, where the cost to one objective is decidedly smaller than the benefit generated for the other.

Repeatability The repeatability of a result is as important as the result itself in terms of solidifying its scientific legitimacy. To this order the random seeds used by the MOEA/D-algorithm during each optimization run were obtained. The repeatability of the results generated by the mission planner was established. In addition, the results were shown to be consistent in quality for initializations with different random seeds.

Sensitivity The sensitivity of the mission planner was analyzed in terms of parameters relating to the initial condition – both the state and attitude – of the vehicle. The mission planner offers a good performance in the handling of perturbations to the initial condition, with the exception of an increase in the initial velocity that resulted in an exo-atmospheric skip. Guidance commands have no effect during an exo-atmospheric phase, which means that the guidance profile should be designed such that the vehicle has exit conditions that lead to a reasonable set of entry conditions. The mission planner does not take this into account, which has two consequences: futile guidance commands are issued during the exo-atmospheric phase due to which insufficient nodes are "left over" to control the vehicle after the second entry point, and the conditions at the second entry point – specifically the flight path angle – were too extreme in the investigated trajectory for the mission planner to generate an adequate guidance profile for. It is probable that a larger number of iterations might allow for the algorithm to determine a guidance profile that corresponds to these conditions, however within the allotted amount of evaluations this was not the case. A possibility for improvement in this respect would be to specify for the mission planner to not issue any guidance commands during the exo-atmospheric phase. This would likely decrease the solution space enough for a guidance profile to be found that leads to favorable second-entry conditions.
The mission planner offers a good performance with regard to the other perturbed parameters, both relating to the initial condition as well as the initial attitude.

To answer the research question: To what extent can optimal re-entry trajectories be developed in the design-time phase of mission development for a winged entry vehicle that provide a maximum-range capability under the objective of minimizing the heat load and adhering to operational constraints? – it is definitely possible to achieve extended-range capabilities for trajectories defined by minimum heat load. Despite the inherently conflicting objectives of minimum heat load and maximum range, a good range of solutions are generated by the mission planner that offer a varied selection in terms of trajectory downrange and heat load properties. Long re-entry durations generally lead to a relatively higher heat load simply because of the time spent incurring heat flux, though all solutions offer good characteristics in terms of the profiles of both heat flux and the g-load. This increased heat load must be taen into account in the design of the vehicle's TPS, should such an extended-range trajectory be preferred. A subset of the mission planner output trajectories represent an improvement to the HORUS reference trajectory – in terms of the mission objectives as defined in this thesis, and disregarding any additional merits of the reference trajectory that do not form a basis for comparison. Therefore it can be concluded that the mission planner is capable of generating feasible and qualitatively good re-entry trajectories that provide a maximum range capability under the objective of minimizing the heat load.

Recommendations

Recommendations for future work are:

Lateral guidance Inclusion of a lateral guidance capability by bank angle modulation would allow for the vehicle to be steered to a specific location – or in the general case of range-maximization without a specific destination – fly a trajectory with a defined heading profile. The guidance profile as generated by the optimization algorithm can in this case still be defined in terms of the absolute bank angle only, as bank angle modulation is based on changing the sign – and not the magnitude – of the bank angle to adjust the vehicle's heading.

Skip capability Including the capability of managing an exo-atmospheric skip into the mission planner relates to the generation of a guidance profile that leads the vehicle to a certain exit condition that results in a favorable second-entry condition. In addition, the optimization algorithm should be defined as such that guidance profiles are created wherein no guidance commands are issued during the exo-atmospheric phase, as these are a "waste" of control nodes.

Thrust The guidance system could be adjusted to provide a thrust command in addition to the attitude commands; this could be utilized to induce skips and lofts in the trajectory that result in additional range. The time of occurrence of the thrust command would then be defined as an additional optimization variable.

Tracking The mission planner could be extended with a tracking capability wherein guidance commands are issued based on following a predefined (optimal) reference trajectory. This capability can be implemented as a part of an on-board real-time mission planner that generates commands during the re-entry based on the vehicle's instantaneous state in relation to this reference trajectory.

On-board re-optimization The on-board real-time mission planner can also be supplemented with an algorithm that generates guidance commands based on an optimal trajectory computed in relation to the vehicle's instantaneous state.

Appendices

Appendix A

Transformations

A.1 Reference Frame Transformations

For the purpose of readability, '**c**' will be used as abbreviation for cosine and '**s**' for sine.

Rotating (Planetocentric) to Inertial (Planetocentric) Frame

$$\begin{aligned}\mathbf{C_{I,R}} &= \mathbf{C_3}(-\omega_{cb}t) \\ &= \begin{bmatrix} \mathbf{c}\omega_{cb}t & -\mathbf{s}\omega_{cb}t & 0 \\ \mathbf{s}\omega_{cb}t & \mathbf{c}\omega_{cb}t & 0 \\ 0 & 0 & 1 \end{bmatrix}\end{aligned} \tag{A.1}$$

Vertical to Rotating (Planetocentric) Frame

$$\begin{aligned}\mathbf{C_{R,V}} &= \mathbf{C_3}(-\tau)\mathbf{C_2}(\tfrac{\pi}{2}+\delta) \\ &= \begin{bmatrix} -\mathbf{c}\tau\mathbf{s}\tau & -\mathbf{s}\tau & -\mathbf{c}\tau\mathbf{c}\delta \\ -\mathbf{s}\tau\mathbf{s}\delta & \mathbf{c}\tau & -\mathbf{s}\tau\mathbf{c}\delta \\ \mathbf{c}\delta & 0 & -\mathbf{s}\delta \end{bmatrix}\end{aligned} \tag{A.2}$$

Trajectory to Vertical Frame

$$\begin{aligned}\mathbf{C_{V,T}} &= \mathbf{C_3}(-\chi)\mathbf{C_2}(-\gamma) \\ &= \begin{bmatrix} \mathbf{c}\chi\mathbf{c}\gamma & -\mathbf{s}\chi & \mathbf{c}\chi\mathbf{s}\gamma \\ \mathbf{s}\chi\mathbf{c}\gamma & \mathbf{c}\chi & \mathbf{s}\chi\mathbf{s}\gamma \\ -\mathbf{s}\gamma & 0 & \mathbf{c}\gamma \end{bmatrix}\end{aligned} \tag{A.3}$$

Aerodynamic to Trajectory Frame

$$\begin{aligned}\mathbf{C_{T,A}} &= \mathbf{C_1}(\sigma) \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathbf{c}\sigma & \mathbf{s}\sigma \\ 0 & -\mathbf{s}\sigma & \mathbf{c}\sigma \end{bmatrix}\end{aligned} \tag{A.4}$$

Body to Aerodynamic Frame

$$\begin{aligned}\mathbf{C_{A,B}} &= \mathbf{C_3}(\beta)\mathbf{C_2}(-\alpha)\\ &= \begin{bmatrix} \mathbf{c}\alpha\mathbf{c}\beta & \mathbf{s}\beta & \mathbf{s}\alpha\mathbf{c}\beta \\ -\mathbf{c}\alpha\mathbf{s}\beta & \mathbf{c}\beta & -\mathbf{s}\alpha\mathbf{s}\beta \\ -\mathbf{s}\alpha & 0 & \mathbf{c}\alpha \end{bmatrix}\end{aligned} \tag{A.5}$$

Aerodynamic to Vertical Frame

$$\begin{aligned}\mathbf{C_{V,A}} &= \mathbf{C_{V,T}}\mathbf{C_{T,A}}\\ &= \mathbf{C_3}(-\chi)\mathbf{C_2}(-\gamma)\mathbf{C_1}(-\sigma)\end{aligned} \tag{A.6}$$

Vertical to Inertial (Planetocentric) Frame

$$\begin{aligned}\mathbf{C_{I,V}} &= \mathbf{C_{I,R}}\mathbf{C_{R,V}}\\ &= \mathbf{C_3}(-\omega_{cb}t)\mathbf{C_3}(-\tau)\mathbf{C_2}(\tfrac{\pi}{2}+\delta) = \mathbf{C_3}(-\bar{\tau})\mathbf{C_2}(\tfrac{\pi}{2}+\delta)\end{aligned} \tag{A.7}$$

defining the celestial longitude $\bar{\tau} = \tau + \omega_{cb}t$.

Aerodynamic to Rotating (Planetocentric) Frame

$$\begin{aligned}\mathbf{C_{R,A}} &= \mathbf{C_{R,V}}\mathbf{C_{V,A}}\\ &= \mathbf{C_3}(-\tau)\mathbf{C_2}(\tfrac{\pi}{2}+\delta)\mathbf{C_3}(-\chi)\mathbf{C_2}(-\gamma)\mathbf{C_1}(\sigma)\end{aligned} \tag{A.8}$$

Body to Rotating (Planetocentric) Frame

$$\begin{aligned}\mathbf{C_{R,B}} &= \mathbf{C_{R,V}}\mathbf{C_{V,A}}\mathbf{C_{A,B}}\\ &= \mathbf{C_3}(-\tau)\mathbf{C_2}(\tfrac{\pi}{2}+\delta)\mathbf{C_3}(-\chi)\mathbf{C_2}(-\gamma)\mathbf{C_1}(\sigma)\mathbf{C_3}(\beta)\mathbf{C_2}(-\alpha)\end{aligned} \tag{A.9}$$

A.2 Coordinate Transformations

Conversions between Cartesian and spherical components both in terms of position and velocity are given in

A.2.1 Cartesian to Spherical

The relation between the Cartesian position in $\mathcal{F}_\mathbf{I}$ and $\mathcal{F}_\mathbf{R}$ is given by

$$\mathbf{X_R} = \mathbf{C_{R,I}}\mathbf{X_I} \tag{A.10}$$

The transformation matrix $\mathbf{C_{R,I}}$ is simply the inverse (and transpose) of $\mathbf{C_{I,R}}$ as defined in Equation A.1. The result is

$$\begin{aligned} x_R &= \cos(\omega_{cb}\cdot t\cdot x_I) + \sin(\omega_{cb}\cdot t\cdot y_I)\\ y_R &= -\sin(\omega_{cb}\cdot t\cdot x_I) + \cos(\omega_{cb}\cdot t\cdot y_I)\\ z_R &= z_I \end{aligned}$$

Before performing the same transformation on the inertial velocity $\mathbf{V_I}$, an extra step must be taken to account for the Earth's rotation:

$$\mathbf{V_R} = \mathbf{C_{R,I}}\left(\mathbf{V_I} - \boldsymbol{\omega}_\mathbf{cb}\times\mathbf{r_I}\right) \tag{A.11}$$

The Cartesian position (as defined in $\mathcal{F}_\mathbf{R}$) can then be converted to the spherical position using

$$R = \sqrt{x_R^2 + y_R^2 + z_R^2} \tag{A.12a}$$

$$\tau = \arctan\left(\frac{y_R}{x_R}\right) \tag{A.12b}$$

$$\delta = \arcsin\left(\frac{z_R}{\sqrt{x_R^2 + y_R^2 + z_R^2}}\right) \tag{A.12c}$$

The spherical velocity V (which is the modulus of the groundspeed vector $\mathbf{V_G}$) is easily obtained:

$$V = \sqrt{u^2 + v^2 + w^2} \tag{A.13}$$

The flight path angle γ and heading χ are calculated by first transforming the Cartesian velocity $\mathbf{V_R}$ to the vertical frame $\mathcal{F}_\mathbf{V}$.

$$\mathbf{V_V} = \mathbf{C_{V,R}} \mathbf{V_R} \tag{A.14}$$

where $\mathbf{C_{V,R}}$ is the inverse of $\mathbf{C_{R,V}}$, given in Equation A.2. The vector sum of the components v_x and v_y is equal to the projection of V_G in the local horizontal plane. The direction angles then follow:

$$\gamma_G = \arccos\left(\frac{\sqrt{v_\delta^2 + v_\tau^2}}{V_G}\right) \tag{A.15}$$

$$\chi_G = \arctan\left(\frac{v_y}{v_x}\right) \tag{A.16}$$

A.2.2 Spherical to Cartesian

The spherical position coordinates can be converted to $\mathcal{F}_\mathbf{R}$-referenced Cartesian coordinates as follows:

$$x_R = R\cos\delta\cos\tau \tag{A.17a}$$

$$y_R = R\cos\delta\sin\tau \tag{A.17b}$$

$$z_R = R\sin\delta \tag{A.17c}$$

These equations are the inverse of the Equations A.12.

As the vector sum of the Cartesian velocity components v_x and v_y is equal to the projection of V_G in the local horizontal plane, the Cartesian velocity components follow from

$$v_x = V_G\cos\gamma_G\cos\chi_G \tag{A.18a}$$

$$v_y = V_G\cos\gamma_G\sin\chi_G \tag{A.18b}$$

$$v_z = -V_G\sin\gamma_G \tag{A.18c}$$

$\mathbf{V}_R$ is then computed using

$$\mathbf{V}_R = \mathbf{C_{R,V}} \mathbf{V_V} \tag{A.19}$$

where $\mathbf{C_{R,V}}$ is given by A.2.

Appendix B

Horus Aerodynamic Data

Table B.1: Range of vehicle parameters and aerodynamic variables.

α (°)	M (-)	δ_b (°)
0	1.2	-20
5	1.5	-10
10	2	0
15	3	10
20	5	20
25	10	30
30	20	
35		
40		
45		

Table B.2: Drag coefficient clean configuration $C_{D_0}(\alpha, M)$.

α, M	1.2	1.5	2	3	5	10	20
0	0.10	0.09	0.08	0.07	0.07	0.07	0.05
5	0.12	0.10	0.09	0.08	0.07	0.07	0.05
10	0.20	0.14	0.12	0.10	0.09	0.08	0.06
15	0.33	0.23	0.18	0.15	0.14	0.12	0.09
20	0.52	0.36	0.29	0.23	0.20	0.18	0.14
25	0.77	0.54	0.42	0.35	0.31	0.29	0.24
30	1.08	0.77	0.60	0.51	0.45	0.42	0.36
35	1.08	1.03	0.82	0.69	0.63	0.60	0.52
40	1.08	1.32	1.07	0.90	0.82	0.79	0.70
45	1.08	1.32	1.33	1.21	1.05	1.02	0.92

Table B.3: Lift coefficient clean configuration $C_{L_0}(\alpha, M)$.

α, M	1.2	1.5	2	3	5	10	20
	1.2	1.5	2	3	5	10	20
0	-0.02	-0.02	-0.03	-0.04	-0.04	-0.05	-0.05
5	0.27	0.16	0.10	0.06	0.05	0.03	0.01
10	0.56	0.36	0.26	0.19	0.14	0.12	0.07
15	0.86	0.57	0.42	0.33	0.26	0.23	0.16
20	1.17	0.79	0.60	0.48	0.40	0.36	0.28
25	1.17	1.01	0.78	0.62	0.54	0.50	0.40
30	1.17	1.21	0.94	0.77	0.68	0.63	0.53
35	1.17	1.38	1.08	0.90	0.80	0.75	0.66
40	1.17	1.38	1.20	1.01	0.91	0.86	0.77
45	1.17	1.38	1.28	1.08	0.99	0.95	0.85

Table B.4: Pitch moment coefficient clean configuration $C_{m_0}(\alpha, M)$.

α, M	1.2	1.5	2	3	5	10	20
	1.2	1.5	2	3	5	10	20
0	-0.020	-0.016	-0.013	-0.011	-0.010	-0.010	-0.007
5	-0.051	-0.028	-0.016	-0.008	-0.004	-0.001	0.001
10	-0.067	-0.033	-0.016	-0.004	0.003	0.007	0.008
15	-0.079	-0.036	-0.014	0.000	0.008	0.012	0.014
20	-0.086	-0.038	-0.013	0.003	0.012	0.016	0.017
25	-0.092	-0.038	-0.012	0.005	0.013	0.017	0.018
30	-0.095	-0.039	-0.011	0.005	0.013	0.016	0.018
35	-0.099	-0.040	-0.012	0.004	0.011	0.014	0.017
40	-0.103	-0.042	-0.014	0.002	0.009	0.010	0.014
45	-0.107	-0.045	-0.017	-0.002	0.005	0.006	0.010

Table B.5: Drag increment due to body flap $\Delta C_{D_b}(\alpha, M, \delta_b)$ for $\delta_b = -20°$.

α, M	1.2	1.5	2	3	5	10	20
	1.2	1.5	2	3	5	10	20
0	0.040	0.021	0.011	0.006	0.002	0.000	0.000
5	0.030	0.016	0.009	0.004	0.001	0.000	0.000
10	0.018	0.010	0.006	0.003	0.001	0.000	0.000
15	0.008	0.004	0.002	0.001	0.000	-0.001	-0.001
20	0.000	-0.001	-0.002	-0.002	-0.002	-0.003	-0.002
25	-0.005	-0.006	-0.006	-0.006	-0.006	-0.006	-0.004
30	-0.011	-0.011	-0.011	-0.011	-0.011	-0.011	-0.009
35	-0.018	-0.018	-0.018	-0.018	-0.018	-0.018	-0.016
40	-0.026	-0.025	-0.025	-0.025	-0.025	-0.025	-0.023
45	-0.033	-0.033	-0.033	-0.033	-0.033	-0.033	-0.031

Table B.6: Drag increment due to body flap $\Delta C_{D_b}(\alpha, M, \delta_b)$ for $\delta_b = -10°$.

α, M	1.2	1.5	2	3	5	10	20
	1.2	1.5	2	3	5	10	20
0	0.016	0.009	0.005	0.003	0.001	0.000	0.000
5	0.010	0.006	0.003	0.002	0.000	0.000	0.000
10	0.004	0.002	0.001	0.000	0.000	0.000	0.000
15	0.000	-0.001	0.000	-0.001	-0.001	-0.001	-0.001
20	-0.003	-0.003	-0.002	-0.002	-0.002	-0.003	-0.002
25	-0.005	-0.005	-0.005	-0.005	-0.005	-0.005	-0.004
30	-0.009	-0.009	-0.009	-0.009	-0.009	-0.009	-0.007
35	-0.013	-0.013	-0.013	-0.013	-0.013	-0.013	-0.011
40	-0.017	-0.017	-0.017	-0.017	-0.017	-0.017	-0.016
45	-0.021	-0.021	-0.021	-0.021	-0.021	-0.021	-0.020

Table B.7: Drag increment due to body flap $\Delta C_{D_b}(\alpha, M, \delta_b)$ for $\delta_b = 0°$.

α, M	1.2	1.5	2	3	5	10	20
	1.2	1.5	2	3	5	10	20
0	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000	0.000	0.000	0.000
10	0.000	0.000	0.000	0.000	0.000	0.000	0.000
15	0.000	0.000	0.000	0.000	0.000	0.000	0.000
20	0.000	0.000	0.000	0.000	0.000	0.000	0.000
25	0.000	0.000	0.000	0.000	0.000	0.000	0.000
30	0.000	0.000	0.000	0.000	0.000	0.000	0.000
35	0.000	0.000	0.000	0.000	0.000	0.000	0.000
40	0.000	0.000	0.000	0.000	0.000	0.000	0.000
45	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table B.8: Drag increment due to body flap $\Delta C_{D_b}(\alpha, M, \delta_b)$ for $\delta_b = 10°$.

α, M	1.2	1.5	2	3	5	10	20
	1.2	1.5	2	3	5	10	20
0	-0.005	-0.003	-0.002	-0.001	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000	0.000	0.000	0.000
10	0.016	0.002	0.002	0.002	0.002	0.002	0.002
15	0.005	0.005	0.005	0.005	0.005	0.005	0.004
20	0.009	0.009	0.009	0.009	0.009	0.009	0.008
25	0.013	0.013	0.013	0.013	0.013	0.013	0.012
30	0.018	0.018	0.017	0.017	0.017	0.017	0.016
35	0.022	0.022	0.022	0.022	0.022	0.022	0.021
40	0.026	0.026	0.026	0.026	0.026	0.026	0.025
45	0.029	0.029	0.030	0.030	0.030	0.030	0.029

Table B.9: Drag increment due to body flap $\Delta C_{D_b}(\alpha, M, \delta_b)$ for $\delta_b = 20°$.

α, M	1.2	1.5	2	3	5	10	20
	1.2	1.5	2	3	5	10	20
0	-0.003	0.000	0.000	0.001	0.002	0.002	0.002
5	0.004	0.005	0.005	0.005	0.005	0.005	0.004
10	0.002	0.011	0.011	0.011	0.011	0.011	0.009
15	0.018	0.018	0.018	0.018	0.018	0.018	0.015
20	0.027	0.027	0.027	0.027	0.027	0.027	0.024
25	0.037	0.036	0.036	0.036	0.036	0.036	0.033
30	0.046	0.045	0.045	0.045	0.045	0.045	0.043
35	0.053	0.053	0.053	0.053	0.053	0.053	0.051
40	0.060	0.060	0.060	0.060	0.060	0.060	0.059
45	0.065	0.065	0.065	0.065	0.065	0.065	0.065

Table B.10: Drag increment due to body flap $\Delta C_{D_b}(\alpha, M, \delta_b)$ for $\delta_b = 30°$.

α, M	1.2	1.5	2	3	5	10	20
	1.2	1.5	2	3	5	10	20
0	0.005	0.008	0.008	0.009	0.010	0.010	0.008
5	0.017	0.017	0.017	0.017	0.017	0.017	0.015
10	0.028	0.028	0.028	0.028	0.028	0.028	0.024
15	0.040	0.040	0.040	0.040	0.040	0.040	0.036
20	0.053	0.053	0.053	0.053	0.053	0.053	0.049
25	0.067	0.066	0.066	0.066	0.066	0.066	0.062
30	0.078	0.078	0.078	0.078	0.078	0.078	0.076
35	0.089	0.089	0.089	0.089	0.089	0.089	0.086
40	0.096	0.096	0.096	0.096	0.096	0.096	0.095
45	0.100	0.100	0.100	0.100	0.100	0.100	0.100

Table B.11: Lift increment due to body flap $\Delta C_{L_b}(\alpha, M, \delta_b)$ for $\delta_b = -20°$.

α, M	1.2	1.5	2	3	5	10	20
	1.2	1.5	2	3	5	10	20
0	-0.054	-0.032	-0.016	-0.006	-0.002	0.000	0.000
5	-0.070	-0.037	-0.021	-0.010	-0.003	-0.001	0.000
10	-0.067	-0.037	-0.022	-0.012	-0.006	-0.002	-0.001
15	-0.053	-0.032	-0.021	-0.013	-0.009	-0.006	-0.003
20	-0.037	-0.026	-0.021	-0.016	-0.014	-0.012	-0.007
25	-0.023	-0.021	-0.020	-0.020	-0.020	-0.019	-0.014
30	-0.025	-0.025	-0.025	-0.025	-0.025	-0.025	-0.021
35	-0.029	-0.029	-0.029	-0.029	-0.029	-0.029	-0.026
40	-0.031	-0.031	-0.031	-0.031	-0.031	-0.030	-0.030
45	-0.030	-0.030	-0.030	-0.030	-0.030	-0.030	-0.030

Table B.12: Lift increment due to body flap $\Delta C_{L_b}(\alpha, M, \delta_b)$ for $\delta_b = -10°$.

α, M	1.2	1.5	2	3	5	10	20
	1.2	1.5	2	3	5	10	20
0	-0.036	-0.019	-0.011	-0.005	-0.002	0.000	0.000
5	-0.039	-0.021	-0.012	-0.006	-0.003	-0.001	0.000
10	-0.033	-0.020	-0.012	-0.007	-0.004	-0.002	-0.001
15	-0.016	-0.011	-0.009	-0.007	-0.006	-0.005	-0.003
20	-0.010	-0.010	-0.010	-0.009	-0.010	-0.009	-0.007
25	-0.013	-0.013	-0.013	-0.013	-0.013	-0.013	-0.011
30	-0.016	-0.016	-0.016	-0.016	-0.016	-0.016	-0.014
35	-0.017	-0.017	-0.017	-0.017	-0.017	-0.017	-0.016
40	-0.017	-0.017	-0.017	-0.017	-0.017	-0.016	-0.017
45	-0.016	-0.016	-0.016	-0.016	-0.016	-0.016	-0.016

Table B.13: Lift increment due to body flap $\Delta C_{L_b}(\alpha, M, \delta_b)$ for $\delta_b = 0°$.

α, M	1.2	1.5	2	3	5	10	20
	1.2	1.5	2	3	5	10	20
0	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000	0.000	0.000	0.000
10	0.000	0.000	0.000	0.000	0.000	0.000	0.000
15	0.000	0.000	0.000	0.000	0.000	0.000	0.000
20	0.000	0.000	0.000	0.000	0.000	0.000	0.000
25	0.000	0.000	0.000	0.000	0.000	0.000	0.000
30	0.000	0.000	0.000	0.000	0.000	0.000	0.000
35	0.000	0.000	0.000	0.000	0.000	0.000	0.000
40	0.000	0.000	0.000	0.000	0.000	0.000	0.000
45	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table B.14: Lift increment due to body flap $\Delta C_{L_b}(\alpha, M, \delta_b)$ for $\delta_b = 10°$.

α, M	1.2	1.5	2	3	5	10	20
	1.2	1.5	2	3	5	10	20
0	0.034	0.020	0.012	0.007	0.004	0.002	0.000
5	0.017	0.012	0.010	0.008	0.006	0.005	0.002
10	0.010	0.010	0.010	0.010	0.009	0.010	0.007
15	0.013	0.013	0.013	0.014	0.013	0.013	0.011
20	0.016	0.016	0.016	0.016	0.016	0.017	0.015
25	0.017	0.018	0.017	0.018	0.018	0.018	0.016
30	0.018	0.018	0.018	0.018	0.018	0.019	0.018
35	0.017	0.018	0.017	0.018	0.017	0.018	0.018
40	0.015	0.015	0.015	0.016	0.015	0.016	0.015
45	0.011	0.011	0.011	0.011	0.011	0.011	0.012

Table B.15: Lift increment due to body flap $\Delta C_{L_b}(\alpha, M, \delta_b)$ for $\delta_b = 20°$.

α, M	1.2	1.5	2	3	5	10	20
	1.2	1.5	2	3	5	10	20
0	0.044	0.030	0.022	0.017	0.013	0.011	0.004
5	0.032	0.026	0.023	0.021	0.020	0.020	0.014
10	0.027	0.027	0.026	0.026	0.026	0.026	0.021
15	0.032	0.032	0.032	0.033	0.032	0.032	0.029
20	0.035	0.036	0.036	0.036	0.036	0.036	0.033
25	0.037	0.037	0.037	0.037	0.037	0.037	0.036
30	0.034	0.035	0.035	0.035	0.035	0.035	0.035
35	0.030	0.031	0.031	0.032	0.031	0.031	0.032
40	0.023	0.024	0.024	0.024	0.024	0.022	0.026
45	0.013	0.014	0.014	0.015	0.014	0.015	0.017

Table B.16: Lift increment due to body flap $\Delta C_{L_b}(\alpha, M, \delta_b)$ for $\delta_b = 30°$.

α, M	1.2	1.5	2	3	5	10	20
	1.2	1.5	2	3	5	10	20
0	0.060	0.046	0.039	0.033	0.030	0.028	0.021
5	0.050	0.045	0.041	0.040	0.038	0.037	0.031
10	0.046	0.047	0.046	0.046	0.046	0.046	0.041
15	0.052	0.052	0.052	0.052	0.052	0.052	0.047
20	0.053	0.053	0.053	0.053	0.053	0.054	0.052
25	0.052	0.052	0.052	0.052	0.052	0.052	0.051
30	0.046	0.046	0.046	0.046	0.046	0.046	0.047
35	0.036	0.037	0.037	0.037	0.037	0.037	0.039
40	0.024	0.024	0.024	0.024	0.024	0.023	0.027
45	0.008	0.008	0.008	0.009	0.008	0.010	0.012

Table B.17: Pitch moment increment due to body flap $\Delta C_{m_b}(\alpha, M, \delta_b)$ for $\delta_b = -20°$.

α, M	1.2	1.5	2	3	5	10	20
	1.2	1.5	2	3	5	10	20
0	0.0191	0.0094	0.0048	0.0017	0.0003	0.0000	0.0000
5	0.0179	0.0109	0.0058	0.0025	0.0007	0.0000	0.0000
10	0.0167	0.0111	0.0065	0.0034	0.0013	0.0003	0.0000
15	0.0154	0.0092	0.0060	0.0039	0.0027	0.0017	0.0007
20	0.0109	0.0077	0.0061	0.0049	0.0041	0.0037	0.0020
25	0.0073	0.0068	0.0066	0.0065	0.0065	0.0061	0.0044
30	0.0087	0.0087	0.0089	0.0089	0.0089	0.0089	0.0075
35	0.0113	0.0113	0.0113	0.0113	0.0113	0.0113	0.0099
40	0.0133	0.0133	0.0133	0.0133	0.0133	0.0133	0.0123
45	0.0152	0.0150	0.0150	0.0160	0.0150	0.0150	0.0143

Table B.18: Pitch moment increment due to body flap $\Delta C_{m_b}(\alpha, M, \delta_b)$ for $\delta_b = -10°$.

α, M	1.2	1.5	2	3	5	10	20
	1.2	1.5	2	3	5	10	20
0	0.0109	0.0056	0.0029	0.0012	0.0003	0.0000	0.0000
5	0.0118	0.0065	0.0034	0.0017	0.0007	0.0000	0.0000
10	0.0102	0.0058	0.0036	0.0020	0.0010	0.0003	0.0000
15	0.0049	0.0036	0.0025	0.0022	0.0017	0.0013	0.0007
20	0.0031	0.0031	0.0031	0.0029	0.0031	0.0031	0.0020
25	0.0044	0.0044	0.0044	0.0044	0.0044	0.0044	0.0034
30	0.0058	0.0058	0.0058	0.0058	0.0058	0.0058	0.0051
35	0.0068	0.0068	0.0068	0.0068	0.0068	0.0068	0.0061
40	0.0078	0.0078	0.0078	0.0078	0.0078	0.0078	0.0075
45	0.0087	0.0085	0.0087	0.0089	0.0085	0.0089	0.0085

Table B.19: Pitch moment increment due to body flap $\Delta C_{m_b}(\alpha, M, \delta_b)$ for $\delta_b = 0°$.

α, M	1.2	1.5	2	3	5	10	20
	1.2	1.5	2	3	5	10	20
0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
15	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
20	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
25	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
30	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
35	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
40	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
45	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table B.20: Pitch moment increment due to body flap $\Delta C_{m_b}(\alpha, M, \delta_b)$ for $\delta_b = 10°$.

α, M	1.2	1.5	2	3	5	10	20
	1.2	1.5	2	3	5	10	20
0	-0.0120	-0.0070	-0.0046	-0.0028	-0.0017	-0.0014	0.0000
5	-0.0065	-0.0046	-0.0038	-0.0029	-0.0024	-0.0024	-0.0014
10	-0.0041	-0.0038	-0.0038	-0.0038	-0.0038	-0.0038	-0.0028
15	-0.0053	-0.0051	-0.0055	-0.0053	-0.0055	-0.0055	-0.0045
20	-0.0069	-0.0069	-0.0069	-0.0069	-0.0069	-0.0069	-0.0062
25	-0.0082	-0.0082	-0.0082	-0.0082	-0.0082	-0.0082	-0.0075
30	-0.0094	-0.0094	-0.0094	-0.0094	-0.0092	-0.0096	-0.0086
35	-0.0103	-0.0103	-0.0103	-0.0103	-0.0103	-0.0103	-0.0099
40	-0.0108	-0.0108	-0.0108	-0.0110	-0.0110	-0.0110	-0.0106
45	-0.0110	-0.0111	-0.0111	-0.0113	-0.0110	-0.0110	-0.0113

Table B.21: Pitch moment increment due to body flap $\Delta C_{m_b}(\alpha, M, \delta_b)$ for $\delta_b = 20°$.

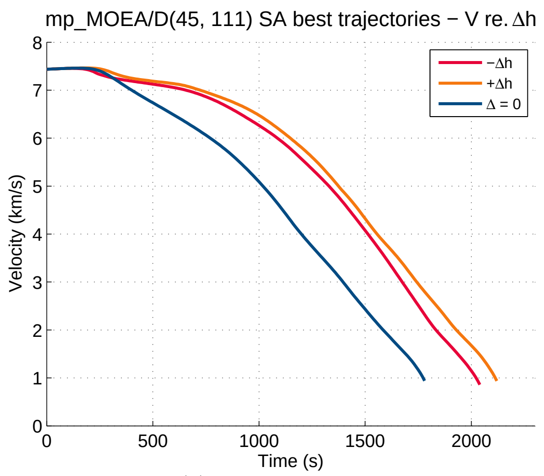
α, M	1.2	1.5	2	3	5	10	20
	1.2	1.5	2	3	5	10	20
0	-0.0157	-0.0106	-0.0084	-0.0065	-0.0051	-0.0048	-0.0031
5	-0.0116	-0.0098	-0.0089	-0.0082	-0.0075	-0.0075	-0.0055
10	-0.0108	-0.0106	-0.0106	-0.0104	-0.0106	-0.0106	-0.0089
15	-0.0133	-0.0135	-0.0135	-0.0135	-0.0133	-0.0137	-0.0120
20	-0.0163	-0.0164	-0.0164	-0.0164	-0.0164	-0.0164	-0.0151
25	-0.0185	-0.0186	-0.0185	-0.0186	-0.0185	-0.0185	-0.0174
30	-0.0202	-0.0205	-0.0204	-0.0205	-0.0205	-0.0205	-0.0195
35	-0.0215	-0.0217	-0.0215	-0.0215	-0.0215	-0.0219	-0.0212
40	-0.0224	-0.0224	-0.0224	-0.0226	-0.0222	-0.0226	-0.0222
45	-0.0217	-0.0226	-0.0224	-0.0226	-0.0222	-0.0226	-0.0229

Table B.22: Pitch moment increment due to body flap $\Delta C_{m_b}(\alpha, M, \delta_b)$ for $\delta_b = 30°$.

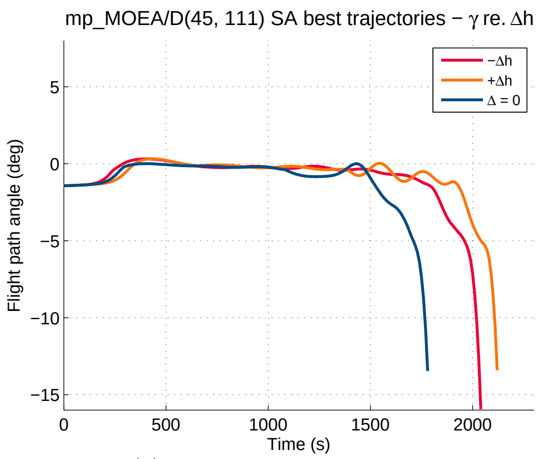
α, M	1.2	1.5	2	3	5	10	20
	1.2	1.5	2	3	5	10	20
0	-0.0217	-0.0169	-0.0147	-0.0128	-0.0116	-0.0113	-0.0086
5	-0.0195	-0.0174	-0.0166	-0.0159	-0.0151	-0.0154	-0.0127
10	-0.0195	-0.0195	-0.0195	-0.0195	-0.0192	-0.0195	-0.0168
15	-0.0234	-0.0234	-0.0236	-0.0233	-0.0236	-0.0236	-0.0215
20	-0.0267	-0.0270	-0.0267	-0.0268	-0.0267	-0.0270	-0.0253
25	-0.0292	-0.0294	-0.0294	-0.0296	-0.0294	-0.0294	-0.0284
30	-0.0315	-0.0316	-0.0315	-0.0318	-0.0315	-0.0318	-0.0308
35	-0.0325	-0.0328	-0.0325	-0.0328	-0.0325	-0.0328	-0.0325
40	-0.0330	-0.0332	-0.0330	-0.0332	-0.0332	-0.0332	-0.0335
45	-0.0321	-0.0325	-0.0323	-0.0325	-0.0321	-0.0325	-0.0332

Appendix C

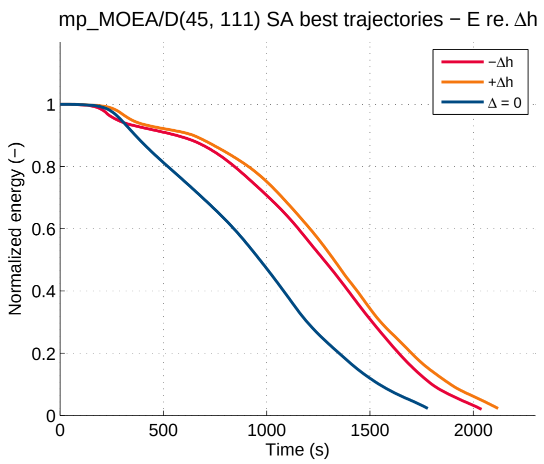
Sensitivity Analysis



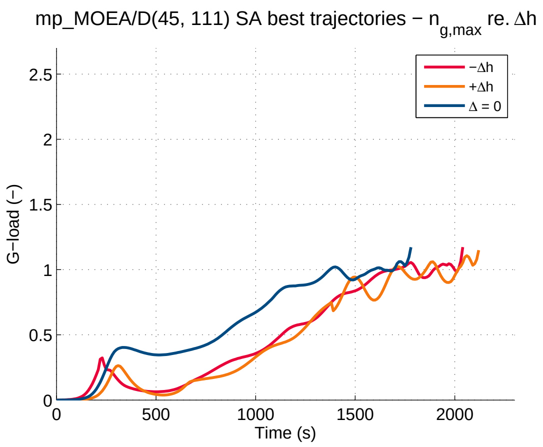
(a) Velocity v. time



(b) Flight path angle v. time

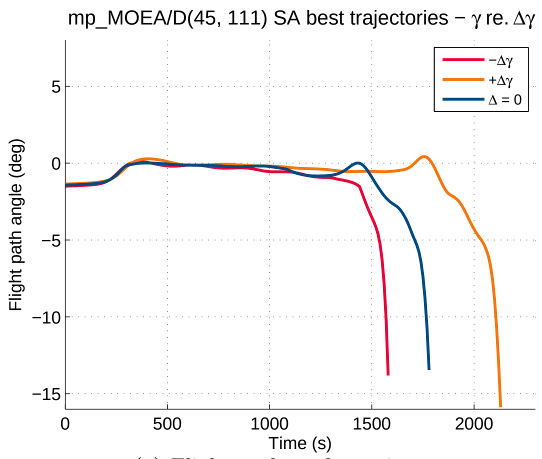


(c) Normalized energy v. time

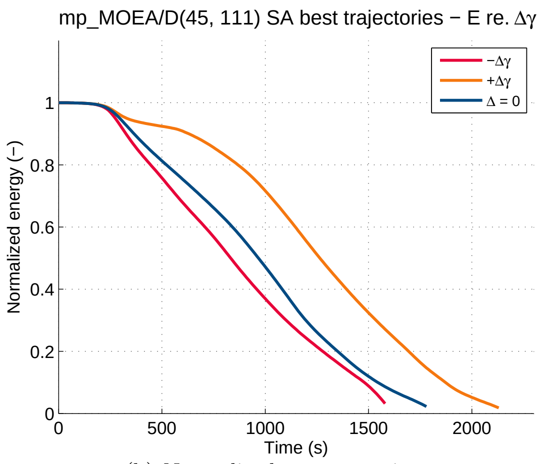


(d) g-load v. time

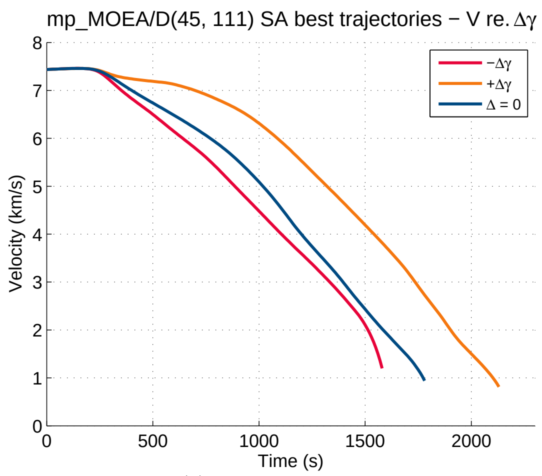
Figure C.1: Sensitivity analysis – effect of h_0 perturbation.



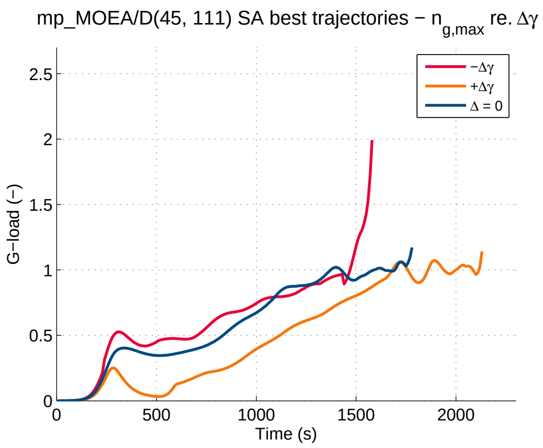
(a) Flight path angle v. time



(b) Normalized energy v. time

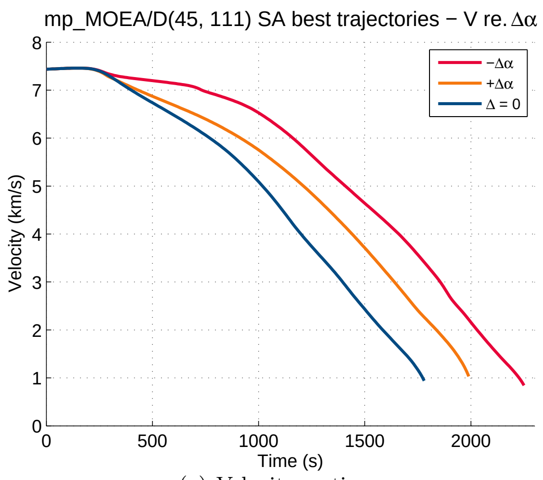


(c) Velocity v. time

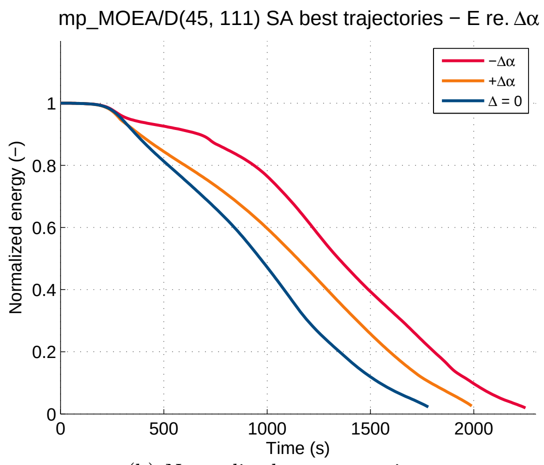


(d) g-load v. time

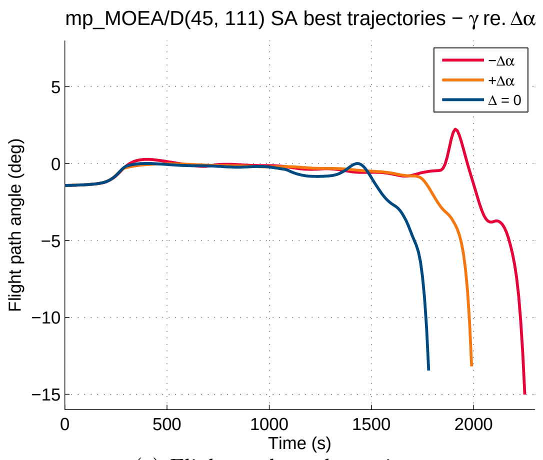
Figure C.2: Sensitivity analysis – effect of γ_0 perturbation.



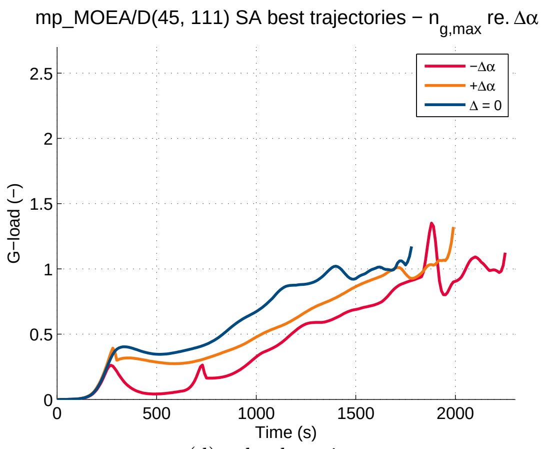
(a) Velocity v. time



(b) Normalized energy v. time

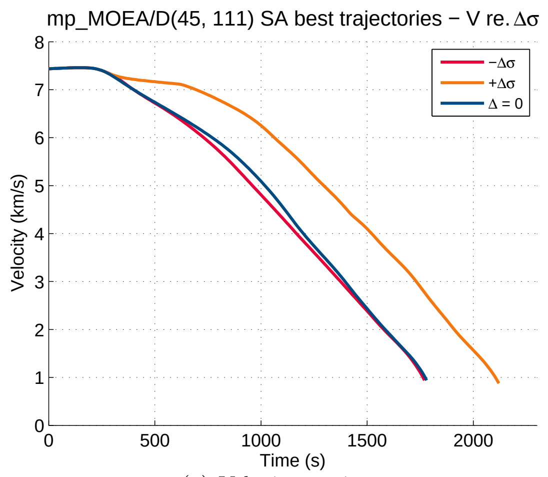


(c) Flight path angle v. time

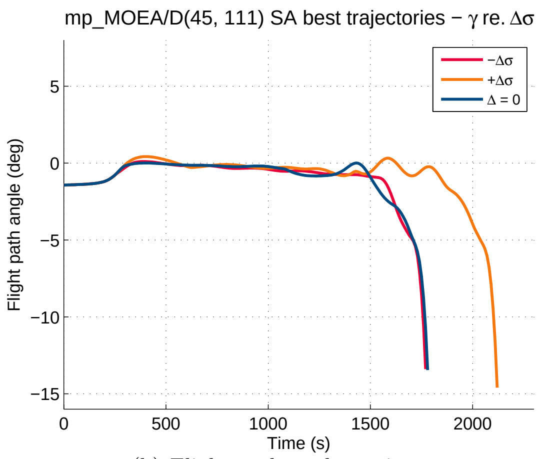


(d) g-load v. time

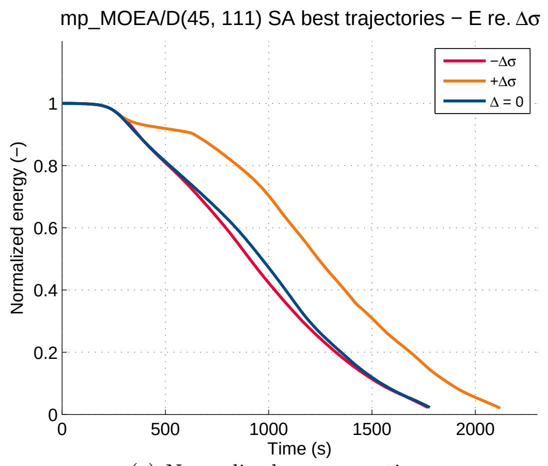
Figure C.3: Sensitivity analysis – effect of α_0 perturbation.



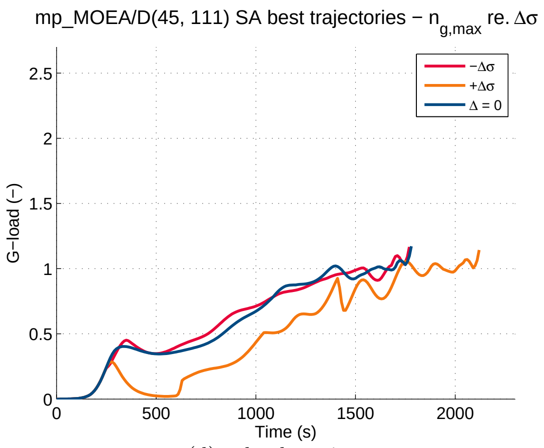
(a) Velocity v. time



(b) Flight path angle v. time

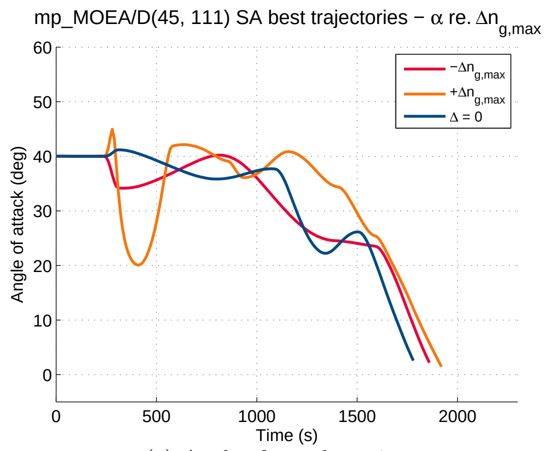


(c) Normalized energy v. time

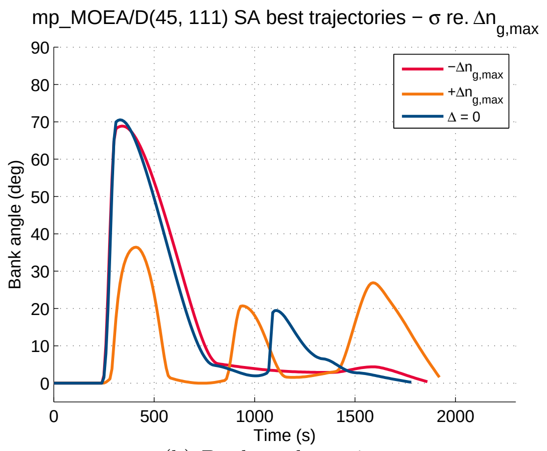


(d) *g*-load v. time

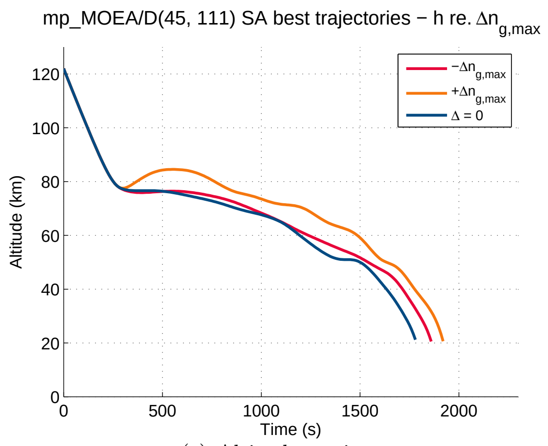
Figure C.4: Sensitivity analysis – effect of σ_0 perturbation.



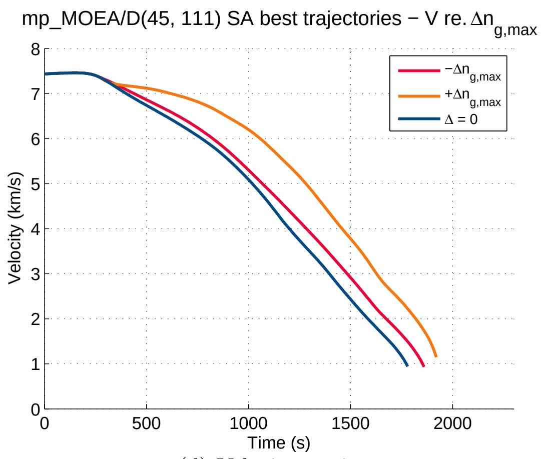
(a) Angle of attack v. time



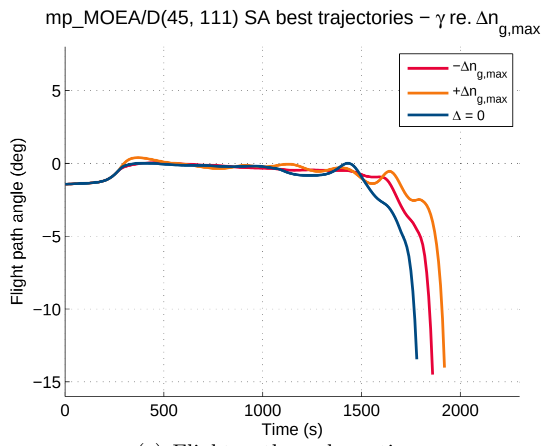
(b) Bank angle v. time



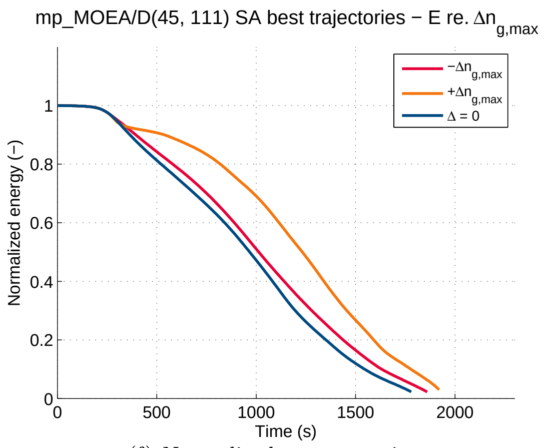
(c) Altitude v. time



(d) Velocity v. time

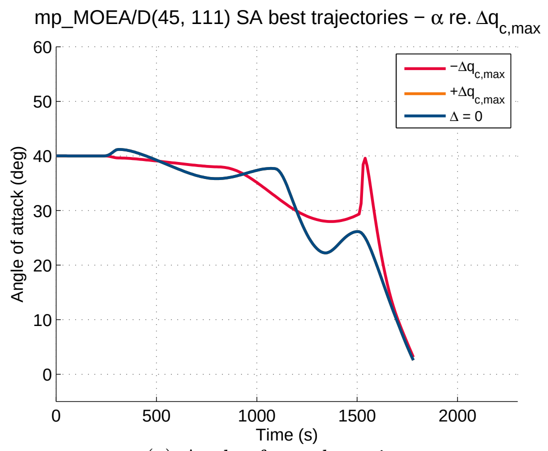


(e) Flight path angle v. time

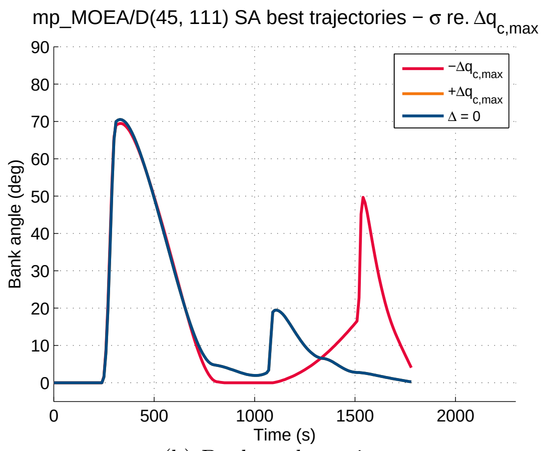


(f) Normalized energy v. time

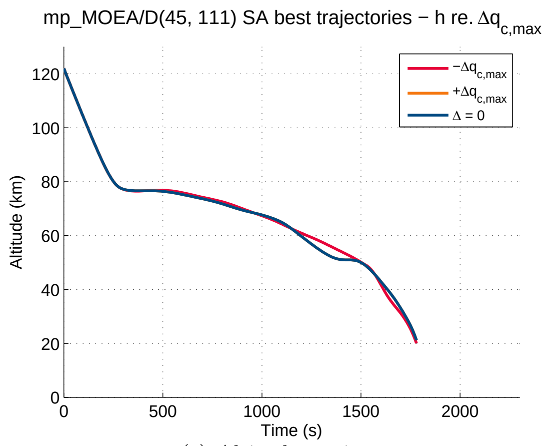
Figure C.5: Sensitivity analysis – effect of c_{n_g} perturbation.



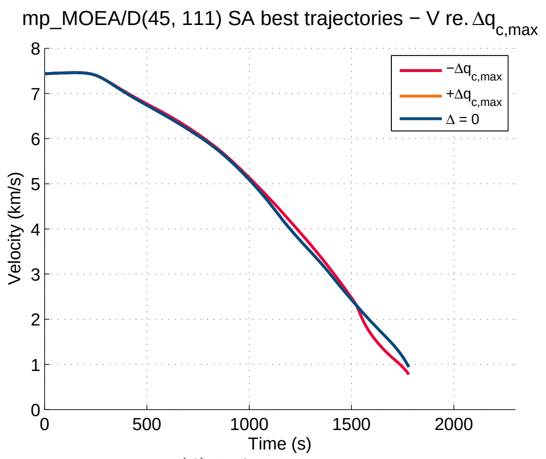
(a) Angle of attack v. time



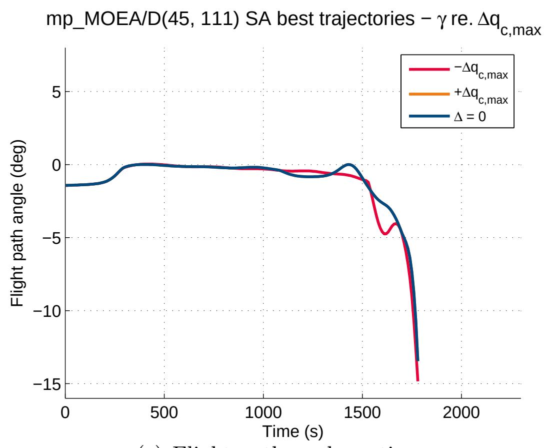
(b) Bank angle v. time



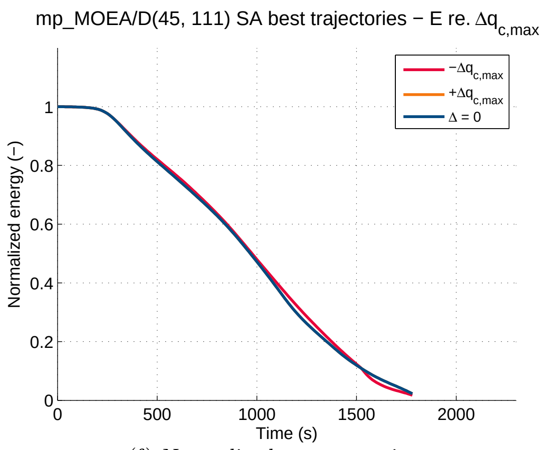
(c) Altitude v. time



(d) Velocity v. time



(e) Flight path angle v. time



(f) Normalized energy v. time

Figure C.6: Sensitivity analysis – effect of c_{q_c} perturbation.

Bibliography

Abraham, A., L. Jain, and R. Goldberg, eds. (2005), *Evolutionary Multiobjective Optimization – Theoretical Advances and Applications*. Springer-Verlag London Limited.

Bairstow, S.H. (2006), *Reentry Guidance with Extended Range Capability for Low L/D Spacecraft*. Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics.

Becker, J.V. (1961), "Re-Entry From Space." *Scientific American*, Vol. 204, pp. 49–57.

Biscani, F. (2014), "pagmo::problem::base class reference (pagmo .1.1.5)." Online, URL `http://esa.github.io/pagmo/classpagmo_1_1problem_1_1base.html`. Last accessed: 11/10/2014 13.09.

Biscani, F., D. Izzo, and C. H. Yam (2010), "A Global Optimisation Toolbox for Massively Parallel Engineering Optimisation." *arXiv preprint arXiv:1004.3824*.

Bloom, M.H. and A. Miele (1962), *Flight Mechanics Volume 1 - Theory of Flight Paths*, chapter Chapter 14: Aerodynamic Heating of Hypervelocity Vehicles, 308–334. Add.

Brunner, C.W. and P. Lu (2008), "Skip Entry Trajectory Planning and Guidance." *Journal of Guidance, Control, and Dynamics*, 31, No. 5.

Brunner, C.W and P. Lu (2010), "Comparison of Numerical Predictor-Corrector and Apollo Skip Entry Guidance Algorithms." *AIAA Guidance, Navigation and Control Conference*.

Chapman, D.R. (1959), "An Approximate Analytical Method for Studying Entry into Planetary Atmospheres." *NASA TR R-11*.

Chapman, D.R. (1960), "An Analysis of the Corridor and Guidance Requirements for Supercircular Entry into Planetary Atmospheres." *GPO*, Vol. 55.

Coello, C. A. Coello (1999), "A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques." *International Journal of Knowledge and Information Systems*, Vol. 1.

Cormen, T.H., C.E. Leiserson, and R.L. Rivest (2009), *Introduction to Algorithms*, 3$^{\text{rd}}$ edition, chapter Chapter 12, 289–291. MIT Press and McGraw-Hill.

Cox, K.J. (1973), "Space Shuttle Guidance, Navigation and Control Design Equations, Vol. III." *Prepared by Systems Analysis Branch Guidance and Control Division, NASA*.

Dijkstra, M. (2012), *Trajectory Optimization of Hyperion-II for the Study of Hypersonic Aerothermodynamic Phenomena*. Msc thesis, Delft University of Technology.

Dijkstra, M., E. Mooij, and K.J. Sudmeijer (2013), "Trajectory Optimization to Support the Study of Hypersonic Aerothermodynamic Phenomena." *AIAA Atmospheric Flight Mechanics (AFM) Conference*. Ref.: AIAA 2013–4501.

Doornbos, E. (2014), "TU Delft Astrodynamics Toolbox (Tudat)." URL `http://tudat.tudelft.nl/projects/tudat/wiki`.

Eggers, A. J., H.J. Allen, and S.E. Neice (1957), "A Comparative Analysis of the Performance of Long-Range Hypervelocity Vehicles." *NACA TN*.

Ely, L.D. (1966), *Return from Space; an Explanation of Re-Entry Problems and Factors of Re-Entry Vehicle Design and Performance.* Thomas.

ESA (2009), "ESA Launches First Earth Explorer Mission GOCE." URL `http://gdsc.nlr.nl/gdsc/en/news/esa_launches_first_earth_explorer_mission_goce`.

Fehlberg, E. (1968), *Classical Fifth-, Sixth-, Seventh-, and Eighth-Order Runge-Kutta Formulas with Stepsize Control.* National Aeronautics and Space Administration.

Feoktistov, V. (2006), *Differential Evolution: In Search of Solutions.* Springer.

Foerste, C., F. Flechtner, F. Schmidt, R. Meyer, and R. Stubenvoll (2005), "A New High Eesolution Global Gravity Field Model from the Combination of GRACE satellite Mission and Altimetry/Gravimetry Surface Gravity Data." *Geophysical Research Abstracts*, Vol. 7, p. 04561.

Fonseca, C. M. and P. J. Fleming (1995), "An Overview of Evolutionary Algorithms in Multiobjective Optimization." *Evolutionary Computation*, Vol. 3, pp. 1–16.

Graves, C.A. and J.C. Harpold (1970), "Reentry Targeting Philosophy and Flight Results from Apollo 10 and 11." *AIAA 8th Aerospace Sciences Meeting*, AIAA Paper No. 70–28.

Harpold, J.C. and C.A. Graves (1979), "Shuttle Entry Guidance." *Journal of Aeronautical Sciences*, Vol. 27, pp. 239–268.

Izzo, D. (2014), "PaGMO 1.1.5 (Parallel Global Multiobjective Optimizer)." URL `http://esa.github.io/pagmo/`. Last accessed: 25/11/2014.

Lambeck, K. (1990), "Aristoteles – An ESA Mission to Study the Earth's Gravity Field." *ESA Journal*, pp. 1–21.

Li, X. and H.J. Götze (2001), "Ellipsoid, Geoid, Gravity, Geodesy, and Geophysics." *Geophysiscs*, Vol. 66, pp. 1660–1668.

Loh, W.H.T. (1968), *Re-Entry and Planetary Entry: Physics and Technology Vol. 1.* Springer.

Marec, J.P. (1979), *Optimal Space Trajectories.* Elsevier Scientific Pub. Co.

Mathews, J.H. and K.K. Fink (2004), *Numerical Methods Using Matlab*, 4$^{\text{th}}$ edition, chapter Chapter 9, pp. 497–499. Prentice-Hall.

MBB (1988), "Study on Re-Entry Guidance and Control." Final report; esa report reference: Esa cr (p) 2652, MBB Space Communication and Propulsion Systems Division.

Mooij, E. (1994), *The Motion of a Vehicle in a Planetary Atmosphere.* Delft University of Technology, Faculty of Aerospace Engineering.

Mooij, E. (1995), "The HORUS-2B Reference Vehicle." Technical report, Delft University of Technology.

Mooij, E. (1997), "Linear Quadratic Regulator Design for an Unpowered, Winged Re-Entry Vehicle." Report-806.

Mooij, E. (1998), *Aerospace-Plane Flight Dynamics – Analysis of Guidance and Control Concepts.* Ph.D. thesis, Delft University of Technology (TUDelft).

Mooij, E. (2013), *Re-Entry Systems (AE4870B) - Lecture Notes.* Delft University of Technology, Faculty of Aerospace Engineering.

Mooij, E. (2014), "Personal communication."

Mooij, E. and P.G. Hänninen (2009), "Distributed Global Trajectory Optimization of a Moderate Lift-to-Drag Re-entry Vehicle." *AIAA Guidance, Navigation, and Control Conference.* Ref.: AIAA 2009–5770.

NASA (1976), *U.S. Standard Atmosphere 1976.* U.S. Government Printing Office, Washington, D.C.

NASA (1980), "Shuttle Program MCC Level C Formulation Requirements: Entry Guidance and Entry Autopilot (NASA TM-81093)." *Mission Planning and Analysis Division.*

Noomen, R. (2011), *Mission Geometry and Orbit Design (AE4878) - Lecture Slides Integrators v.4-1.* Delft University of Technology, Faculty of Aerospace Engineering.

O'Keefe, J.A., A Eckeis, and R.K. Squires (1959), "Vanguard Measurements Give Pear-Shaped Component of Earth's Figure." *Science*, Vol. 129, pp. 565–566.

Popova, O.P., P. Jenniskens, and V. Emel'yanenko (2013), "Chelyabinsk Airburst, Damage Assessment, Meteorite Recovery, and Characterization." *Science.*

Press, W.H., S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery (1989), *Numerical Recipes in C: the Art of Scientific Computing.* Cambridge University Press.

Qi, Y., X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu (2014), "MOEA/D with Adaptive Weight Adjustment." *Evolutionary Computation*, Vol. 22, pp 231–264.

Regan, F.J. (1993), *Dynamics of Atmospheric Re-Entry.* AIAA.

Robert, C. P. and G. Casella (2004), *Monte Carlo Statistical Methods.* Springer, New York.

Saltelli, A. and M. Ratto (2008), *Global Sensitivity Analysis. the Primer.* John Wiley & Sons.

Saraf, A., J.A. Leavitt, D.T. Chen, and K.D. Mease (2004), "Design and evaluation of an acceleration guidance algorithm for entry." *Journal of Spacecraft and Rockets*, Vol. 41, pp. 986–996.

Stacey, F.D. and P.M. Davis (2008), *Physics of the Earth*, 4th edition. Cambridge University Press.

Standish, E.M. (1995), "Report of the iau wgas sub-group on numerical standards." *Highlights of Astronomy*, Vol. 12, pp. 180.

Tigges, M., T. Crull, J. Rea, and Dr. W. Johnson (2006), "Numerical Skip-Entry Guidance." *29th Anuual AAS Guidance and Control Conference.*

Vallado, D.A. (2001), *Fundamentals of Astrodynamics and Applications*, 2nd edition. Kluwer Academic Publishers Group.

Vinh, N.X. (1981), *Optimal Trajectories in Atmospheric Flight.* Elsevier Scientific Software.

Watanabe, S. (2010), "Boost library: mt19937." URL `http://www.boost.org/doc/libs/1_57_0/doc/html/boost/random/mt19937.html`.

Yeomans, D. and P. Chodas (2013), "Additional details on the large fireball event over russia on feb. 15, 2013." URL `http://neo.jpl.nasa.gov/news/fireball_130301.html`. Last accessed 21-10-2014.

Zhang, Q. and H. Li (2007), "Moea/d: A multiobjective evolutionary algorithm based on decomposition." *IEEE Transactions on Evol*, Vol. 11, pp. 712–731.

Zhang, Q. and H. Li (2009), "Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii." *Evolutionary Computation, IEEE Transactions*, pp. 284–302.

Zitzler, E. and L. Thiele (1998), "Multiobjective optimization using evolutionary algorithms – a comparative case study." *Parallel Problem Solving from Nature – PPSN V, Springer Berlin Heidelberg*, pp. 292–301.

Zuiani, F. and M. Vasile (2013), "Multi agent collaborative search based on tchebycheff decomposition." *Computational Optimization and Applications*, Vol. 56, pp. 189–208.