Mars Ascent Vehicle trajectory optimisation

Investigation of the optimum trajectory for the Mars Ascent Vehicle to bring samples from the Martian surface to its orbit

Jérémie Gaffarel
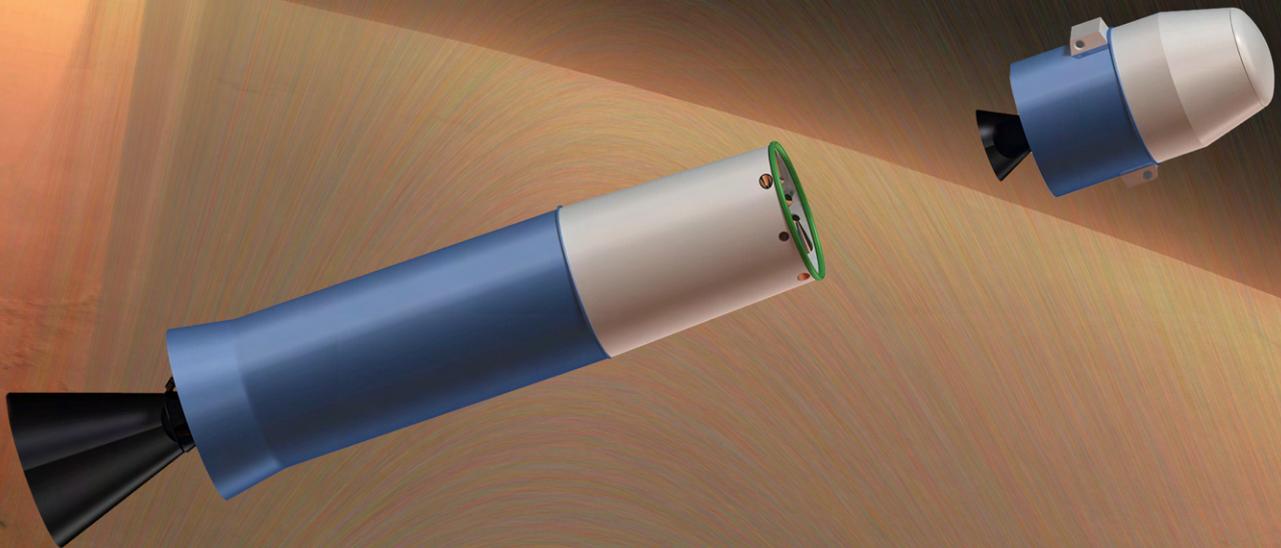
TUDelft

Mars Ascent Vehicle trajectory optimisation

Investigation of the optimum trajectory for the Mars Ascent Vehicle to bring samples from the Martian surface to its orbit

by

Jérémie Gaffarel

To obtain the degree of Master of Science in Aerospace Engineering
at the Delft University of Technology, and to be defended publicly
on Thursday Augustus 25, 2022 at 10:00 AM CEST.

Student number:	4651634			
Duration:	January 2022 – August 2022			
Defense committee:	Ir. M.C. Naeije,	TU Delft,	Thesis supervisor	
	Dr. S.M. Cazaux,	TU Delft,	Chair	
	Dr. J. Guo,	TU Delft,	Examiner	
	Dr. D. Dirkx,	TU Delft,	Additional examiner	

Version	Date	Comment
0.1	23/07/2022	Ch.1 to 3: first draft.
0.2	27/07/2022	Sec.4.1 to 4.3: first draft. Sec.4.4 and 4.5: first figures.
0.3	28/07/2022	Sec.4.3: added more results. Ch.5: first draft.
0.4	08/08/2022	Sec.2.1: adapted discussion for new helicopters plan. Ch.3 and 4: added anchor SRM, results from all optimisations. Sec.4.4 and 4.5: first draft. Ch.A: results appendix first draft.
1.0	11/08/2022	Sec.4.5.2: first draft. Ch.5 and 6: first complete drafts. Entire report: refactoring.
1.1	18/08/2022	Sec.4.5.2 and Ch.5: final results. Appendix A: described all appendices. Entire report: implemented comments on v1.0 from M. Naeije.

Cover image of the Mars Ascent Vehicle discussed by Yaghoubi et al. [1] during stage separation, modelled using CATIA [2], with 634 thousand distinct simulated ascent trajectories represented.

An electronic version of this thesis is available at `repository.tudelft.nl`.

TUDelft

Preface

With this report, I conclude 5 years of studies at the Delft University of Technology. While my Bachelor widened my views on aeronautical engineering, space had a special place in my heart from the beginning. Since a long time, I am interested in how daring missions are planned, carried by complex spacecrafts, to explore space. To me, this is one of the hardest and most inspiring challenge in which humanity can unite towards a common goal. Throughout my studies in Delft, the projects that I most enjoyed were the pre-phase A designs of various spacecrafts, during which I had the opportunity of designing and optimising both the vehicle itself, as well as its mission profile. This required me to explore all paths that could lead to a feasible solution, with their associated cost, complexity, and risk. My Master thesis that concludes these studies encompasses the aspects that challenge me, and in which I have specialised myself: modelling, simulation, and optimisation.

My graduation research is on the study of the ascent trajectory of the Mars Ascent Vehicle and on the effect that its solid rocket motors have. This rocket, currently designed by the Jet Propulsion Laboratory, is set to bring samples from the surface of Mars to its low orbit in 2031. Studying the different reports on the progress of the design of this vehicle, it appeared that no precise definition was made for the geometry of the solid motors used. I took this gap as an opportunity to do what I learned in my Master and like best: numerically represent the rocket and its motors, simulate the motor burn and ascent to orbit, and optimise both the ascent trajectory and the motor geometry.

I would like to thank my daily supervisor, Marc Naeije, for his support and guidance during not only this thesis, but also during my Bachelor thesis, as well as throughout my first year of master's studies in the context of the publication of my first scientific article. His weekly investment greatly helped to keep me on track on my work, motivating me to keep investing time in my work even when getting stuck for weeks on end. Our weekly meetings also helped me stay on track, and see that, after all, progress was sometime slow but still very present. I also thank my daily supervisor for his trust, leaving me room to develop my own process and methodology, while giving advice and critical comments to prevent me from losing track of my research goal. Lastly, thank you to both Marc Naeije and the Astrodynamics and Space Missions department, for allowing me to make use of their server, considerably reducing the time required to run my simulations and optimisations, allowing me to explore more solutions.

I am also very grateful to Dominic Dirkx for the opportunity to work as an assistant on the TU Delft Astrodynamics Toolbox, in parallel to my master's thesis. Amongst other benefits, this allowed me to be a lot more familiar into the capabilities of this software, making my graduation work more efficient, and allowing me to explore more possibilities for my models and simulation. Working in the team for this software also motivated me to go more in-depth into the numerical aspect of this research, studying for instance the limits of feasible accuracy, or working with a high number of integration schemes. Moreover, working on this code also made me trust my results more, since I was seeing this tool less as a black box, having worked on its source.

While my graduation project is not related to the von Karman Institute for Fluid Dynamics, I thank them for the internship the I undertook preceding my master's thesis. In particular, would like to thank Thierry Magin, Damien Le Quang, Amandine Denis, and Pietro Parodi. This internship allowed me to gain a much deeper understanding of the physics of the Martian atmosphere and gave me the understanding and tools to study a spacecraft that flies through it. The academic setting in which I was during my internship also motivated me to be more rigorous in my work and revealed how much I enjoy theoretical research.

Last but not least, I would like to thank my family. First, I thank my parents for their continued support, and for offering me the occasion of studying in Delft for 5 years. I am also grateful for the way they allowed me to grow curious about the world I live in, and more specifically Science and engineering. Thanks also go to my brother for keeping me on my toes, motivating me to never fear being challenged, but to always give my all instead. Finally, I hearty thank my partner for supporting me daily and being patient during the entirety of my studies, allowing me to stay positive whenever I got stuck on my project, to get some fresh air when needed, and to make sure that I did not get too easily distracted from my work.

Ad astra per audacia et scientiam.

Jérémie Gaffarel
The 17th of August 2022

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Al	Aluminium
AP	Ammonium Perchlorate
CAD	Computer-Aided Design
CEA	Chemical Equilibrium Applications
DSE	Design Space Exploration
DSMC	Direct Simulation Monte Carlo
EM	Exponential Map
ERO	Earth Return Orbiter
GLOM	Gross Lift-Off Mass
HTPB	Hydroxyl Terminated PolyButadiene
JPL	Jet Propulsion Lab
MAV	Mars Ascent Vehicle
MCD	Martian Climate Database
MHACO	Multi-objective Hypervolume-based Ant Colony Optimisation
MON	Mixed Oxide of Nitrogen
MRP	Modified Rodrigues Parameters
MSR	Mars Sample Return
N.A.	Not Applicable
NSGA2	Non-dominated Sorting Genetic Algorithm
NSPSO	Non-dominated Sorting Particle Swarm Optimisation
OS	Orbiting Samples
Pagmo	Parallel Global Multiobjective Optimizer library
PPC	Particles Per Cell
RCS	Reaction Control System
RK	Runge-Kutta
RPA-C	Rocket Propulsion Analysis Combustion
SFR	Sample Fetch Return
SH up to D/O	Spherical Harmonics up to Degree and Order
SPARTA	Stochastic PArallel Rarefied-gas Time-accurate Analyzer
SRL	Sample Retrieval Lander
SRM	Solid Rocket Motor
Tudat	TU Delft Astrodynamics Toolbox
TVC	Thrust Vectoring Control
USM	Unified State Model
VECTOR	Vertically Ejected, Controlled Tip-Off Rate

Symbols

α	Accommodation coefficient	[-]
$\bar{C}$	Cosine spherical harmonic coefficient	[-]

$\bar{P}_{lm}$	Legendre polynomial at a degree l and order m	[-]
$\bar{S}$	Sine spherical harmonic coefficient	[-]
β, γ	Parametrised post-Newtonian parameters[1]	[-]
χ	Heading	[rad]
Δb	Change in burned distance	[m]
Δh	Difference in altitude	[h]
Δt	Time step	[s]
ΔV	Change in velocity	[m/s]
Δx	Simulation grid size	[m]
δ	Initial kick angle	[rad]
δ_s	Anchor shank width	[m]
$\dot{m}$	Mass flow	[kg/s]
η_c	Combustion efficiency factor	[-]
Γ	Vandenkerckhove function	[-]
γ	Flight path angle[2]	[rad]
γ	Specific heat ratio[3]	[-]
Λ	Mass ratio	[-]
λ	Mean free path	[m]
$\mathrm{a_{BA}}$	Acceleration of body B on body A	[m/s^2]
f	State derivative	[-]
F_p	Perturbation force	[N]
$\mathrm{r_{BA}}$	Difference in position between body B and A	[m]
x	State	[-]
x_0	Initial state	[-]
μ_{Mars}	Gravitational parameter of Mars	[kg·m^3/s^2]
μ_B	Gravitational parameter of body B	[kg·m^3/s^2]
ϕ	Latitude	[rad]
ρ	Density	[kg/m^3]
ρ_0	Density at the bottom of an atmospheric layer	[kg/m^3]
ρ_c	Combustion gas density	[kg/m^3]
ρ_p	Propellant density	[kg/m^3]
τ	Mean collision time	[s]
θ	Longitude	[rad]
θ_y	Nozzle deflection angle around the y axis	[rad]
θ_z	Nozzle deflection angle around the z axis	[rad]
a	Burning rate coefficient[4]	[m/s/Pa^n]
a	Semi-major axis[5]	[m]
A_{occulted}	Occulted area	[m^2]
a_{thrust}	Thrust acceleration	[m/s^2]
A_e	Exhaust area	[m^2]

[1]Used related to General Relativity.
[2]Used related to spherical elements.
[3]Used related to propellant chemistry.
[4]Used related to solid motor burning.
[5]Used related to orbital elements.

A_p	Port surface area	[m^2]
A_t	Throat area	[m^2]
b	Burned distance	[m]
c	Speed of light in vacuum	[m/s]
c^*	Characteristic velocity	[m/s]
c_{eff}	Effective exhaust velocity	[m/s]
C_r	Radiation pressure coefficient	[-]
d_{app}	Apparent separation	[m]
e	Eccentricity	[-]
f_{Sun}	Shadow fraction	[-]
F_T	Thrust	[N]
g	Gravitational acceleration	[m/s^2]
g_0	Earth graviational acceleration at sea level	[m/s^2]
h	Altitude	[m]
h	Integration step	[s]
h_0	Initial altitude	[m]
h_s	Scale height	[m]
I	Shadow parameter	[-]
i	Inclination	[rad]
I_{sp}	Specific impulse	[s]
Kn	Knudsen number	[-]
Kn_{grid}	Knudsen number with $L_{\text{ref}} = \Delta x$	[-]
L	Length of a solid rocket motor[6]	[m]
L_{ref}	Reference length	[m]
L_f	Fin length	[m]
M	Molar mass	[kg/mol]
m	Mass	[kg]
M_0	Initial mass	[kg]
M_{inert}	Motor inert mass	[kg]
m_A	Mass of body A	[kg]
M_P	Propellant mass	[kg]
n	Burning rate exponent	[-]
N_a	Number of anchors	[-]
N_f	Number of fins	[-]
n_O	Atomic oxygen number density	[1/m^3]
P	Burning perimeter[7]	[m]
P	Total power output	[W]
p_a	Ambient pressure	[Pa]
p_c	Chamber pressure	[Pa]
p_e	Exhaust pressure	[Pa]
R	Reference radius of a body	[m]
r	Solid propellant regression rate	[m/s]

[6]Used related to SRM geometry.
[7]Used related to solid motor burning.

$R_{app,M}$	Apparent radius of Mars	[m]
$R_{app,S}$	Apparent radius of the Sun	[m]
R_{Mars}	Mars radius	[m]
R_{mid}	Intermediate solid propellant radius	[m]
R_A	Absolute gas constant	[J/K/mol]
R_e	SRM exhaust conduit radius	[m]
r_f	Fillet radius	[m]
R_i	Inner solid propellant radius	[m]
R_o	Outer solid propellant radius	[m]
S	Burning surface	[m^2]
S_{ref}	Reference surface area	[m^2]
T	Temperature	[K]
t	Time	[s]
t_b	Burn time	[s]
T_c	Chamber temperature	[K]
u	Flow velocity	[m/s]
V	Velocity	[m/s]
V_0	Initial velocity	[m/s]
$V_{Mars,circ}$	Circular velocity of a body in Martian orbit	[m/s]
V_e	Exhaust velocity	[m/s]
V_p	Propellant volume	[m^3]
w	Web thickness	[m]
w_{ds}	Web trough the detached silver	[m]
w_f	Fin width	[m]
w_t	Total web thickness	[m]

Abstract

In 2020, the Perseverance rover landed on Mars, starting the Mars Sample Return mission. These samples are then to be returned to Earth, being the first ones from a different planet. This is the work that will be carried by the Mars Ascent Vehicle, a two-stage rocket that uses solid propellant, and of which a preliminary design has been made by the Jet Propulsion Lab. Once in Martian orbit, the payload will then be caught by the Earth Return Orbiter that will take care of the transfer from Low Mars Orbit to the surface of the Earth.

However, a research gap has been identified: finding what the optimum solid rocket motors geometries are to bring the Martian samples to a target orbit between 300 km and 375 km above Mars. Trough the modification of the solid rocket motors, the thrust profile itself is altered, which highly affects the ascent trajectory.

To fill in the research gap, a simulation of both the ascent and the propellant burn has been setup. The thrust profile is thus computed from the geometry of both motors and is then included in the ascent simulation. The model that has been setup also includes the launch angle of the rocket, which can vary between 30 deg and 60 deg, as well as the stage separation angle. Thrust vectoring control has also been implemented, with nozzle deflection angles between -4 deg and 4 deg.

A benchmark simulation has then been setup, studying the highest accuracy feasible for the thrust simulation and 4 sections of the ascent simulation, splitting the ascent into first and second stage, and powered and unpowered. Using this benchmark, the integrators used for the simulations has been tuned to achieve an accuracy of 5 km and 5 m/s with the least number of function evaluations possible, to save as much CPU time as possible during the optimisation that will follow. The same requirement and benchmark have been used to tune the numerical representation of the Martian environment used in the ascent simulation.

Using the tuned simulation, an optimisation has then been setup, with the following design variables: the motor geometry of both stages, the launch and separation angles, and the nozzle deflection angles during first stage burn. The objectives of this optimisation were to minimise to deviation to the target periapsis and apoapsis altitude range, and to minimise the launch mass.

From the optimisation setup, and running the optimisation itself, more than 634 thousand propellant burn and ascent simulations were run, amongst which thousand satisfy the requirement in target orbit, and in mass. From these, multiple cases were selected. First, the absolute optimum reached an orbit with a periapsis altitude of 314 km, a periapsis of 339 km, and a launch mass of 343 kg, 57 kg less than required.

Then, while the optimisation aimed at minimising the launch mass, the solutions database was used to look for other types of solutions. One of these is the ascent that has a minimum total burn time, which is of 32.2 s: 24.9 s for the first stage, and 7.3 s for the second stage. In this case, the launch mass is 27 kg higher, at 370 kg. One last feasible solution that has been searched in the solutions database is the one that leads to the lowest second stage burn time. Because that stage is uncontrolled, reducing the burn time can reduce the orbital deviation. This lowest second stage burn time is of 5.2 s, following a first stage burn time of 34.7 s.

Following the optimisation, a sensitivity analysis has been run with two goals. This first is to ensure the robustness of the solutions that were found to various errors in initial state, in environment, and in vehicle design. The second goal is to study the orbital decay of the final orbits. From this process, it was found that the solutions were robust to initial errors of the levels that could be expected from similar missions. Also, the final orbit that is reached decays below the altitude at which the Earth Return Orbiter can catch the samples after 84 days. During this period, the inclination of the orbit repeats back to the target one 5 times, with the last opportunity to catch the samples being 78 days after launch.

All results from the optimisation and the sensitivity analysis have then been used to fill the research gap that was identified. At the same time, the setup of the numerical simulation was used as a context to study the importance of numerical accuracy in details.

1

Introduction

One of the most fundamental questions one can ask is whether life is present somewhere else in the universe. The latter being so vast, it is likely that life is indeed present somewhere else, in one form or another.

For years, scientists and engineers have wondered what secrets Mars could be keeping, and if life was one day on it. In this context, the Mars Sample Return mission that started in 2020 aims at returning samples from relevant localisation of the surface of Mars, back to Earth. An overview of this mission is given in Chapter 2, in Section 2.1.

This sample return mission bears second importance: it will be the first time that a rocket takes off from the surface of Mars. Indeed, the Mars Ascent Vehicle (MAV) is a two-stage solid propellant rocket that will bring 16 kg of samples from the surface of Mars to its orbit. The requirements for this rocket, as well as its preliminary design, are discussed also in Chapter 2, in Section 2.2.

This Chapter shows that a lot has already been designed for this vehicle. However, optimising its trajectory has not been done yet. In particular, it has been identified that the geometry of the two solid rocket motors that propel the MAV can be optimised to create a thrust profile that will inject the Martian samples into a pre-defined orbit with the lightest vehicle possible.

This task of optimising the ascent trajectory of the MAV, as well as its motors, is at the core of the research that is described in this report. The objectives of this research can also be found in Chapter 2, in Section 2.3.

Chapter 3 then describes the methodology that was followed, with the objective of answering the different research questions. This methodology is split into three distinct sections: modelling, simulation, and optimisation. These three sections have been executed in that order. First, the MAV, its motors, and its environment have been modelled, as described in Section 3.1.

Then, these models have been used to set up a numerical simulation of the ascent trajectory, including a simulation of the solid motors. The methodology followed for this is discussed in Section 3.2, including the methodology used to ensure the accuracy of the numerical simulation.

Finally, Section 3.3 reflects on the methodology followed to set up the optimisation of the solid motors and of the ascent trajectory. This includes the design variables and objectives selection, as well as the optimiser algorithm tuning, and the method for picking optimums.

The results that directly follow from applying the different methodologies are then laid out in Chapter 4. The goal of this Chapter is to give an in-depth report of the results, as objectively as possible. This includes the definition of the accuracy of the numerical simulation, the optimisation results, validation of the models, and the sensitivity analysis of the results.

Chapter 5 then reflects on the results, explaining how they answer the research questions. A discussion is also made in this Chapter, to explain how the results of the simulations can be used to adapt the design of the MAV, or to prove its feasibility up to a certain extent.

Last but not least, Chapter 6 compiles recommendations that are made, should anyone push this research further. Recommendations are especially made regarding possible improvements in the models, and in the research setup. A discussion is also made on further investigations that could be made based on the set of results that have been uncovered thanks to this research.

2

Problem statement

This chapter sets up the scene for the research that is reported in this document First, an overview of the overall mission in which the research fits is given in Section 2.1. Then, the most up-to-date design and requirements of the Mars Ascent Vehicle (MAV) are given in Section 2.2. Follows a discussion on the proposed research in Section 2.3, concluded by Section 2.4 in which relevant previous work on the MAV and on similar topics is described.

2.1. Mission overview

This section presents a global overview of the Mars Sample Return (MSR) mission in which the Mars Ascent Vehicle registers itself. As seen in Figure 2.1, the MSR mission contains different segments, some of them developed by ESA, and some of them by NASA. These four segments are Mars 2020, the Earth Return Orbiter, the Sample Retrieval Lander, and the Sample Return and Science. These are explained in Sections 2.1.1 to 2.1.5 respectively.
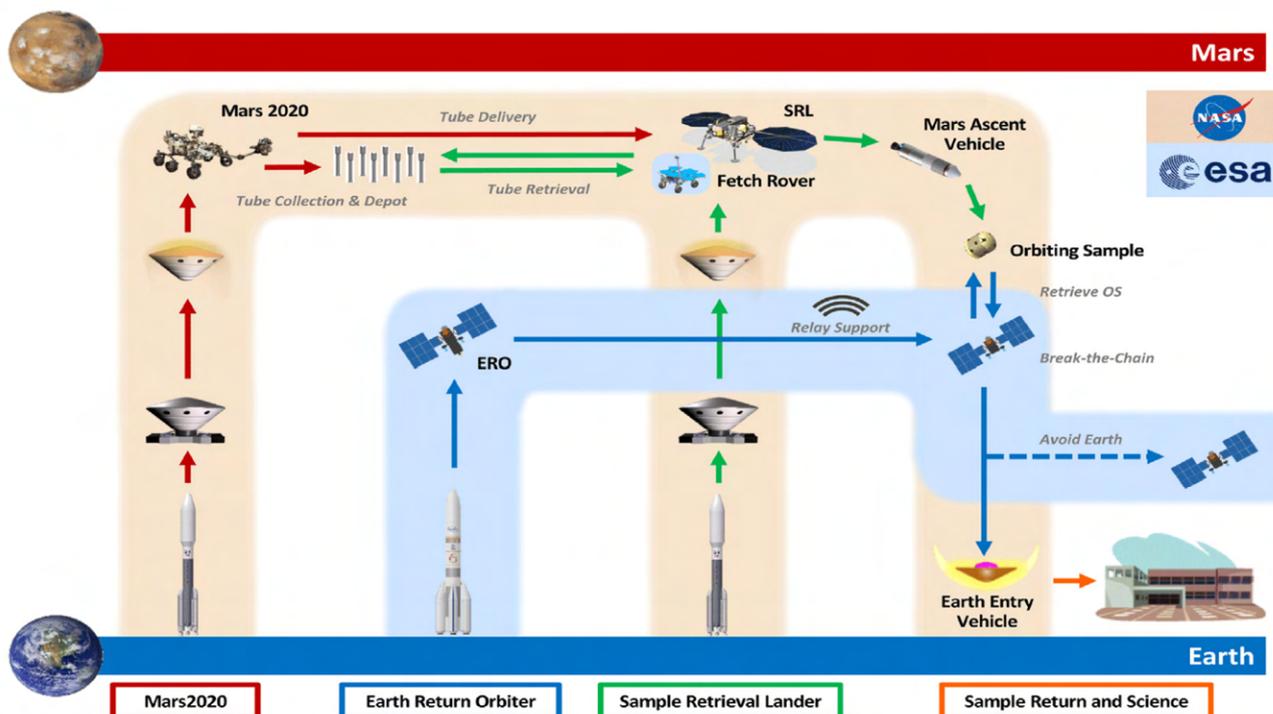
Figure 2.1: Different segments of the Mars Sample Return mission campaign [3].

As specified by Muirhead et al. [3], the Mars 2020 segment has already started and will end in mid-2028. The Sample Retrieval Lander (SRL) will be launched at the end of 2026, and the samples will be launched from Mars in mid-2029. The Earth Return Orbiter will also be launched at the end of 2026 and will return to Earth towards the middle of 2031.

To be noted is that this mission plan has evolved. In November 2020, an Independent Review Board recommended that two distinct landers be used for the SRL and the MAV, as to keep the interplanetary capsule the same size as previous missions such as Perseverance or Curiosity [4]. Later, in July 2022, NASA decided to scrap the SRL altogether, fitting everything once again into

the MAV lander, and using Perseverance and two helicopters to retrieve the samples [5]. This is discussed in Section 2.1.2.

2.1.1. Mars 2020

The MSR mission first starts with the Mars 2020 segment. It consists in the collection of the samples, and their depot on the Martian surface.

One of the assignments of the Perseverance rover that landed on Mars in February 2021 is to take samples of the surface of Mars. As discussed by Farley et al. [6], Perseverance is equipped with an arm that can take a sample tube and put it behind a drill bit. Then, the samples that it drills from the surface of Mars go directly into sample tubes.Perseverance then degases the tubes and seals them. Furthermore, to preserve the scientific interest of the samples, a great deal of attention was paid so that every element of Perseverance that gets in contact with the samples does not alter nor pollute them in any way. A render of such a sampling tube can be seen in Figure 2.2.

Figure 2.2: Render of a sample tube made using the file from NASA/JPL-Caltech [7].

2.1.2. Sample retrieval

After Perseverance takes a sample from the Martian surface, it splits the sample in two, and places it in two samples tube. Each sample tube weights a total of 150 gr [8]. Perseverance then leaves one sample tube on the surface and keeps one inside of it.

Then, there are two distinct ways to bring the samples to the MAV. The first is to make Perseverance drive to the MAV platform, and make it place the samples directly into the sample container that, together, constitute the MAV payload. The second way is to use two helicopters to retrieve the samples [5]. These helicopters, inspired by the Ingenuity helicopter that landed with Perseverance, can fly to the samples that were left on the surface of Mars, land close to them, then drive a few meters and grab the sample. They then fly back to the MAV, and a robotic arm places the sample tubes into the MAV sample container.

In this mission scenario, Perseverance could fail, as well as one helicopter, and it would still be possible to place the samples in the MAV. In the original MSR plan, a Sample Fetch Rover was to be used to drive to the samples left behind by Perseverance, and drive back to the MAV with them [9]. Bringing the samples back to the MAV should be done in 111 sols, as to avoid solar conjunction. If a hibernation mode is used, up to 180 sols can be used to fulfil the sample retrieval mission [9].

However, in July 2022, it was decided to use Perseverance or the two helicopters, depending on which system was most successful [5, 8]. This is because Perseverance has proven to be more persevering than planned. It is then thought that it would still be functional in 2031. The Ingenuity helicopter surviving on Mars for more than a year more than planned inspired the use of the two helicopters.

2.1.3. Mars Ascent Vehicle

Once the samples are collected and brought back to the SRL, up to 30 are placed in a container that houses all of them together, such that they can be transferred to Earth. The Mars Ascent Vehicle is a rocket that is then used to transfer this payload of samples from the surface of Mars to Martian orbit [3]. The design requirements and aspects of this MAV are discussed in more depth in Section 2.2.

2.1.4. Earth Return Orbiter

After the MAV brings the samples into Mars orbit, the Orbiting Samples (OS) container is released from it. The Earth Return Orbiter (ERO), which is placed in a Low Mars Orbit similar to the one in which the MAV injects itself, then has to detect, rendezvous, and capture the OS [10]. The system to capture, contain, and return the OS, as part of the ERO, can be seen in Figure 2.3.

In addition, the ERO acts as a communication relay between the SRL, the MAV, and the Earth during the entire mission leading to the orbiting sample retrieval [3]. Finally, after the ERO catches the OS, it departs from Mars, places the sample container into an Earth entry capsule, and places the capsule in an Earth re-entry trajectory before executing an Earth avoidance manoeuver [10].
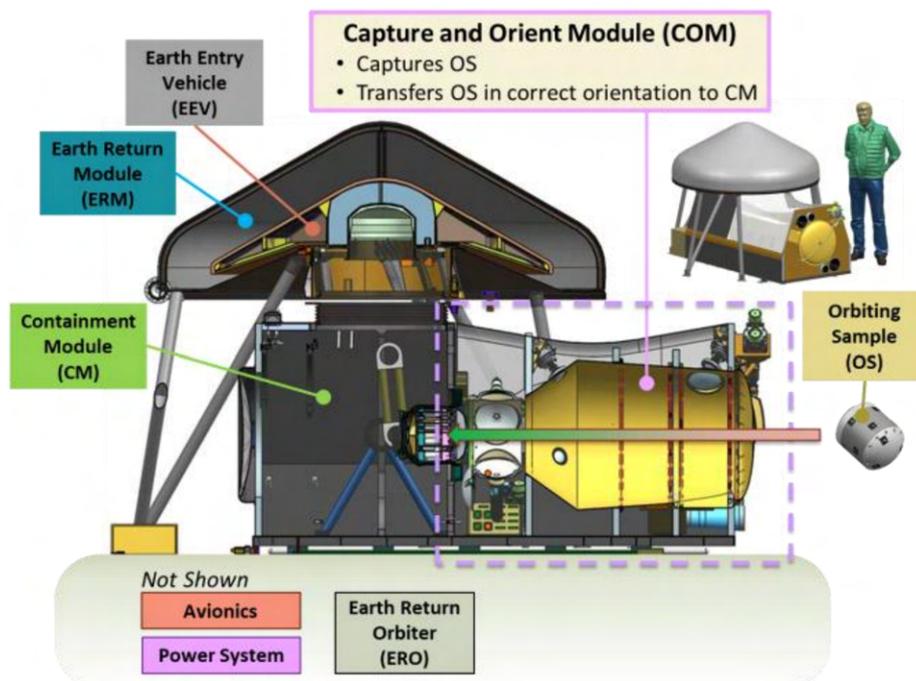
Figure 2.3: Capture, Contain, and Return System of the Earth Return Orbiter [11].

2.1.5. Sample Return and Science

The last step in the Mars Sample Return mission is to get the samples safely to Earth and make Science. The entry capsule that allows for the samples to get into the Earth atmosphere will also serve as an aerodynamic decelerator up to the Earth surface, as discussed by Corliss [12].

Furthermore, shock absorbers are placed below the sample container, such that their scientific integrity is preserved by limiting off nominal landing loads below 3000 g [12]. The layout of the entry capsule can be seen in Figure 2.4, with the sample container in its core and the pink shock absorbers.
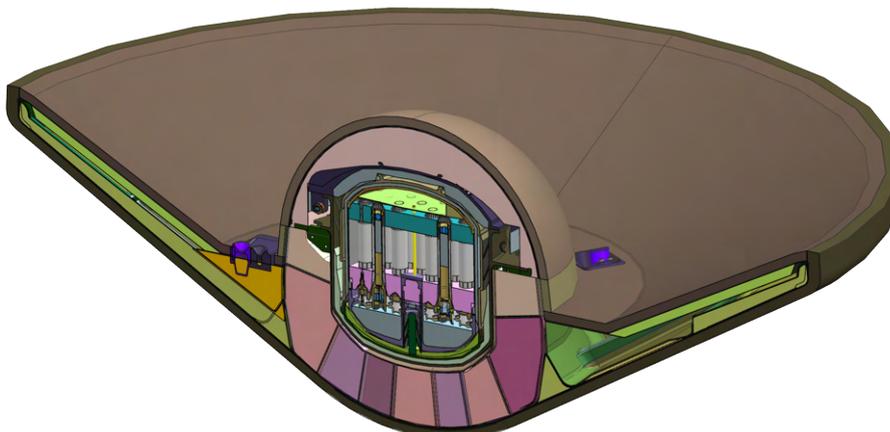
Figure 2.4: Layout of the entry capsule with the Mars sample container [12].

Finally, once the samples of the Martian surface have completed their journey to the surface of the Earth, they can be analysed. It is then of high importance that the samples are not contaminated by Earth, and that they do not contaminate Earth. Indeed, the Martian samples are a category V restricted Earth return [9]. According to Planetary Protection Policy [13], this means that, for such samples, total containment of the unsterilised samples and material being in contact with them is required. Moreover, if any sign of life is detected, the samples must remain contained unless effective sterilisation has been made.

2.2. Requirements and design from JPL

This section first discusses the requirements that have been elaborated for the MAS, as part of the MSR mission. Then, the preliminary design that has already be made by the Jet Propulsion Lab (JPL) for different systems and subsystems is elaborated.

2.2.1. Requirements

Multiple requirements have been identified by JPL for the Mars Ascent Vehicle. These requirements first concern the quality of the Orbiting Sample's final orbit. They are identified by **MAV-OSO**, and have been established by Jackman et al. [14]. Of interest, **MAV-OSO-01** is defined so that the OS does not decay too fast into the atmosphere, and **MAV-OSO-03** because the ERO will be placed in a similarly inclined orbit before capturing the OS.

Then, requirements regarding the volume and mass have been established. These are identified by **MAV-VM**, and have been established by Jackman et al. [14]. The **MAV-VM** requirements are mostly related to the logistics associated with bringing the MAV from Earth to Mars.

Finally, the requirements identified by **MAV-LL** concern the launch of the MAV from Mars. Requirements **MAV-LL-01** to **MAV-LL-03** have been established by Jackman et al. [14], while Requirements **MAV-LL-04** to **MAV-LL-06** have been established by Prince et al. [15].

MAV-OSO-01	The final orbit of the payload of the MAV shall be at a hard lower altitude of 300 km.
MAV-OSO-02	The final orbit of the payload of the MAV shall be at a soft upper altitude of 375 km.
MAV-OSO-03	The final orbit of the payload of the MAV shall be at an inclination of 20 deg.
MAV-VM-01	The MAV shall have a height of a maximum of 2.8 m.
MAV-VM-02	The MAV shall have a diameter of a maximum of 0.57 m.
MAV-VM-03	The MAV shall have a gross lift-off mass (GLOM) of a maximum of 400 kg.
MAV-VM-04	The MAV shall carry a payload of a mass of 16 kg.
MAV-LL-01	The earliest launch date from the surface of Mars shall be 2031.
MAV-LL-02	The launch of the MAV shall be done from the Jezero crater (18.5°N,77.5°E), at an altitude of -2.55 km
MAV-LL-03	During ascent, the angle of attack of the MAV shall not exceed 4 deg.
MAV-LL-04	The MAV shall be launched by the SRL at an angle between 30 deg and 60 deg from the local horizon.

2.2.2. MAV preliminary design

This section describes the design trade-off that has been carried to select the overall architecture of the MAV. Then, the selected architecture is expanded upon in more depth, as well as key elements related to it. Finally, the latest design update is elaborated upon.

Design trade-off

Two distinct concepts have been assessed and traded off by NASA Marshall Space Flight Center and JPL to fulfil the mission of the Mars Ascent Vehicle. Firstly, a single stage hybrid propulsion

concept has been assessed by McCollum et al. [16]. The oxidiser is liquid, composed of Mixed Oxide of Nitrogen (MON), pressurised by helium. The MON is also used for Thrust Vectoring Control (TVC), by injecting it directly into the nozzle to redirect the thrust. Also, helium is used as a monopropellant for the Reaction Control System (RCS). The fuel is solid and made of SP7 wax. The engine can be throttled by throttling the oxidiser flow. This first concept can be seen illustrated in Figure 2.5, and it has preliminarily been computed by McCollum et al. [16] that it has a GLOM of 372 kg.

The second concept that has been assessed by McCollum et al. for the MAV is a two-stage solid propellant rocket [16]. Solid rockets have already flown on Mars for instance to propulsively land the Curiosity or Perseverance rovers, making this technology flight-proven, and this concept overall less complex. This concept has a preliminary GLOM of 384 kg and can be seen illustrated in Figure 2.6.
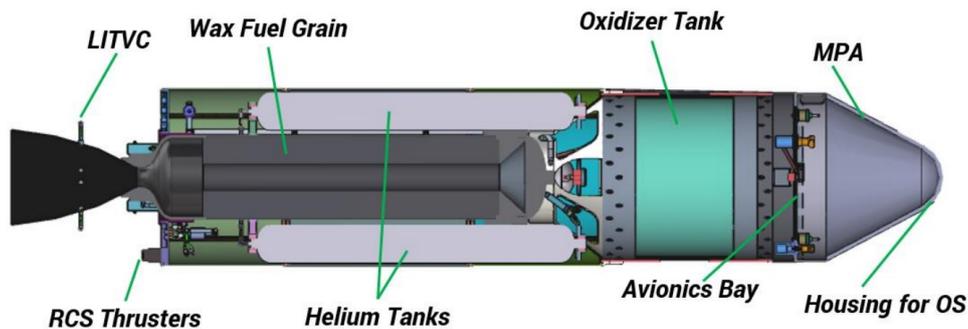
Figure 2.5: Hybrid propulsion concept for the Mars Ascent Vehicle [16].
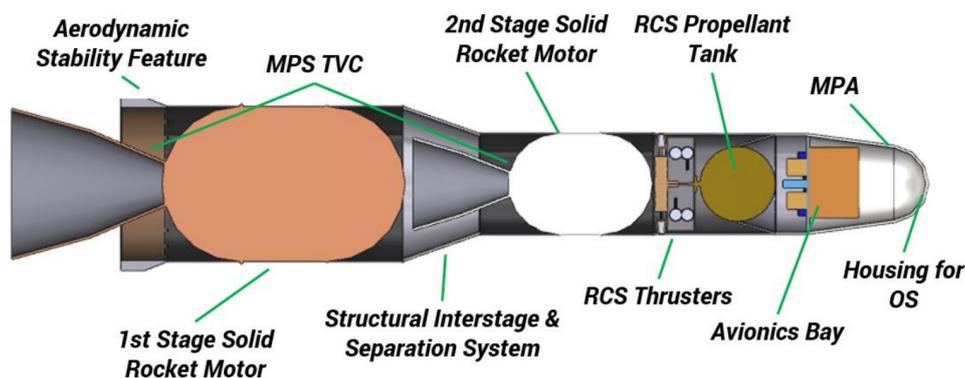
Figure 2.6: Solid propulsion concept for the Mars Ascent Vehicle [16].

Yaghoubi [17] has summarised the advantages of disadvantages of both concepts. Ultimately, the solid propellant rocket has been selected as the concept to develop further for the MSR mission, mainly due to the Martian flight heritage of solid rockets and the higher Technology Readiness Level of electro-mechanical TVC. The main disadvantage of the two-stage solid rocket propellant rocket is that it is impossible to throttle its engine during flight. This decision has been made official in December 2019, and only the design of the solid MAV has been evaluated further since [1].

Subsystems

With the two-stage solid rocket propellant concept selected, a more in-depth preliminary design has been made by JPL, the latest update of it found to be in September 2020 [18]. This section describes the following relevant subsystems of the MAV: the solid motors, the thrust vectoring control, the reaction control system, and the launch system.

Motors   The solid motor of the first stage is cylindrical. The solid motor of the first stage is cylindrical. As identified by Jackman et al. [14], in the simulation that has been run to assess the feasibility of this concept, it needs to provide a specific impulse of about 293 s, an average of 9854 N of thrust during a burn time of 75.9 s, and using 216 kg of solid propellant. The second stage motor has been decided to be spherical, with a specific impulse computed to be of about 282 s, burning 54 kg of propellant

during 24.5 s to generate an average thrust of 6937 N. In this first design, the area ratio of the first stage nozzle is 81 [15], while it is 73 for the nozzle of the second stage [19]. Also, the throat area of the first stage is 0.065 m^2 [19], while the throat area of the second stage is 0.005 m^2 [19]. This means that the throat of the nozzle has a radius of 14.4 cm for the first stage, and of 4.55 cm for the second stage. Both of these motors are to burn from the inside out [1]. Furthermore, all elements of the structure of the MAV are to be made of metal, mainly in aluminium. The main exception is with the titanium enclosure of the solid propellant motors [14]. Last but not least, the solid propellant that has been selected for both stages is a Hydroxyl Terminated PolyButadiene (HTPB) propellant, with a mixture of Ammonium Perchlorate (AP) and Aluminium (Al), named TP-H-3544 [20].

Thrust Vectoring Control   As discussed earlier, the two-stage solid MAV makes use of an electro-mechanically activated TVC. As discussed by Yaghoubi et al. [1], the TVC provides pitch and yaw control during the motor burn. It does so using a similar nozzle to the one of the ASAS 13-30V motor from Northrop Grumman, offering thrust vectoring of 5 deg from the exhaust centreline [14]. Furthermore, as discussed by Yaghoubi and Ma [21], the vectored nozzle uses a ball joint supersonic split line design with a trapped ball. Such design can be seen illustrated in Figure 2.7. Lastly, in September 2020, it has been decided to equip only the first stage motor with TVC, in an attempt to keep the mass of the MAV lower [18]. However, to keep the second stage stabilised during the burn of its motor, 2 spin and 2 de-spin solid motors have been added, to rotate the entire stage between 5 and 10 rpm before firing its motor. This update in the architecture of the MAV can be seen in Figure 2.8.
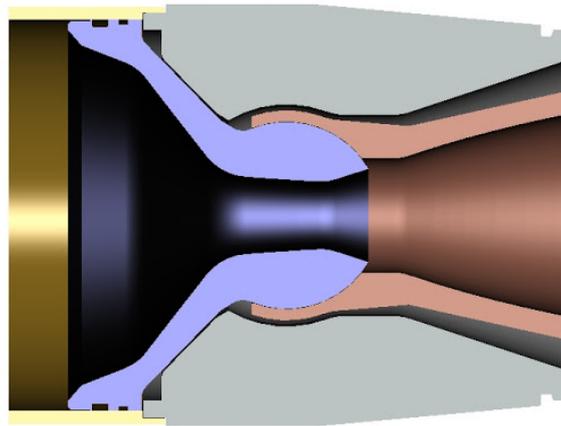
Figure 2.7: Example of supersonic split line nozzle with a trapped bal design [21].
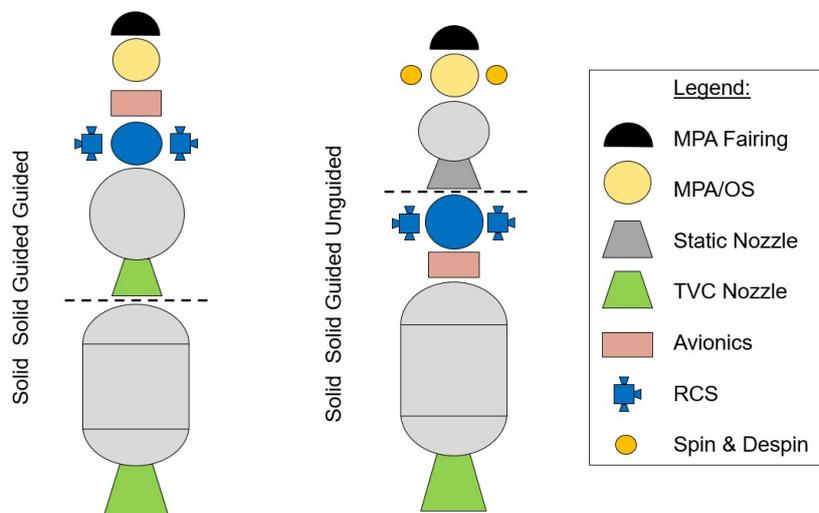
Figure 2.8: Updated architecture for the MAV [18].

Monopropellant Reaction Control System   As discussed by Yaghoubi and Ma [21], the RCS of the preliminarily designed MAV use a hydrazine monopropellant to provide roll control during first stage

motor burn, and full control during coasting. The assembly for the RCS can be seen in Figure 2.9, showing the hydrazine tank in orange, three thrusters on each side in red for roll, yaw, and pitch control. Also, the original design included two axial thrusters, as seen in red at the bottom of Figure 2.9. These thrusters were placed to fine-tune the final orbit, as to reduce orbital deviations [21]. However, with the design update of September 2020, it was decided to remove these axial thrusters, and for RCS to be placed only in the first stage. As previously discussed and as shown in Figure 2.8, spin and de-spin thrusters have been placed on the second stage, as to spin the vehicle during the second stage burn and reduce deviations due to thrust misalignment.
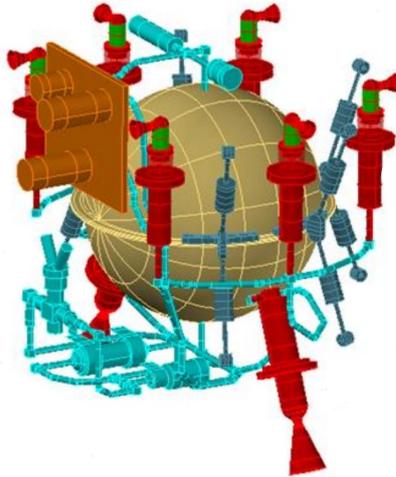
Figure 2.9: RCS assembly showing the hydrazine tanks and the thrusters [21].

Launch system   It was first assumed that the MAV would be launched using a launch rail. However, as discussed by Yaghoubi [18], a Vertically Ejected, Controlled Tip-Off Rate (VECTOR) launch has been selected instead. With it, the MAV will be launched from the surface of Mars by being launched in the air by a system of spring-loaded arms, placing the MAV in a ballistic trajectory at an angle of 30 deg to 60 deg from the horizontal when the first stage motor ignites [21]. This is illustrated in Figure 2.10.
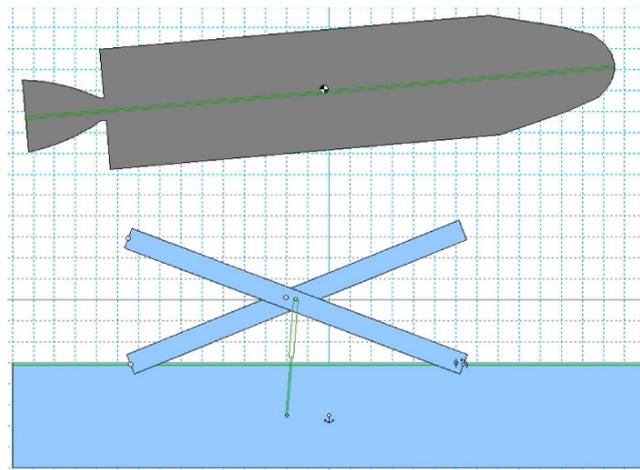
Figure 2.10: Vertically Ejected, Controlled Tip-Off Rate launch of the MAV [21].

2.3. Research proposal

NASA and ESA have laid a plan to return samples from Mars in an ambitious mission that consists of the different sub-missions discussed in Section 2.1. The first one, Mars 2020, is already in operation. The last one, consisting of the re-entry of the Mars samples on Earth, is set to take place in mid-2031. As part of this global mission, a so-called Mars Ascent Vehicle will be used to bring the samples from

the Martian surface to an orbit around Mars.

It has already been pre-defined that the MAV will consist of a two-stage solid rocket, as mentioned in Section 2.2. However, a detailed launch profile still has to be established. This is the research that is proposed in this report. Such launch profile is to be defined on one hand by optimising the use of thrust vectoring control during ascent, and on the other hand by optimising the cross-sectional geometry and size of the solid motors. The latter directly characterises the MAV thrust profile.

The research will thus consist in modelling the vehicle, simulating its environment and launch, and finally optimising the different design variables that parametrise its ascent and solid motor geometry with the goal of minimising the mass, and the deviation from the target orbit.

Also, this research offers the pretext of studying how accurate a numerical simulation needs to be in such a context, where two-stages are used, both of which are quite light, with powerful solid motors. This means that the acceleration on the vehicle is high, making the dynamics of the problem at hand more complex, and offering the opportunity of studying the feasible and required accuracy of such numerical simulation.

This section presents the research proposal. In a first time, the research questions are laid out. The research objectives are then discussed in a second time.

2.3.1. Research questions

The main research question has first been formulated. It consists of the following:

Main research question	*What is the optimum trajectory and solid propellant geometry for the Mars Ascent Vehicle to bring samples from the Martian surface to a defined Martian orbit?*

From this research question, two main sub-questions arise. The first one is axed more on the physical vehicle itself, and more practical manners, while the second one relates more to the sensitivity of the design and of the simulation that will be carried during the research. Both of these sub-questions have then been detailed into 5 more detailed questions. Once all of these detailed questions are answered, the main research question will be answered too.

Sub-question 1	*What vehicle configuration and control allows to reach the target orbit with the highest accuracy and lowest mass?*
Sub-question 1.1	*What cross-sectional geometry and length of the first stage solid motor gives the best orbital accuracy at lowest mass?*
Sub-question 1.2	*What inner and outer diameter of the second stage spherical solid motor gives the best orbital accuracy at lowest mass?*
Sub-question 1.3	*How should thrust vectoring control be used through ascent to reach the target orbit with the highest accuracy?*
Sub-question 1.4	*What is the minimum burn time of both stages combined that allows to reach the target orbit?*
Sub-question 1.5	*What is the minimum Mars Ascent Vehicle Gross Lift-Off Mass that allows to reach the target orbit?*
Sub-question 1.6	*What are the optimum launch and stage separation angles to reach the target orbit with minimum Gross Lift-Off Mass?*
Sub-question 2	*How do key simulation variations impact the overall propagated states of the Mars Ascent Vehicle and the final payload orbit?*

| **Sub-question 2.1** | *How does the initial launch angle impact the final propagated state?* |

| **Sub-question 2.2** | *What choice of integrator, propagator, and environment leads to the best combination of speed and accuracy for the numerical simulation of the propellant burn and ascent trajectory?* |

| **Sub-question 2.3** | *How do deviations in the environment affect the simulated states of the Mars Ascent Vehicle?* |

| **Sub-question 2.4** | *How do deviations in the initial state affect the final propagated state and final orbit?* |

| **Sub-question 2.5** | *How stable is the final orbit of the second stage at the minimum target altitude?* |

2.3.2. Research objectives

With the research questions elaborated, the objectives of the proposed research can now be set up. The main research objective is related to the main research question and consists in the following:

| **Main research objective** | *To design the lightest possible solid motors for the Mars Ascent Vehicle to reach a target orbit, by finding the best compromise in CPU speed and numerical accuracy for an ascent and propellant burn simulations, and optimising the launch profile and solid motor geometry.* |

Once again, multiple sub-objectives can be identified, which are smaller goals that have to be reached in order to fulfil the main research objective.

| **Sub-objective 1** | *To setup an alterable thrust profile by modelling a parametrised solid motor geometry and simulating its burn.* |

| **Sub-objective 2** | *To setup an ascent trajectory simulation framework by selecting appropriate integrators, propagators, and environmental accelerations.* |

| **Sub-objective 3** | *To setup an optimisation framework of both the trajectory and the solid propellant by selecting appropriate design variables and their allowable range, and appropriate optimisation objectives.* |

| **Sub-objective 4** | *To find the solid propellant geometry and control during ascent that lead to the lowest mass and highest orbital accuracy by optimising the numerical simulation of the solid propellant and ascent.* |

| **Sub-objective 5** | *To investigate the sensitivity of the solutions by varying different combinations of simulation, optimisation, and design parameters.* |

2.4. Relevant previous work

As described in Section 2.2, the work carried out by JPL offers a strong basis for the research described in this report. However, the research gap that was identified in this feasibility study is that no accurate ascent simulations were run, that only an estimated and simplified thrust profile was used, and that no optimisation of neither the thrust profile nor the MAV was carried [19].

In addition to the design from JPL, the work of S. D. Petrović [22] from 2017 has been studied. In it, the ascent trajectory of the MAV has been studied, with a focus on the use of Taylor Series Integration for the numerical simulation. From this study, multiple recommendations were given, the ones being most relevant to the work of this report being In addition of the design from JPL, the work of S. D. Petrović [22]:

- Perturbing accelerations such as higher-order gravity terms or third-body effects should be added to the simulation.
- A more complex model of thrust should be explored, whereas constant thrust was used.
- Thrust angles in the form of for instance TVC should be incorporated into the simulation to tune the ascent.
- Optimisation of the trajectory should be attempted.

Furthermore, different papers have been written about the Mars Ascent Vehicle, studying different mission scenarios, in different ways. Desai, P. et al. [23] studied a different MAV configuration entirely in 1998, setting the vehicle more as a lifting platform than as a rocket. Then, Whitehead, J. [24] studied propulsion trades for the MAV with a maximum GLOM of 100 kg, whereas that value was set to 400 kg by JPL. Further, Di Sotto, E. et al. [25] studied the MAV system and GNC for rendezvous in a target orbit with a periapsis of 300 km and apoapsis of 2000 km, where JPL set the target altitude range of 300 km to 375 km. Finally, the research carried out by Benito, J. and Johnson, B. [26] is the closest to the one carried out in this report. Still, their target altitude was 500 km, they only explored one solution for the configuration discussed in Section 2.2, and they used a constant thrust.

Lastly, in the study of JPL on the MAV, optimisations of the motor have been started, using the Solid Performance Program to carry the SRM burn simulation [27]. This software simulates the burn of the propellant grain itself, in 3D. It was found by Hetterich [27] that the burn simulation is then very CPU intensive, taking considerate CPU time to run, making the optimisation of the motor challenging because a high number of simulations are to be run. This also means that coupling the SRM optimisation with an ascent optimisation was out of the question.

The gap from the research of JPL, the thesis of S. D. Petrović, and other studies, left the door open for the research of this report as proposed in Section 2.3. In particular, the coupling of a fast propellant burn simulation with the ascent simulation, and with an optimisation, allow for a more complex MAV ascent model. In turn, this allows exploiting how drag, gravity, and thrust all affect one another, opening the doors to finding new ascent trajectories for a MAV at a lower mass, while fitting the design envisioned by JPL.

3

Methodology

This chapter describes the different methods that have been used to simulate and optimise the trajectory and design of the MAV, with the goal of answering the research laid out in Chapter 2. First, all of the models that have been setup to numerically represent the MAV, its environment, and its motors, are detailed in Section 3.1. Then, a discussion is made in Section 3.2 on how the models are assembled in a numerical simulation framework, to simulate the ascent of the MAV. Finally, in Section 3.3, details are given on the setup that has been established to optimise both the motors and the trajectory of the MAV.

3.1. Modelling

This section describes the different models that have been made and implemented to numerically represent the MAV with respect to its aerodynamics and mass. Then, this section presents the models used to represent the Martian environment in which the MAV will launch and orbit. Finally, a discussion is made on the numerical models setup to represent the performance of the solid motors used by the MAV.

3.1.1. Vehicle models

This section discusses three different models that have been set up for the MAV itself. These models cover aerodynamics using two different methods, one for each stage, as well as how mass was modelled. The aerodynamics properties of the MAV have been modelled in two distinct ways. From lift-off to stage separation, the different aerodynamics coefficients have been obtained using the Missile DATCOM software [28]. Then, after stage separation, only the drag of the second stage is taken into account, using a Direct Simulation Monte Carlo (DSMC). Aerodynamics from both methods are to be used for the ascent trajectory, while aerodynamics of the second stage, from DSMC, is also of particular interest to analyse the stability of the final orbit. Then, this section discusses how the mass of the MAV has been modelled as a function of its motors. Lastly, it is worth noting that, while the solid motors are an important part of the MAV, their model is not described in this section, but in Section 3.1.3.

Aerodynamics from Missile DATCOM
This software can be used to input the geometry of the vehicle, and to get aerodynamic coefficients as a function of the angle of attack and Mach number. To do so, Missile DATCOM uses semi-empirical relationships [28]. While this method allows for both force and moment coefficients to be gathered, only the force coefficients will be used for this research.

To specify the geometry of the MAV to Missile DATCOM, a Computer-Aided Design (CAD) has been generated, based on the literature discussed in Section 2.2. Discrete points have then been taken along this CAD so that their coordinate can be input to Missile DATCOM. Then, the software interprets these points, fits a line through them, and revolves this line around a centerline, which is the centerline of the rocket. This means that, in the implementation used, Missile DATCOM computes the aerodynamic properties of the MAV as a vehicle that is radially symmetric.

Furthermore, as inputs of Missile DATCOM, angles of attacks ranging from -4 deg to 4 deg have been specified, with increments of 0.5 deg. This is because, as mentioned in Section 2.2, the angle of attack of the vehicle shall not exceed 4 deg.

Also, the Mach number have been set to vary between 0.3 and 12. The orbital velocity of a satellite in a circular orbit around Mars at the minimum required MAV altitude can be computed using

Equation 3.1, with said velocity $V_{\text{Mars,circ}}$ [m/s], the orbital altitude h [m], Mars radius R_{Mars} [m], and the gravitational parameter of Mars μ_{Mars} [kg·m^3/s^2] [29]. This gives an orbital velocity of around 3.4 km/s or Mach 9.9. Adding some margins, and starting with a non-zero velocity otherwise aerodynamics would be irrelevant, this is why the Mach number has been varied between 0.3 and 12.

$$V_{\text{Mars,circ}} = \sqrt{\frac{\mu_{\text{Mars}}}{r}} = \sqrt{\frac{\mu_{\text{Mars}}}{R_{\text{Mars}} + h}} = \sqrt{\frac{4.282837 \cdot 10^{13}}{(3389.5 + 300) \cdot 10^3}} = 3407.078 \text{ [m/s]} \qquad (3.1)$$

Finally, the flight conditions input of Missile DATCOM has been set to a given altitude on Earth, as Missile DATCOM only supports flight conditions on Earth. The atmospheric density at zero altitude on Mars is of an average of 0.02 kg/m^3. According to the U.S. Standard Atmosphere, this corresponds to an altitude on Earth of 30 km [30], which is then the altitude used to input the flight conditions.

Aerodynamics from DSMC

In a successful ascent trajectory, the second stage of the MAV flies at orbital altitude. While the effect of the atmosphere at such altitude becomes almost insignificant for the ascent, the research **Subquestion 2.5** from Section 2.3.1 requires the orbital decay of the second stage to be studied with accuracy. This means that the aerodynamics of the second stage of the MAV are still to be studied.

Given the altitude, the atmospheric flow that the second stage encounters is most likely to be rarefied. It has thus been decided to analyse the aerodynamic drag of the vehicle using a Direct Simulation Monte Carlo.

Kinetic theory is the foundation on which the DSMC method is built [31]. In essence, DSMC simulates the macroscopic displacement of individual particles in straight lines, checking for collisions between the particles or with surfaces that are placed in the simulation volume. Furthermore, in kinetic theory, three assumptions are made:

- The density of the rarefied flow is very low, and the gas is dilute. The mean free path is thus higher than the size of the particles.
- The high number of particles justifies the use of a probabilistic model for the kinetic problem.
- The particles exert no forces on each other besides when they collide.

Because simulating all of the particles would be too CPU intensive in most applications, only a fraction of the particles are simulated. Thanks to the aforementioned assumptions, DSMC allows for a reduction in the number of simulated particles without sacrificing the accuracy of the results. This is because a decrease in particle number can be statistically compensated by an increase in the size of the particles, carefully treating how the collisions are computed [32].

Additionally, a Variable Soft Sphere model has been used to approximate the elastic interactions between the particles [33]. This helps fit the Schmidt number of the interactions, representing how much momentum and mass are being diffused.

More details about the models of kinetic theory and DSMC are given by P. Parodi in its thesis about the analysis and simulation of an ABEP intake in very low Earth orbit [34]. The books of Boyd and Schwartzentruber [32], and of Bird [35] about DSMC are also recommended.

$\alpha = 0$ $\qquad\qquad\qquad\qquad\qquad$ $\alpha = 1$
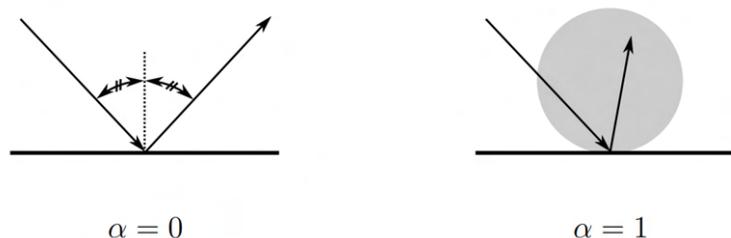
Figure 3.1: Specular and diffuse reflection models [34].

Finally, the concept of specular and diffuse reflection is of importance. As illustrated in Figure 3.1, a surface with an accommodation coefficient α of 0 means that a particle colliding with a surface will

reflect in a perfectly specular way, with the reflection angle equal to the incidence angle. Conversely, when $\alpha = 1$, the reflection of particles colliding with a surface is done in a perfectly diffuse manner, according to Lambert's cosine law.

For the second stage of the MAV, it has been decided to use an accommodation coefficient based on the absorption of atomic oxygen. This is following the semi-empirical model established by Pilinski et al. [36] that suggests the use of Equation 3.2 to use as the accommodation coefficient for a satellite in low Earth orbit, with the temperature T [K] and the oxygen number density n_o [1/m³]. It has been decided to use the same law for the CubeSat in VLMO. As shown in Figure 3.2, this leads to accommodation coefficients above 0.92 for temperatures and oxygen number densities corresponding to altitudes above Mars from 90 km to 200 km.

$$\alpha \approx \frac{7.5 \cdot 10^{-17} \cdot n_0 \cdot T}{1 + 7.5 \cdot 10^{-17} \cdot n_O \cdot T} \tag{3.2}$$
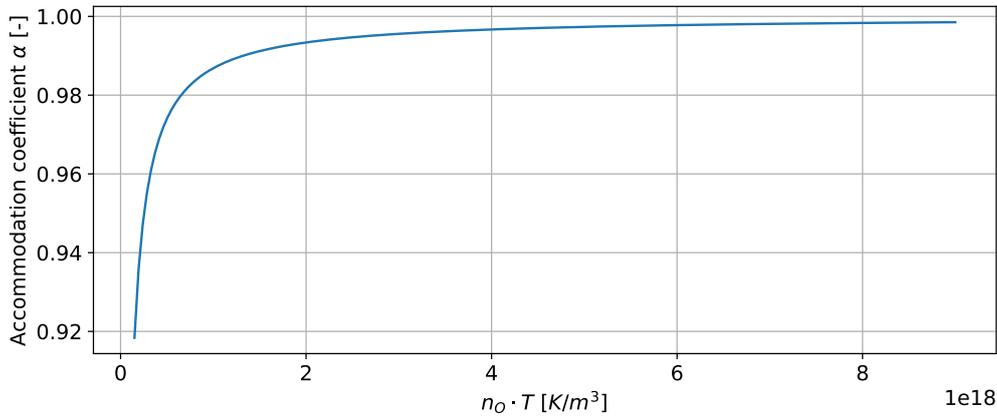
Figure 3.2: Accommodation coefficients computed for oxygen number densities and temperatures corresponding to altitudes above Mars from 90 km to 200 km.

Simulation constraints   have to be respected for the DSMC simulation to be valid, as discussed by P. Parodi [34] First of all, the simulation time step Δt [s] has to be smaller than the mean collision time τ [s] divided by 5, as shown in Equation 3.3. This is to ensure that the DSMC method has enough time to properly compute the collision between the particles.

Then, Equation 3.4 shows that the spacing of the simulation grid Δx [m] has to be at least five times smaller than the mean free path of the flow λ [m]. This is to reduce the error associated with the cell dimension.

In addition, a constraint is set on both Δx and Δt, so that the flow velocity u [m/s] multiplied by the simulation time step is above the grid size, as in Equation 3.5. Then, a constraint is stating that there should be at least 10 simulated Particles Per Cell (PPC). This is to allow for particle interactions to develop, and is represented by Equation 3.6.

$$\Delta t < \frac{\tau}{5} \quad (3.3) \qquad \Delta x < \frac{\lambda}{5} \quad (3.4) \qquad \Delta x < u \cdot \Delta t \quad (3.5) \qquad \text{PPC} > 10 \quad (3.6)$$

Lastly, a DSMC method can be applied for any flow, but is most efficient when used for a rarefied flow. The flow can be characterised using Equation 3.7 to compute the Knudsen Kn [-], using the mean free path λ [m] and the reference length of the MAV L_{ref} [m]. Then, from the following categorisation can be made [37]:

- $Kn < 0.01$: Continuum flow.
- $0.01 < Kn < 0.1$: Slip flow.
- $0.1 < Kn < 10$: Transitional flow.
- $Kn > 10$: Free-molecular flow.

$$Kn = \frac{\lambda}{L_{\text{ref}}} \tag{3.7}$$

Stochastic PArallel Rarefied-gas Time-accurate Analyzer   (SPARTA) is the open-source software that has been used to run the DSMC [38]. Specialised for rarefied flow simulations, it can run on multiple processors in parallel, allow for triangulated objects to be placed in the 3D simulation volume, and can automatically split the simulation domain and balance the load between each processor.

A script has been written in Python so that the input file required to run the DSMC related to each satellite configuration at each altitude can be automated. This script automatically adjusts all of the parameters related to the atmosphere at the given altitude, such as the flow velocity, the molecular composition, the temperature, and the density. It also adjusts the size of each satellite to generate an adapted simulation volume.

Furthermore, running SPARTA with a uniform grid that respects all the constraints from Equations 3.3 to 3.6 would be too CPU intensive. Instead, the simulation grid size Δx is initially adapted to the free stream, resulting in a very coarse grid. The time step Δt is then taken equal to the minimum of $\tau/5$ and $\Delta x/u$. Then, the number of simulated particles is set relative to the number of cells in the simulation volume so that the number of PPC is above 10.

This DSMC is then run until the number of particles that get into the simulation volume at orbital velocity equals the number of particles getting out of it, hinting that the simulation stabilises. This ensures that the simulation volume is filled with particles and that these particles started accumulating where they should when they get compressed into the atmosphere inlet.

As this point, the Knudsen number of the grid Kn_{grid} [-] is computed everywhere on the grid, using Equation 3.8. The grid size Δx is then halved where this Kn_{grid} is below 5, to enforce the constraint from Equation 3.4. At the same time, the simulation time step Δt is halved. Also, the macro-particle weight is decreased, which can be analogously seen as increasing the number of particles in the simulation volume everywhere as well as of the particles that enter it. This refinement is tuned so that the number of PPC is still above 10 after it is carried. The grid is finally re-balanced between the processors to keep the operation as efficient as possible.

This grid refinement process is carried 3 times so that all of the aforementioned conditions are respected. Using progressive refinement based on Kn_{grid} allows for some places in the simulation grid to be refined more where needed, instead of uniformly refining the grid everywhere.

$$Kn_{\text{grid}} = \frac{\lambda}{\Delta x} \qquad\qquad (3.8)$$

Atmospheric conditions   needed for a DSMC are more elaborate than for Missile DATCOM. This is explained as a fraction of the rarefied flow itself is simulated, and so are the particles composing the atmosphere. This contrasts with the semi-empirical relations used by Missile DATCOM.

The first parameter that is required for the DSMC is the velocity of the vehicle. This is computed as a function of altitude using once again Equation 3.1, assuming that the vehicle is at orbital velocity. Then, the atmospheric mass density, temperature, and pressure are all required. Finally, the mixture of the atmosphere is an important input, representing the molecular composition of the atmosphere. In the implemented setup, the mixture is input as mol per mol, and then converted to a number density[1], making use of the atmospheric mass density and of the molar mass of each atmosphere component. All of these inputs, with the exception of the velocity, have been taken from the Mars Climate Database [39], averaging values over latitude, longitude, and time.

Finally, while the target altitude of the MAV is between 300 km and 375 km, it has been decided to sample these atmospheric conditions for 13 points between 100 km and 500 km above Mars. This allows for the simulation that will study the orbital decay to run at slightly higher altitudes if needed, and to be more precise when the MAV starts de-orbiting to lower altitudes.

Mass

The mass of the MAV is modeled by following analytical relations based on similar vehicles tuned based on the preliminary design of the MAV by JPL.

First of all, the propellant mass of both stages is computed by multiplying the volume of propellant by the propellant density. Since the geometry of the solid motors is parametrised, it is possible to compute the propellant volume for each set of geometry parameters. This is described in Section 3.1.3.

---

[1]As opposed to the mass density that represents the mass of the atmosphere in a given volume, the number density represents the number of particles is a given volume. The number density then has a unit of $1/m^3$.

The density of the propellant is also discussed in Section 3.1.3, in the propellant specifications. This then allows to compute the propellant mass of both the first and second stage motors for any motor parametrisation.

Next, the mass of the casing and nozzle of the motors are to be computed. To do so, Equation 3.9 has been used, adapted from [40]. This equation can be used to get the motor inert mass M_{inert} [kg] from the propellant mass M_P [kg], and is fitted from various different solid motors that all have a case made of titanium, as will be for the MAV. This relation was made based on 20 data points and has a coefficient of determination of 0.985. The equation has been adapted by adding a constant mass of 10 kg to account for TVC, and to fit the motor mass indicated by JPL in its MAV preliminary design [19].

$$M_{\text{inert}} = 10 + 0.2833 \cdot M_P^{0.789} \tag{3.9}$$

Finally, the mass of each stage of the MAV without any motors is still to be computed. This has been done by taking values given in the preliminary design of the MAV by JPL [19]. From it, it was found that the entire vehicle without any motors has a mass of 112.5 kg. To split this mass between the first and second stage, the following has been taken into account: RCS is only in the first stage and has a mass of 9.4 kg [19]; the second stage contains the 16 kg payload, avionics, and spin thrusters; the first stage shall have a sturdier structure than the second stage due the higher mechanical load on it during ascent. Given all these considerations, it has been decided to split the motor-less mass of the MAV to 65 kg for the first stage, and 47.5 kg for the second stage.

In conclusion, the empty mass of both stages is fixed, the propellant mass of both stages can be computed from the motor geometry, and the inert mass of the motors can be computed from the propellant mass. Combining all of this together allows to compute the GLOM of the MAV for any solid motor configuration, as well as the mass of the second stage only, after stage separation.

3.1.2. Environment models

Following the models used to represent the MAV, the modelling of the environment in which it flies is discussed in this section.

Mars and Earth comparison

It is important that the main differences between the Earth and Mars are kept in mind. To this extent, Table 3.1 has been created.

Parameter	Unit	Earth	Mars (% of Earth)
Surface gravity	m/s^2	9.81	3.71 (37.8%)
Surface air density	kg/m^3	1.225	0.02 (1.6%)
Average temperature	K	288	210 (72.9%)
Mean radius	km	6371.0	3389.5 (53.2%)
Escape velocity	m/s	11.19	5.03 (45%)
Gravitational parameter	$10^6 \cdot$ km^3/s^2	0.3986	0.04283 (10.7%)
Solar irradiance	W/m^2	1361.0	586.2 (43%)
Day length	hrs	24	24.6597 (102.7%)
Distance from Sun	AU	1	1.5236 (152.4%)

Table 3.1: Mars and Earth main differences [29].

Ephemeris

If the launch of the MAV is to be simulated, it is important that the state of certain bodies be known as a function of time. Such bodies include for instance Mars, its moons, and the Sun.

The most complete and open way to do so is by using the SPICE Toolkit [41, 42]. Part of this library compiles the state of all of the important bodies in the Solar system. This state include information such as the position and orientation at a given moment in time. As explained by Dirkx and Mooij [43, p.28], full state consistency of a certain set of bodies can be reached only if all of the bodies that exert an acceleration are propagated together. This means that, taking the Solar system as an example, the state of all of the planets, massive Moons and planetoids, and of the Sun need to be propagated together, using an n-body simulation. This explains why using pre-compiled data from a toolkit such as SPICE is useful thanks to the precious CPU time that can be saved.

Mars atmosphere

The atmosphere of Mars is of importance for the MAV flight since it is expected that aerodynamic effects will influence the ascent trajectory. This subsection focuses on the ways in which the Martian atmosphere can be modelled. More details on the aerodynamic of the MAV itself are given in Section 3.1.1.

Exponential atmosphere is a simplified model that can be used to compute the density ρ [kg/m^3] of the Martian atmosphere at a given altitude h [m]. The single equation of this model is given in Equation 3.10, taken from Wertz et al. [44]. In it, ρ_0 [kg/m^3] is the density at the bottom of an arbitrary layer of the atmosphere, h_s [m] is the so-called scale height for the given layer, and Δh [m] is the difference between the altitude of the vehicle and the bottom of the atmosphere layer in which it is.

$$\rho(h) = \rho_0 \cdot \exp\left(\frac{-\Delta h}{h_s}\right) = \rho_0 \cdot \exp\left(\frac{-(h - h_0)}{h_s}\right) \tag{3.10}$$

The most common values used for the Martian atmosphere is to use a value of $\rho_0 = 0.02$ kg/m^3 and a scale height of $h_s = 11100$ m [45]. These values then assume that the Martian atmosphere follows an exponential profile across its entire altitude range.

A different exponential model has been fitted to increase the accuracy of the density for higher altitudes. Indeed, it was found by P. Gallais that splitting the model in two distinct steps, with a discontinuity at 25 km, allows for a better fit of the Martian atmosphere up to at least 125 km [45]. The first part of the atmosphere then starts with a density at 0 km of $\rho_0 = 0.0159$ kg/m^3, and has a scale height of $h_s = 11049$ m. The second part of the atmosphere starts at 25 km with a density of $\rho_0 = 0.0525$ kg/m^3, and has a scale height of $h_s = 7295$ m.

Mars Climate Database (MCD) is a more detailed atmospheric model can be obtained for the Martian atmosphere [46, 47]. Much like Earth, Mars has a predictable weather. Variations in the physical and chemical properties of its atmosphere due to for instance the topography or seasonal cycles can be modelled. From this, the MCD contains information such as air density and temperature at given altitudes, times, and localisation on Mars. The MCD also provides an online interface [48], which allows to easily get information such as the air density as a function of the altitude (at a fixed localisation) or as a function of the localisation (at a fixed altitude). This is graphically illustrated in Figure 3.3. Atmosphere information is available from the MCD tabulated as a function of the altitude, with logarithmic spacing varying from 250 m to 50 km, and averaged over longitude, latitude, and time. Also, the same information is available tabulated as a function of the latitude, longitude, and altitude, averaged over time [49].
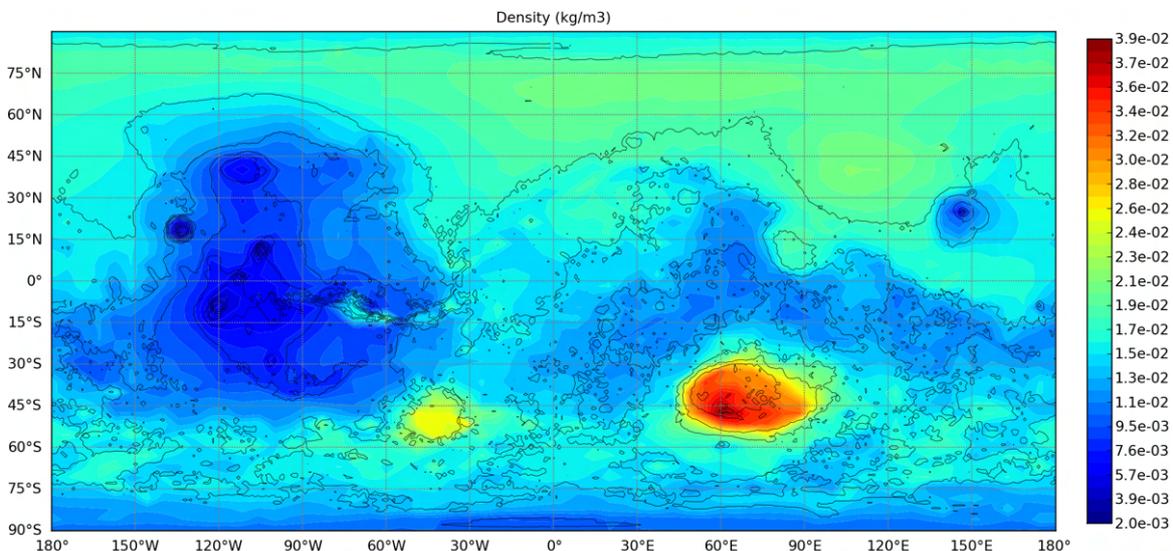
Figure 3.3: Predicted air density on Mars at an altitude of 10 m for the 6th of June 2029 [48].

Mars GRAM 2010   is another detailed atmospheric model for Mars. As discussed by Justus and Johnson [50], this engineering-oriented atmospheric model was first based on the data gathered during the Mariner and Viking missions. However, this model is nowadays improved with data from the NASA Ames Mars General Circulation Model and the University of Arizona Mars Thermospheric General Circulation Model [51]. The input of both of these models has been tweaked so that the Mars-GRAM model fits the measurements made by the Mariner and Vikings missions. Unfortunately, it has been decided from the start to not use the Mars-GRAM model, since it is found to be less accurate than the MCD [45], and since this model is not available at altitudes above 250 km.

Mars gravitational field

There are two distinct way to account for the impact of a massive body acting on another one. First, the point mass model will be described, followed by spherical harmonics (SH).

Point mass model   The acceleration that a certain body B exert on a smaller body A can be written as Equation 3.11, with the gravitational parameter of body B, μ_B, and the difference in position between the two bodies, r_{BA}. This relation holds only if the mass of body A can be assumed negligible compared to the mass of body B. From it, one can see that the acceleration between the two bodies varies with the inverse of the square of the distance between them.

$$a_{BA} = \frac{\mu_B}{||r_{BA}||^3} r_{BA} \tag{3.11}$$

Spherical harmonics   While the point mass contributions approximate a complete body to a point, variations in the gravitational field around the body are most often important. For instance, mountains, valleys, or crust density variations will induce fluctuations in the gravity field. These fluctuations can be modelled using so-called spherical harmonic expansions [52, pp.63-67]. These are governed by Equation 3.12, as taken from [53]. In it, R [m] is the reference radius of the body exerting the acceleration, $\bar{P}_{lm}$ is the associated Legendre polynomial, $\bar{C}$ and $\bar{S}$ are the spherical harmonic coefficients, and ϕ and θ [deg] are the latitude and longitude of the spacecraft in orbit as seen from the body. The spherical harmonic coefficients $\bar{C}$ and $\bar{S}$ can be taken from the SPICE Toolkit [41, 42] that was discussed earlier in this section.

$$a_{BA} = \frac{\mu_B}{r_{BA}} \sum_{l=0}^{\infty} \sum_{m=0}^{l} \left[\left(\frac{R}{r_{BA}} \right)^l \bar{P}_{lm}(\sin \phi) \cdot (\bar{C}_{lm} \cos m\theta + \bar{S}_{lm} \sin m\theta) \right] \tag{3.12}$$

The spherical harmonic expansions can be grouped in three types: zonal, when $m = 0$, tesseral, when $l > m$, and sectoral, when $l = m$. An illustration of where this represents can be seen in Figure 3.4. Using Equation 3.12 with $l = m = 0$ then comes down to the same acceleration magnitude as with Equation 3.11.

zonal
$l_{10}m_0$

tesseral
$l_{10}m_5$

sectoral
$l_{10}m_{10}$
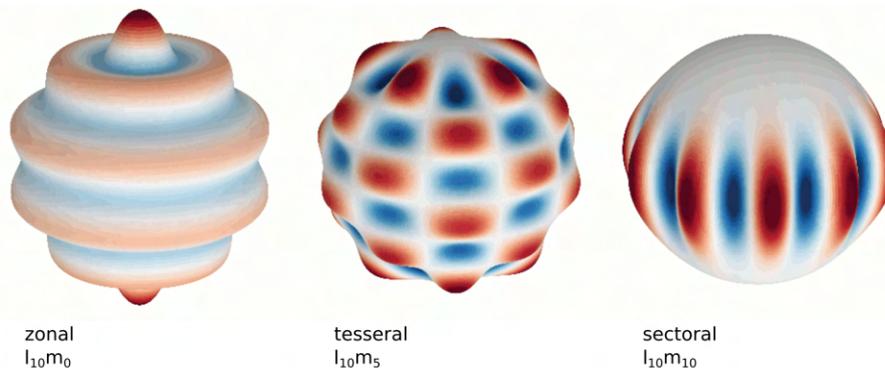
Figure 3.4: Illustration of the three different type of spherical harmonic expansions [54].

Other considerations

The atmosphere and the gravitational field of Mars are the two most important factors that will affect the ascent of the MAV. Still, other accelerations have been considered. These are the pressure of the Sun on the MAV, and the relativistic correction due to the mass of Mars bending space.

Solar radiation pressure   is the phenomenon caused by element of a spacecraft that absorbs, emits, or reflects photons, inducing a force on it. This causes an acceleration on the vehicle that can be modelled.

A simplified model has been used, to avoid having to model every antenna, panel, thruster and similar subsystems of the spacecraft [43, Slide 35]. This first-order model is the cannonball radiation pressure model, approximating the vehicle as a sphere and computing the acceleration of the Solar pressure on it using Equation 3.13 from [55, Slide 8]. In this equation, P [W] is the total power output of the Sun, c [m] is the speed of light in vacuum, C_r is the dimensionless radiation pressure coefficient, S_{ref} [m²] is the effective spacecraft area, m [kg] is the spacecraft mass, and $\hat{r}_{\text{BA}}$ is the unit vector pointing from the Sun to the vehicle.

A value of 1 for C_r means that all of the radiation is absorbed by the vehicle. A value of 2 means that the radiation is completely reflected. These are the two limiting values for C_r.

$$a_{\text{BA}} = \frac{P}{4 \cdot \pi \cdot c} \cdot \frac{C_r \cdot S_{\text{ref}}}{m} \cdot \frac{r_{\text{BA}}}{||r_{\text{BA}}||^3} \tag{3.13}$$

Then, an additional computation is made to check whether the satellite is fully exposed to the Sun, fully in the umbra of Mars, or in the penumbra in-between. This is done by first computing the apparent radius of the Sun $R_{\text{app,S}}$ [m] and of Mars $R_{\text{app,M}}$ [m], and the apparent separation between them d_{app} [m]. This is done using Equations 3.14 to 3.16.

Then, the occulted area A_{occulted} [m²] is computed using Equation 3.19, making use of Equations 3.17 and 3.18 as helpers. Finally, Equation 3.20 is used to computed the fraction of Sun f_{Sun} that gets to the MAV. Equations 3.14 to 3.20 have been taken from Montenbruck and Gill [56].

$$R_{\text{app,S}} = \arcsin\left(\frac{R_{\text{Sun}}}{|r_{\text{Sun}} - r|}\right) \tag{3.14}$$

$$R_{\text{app,M}} = \arcsin\left(\frac{R_{\text{M}}}{|r - r_{\text{M}}|}\right) \tag{3.15}$$

$$d_{\text{app}} = \arccos\left(\frac{(r_{\text{M}} - r) \cdot (r_{\text{S}} - r)}{|r_{\text{M}} - r| \cdot |r_{\text{S}} - r|}\right) \tag{3.16}$$

$$X_s = \frac{d_{\text{app}}^2 + R_{\text{app,S}}^2 - R_{\text{app,M}}^2}{2 \cdot d_{\text{app}}} \tag{3.17}$$

$$Y_s = \sqrt{R_{\text{app,S}}^2 - X_s^2} \tag{3.18}$$

$$A_{\text{occulted}} = R_{\text{app,S}}^2 \cdot \arccos\left(\frac{X_s}{R_{\text{app,S}}}\right) + R_{\text{app,M}}^2 \cdot \arccos\left(\frac{d_{\text{app}} - X_s}{R_{\text{app,M}}}\right) - d_{\text{app}} \cdot Y_s \tag{3.19}$$

$$\begin{cases} f_{\text{sun}} = 0 & if \ (d_{\text{app}} < (R_{\text{app,M}} - R_{\text{app,S}}) \ \text{and} \ R_{\text{app,S}} < R_{\text{app,M}}) \\ f_{\text{sun}} = 0 & if \ (d_{\text{app}} < (R_{\text{app,S}} - R_{\text{app,M}}) \ \text{and} \ R_{\text{app,S}} > R_{\text{app,M}}) \\ f_{\text{sun}} = 1 - \dfrac{A_{\text{occulted}}}{\pi \cdot R_{\text{app,S}}^2} & if \ |R_{\text{app,S}} - R_{\text{app,M}}| < d_{\text{app}} \ and \ d_{\text{app}} < (R_{\text{app,S}} + R_{\text{app,M}}) \\ f_{\text{sun}} = 1 & otherwise \end{cases} \tag{3.20}$$

Relativistic corrections   are needed to correct for the difference between the somewhat simplified Newtonian mechanics that are used over General Relativity. The most important correction is the Schwarzschild term that accounts for the difference in acceleration caused by a point mass in Newtonian and relativistic dynamics. This correction can be computed as in Equation 3.21, taken from [57, p.155]. In this equation, r_{BA} is the position of the spacecraft with respect to the massive body B, and β and γ are the parameterised post-Newtonian parameters, both equal to 1 in General Relativity.

$$\Delta a_{\text{BA}} = \frac{\mu_B}{c^2 r_{\text{BA}}^3} \left\{ \left[2(\beta + \gamma) \frac{\mu_B}{r_{\text{BA}}} - \gamma \dot{r}_{\text{BA}} \cdot \dot{r}_{\text{BA}} \right] r_{\text{BA}} + 2(1 + \gamma)(r_{\text{BA}} \cdot \dot{r}_{\text{BA}}) \dot{r}_{\text{BA}} \right\} \tag{3.21}$$

3.1.3. Solid rocket motor

A model is needed to translate the 3D geometry of a given Solid Rocket Motor (SRM) into a thrust profile. This model first needs to compute the thrust at any given time from the burning surface area of the motor. Then, this model needs to simulate the evolution of the burning surface area of the motor over time. Finally, it is important that this model can be tuned with the parameters that are specific to the type of propellant used. All three of these aspects relative to the required model are discussed in this section.

Rocket thrust

The thrust of a rocket is produced by the combination of the matter that it ejects at a certain velocity and of the difference of pressure between the rocket motor and the environment. As discussed by Sutton [58, Section 2.2], the thrust F_T [N] can thus be computed as in Equation 3.22, with the engine mass flow $\dot{m}$ [kg/s], the exhaust velocity V_e [m/s], the exhaust pressure p_e [Pa], the ambient pressure p_a [Pa], the exhaust area A_e [m^2], and the thrust efficiency η_T [-].

$$F_T = \eta_T \cdot F_{T_{\text{ideal}}} = \eta_T \cdot (\dot{m} \cdot V_e + (p_e - p_a) \cdot A_e) \tag{3.22}$$

The simulation of the solid propellant burn is discretised in time. As such, the mass flow $\dot{m}$ in a time step Δt [s] is computed from the propellant density ρ_p [kg] multiplied by the volume of propellant burned between two time steps, equal to $S \cdot \Delta b$, with S [m^2] and Δb [m] being the burning surface and the change in burned distance between two time steps. This results in Equation 3.23. As shown, this equation can be simplified further, taking into account that the solid propellant regression rate r [m/s] is equal to the burning distance over time: $r = \Delta b / \Delta t$.

$$\dot{m} = \frac{S \cdot \Delta b \cdot \rho_p}{\Delta t} = S \cdot r \cdot \rho_p \tag{3.23}$$

In Equation 3.23, the propellant density ρ_p is a fixed value, related to the propellant type. The burning surface S is related to the geometry of the solid motor and varies over time, as discussed further in this section.

This regression rate r at which the propellant burns is computed using Equation 3.24 [40]. As shown in this equation, r varies with the chamber pressure p_c [Pa]. This equation also makes use of the burning rate exponent n [-] and of the burning rate coefficient a [m/s/Pa^n]. Both of these values are also constant and associated with the type of solid propellant used.

$$r = a \cdot p_c^n \tag{3.24}$$

In the last equation, the chamber pressure is a value that is not constant over time. It is computed during the simulation using Equation 3.25 from [40], with the characteristic velocity c^* [m/s], the propellant density ρ_p, the chamber gas density ρ_c, the burning rate coefficient a [m/s/Pa^n], the burning surface S [m^2], the motor throat area A_t [m^2], and the burning rate exponent n [-]. Under the assumption that the chamber gas density is much lower than the propellant density ($\rho_c << \rho_p$), Equation 3.25 has been simplified.

$$p_c = \left(c^* \cdot (\rho_p - \rho_c) \cdot a \cdot \frac{S}{A_t} \right)^{\frac{1}{1-n}} = \left(c^* \cdot \rho_p \cdot a \cdot \frac{S}{A_t} \right)^{\frac{1}{1-n}} \tag{3.25}$$

All of the parameters in the simplified Equation 3.25 are then either already computed, or fixed values, with the exception of the characteristic velocity c^* [m/s]. This value is computed using Equation 3.26, with the combustion efficiency factor η_c [-], the Vandenkerckhove function Γ [-], the gas constant R_A [J/mol/K], the propellant molar mass M [kg/mol], and the chamber temperature T_c [K]. Assuming a constant chamber temperature, all of the parameters used in this equation have constant values and are related to the propellant type. The Vandenkerckhove function Γ is a constant value that is computed solely based on the specific heat ratio of the propellant γ [-], as shown in Equation 3.27 [40].

$$c^* = \eta_c \cdot c^*_{\text{ideal}} = \frac{\eta_c}{\Gamma} \cdot \sqrt{\frac{R_A}{M} \cdot T_c} \tag{3.26}$$

$$\Gamma = \sqrt{\gamma} \cdot \left(\frac{2}{\gamma + 1}\right)^{\frac{\gamma+1}{2 \cdot (\gamma-1)}} \tag{3.27}$$

Last but not least, the exhaust velocity V_e used in Equation 3.22 is to be computed. This is done using Equation 3.28, which makes use of propellant-specific properties that have been discussed in the previous equations, and of the ratio of the exhaust pressure p_e [Pa] and the chamber pressure p_c [Pa]. The computation of the chamber pressure has already been discussed with Equation 3.25. The exhaust pressure is found by numerically solving Equation 3.29, which contains already known and discussed parameters, and the engine exhaust area A_e [m^2].

$$V_e = \sqrt{\frac{2 \cdot \gamma}{\gamma - 1} \cdot \frac{R_A}{M} \cdot T_c \cdot \left(1 - \left(\frac{p_e}{p_c}\right)^{\frac{\gamma-1}{\gamma}}\right)} \tag{3.28}$$

$$\frac{A_e}{A_t} = \frac{\Gamma}{\sqrt{\frac{2\gamma}{\gamma - 1}\left(\frac{p_e}{p_c}\right)^{\frac{2}{\gamma}}\left(1 - \left(\frac{p_e}{p_c}\right)^{\frac{\gamma-1}{\gamma}}\right)}} \tag{3.29}$$

During the propellant burn simulation, it is important to notice that the results of Equations 3.26 to 3.29 is constant. This means that CPU time can be saved by computing c^*, V_e and p_e/p_c only once.

Also, from Equations 3.22 to 3.29, it becomes apparent that the following propellant-related parameters shall be known before starting to simulate the burn of the solid propellant: γ, a, n, ρ_p, A_t, A_e, M, η_c, η_T, and T_c. The throat and exhaust areas A_t and A_e are also to be known before the propellant burn simulation. From Section 2.2.2, the throat area of the first stage motor was set to be of 0.065 m^2, and the throat area of the second stage motor was set to be of 0.005 m^2.

Finally, the burning propellant surface S as a function of the propellant geometry and time is also to be modeled. The propellant properties and geometric models are discussed in the two following sub-sections.

Propellant specification

As discussed in Section 2.2.2, the propellant selected for both stages of the MAV is TP-H-3544. Since this propellant has been custom-made to answer the needs of the MAV, finding documentation for it was challenging. Also, JPL has been contacted directly for the need of this research, in the hope of sharing information, but to no avail.

This means that the required properties of the TP-H-3544 propellant will have to be derived from different sources and proxies. Also, the code supporting this research will be made in such a way that the propellant properties can be replaced at only one place in the code to adapt the study, if later needed. The following sources have been selected:

- In their report on the preliminary design of the MAV, Q. Bean et al. explicitly mentions that the propellant selected for the MAV burns at a high chamber temperature of 3645 K [19].
- From the Rocket Propulsion Elements book, the density of HTPB/AP/Al propellant is given to be in the ballpark of 1860 kg/m^3 [58].
- From their work on the TP-H-3544 propellant, Dankanich, J. et al. mention that the density of this propellant is higher than the one of TP-H-3062 [20]. Then, in the US Patent 5071495, the density of TP-H-3062 is given to be of 1749 kg/m^3 [59].
- In their studies on internal ballistic of solid propellant motors, Terzic, J. et al. report that the TP-H-3062 propellant has a burning rate exponent of 0.31, and a burning rate coefficient of 4.202 mm/s/Mpa^n [60]. This propellant is used in the STAR 8 motor, that is similar to the one that will be used in the MAV [61].

In addition, the Chemical Equilibrium Application (CEA) [62] and Rocket Propulsion Analysis Combustion (RPA-C) [63] software can both be used to compute the molar mass and the specific heat ratio of a propellant, given its composition, burn temperature, and burn pressure.

Solid propellant geometry

As seen in the previous section, the thrust F_T of a SRM depends on the burning propellant surface area S over time. In the case of a cylindrical SRM, this burning area can be computed by multiplying the perimeter of the burning propellant P [m] by the length of the motor L [m], as in Equation 3.30.

$$S = P \cdot L \tag{3.30}$$

In addition to the variation of perimeter over time, the port surface area A_p [m²] before the motor starts burning is also of importance. This area represents the cross-sectional surface of the propellant. Multiplying the port surface area A_p by the length of the motor L then allows to get the initial propellant volume V_p [m³] and, in turn, the initial propellant mass for a given SRM geometry.

The influence of the cross-sectional geometry of a SRM on the burning perimeter over time, and thus on the thrust profile, can be seen in Figure 3.5.
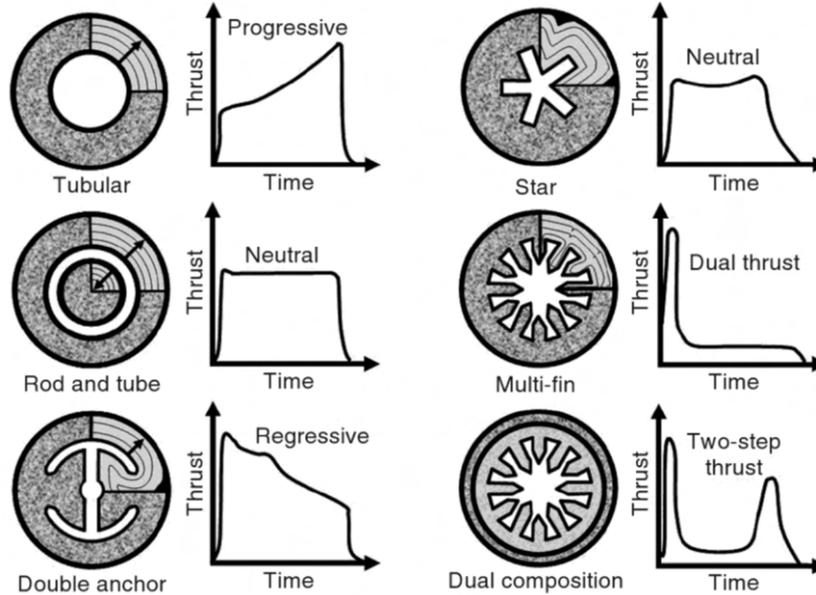
Figure 3.5: Cross-sectional geometry and thrust profile of the most common solid rocket motors [64].

From these cross-sectional geometries, the four selected geometries and associated thrust profiles are the following: tubular, with a progressive thrust; rod and tube, with a neutral thrust; anchor, with a regressive thrust; and multi-fin, with a dual thrust. These geometries are selected as they allow to cover progressive, neutral, dual thrust, and regressive thrust profiles. The evolution of the burning perimeter over time for each of these four geometries is described in the following paragraphs.

Tubular   The propellant burns uniformly from the inside to the outside. Setting R_i [m] as the initial inner radius, and R_o [m] as the outer radius, the burning perimeter is equal to Equation 3.31, with the burned distance b. This parametrisation is graphically represented in Figure 3.6, with the propellant in grey and the burning hot gases in orange. This motor stops burning when $b = R_o - R_i$ is reached. Also, the initial propellant port area is computed through Equation 3.32 for this tubular motor geometry.

$$P_{\text{tubular}}(b) = 2\pi \cdot (R_i + b) \tag{3.31}$$

$$A_{p_{\text{tubular}}} = \pi \cdot (R_o^2 - R_i^2) \tag{3.32}$$
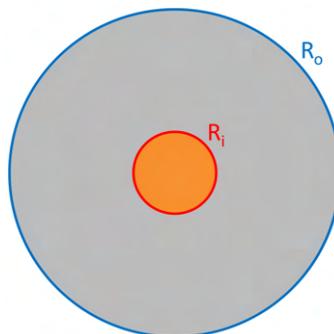
Figure 3.6: Parametrisation of the tubular SRM geometry.

Rod and tube   The solid propellant is composed of two rings. The outer ring, called the tube, has an outer radius R_o and inner radius R_{mid}, and it burns like the tubular SRM, until $b = R_o - R_{mid}$. The inner rod, of radius R_i, burns from the outside to the inside, with a burning perimeter equal to Equation 3.33, until $b = R_i$. As long as none of the tube or rod has burned off, both of these perimeters need to be added together to get the total burning perimeter of the motor, as in Equation 3.34. For this SRM geometry, the propellant port area before ignition is as in Equation 3.35. This geometry and its parametrisation are represented in Figure 3.7.

$$P_{\text{rod and tube, i}}(b) = 2\pi \cdot (R_i - b) \tag{3.33}$$

$$P_{\text{rod and tube}}(b) = P_{\text{tubular}}(b) + P_{\text{rod and tube, i}}(b) \tag{3.34}$$

$$A_{p_{\text{rod and tube}}} = \pi \cdot (R_i^2 + R_o^2 - R_{mid}^2) \tag{3.35}$$
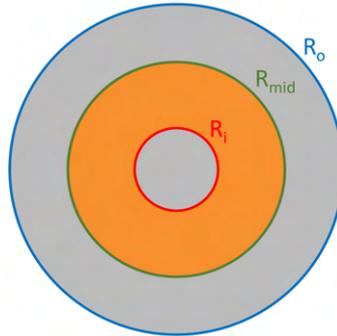
Figure 3.7: Parametrisation of the rod and tube SRM geometry.

Anchor   This solid propellant is more complex to model than the previous ones. To this extent, the parametrisation derived by Umbel [65] is used. Though the double anchor is most commonly seen, as in Figure 3.5, Umbel's model is based on a generalised anchor solid propellant. Some examples of the generalised anchor can be seen in Figure 3.8, with the following number of anchors N_a illustrated: 2, 3, 5, and 6.
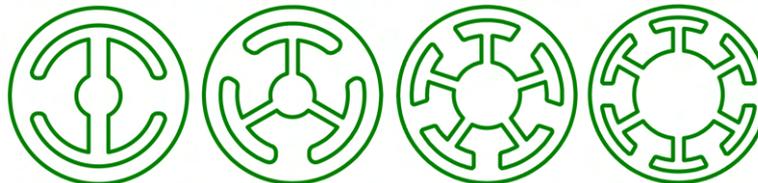
Figure 3.8: Example of variations of the generalised anchor geometry for solid propellant [65].

The generalised anchor is parametrised as in Figure 3.9, which represents half of an anchor. In this figure, R_o [m] is the outer radius of the propellant, R_i [m] is the radius of the initial empty cylinder at the centre of the propellant, w [m] is the web thickness, r_f [m] is the fillet radius, and δ_s [m] is the width of the anchor shank.
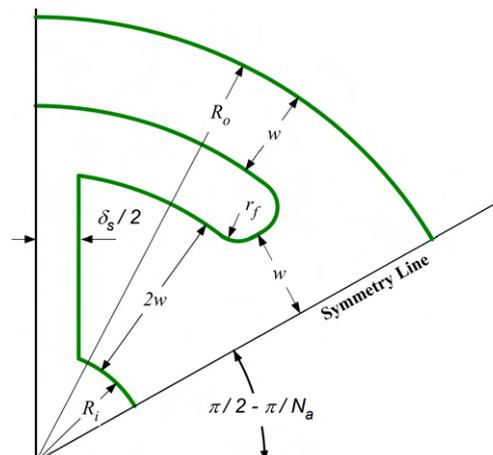
Figure 3.9: Parameters describing the solid propellant geometry of a generalised anchor [65].

From the geometry, Umbel [65] has identified the 6 constraints of Equations 3.36 to 3.41. In particular, Equation 3.40 enforces the fact that the arc defined by R_i is more than just a point, and Equation 3.41 ensures that the arc defined by R_i does not burn out before w is reached.

$$0 < R_i < R_o \tag{3.36}$$

$$0 < w < \frac{R_o - R_i}{3} \tag{3.37}$$

$$0 < r_f < \frac{R_o - 3 \cdot w - R_i}{2} \tag{3.38}$$

$$2 \leq N_a \tag{3.39}$$

$$0 < \delta_s < 2 \cdot R_i \sin\left(\frac{\pi}{N_a}\right) \tag{3.40}$$

$$\arcsin\left(\frac{\delta_s + 2 \cdot w}{2 \cdot (R_i + w)}\right) + \arcsin\left(\frac{r_f + w}{R_i + 2 \cdot w + r_f}\right) < \frac{\pi}{N_a} \tag{3.41}$$

Seven distinct sections of the perimeter have then been defined by Umbel [65], resulting in the total burning perimeter to be computed as in Equation 3.42, as a function of the burned distance b. These seven perimeter sections can be seen illustrated in Figure 3.10.

$$P_{\text{anchor}}(b) = 2 \cdot N_a \cdot \sum_{i=1}^{7} P_i(b) \tag{3.42}$$
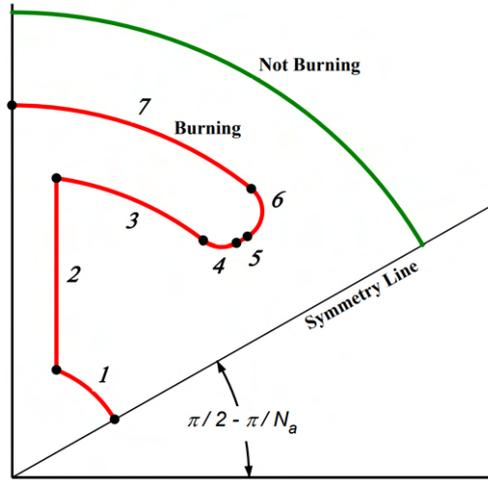
Figure 3.10: Sections of the anchor SRM defining the burning perimeter [65].

To compute the perimeter of the burning anchor SRM, the total web thickness w_t [m] and the web through the detached sliver w_{ds} [m] first have to be computed. This can be done as in Equations 3.43 and 3.44, taken from [65].

$$w_t = \sqrt{(w + r_f)^2 + 2 \cdot R_o^2 - 2 \cdot R_o\left(\sqrt{R_o^2 - 2 \cdot R_o(w + r_f)} + w + r_f\right)} - r_f \tag{3.43}$$

$$w_{ds} = \frac{R_i^2 + \left(2 \cdot w + r_f - \sqrt{R_i^2 + 2 \cdot R_i (2 \cdot w + r_f) + 3 \cdot w^2 + 2 \cdot r_f \cdot w}\right) \cdot R_i + 2 \cdot w(w + r_f)}{\sqrt{R_i^2 + 2 \cdot R_i (2 \cdot w + r_f) + 3 \cdot w^2 + 2 \cdot r_f \cdot w} + r_f - R_i} \tag{3.44}$$

The length of the perimeters P_1 to P_7 can finally be computed as a function of the burning distance b using Equations 3.45 to 3.51.

$P_1(b) = ...$	if
$(R_i + b) \cdot \left(\frac{\pi}{N_a} - \arcsin\left(\frac{\delta_s + 2 \cdot b}{2(R_i + b)} \right) \right)$	$0 \leq b \leq w$
$(R_i + b) \cdot \left(\arcsin\left(\frac{r_f + w}{R_i + 2 \cdot w + r_f} \right) - \arccos\left(\frac{R_i^2 + (2 \cdot w + b + r_f)R_i + (2 \cdot w - b)r_f + 2 \cdot w^2}{(R_i + 2 \cdot w + r_f)(R_i + b)} \right) \right)$	$w < b \leq w_{ds}$
0	$w_{ds} < b$

$$(3.45)$$

$P_2(b) = ...$	if
$\sqrt{(R_i + 2 \cdot w - b)^2 - \left(\frac{\delta_s}{2} + b \right)^2} - \sqrt{(R_i + b)^2 - \left(\frac{\delta_s}{2} + b \right)^2}$	$0 \leq b \leq w$
0	$w < b$

$$(3.46)$$

$P_3(b) = ...$	if
$(R_i + 2 \cdot w - b)\left(\frac{\pi}{N_a} - \arcsin\left(\frac{\delta_s + 2 \cdot b}{2 \cdot (R_i + 2 \cdot w - b)} \right) - \arcsin\left(\frac{r_f + w}{R_i + 2 \cdot w + r_f} \right) \right)$	$0 \leq b \leq w$
0	$w < b$

$$(3.47)$$

$P_4(b) = ...$	if
$(r_f + b) \arccos\left(\frac{r_f + w}{R_i + 2 \cdot w + r_f} \right)$	$0 \leq b \leq w$
$(r_f + b) \left(\arccos\left(\frac{r_f + w}{R_i + 2 \cdot w + r_f} \right) - \arccos\left(\frac{r_f + w}{r_f + b} \right) - \arccos\left(\frac{(R_i + 2 \cdot w + r_f)^2 + (r_f + b)^2 - (R_i + b)^2}{2(R_i + 2 \cdot w + r_f)(r_f + b)} \right) \right)$	$w < b \leq w_{ds}$
0	$w_{ds} < b$

$$(3.48)$$

$P_5(b) = ...$	if
$\sqrt{(R_o - w - r_f)^2 - (r_f + w)^2} - \sqrt{(R_i + 2 \cdot w + r_f)^2 - (r_f + w)^2}$	$0 \leq b \leq w$
0	$w < b$

$$(3.49)$$

$P_6(b) = ...$	if
$(r_f + b)\left(\frac{\pi}{2} + \arcsin\left(\frac{r_f + w}{R_o - w - r_f} \right) \right)$	$0 \leq b \leq w$
$(r_f + b)\left(\arccos\left(\frac{(R_o - w - r_f)^2 + (r_f + b)^2 - R_o^2}{2(R_o - w - r_f)(r_f + b)} \right) - \arccos\left(\frac{r_f + w}{r_f + b} \right) - \arccos\left(\frac{r_f + w}{R_o - w - r_f} \right) \right)$	$w < b \leq w_t$
0	$w_t < b$

$$(3.50)$$

$P_7(b) = ...$	if
$(R_o - w + b)\left(\frac{\pi}{N_a} - \arcsin\left(\frac{r_f + w}{R_o - w - r_f} \right) \right)$	$0 \leq b \leq w$
0	$w < b$

$$(3.51)$$

Finally, the initial port area of the anchor SRM can be computed by splitting the port area into the 7 areas shown in Figure 3.11, as discussed by Umbel [65]. Then, Equation 3.52 can be used to

sum the sections together, and multiply the areas by the number of anchors. This makes uses of Equations 3.53 to 3.59 to compute the 7 individual areas.
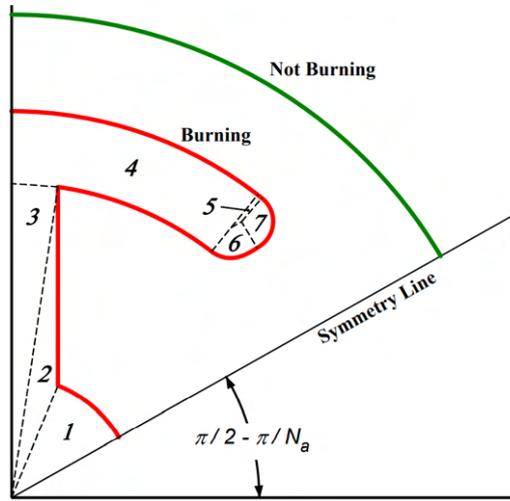
Figure 3.11: Zones of the anchor SRM fractionating the port area [65].

$$A_{p_{\text{anchor}}} = 2 \cdot N_a \cdot \sum_{i=1}^{7} A_{p,i} \tag{3.52}$$

$$A_{p,1} = \frac{R_i^2}{2} \cdot \left(\frac{\pi}{N_a} - \arcsin\left(\frac{\delta_s}{2 \cdot R_i} \right) \right) \tag{3.53}$$

$$A_{p,2} = \frac{(R_i + 2w) \cdot R_i}{2} \cdot \sin\left(\arcsin\left(\frac{\delta_s}{2 \cdot R_i} \right) - \arcsin\left(\frac{\delta_s/2}{R_i + 2w} \right) \right) \tag{3.54}$$

$$A_{p,3} = \frac{(R_i + 2w)^2}{2} \cdot \arcsin\left(\frac{\delta_s}{2(R_i + 2w)} \right) \tag{3.55}$$

$$A_{p,4} = \frac{(R_o - w)^2 - (R_i + 2w)^2}{2} \cdot \left(\frac{\pi}{N_a} - \arcsin\left(\frac{w + r_f}{R_i + 2w + r_f} \right) \right) \tag{3.56}$$

$$A_{p,5} = \frac{(R_o - w)^2}{2} \cdot \left(\arcsin\left(\frac{w + r_f}{R_i + 2w + r_f} \right) - \arcsin\left(\frac{w + r_f}{R_o - w - r_f} \right) \right) -$$
$$\frac{(R_o - w - r_f) \cdot (R_i + 2w + r_f)}{2} \cdot \sin\left(\arcsin\left(\frac{w + r_f}{R_i + 2w + r_f} \right) - \arcsin\left(\frac{w + r_f}{R_o - w - r_f} \right) \right) \tag{3.57}$$

$$A_{p,6} = \frac{r_f^2}{2} \arccos\left(\frac{w + r_f}{R_i + 2w + r_f} \right) + r_f \cdot \left(\sqrt{(R_o - w - r_f)^2 - (w + r_f)^2} - \sqrt{(R_i + 2w + r_f)^2 - (w + r_f)^2} \right) \tag{3.58}$$

$$A_{p,7} = \frac{r_f^2}{2} \left(\frac{\pi}{2} + \arcsin\left(\frac{w + r_f}{R_o - w - r_f} \right) \right) \tag{3.59}$$

Multi-fin   This solid propellant geometry can be approximated by a modified tubular geometry, with N_f added burning fins. These fins can be approximated by thin rectangles of which only the two longer sides burn. This can be done by virtually moving the innermost burning side of the fin back to the tubular section, meaning that the burning perimeter is the sum of the tubular burning perimeter and the two lengths of the fins. The total burning perimeter can then be computed as Equation 3.60, with the length of the fins L_f [m]. This can be used until $b = w_f/2$, with the fin width w_f [m]. Afterwards, Equation 3.60 can be used, as for the tubular burning propellant. The initial port area of a multi-fin geometry can be computed from Equation 3.61. The graphical representation of the parametrisation used for multi-fin SRM can be seen in Figure 3.12.

$$P_{\text{multi-fin}} = P_{\text{tubular}} + N_f \cdot P_{\text{fin}} = 2\pi \cdot (R_i + b) + 2 \cdot N_f \cdot L_f \tag{3.60}$$

$$A_{p_{\text{multi-fin}}} = \pi \cdot (R_o^2 - R_i^2) + N_f \cdot w_f \cdot L_f \tag{3.61}$$
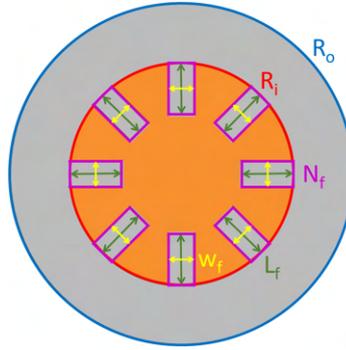
Figure 3.12: Parametrisation of the multi-fin SRM geometry.

This model imposes two conditions. First, the length of the fin shall be smaller than the inner radius of the tubular section, as in Equation 3.62. Second, the fins shall be spaced such that they do not spatially interfere with each other, as in the equality of Equation 3.63.

$$L_f < R_i \tag{3.62}$$

$$N_f \cdot w_f < 2\pi \cdot (R_i - L_f) \tag{3.63}$$

Spherical   As discussed in Section 2.2.2, the first design iteration from JPL imposes that the SRM of the second stage is a spherical motor. These spherical SRM differ from the cylindrical SRM of which cross-sectional geometries were discussed in the previous paragraphs. The geometry of a spherical SRM is shown in Figure 3.13. To visualise this geometry in 3D, one has to imagine that this figure is made to revolve around the dotted line. This then forms a propellant shell in grey, with outer radius R_o [m] and inner radius R_i. Also, this geometry contains an exhaust conduit of radius R_e [m]. The burning surface S of a spherical SRM can directly be computed using Equation 3.64, without first computing the burning perimeter. Similarly, the volume of propellant before the motor starts can be computed using Equation 3.65.

$$S = 4 \cdot \pi \cdot (R_i + b)^2 - \pi \cdot (R_e + b)^2 + 2 \cdot \pi \cdot (R_e + b) \cdot (R_o - R_i - b) \tag{3.64}$$

$$V_{p_{\text{spherical}}} = \frac{4}{3} \cdot \pi \cdot (R_o^3 - R_i^3) - \pi \cdot R_e^2 \cdot (R_o - R_i) \tag{3.65}$$
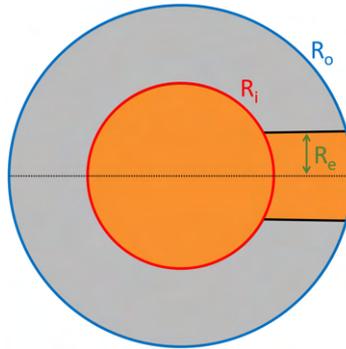
Figure 3.13: Parametrisation of the spherical SRM geometry.

3.2. Simulation

In this section, discussions are made on how the simulation of both the MAV ascent and the propellant burn are implemented. First, a simplified ascent model is discussed, followed by a description of the tools from the TU Delft Astrodynamics Toolbox that have been used. Then, the implementation of the propellant burn simulation and of TVC is discussed. Finally, the methodology used to setup the entire simulation is laid out.

3.2.1. Semi analytical model

First of all, a simulation of the MAV ascent has been made by propagating analytical equations that represent a first order model. This is to ensure that, using baseline values discussed in Section 2.2 results in the MAV reaching an altitude and velocity that is satisfactorily close to the target orbit. Also, this model can be used to discuss first order estimated of the MAV's performance, in the form of a mass and ΔV budget.

This model takes into account a constant thrust, a constant mass flow, drag using a two-step exponential atmosphere model, and a constant gravitational acceleration. Also, this model does not implement any TVC, but rather ensures a gravity turn, forcing the angle of attack to remain at 0. Further, only two dimensions are considered in this model: the y-dimension goes from the centre of Mars up to the vehicle at lift-off; the x-dimension is on the horizontal, and can be used to measure the sideways displacement of the MAV. All of the equations used for this model have been adapted and reworked from the Rocket Motion reader from Wittenberg, H. et al. [66].

This model is initiated with the initial velocity of the MAV V_0 [m/s], its initial altitude h_0 [m], its first stage propellant mass M_P [kg], its lift-off mass M_0, the initial kick angle δ [deg], the effective exhaust velocity c_{eff} [m/s], and the constant thrust F_T [N]. The initial kick angle represents the angle with respect to the vertical by which the vehicle is orientated when the gravity turn starts. The effective exhaust velocity is also the specific impulse I_{sp} [s] multiplied by $g_0 = 9.80665$ [m/s^2].

From these parameters, the burn time of the first stage motor can first be computed using Equation 3.66. In this equations, simplifications are made using Equation 3.67 for the mass ratio of the rocket Λ, and Equation 3.68 for the constant thrust load. Then, the constant mass flow $\dot{m}$ [kg/s] is computed with Equation 3.69.

$$t_b = \frac{c_{\text{eff}}}{g_0 \cdot \psi} \cdot \left(1 - \frac{1}{\Lambda}\right) = \frac{c_{\text{eff}} \cdot M_0}{F_T} \cdot \left(1 - \frac{M_0 - M_P}{M_0}\right) = \frac{c_{\text{eff}} \cdot M_0}{F_T} \cdot \frac{M_P}{M_0} = \frac{c_{\text{eff}} \cdot M_P}{F_T} \tag{3.66}$$

$$\Lambda = \frac{M_0}{(M_0 - M_P)} \tag{3.67}$$

$$\psi = \frac{F_T}{M_0 \cdot g_0} \tag{3.68}$$

$$\dot{m} = \frac{F_T}{c_{\text{eff}}} \tag{3.69}$$

The simulation then starts with computing different accelerations at a given time. The values of these accelerations depend on the mass of the MAV m, its altitude h, its velocity V, and its flight path angle γ.

First of all, the different accelerations that act on the MAV are computed. This is done by first computing the acceleration caused by the thrust in the inertial direction, through the use of Equation 3.70. Then, the acceleration of the vehicle towards the center of Mars, in the inertial direction, can be computed using Equation 3.71, with the Mars gravitational acceleration g [m/s^2]. The third acceleration that is computed is the one caused by the drag of the atmosphere on the vehicle. It can be computed as in Equation 3.72, also in the inertial direction. As this equation needs the air density ρ as a function of altitude, the two-step exponential atmosphere model that is discussed in Section 3.1.2 is selected for use in this simplified ascent simulation. All of these accelerations are then added together into one total inertial acceleration, as in Equation 3.73.

$$a_{\text{thrust}} = \frac{F_T}{m} \tag{3.70}$$

$$a_{\text{gravity}} = g \cdot \sin \gamma \tag{3.71}$$

$$a_{\text{drag}} = \frac{\rho \cdot V^2 \cdot C_D \cdot S_{\text{ref}}}{2 \cdot m} \tag{3.72}$$

$$a_{\text{total}} = a_{\text{thrust}} - a_{\text{gravity}} - a_{\text{drag}} \tag{3.73}$$

With the total inertial acceleration acting on the MAV known, its velocity can be incremented over time using Equation 3.74, with the discrete time step Δt [s]. Then, the individual component of the velocity in the x and y directions can be computed using Equations 3.75 and 3.76 respectively, with the flight path angle γ. The position in the x and y directions is similarly computed using Equations 3.77 and 3.78 respectively.

$$V_{i+1} = V_i + a_{\text{total}} \cdot \Delta t \tag{3.74}$$

$$V_x = V \cdot \cos \gamma \tag{3.75}$$

$$V_y = V \cdot \sin \gamma \tag{3.76}$$

$$X_{i+1} = X_i + V_x \cdot \Delta t \tag{3.77}$$

$$Y_{i+1} = Y_i + V_y \cdot \Delta t \tag{3.78}$$

At each discrete time step, the flight path angle is also to be updated. The rate at which the flight path angle changes is given as a function of the velocity of the vehicle in Equation 3.79. From it, the new flight path angle can be computed using Equation 3.80 at each time step.

$$\dot{\gamma} = -g \cdot \frac{\cos \gamma}{V} \tag{3.79}$$

$$\gamma_{i+1} = \gamma_i + \frac{d\gamma}{dt} \cdot \Delta t \tag{3.80}$$

Last but not least, the mass of the MAV is to be updated at each time step. This is done using Equation 3.81, with the constant propellant mass flow $\dot{m}$.

$$M_{i+1} = M_i - \dot{m} \cdot \Delta t \tag{3.81}$$

Iterating on the previous equations from $t = 0$ s up to $t = t_b$ s with time steps of Δt s allows to get the position, velocity, flight path angle, and mass of the MAV after first stage burnout.

The coasting phase then starts, where no motor is used, and both stages are kept attached so that RCS of the first stage can continued being used. During coasting, Equations 3.74 to 3.80 are iterated, with the exception of a_{thrust} now being equal to 0 m/s^2.

Finally, the same process can be repeated for the second stage burn as for the first stage. The only difference being that, this time, the initial position, velocity, and mass, is now the final position, velocity, and mass of the MAV after coasting. The initial flight path angle is set so that the second stage of the MAV is horizontal at second stage ignition, as to circularize the orbit of the vehicle.

3.2.2. TU Delft Astrodynamics Toolbox

More complex simulations can be setup using the TU Delft Astrodynamics Toolbox (Tudat) [67]. It contains various libraries to simulate a vehicle in many different situations such as a rocket launch, a satellite in orbit, a capsule in re-entry, and everything in-between. Also, this toolbox is open-source, its code being non-compiled and publicly accessible on GitHub [68]. Furthermore, while the code-base is written in C++, a Python interface is available since 2021, called Tudat(Py). This offers the comprehensibility of Python with the speed of execution of C++. Not only does this toolbox allows for the use of all of the environment models described in Section 3.1.2, and for the custom thrust models of Section 3.1.3 to be implemented. Tudat also allows for a more complex initial state to be set for the MAV and for complex simulation termination conditions to be used. Moreover, a wide set of different numerical integration methods can be used, and different state propagators are available.

Initial state

The lift-off position of the MAV is known in spherical elements by its altitude, longitude, and latitude. The initial velocity of the MAV is known to be of 0. However, the initial flight path angle is non-zero, and set to be between 30 deg and 60 deg, as discussed in Section 2.2.1. Since Tudat requires the initial state of the propagated vehicle to be given in cartesian coordinates, conversion is needed. These initial conditions as spherical elements are listed as follows:

- The radial distance from the centre of Mars r, consisting in the average radius of Mars R_{Mars} summed with the altitude in the Jezero crater, which is of $h = $ -2.55 km.
- The latitude ϕ of the MAV in the Jezero crater, which is of about 18.5 deg.
- The longitude θ of the MAV in the Jezero crater, which is of about 77.5 deg.
- The velocity of the MAV V, which is decided to be set to 0.1 m/s, as to be close to 0 given the VECTOR launch system, but different to 0 to avoid any singularity in the initial state.
- The initial flight path angle γ is set to be between 30 deg and 60 deg.
- The heading angle of the MAV χ is set to 0 by default. This value will evolve during ascent as TVC is used.

Termination conditions

Four distinct termination conditions from Tudat have been used. Two of them are to end the simulation when desired, and two of them are used as safeguards to make sure a simulation is not run needlessly long.

Altitude termination is triggered when the altitude of the MAV gets above or below a specified altitude. This is useful to stop the simulation if the vehicle goes to negative altitudes, which is unrecoverable. Similarly, some optimisations may lead to the MAV going to very high orbital altitudes. Since this is not the target, there is no need to continue the simulation above a certain altitude threshold.

Apogee termination is used to stop the simulation when the MAV, after stage 1 burnout, and coasting, reaches apogee. This is done by setting a condition to terminate when the flight path angle becomes negative, hinting that the vehicle starts falling back to Mars. For consistency, and to make sure stage separation occurs exactly at apogee, this condition has been set to terminate at the exact moment the flight path angle becomes 0. A secant root finder is then used to this end.

Time termination stops the simulation after a certain simulation epoch is reached. This is for instance used to propagate the final orbit of the MAV for a predefined amount of time.

CPU time termination is used to ensures that, would anything cause a simulation to get stuck during the optimisation, it will be stopped after a given amount of CPU time is elapsed. This value should be set high enough to ensure that a simulation that is taking a long time but not stuck does not get unnecessarily stopped. Taking a value multiple times higher than what a baseline successful simulation takes is thus recommended.

Propagators

With Tudat, a propagator consists in the differential equations that are integrated [69]. This thus include the way the dynamics of the systems are implemented.

For propagation of translational dynamics, the following propagators are supported by Tudat: Cowell, Encke, Gauss Keplerian, Gauss Modified Equinoctial, Unified State Model with Quaternions, Unified State Model with Modified Rodrigues Parameters, and Unified State Model with Exponential Map [70].

For propagation of rotational dynamics, only the Quaternions, Modified Rodrigues Parameters and Exponential Map propagators are suited, and implemented in Tudat [71]. All of these propagators, and most importantly their singularities, are discussed in the following sub-sections.

Cowell propagator works by directly integrating the differential Equation 3.82, from [72]. In it, m_A [kg] is the mass of the vehicle and F_p is a perturbation force. This propagator propagates the states $x = [r\ \dot{r}]$ in Cartesian frame. This means that the state derivative $\dot{x} = [\dot{r}\ \ddot{r}]$ may contains large values and large deviations. This makes large numerical error more common.

$$m_A \cdot \ddot{r}_{BA} = -\frac{m_A \cdot \mu_B}{||r_{BA}||^3} r_{BA} + \sum F_p \tag{3.82}$$

Finally, the Cowell propagator is mostly singularity-free. As seen from Equation 3.82, the only singularity present is for $r_{BA} = [0\ 0\ 0]$. This only occurs when the position of the vehicle coincides with the centre of mass of the body B that it orbits.

Encke mitigates the issue that arises with the Cowell propagator in which the state derivative may contains large values. Indeed, the Encke propagator makes advantage from the fact that a perturbed orbit-like trajectory is similar to a Kepler orbit for some time.

The differential equation that is then propagated is the one of Equation 3.83, containing the difference between the perturbed orbit from Equation 3.82 and the perfect Kepler orbital state computed with Equation 3.84 [73].

$$\Delta r = r - \boldsymbol{\rho} \tag{3.83}$$

$$\ddot{\boldsymbol{\rho}} = -\mu_B \frac{\boldsymbol{\rho}}{||\boldsymbol{\rho}||^3} \tag{3.84}$$

As discussed by Dirkx and Mooij [74, Slides 21-24], because the propagated state Δr is now small, it does not suffer from numerical error anymore, but may suffer from numerical noise. Also, a high propagation time will result in the same pitfall as with Cowell since Δr will become similar in magnitude as r.

Gauss Keplerian propagator uses Gauss planetary equations with Keplerian elements as the state. The Gauss planetary equations are then propagated, with extra terms added to account for the perturbation forces. Such modified equations can be found derived by Rodríguez [75]. This propagator contains a singularity when $e = 0$. This is because it is not possible to compute the argument of periapsis ω of a perfectly circular orbit, since its ascending node is undefined. A similar singularity exists for a zero orbital inclination 0, when the orbit is equatorial. Moreover, singularities are present when the orbit is parabolic ($e = 1$) or when the orbit is highly eccentric ($e \to \infty$).

Gauss Modified Equinoctial propagator also uses Gauss planetary equations, with the modifications of Equations 3.85 to 3.90, taken from [76]. In them, the parameter I can take the value of 1 or -1, and is a so-called shadow parameter, implemented to avoid an otherwise present singularity.

This propagator contains singularities for $i = 0$ and $i = 180$ [deg]. The value parameter I can either be set at the beginning of the propagation, or be set as a 7th element to propagated, and switched as to avoid the singularities of this integrator.

$$p = \begin{cases} a(1 - e^2) & (e \neq 1) \\ 2a & (e = 1) \end{cases} \tag{3.85}$$

$$f = e \cos(\omega + I\Omega) \tag{3.86}$$

$$g = e \sin(\omega + I\Omega) \tag{3.87}$$

$$h = \tan^I (i/2) \sin \Omega \tag{3.88}$$

$$k = \tan^I (i/2) \cos \Omega \tag{3.89}$$

$$L = \omega + I\Omega + \theta \tag{3.90}$$

Unified State Model   (USM) propagator attempts as a singularity-free propagator, using as propagated state: 3 from an velocity hodograph, and the remaining being used to represent the orientation of the vehicle.

As discussed by Vittaldev, Mooij, and Naeije [77], while a quaternion with 4 elements is commonly used to represent the orientation, it is possible to lower the total number of states from 7 to 6 by replacing the quaternion using Modified Rodrigues Parameters (MRP) or an Exponential Map (EM).

A quaternion is a unit vector that can represents the orientation of a vehicle with 4 elements, as in Equation 3.91 from [77, pp.256-257]. In it, $\hat{\alpha}$ is the Euler axis, ϕ is the Euler angle, η is the real scalar part of the quaternion, and ϵ is the vector representing the imaginary part of the quaternion.

$$\begin{cases} \boldsymbol{\epsilon} = [\epsilon_1 \ \epsilon_2 \ \epsilon_3]' = \hat{a}\sin(\phi/2) \\ \eta = \cos(\phi/2) \end{cases} \tag{3.91}$$

Modified Rodrigues Parameters work by using a projection point to project a 4-dimensional quaternion on a 3-dimensional hyperplane. This method results in a MRP vector instead of the quaternion vector. The MRP vector $\boldsymbol{\sigma}$ can be written as in Equation 3.92, also taken from [77, pp.256-257]. It can however be seen that $\boldsymbol{\sigma}$ contains a singularity. The latter can be solved using the shadowed MRP, as in Equation 3.93.

$$\boldsymbol{\sigma} = \hat{a}\tan(\phi/4) \tag{3.92}$$

$$\boldsymbol{\sigma}^S = \frac{-\boldsymbol{\sigma}}{\sigma^2} \tag{3.93}$$

The Exponential Map of a quaternion, a, is simply the unit Euler axis $\hat{a}$ scaled by the magnitude of the Euler angle ϕ, as in Equation 3.94. However, to avoid the singularity present when converting the quaternion into the EM, and when $\phi = 0$, the shadowed EM a^S from Equation 3.95 can be used [77, p.258].

$$a = \hat{a}\phi \tag{3.94}$$

$$a^S = \left(1 - \frac{2\pi}{||a||}\right)a \tag{3.95}$$

Two singularities remain with the USM. The first one is when the orbit is a retrograde equatorial orbit, having an inclination i of 180 deg. The second singularity is when the angular momentum of the vehicle is close to 0 kg·m²/s [77, pp.265-266]. Both of these singularities are related to the orbital velocity hodograph.

Runge-Kutta integration
Integrators are used to compute the state x of a body at a given time, usually using a combination of a previous state, and of the state derivative evaluated at key moments in time. For this research, Runge-Kutta (RK) integration methods have been exclusively considered. This is due to the large number of RK integration coefficients that have been added to Tudat in parallel to this study, making the room for decision between different RK integration methods large. First of all, the example of the Euler and RK4 integration methods are made.

Euler   integration computes the next state x of a body at an initial time t_0 [s] plus a time step h [s], using Equation 3.96, as written by Hahn [78]. In this equation, $f(t_0, x_0)$ is the derivative of the state at the initial time t_0 and at the initial state x_0. It is important to note that, while it is common that the state derivative f only depends on the time t, it happens that it also depends on the state x itself. For instance, the state derivative of the MAV will depend on its position to compute the gravitational acceleration, and on its velocity to compute the aerodynamic forces.

$$x(t_0 + h) \approx x_0 + f(t_0, x_0) \cdot h \tag{3.96}$$

RK4 integration is less straightforward than Euler integration, and requires more function evaluations, but is a generally more accurate method. Indeed, an RK4 integrator evaluates the derivative f at four different emplacements, and take a weighted average of these derivatives to add them to the initial state x_0 [79]. This is seen in Equation 3.97, with the four derivatives taken as in Equations 3.98 to 3.101.

$$x(t_0 + h) \approx x_0 + \frac{h}{6} \cdot (k_1 + 2 \cdot k_2 + 2 \cdot k_3 + k_4) \tag{3.97}$$

$$k_1 = f(t_0, x_0) \tag{3.98}$$

$$k_2 = f\left(t_0 + \frac{h}{2}, x_0 + \frac{h}{2} \cdot k_1\right) \tag{3.99}$$

$$k_3 = f\left(t_0 + \frac{h}{2}, x_0 + \frac{h}{2} \cdot k_2\right) \tag{3.100}$$

$$k_4 = f(t_0 + h, x_0 + h \cdot k_3) \tag{3.101}$$

An illustration of the advantage of the RK4 integration can be seen in Figure 3.14. As seen, the next state y_{n+1} is close to the true one, on the dotted line. If a Euler integrator was used instead, y_{n+1} would be aligned with the derivative that is computed at y_n.
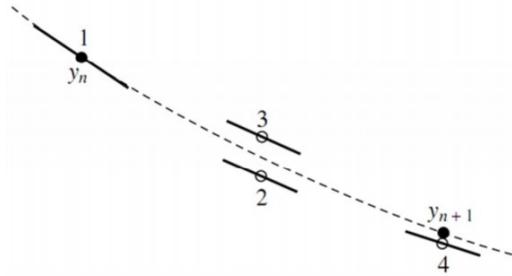
Figure 3.14: Illustration of the benefit of using a Runge-Kutta integrator [79].

Butcher tableaus are used to generalise Runge-Kutta methods. Indeed, both the Euler and RK4 integrators make use of the same mechanisms. The only difference is the state, time, and weights used to evaluate the derivative and combine to the next state. This general formula is given in Equation 3.102, with the evaluated derivatives k_i computed as in Equation 3.103 [80].

$$x(t_0 + h) = x_0 + h \cdot \sum_{i=1}^{s} (b_i \cdot k_i) \tag{3.102}$$

$$k_i = f\left(t_0 + h \cdot c_s, h \cdot \sum_{j=1}^{s-1} (a_{s,j} \cdot k_j)\right) \tag{3.103}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ c_2 & a_{2,1} & 0 & 0 & \dots & 0 \\ c_3 & a_{3,1} & a_{3,2} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s,1} & a_{s,2} & a_{s,3} & \dots & 0 \\ \hline & b_1 & b_2 & b_3 & \dots & b_s \end{bmatrix} \tag{3.104}$$

The Butcher tableau that corresponds to the Euler integration method is then straightforward, as given in Equation 3.105. Indeed, swapping the coefficients for a, b, and c from this tableau into Equations

3.102 and 3.103 gives Equation 3.96. Similarly, the Butcher tableau for the RK4 integration method is given in Equation 3.106.

$$\left[\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}\right] \tag{3.105}$$

$$\left[\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array}\right] \tag{3.106}$$

In Tudat, since the completion of the work made in parallel to this research, the following Butcher tableaus can be used for Runge-Kutta integration:

- Euler forward.
- Heun Euler.
- Explicit midpoint method.
- Explicit trapezoid rule (also called Heun's method or improved Euler's method).
- 3/8th rule of order 4.
- Ralston's method of degree 2, 3, 4.
- Runge-Kutta-Fehlberg method of degree 1, 2, 4, 5, 6, 7, 8, 9.
- Strong Stability Preserving Runge-Kutta of order 3.
- Runge-Kutta Dormand-Prince method of order 7, 8.
- Runge-Kutta-Verner method of order 8, 9.
- Runge-Kutta-Feagin method of order 10, 12, 14.

Variable step size   Runge-Kutta integration is also available through Tudat. While the classic RK method described above uses a fixed step h, an alternative method exists with a step size that adapts itself during the simulation, based on an estimation of the integration error over time. These variable step RK integrator are for instance the Runge-Kutta-Fehlberg integrator [81], or the widely used Runge-Kutta Dormand-Prince integrator [82].

The way the step size is computed is described by Fehlberg in [81]. In essence, two similar integration methods of different orders are used. These methods of different orders are setup in such a way that all of the a and c values from their Butcher tableaus are identical. The only difference between the two methods is that they have different values for the b coefficients, and that the higher order method has more b coefficients than the lower order one. This means that, during an integration step, the state derivative is evaluated at both states and times for both methods, saving computational resources. The only additional step to compute the distinct order method is to compute a different weighted average with different b coefficients.

Then, from the resulting state found by the two methods of different orders, an estimate is made on the integration error, from the difference in state from the two orders. This error is then used to adjust the step size. The following parameters are used to tune the step size control algorithm:

- The minimum and maximum allowed step size.
- The relative and absolute error tolerance, used to determine how much the step size should be adapted based on the estimated error.
- The maximum and minimum factor by which the step size can be adjusted.
- The safety factor, used to ensure that the step size stays low enough to mitigate estimated errors.

3.2.3. Thrust
It has been decided that the model for the thrust described in Section 3.1.3 shall be kept distinct from the ascent simulation, in the sense that they do not run at the same time. Indeed, the burn simulation

is implemented to run first, the thrust and mass flow over time saved, and then these values are used during the ascent simulation. This keeps the simulation setup more straightforward, and allows to know the exact burn time before starting the ascent, which is used to control TVC during the burn.

Besides this, there is still a need to setup the integration of the thrust equations from Section 3.1.3. Thrust vectoring control also needs to be implemented in the simulation. Both of these aspects of the simulation are discussed in the following sections.

Solid propellant burn integration

The state that is propagated to simulate the burn of the solid propellant is, as described in Equation 3.107, the propellant mass M_P and the burned distance b. The initial propellant mass is computed from the propellant density and volume, while the initial burn distance is zero.

Then, the state derivative that is computed is the propellant mass flow $\dot{m}$ and the motor regression rate r, as in Equation 3.108. These values are computed by following the steps detailed in Section 3.1.3. With a representation for the state and state derivative, and with a set of equations to compute the latter, the propellant burn and thrust simulation can be setup. While, in theory, any integration method could be used, it was decided to hard-code an RK4 method, tune its fixed step value, and advise on whether this setup was sufficient for the required accuracy or not.

The value of the mass flow $\dot{m}$ is then saved for each integration step, to be used later in the ascent simulation. The thrust itself is computed at each of the integration sub-steps, at the same time as each derivative evaluation. Then, the weighted average of the thrust from the sub-steps is taken, with the same weights as used to compute the next state. This ensures that a high accuracy is reached for the thrust, without impacting the integration itself.

$$x = [M_P, b] \tag{3.107}$$

$$\dot{x} = [\dot{m}, r] \tag{3.108}$$

Thrust vectoring control

During the ascent, it is assumed that the thrust of the MAV is always aligned with its inertial direction, meaning that the thrust is aligned with the velocity. An exception is made with the addition of TVC. Setting the centerline of the MAV as the x-axis, TVC can be deflected by rotation either around the y-axis or the z-axis. The model used to represent this effect of TVC on the thrust orientation is set up using rotation matrices. A deflection of θ_y [rad] of the nozzle around the y-axis will result in a thrust rotation matrix as in Equation 3.109. The rotation matrix representing a deflection of θ_z [rad] of the nozzle around the z-axis is given in Equation 3.110. Finally, if the nozzle is deflected around both the y and the z axes, matrix multiplication can be used to combine both rotation matrices, as in Equation 3.111.

$$R_y(\theta_y) = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \tag{3.109}$$

$$R_z(\theta_z) = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.110}$$

$$R(\theta_y, \theta_z) = R_y(\theta_y) \cdot R_z(\theta_z) = \begin{bmatrix} \cos\theta_y \cos\theta_z & -\cos\theta_y \sin\theta_z & \sin\theta_y \\ \sin\theta_z & \cos\theta_z & 0 \\ -\sin\theta_y \cos\theta_z & \sin\theta_y \sin\theta_z & \cos\theta_y \end{bmatrix} \tag{3.111}$$

3.2.4. Setup

This section describes how the whole simulation has been implemented and setup. First, a benchmark has been generated, to have a accurate baseline to compare variation in the simulation with. This benchmark was then used to study the effect of tuning the ascent simulation integrator, and the time step for the burn simulation. Similarly, the benchmark has been used to tuned which type of environmental accelerations are relevant. Finally, what propagator should be used has been studied.

This entire simulation setup has been carried based solely on the multi-fin SRM model. This is because this model offers the best compromise in ease of implementation and complexity in the dynamics. Indeed, only few equations are required to setup this model, allowing to carry the simulation setup in parallel to the implementation and validation of the other SRM models. Also, the multi-fin SRM offers complex dynamics through its two distinct level of thrusts, making the simulated dynamics more complex.

Benchmark

The goal of the benchmark is to simulate the ascent of the MAV with an accuracy at least 100 times the accuracy required for the optimisation simulation. This factor of 100 is used to quantify the accuracy of the optimisation simulation with a high degree of certainty, and to allow for the study of the numerical accuracy of the ascent simulation.

To generate this benchmark, simulations for the propellant burn and MAV ascent are run with the highest accuracy possible. The limit for this accuracy is expected to be the numerical accuracy of the code. Indeed, as smaller and smaller time steps are used, the accuracy of the numerical integration is expected to increase up to a certain point. At this point, the state is being incremented by such low values that the numerical integration becomes unstable, introducing errors in the simulation.

This process is run to first find the highest accuracy possible for the motor burn simulation of the first stage of the MAV. The simulation is run with the RK4 method, first with a time step[2] of 9.95 s. Then, a logarithmic scale of base 10 is used, so that 10 time steps are tried for each power of 10. The burn simulation is thus run with increasingly small time steps, and the resulting thrust and mass of the motor is saved for each of these simulations.

Afterwards, each burn simulation is compared to the simulation that uses the highest time step that is at least twice as low. Comparing these simulations, an estimate for the simulation with the higher time step can be made. Finally, this estimate is used to generate a plot of the integration error in thrust and mass as a function of the time step. From this plot, the point at which numerical accuracy starts to degrade is found. The benchmark is then set to be the simulation that achieves the highest accuracy before any numerical accuracy degradation.

The ascent simulation is then run for the first stage ascent up until motor burnout, using the thrust and mass flow values computed from the propellant burn benchmark. This simulation is not run further since it is expected that the high thrust from the SRM will cause the minimum time step to be different than when the thrust acceleration is not present. Once again, this ascent simulation is run with increasingly small time steps, so that the point at which numerical accuracy starts degrading can be found, analysing the error in position and velocity. In this case, the integration method that is used is the Runge-Kutta-Feagin method of order 14, as it is the one that theoretically leads to the highest accuracy amongst all of the integration methods implemented in Tudat.

Then, the same process is repeated to generate the benchmark for the first stage ascent from motor burnout to apogee. Once the benchmark for the entire ascent simulation of the first stage is made, the benchmark for the burn simulation of the second stage motor is generated, following the same procedure as for the first stage. Finally, the benchmark for the ascent of the second stage of the MAV is generated, first only during motor burn using the second stage propellant burn benchmark and starting from the final state of the first stage benchmark. Then, the second stage ascent benchmark is made, starting from motor burnout to a set time in orbit.

With this procedure completed, a benchmark is now available that covers the entire ascent simulation of the MAV, and the propellant burn simulation for both stages. In conclusion, the following separate benchmarks are made:

- Burn simulation of the first stage.
- Ascent simulation of the first stage from lift-off to motor burnout.
- Ascent simulation of the first stage from motor burnout to apogee.
- Burn simulation of the second stage.
- Ascent simulation of the second stage from apogee to motor burnout.

---

[2]This number is guessed, since multiple time steps will be tried. Of importance, it is selected to be non-integer on purpose, to avoid any possible unexpected issue with numerical accuracy due to the time step having no decimals.

- Ascent simulation of the second stage from motor burnout to set time in orbit.

Still, since the benchmarks were generated separately from each other, there is no guarantee that an error that is introduced during the simulation of the first stage will for instance not affect the simulation of the second stage above the required accuracy.

To this extent, a study is made on how sensitive each part of the simulation benchmark is to initial state and thrust deviations. Starting with the simulation of the second stage from motor burnout to set time in orbit. First, the initial state is varied in position and in velocity at the same time. The ratio of the variation in velocity compared to the variation in position is based on the observed ratio in integration error for both of these values, as analysed during benchmark generation.

This variation in position and velocity is tuned until the final error in the second stage simulation from motor burnout to orbit is below the required accuracy. These initial variations are then used as the required accuracy for the previous ascent simulation step, or for the relevant burn simulation. For phases of the ascent that use thrust, in addition to initial deviations in position and velocity, the study of a constant error in thrust and mass flow is made. This error is then used as a requirement for the motor burn simulation.

Running this procedure from the end of the simulation all the way back to its beginning then allows to check whether the higher possible numerical accuracy is indeed below the required accuracy for each part of the benchmark. If it is not, the benchmark cannot, as a whole, be used to detect error above the required accuracy.

Generating these six benchmarks requires countless simulations to be run, of increasing CPU time, due to the increasingly high accuracy. To partially mitigate this, multiprocessing has then been used to run simulations that use different time steps in parallel. This directly reduced the CPU time needed to generate the benchmarks by the number of processes used.

This entire setup aims at ensuring a pre-determined accuracy for the simulation. Still, selected results from the optimisation will be re-run with a higher accuracy integrator and environment. This is to ensure that optimisation results carry the same accuracy, instead of just proving the accuracy of the benchmark.

Integrator tuning

With a benchmark for the entire MAV ascent and propellant burn simulations, the integrator used to carry the simulation can be tuned.

For the propellant burn simulation, the time step was selected to be the highest one possible that still manages to achieve the required accuracy, as from the results of the benchmark accuracy verification process.

For the ascent simulation, the methodology followed has been to try all of the fixed step integration methods that Tudat possesses, with different times steps. The results of the simulations are then compared to the benchmark to compute the error in position and velocity for each of the integrator and time step. The number of function evaluations made for each integration method and time step is also saved. A plot can then be made of the state error as a function of the number of function evaluations, for each of the time steps and integrator.

A similar plot has been made for variable step integration methods, trying different tolerances for step size control. The other parameters of step size control have been set in such a way that the time steps can take value as high or low as they need, and switch as fast as possible between high and low time steps. This is because of the expected drastic change in dynamics when thrust starts or stops.

The plots that compare state error and function evaluations for the different integrators and time steps have then been used to select the combination that results in a state error below the required accuracy, but still relatively close to the requirement as to keep the number of function evaluations low.

Environment tuning

Environment tuning is a more straightforward process than integrator tuning. Models that aim at representing the same aspect of the environment are compared to each other, and the one that leads to an error below the required accuracy is selected, keeping some margin below the required accuracy

to ensure that errors do not accumulate. For instance, gravitational models such as a point mass and different degree and orders of spherical harmonics are compared together, since they are all representing the gravitational field of Mars.

In some cases, it may also be found than adding some other aspect of the environment, such as the gravitational acceleration of a distant planet, does not contribute to the accuracy of the simulation significantly, when looking at the required accuracy. This means that these parts of the environment can be omitted for the simulation.

Propagator selection

Lastly, it is important to ensure that an appropriate propagator is used for the simulation. Up to this step, a Cowell propagator will be used, since it leaves the least opportunity for errors caused by singularities, and it has been used in similar studies successfully in the past.

However, all of the other available propagators will be tried, and their error will be compared to one another. If significant deviations are found, an investigation must then be made to find if the error is caused by the Cowell propagator or by the one that it is compared with. Furthermore, if it is found that the Cowell propagator is not the best choice for the given problem, a different propagator is to be selected. In this case, a new benchmark needs to be generated, the integrator needs to be tuned again, and so does the environment.

3.3. Optimisation

This section describes each step of the optimisation process. To carry the optimisation, the optimisation library described by Biscani and Izzo [83] can be used. This is the Parallel Global Multiobjective Optimizer library, called Pagmo. Similarly to Tudat, the Python interface of Pagmo will be used, called Pygmo [83]. This allows to use only one programming language to make the simulation and optimise the design. Thanks to this, the two processes can be interlinked, offering a more efficient implementation.

3.3.1. Design variables

According to Noël et al. [84], the design variables of an optimisation problem are the variables with which the design is parametrised. These can also be seen called control variables, or decision variables. The design variables are thus the parameters that are tuned by an optimisation algorithm, as to change the studied design.

For this research, the following design variables have been selected:

- Launch angle of the MAV at lift-off. This value is allowed to vary between 30 deg and 60 deg from the horizontal, as discussed in Section 2.2.2.

- MAV angle at stage separation. This value is preliminarily allowed to vary between 60 deg and 120 deg from the horizontal, since a value of 0 would mean that the orbit is being solely circularised, and that increasing the angle by more than 30 deg would result in a non-optimum manoeuver.

- Outer radius of the first stage SRM, allowed to vary between 0.1 m and 0.285 m. The upper value is set from the maximum allowed MAV size, as discussed in Section 2.2.1. The lower value is set to prevent the MAV from being too thin.

- Length of the first stage SRM, allowed to vary between 0.3 m and 1.25 m. The upper value being a restriction due to the allowed length of the MAV, as extrapolated from Section 2.2.2. The lower value is set to ensure that the first stage motor will always be longer than its radius.

- The angle of the nozzle deflection in two planes due to the use of TVC at 5 nodes equispaced through the burn time of the first stage motor. This implementation is described in Section 3.2.3. The deflection angle is allowed to vary between -5 deg and 5 deg, as required by JPL and discussed in Section 2.2.1.

In addition, the geometry of the SRM of both stages that has been parametrised in Section 3.1.3 is also included as design variables. An effort has been made to ensure that the design variables could

take any value within their respective range, so that no unfeasible SRM geometry would be attempted and have to be discarded during optimisation.

For the second stage spherical SRM, two design variables are set, to compute the inner and outer radius of the propellant shell. These are shown in Table 3.2, with the equations used to deduct the SRM geometry parameter from the design variable. Both of these design variables can in theory vary between 0 and 1, meaning that the outer radius of the second stage SRM can vary between 0 m and the outer radius of the first stage SRM, and that the inner sphere can occupy any volume withing the outer one.

In Section 3.1.3, the exhaust conduit of the spherical SRM was also discussed, of radius R_e. It has been decided that this variable would not affect the optimisation significantly, hence it was not included as a design variable. Instead, the value of R_e is fixed at half the inner sphere radius: $R_e = R_{i,2}/2$.

Design variable	Associated equation	Maximum range
Outer radius fraction $R_{o,2,\text{frac}}$	$R_{o,2} = R_{o,2,\text{frac}} \cdot R_{o,1}$	0 to 1
Inner radius fraction $R_{i,2,\text{frac}}$	$R_{i,2} = R_{i,2,\text{frac}} \cdot R_{o,2}$	0 to 1

Table 3.2: Design variables for the spherical SRM geometry.

For the first stage SRM, different cross-sectional geometries can be used: tubular, rod and tube, multi-fin and anchor. To ensure that the optimisation stays as clear as possible, distinct optimisations have been performed for each of these geometries, and the results combined afterwards.

Irrespective of the cylindrical SRM type, two common design variables are used. These are the length of the SRM L, and its outer radius $R_{o,1}$. Different sets of design variables have then been established for the cross-sectional geometry of the first stage SRM, depending on its type.

The tubular SRM type being the most straightforward one, it is characterised by only one additional design variable, the inner radius fraction. This is shown in Table 3.3. Then, the rod and tube SRM has one more design variable, to define the intermediate radius of the inner propellant rod, as shown in Table 3.4. The multi-fin SRM geometry being more complex, it has four design variables to characterise its cross-sectional geometry. The relation between these design variables and the geometry parameters is given in Table 3.5. Most notably, the number of fins N_f must be an integer, and the design variable that defines the fin width w_f is made to ensure that the fins do not interfere with one another. Lastly, the anchor SRM geometry is the most complex. The design variables are translated into the geometry parameters through the equations presented in Table 3.6. Again, the number of anchors N_a must be an integer. All the design variables in this table are setup as to enforce the constraints from Equations 3.36 to 3.40.

Design variable	Associated equation	Maximum range
Inner radius fraction $R_{i,1,\text{frac}}$	$R_{i,1} = R_{i,1,\text{frac}} \cdot R_{o,1}$	0 to 1

Table 3.3: Design variables for the tubular SRM cross-sectional geometry.

Design variable	Associated equation	Maximum range
Intermediate radius fraction $R_{\text{mid,frac}}$	$R_{\text{mid}} = R_{\text{mid,frac}} \cdot R_{o,1}$	0 to 1
Inner radius fraction $R_{i,1,\text{frac}}$	$R_{i,1} = R_{i,1,\text{frac}} \cdot R_{\text{mid}}$	0 to 1

Table 3.4: Design variables for the rod and tube SRM cross-sectional geometry.

Design variable	Associated equation	Maximum range
Inner radius fraction $R_{i,1,\text{frac}}$	$R_{i,1} = R_{i,1,\text{frac}} \cdot R_{o,1}$	0 to 1
Number of fins N_f	N.A.	3 to 20
Fins length fraction $L_{f,\text{frac}}$	$L_f = L_{f,\text{frac}} \cdot R_{i,1}$	0 to 1
Fins width fraction $w_{f,\text{frac}}$	$w_f = w_{f,\text{frac}} \cdot \frac{2\pi \cdot (R_{i,1} - L_f)}{N_f}$	0 to 1

Table 3.5: Design variables for the multi-fin SRM cross-sectional geometry.

Design variable	Associated equation	Maximum range
Inner radius fraction $R_{i,1,\text{frac}}$	$R_{i,1} = R_{i,1,\text{frac}} \cdot R_{o,1}$	0 to 1
Number of anchors N_a	N.A.	2 to 15
Anchor spacing fraction w_{frac}	$w = w_{\text{frac}} \cdot \frac{R_{o,1} - R_{i,1}}{3}$	0 to 1
Fillet radius fraction $r_{f,\text{frac}}$	$r_f = r_{f,\text{frac}} \cdot \frac{R_{o,1} - 3w - R_{i,1}}{2}$	0 to 1
Anchor spoke thickness fraction $\delta_{s,\text{frac}}$	$\delta_s = \delta_{s,\text{frac}} \cdot 2 \cdot R_{i,1} \cdot \sin\left(\frac{\pi}{N_a}\right)$	0 to 1

Table 3.6: Design variables for the anchor SRM cross-sectional geometry.

Depending on the SRM type selected for the first stage, this means that there are between 17 and 21 design variables used to optimise the MAV ascent and SRM geometry.

3.3.2. Objectives

As discussed by Papalambros and Wilde [85], an optimisation objective is a quantity that shall be minimised or maximised. It is also referred to as the cost function, the merit function, or the fitness function. An optimisation problem can be a single-objective or a multiple-objective optimisation. As for the design variables, the objectives are also associated with an objective space that contains all feasible outcomes of the optimisation process [86].

It is worth knowing that, in Pagmo, the objectives are always minimised by default [83]. In the case of the optimisation of the MAV ascent trajectory, the following three objectives are used:

- Minimise the mass of the vehicle, including solid motors.
- Minimise the deviation between the final orbit periapsis and the target orbit.
- Minimise the deviation between the final orbit apoapsis and the target orbit.

These objectives can actually be seen as truly the goal of finding the lowest mass possible, with the other two objectives serving as constraints. However, they are still implemented as objectives, so that the optimiser does not simply see the target altitude as something that is binary (reached or not), but as something that should be targeted.

Because all of these objectives should be minimised, laws have been setup to translate them in functions. The objective of minimising the mass is implemented through Equation 3.112, with the MAV mass m [kg], and a limiting mass m_{lim} [kg]. The value for this limiting mass is set to 400 kg, as this is the maximum MAV mass as discussed in Section 2.2.1. This relationship results in the objective score as a function of the MAV mass shown in Figure 3.15a. It can then be seen that a mass of 400 kg results in an objective score of 1, and that the objective score decreases linearly as the mass gets lower. Also, the slope of the line is higher when the mass is of more than 400 kg, to increase on the importance of first reaching the requirement, and then trying to decrease the mass even lower.

Similarly, the relationship used to computed the final orbital altitudes score is given in Equation 3.113, with the minimum and maximum required altitude of $h_{\text{min}} = 300$ [km] and $h_{\text{max}} = 375$ [km], as discussed in Section 2.2.1. The buffer altitude h_{buffer} [km] is used so that the optimiser tries to stay further from the lower limit than the upper one, as the lower altitude is a hard requirement, and the upper altitude is a soft requirement. It has been decided to set the value of this buffer to 315 km. Both the periapsis and the apoapsis of the final orbit are to be scored using Equation 3.113. The plot of Figure 3.15b shows the mapping of the orbital altitude to the objective score. Once again, the slopes are made in a way to represent the importance of staying clear of low altitudes at all costs, to represent the optimum altitude range, and to be gentler at higher altitudes, since the consequences of going higher are less severe.

$$f_{\text{mass}}(m) = \begin{cases} \frac{m}{m_{\text{limit}}} = \frac{m}{400} & \text{if } m \leq m_{\text{limit}} \\ \frac{5 \cdot m_{\text{limit}} - 4 \cdot m - 500}{m_{\text{limit}} - 500} = \frac{4 \cdot m - 1500}{100} & \text{otherwise} \end{cases} \tag{3.112}$$

$$f_{\text{altitude}}(h) = \begin{cases} 0 & \text{if } h > h_{\text{buffer}} \text{ and } h \leq h_{\text{max}} \\ \frac{h-h_{\text{max}}}{450-h_{\text{max}}} = \frac{h-375}{25} & \text{if } h > h_{\text{max}} \\ \frac{h_{\text{buffer}}-h}{450-h_{\text{max}}} = \frac{315-h}{75} & \text{if } h \geq h_{\text{min}} \text{ and } h < h_{\text{buffer}} \\ \frac{h-h_{\text{buffer}}}{h_{\text{min}}-h_{\text{buffer}}} \cdot \frac{3}{4} - \frac{9}{16} = \frac{1215-4\cdot h}{80} & \text{otherwise} \end{cases} \tag{3.113}$$
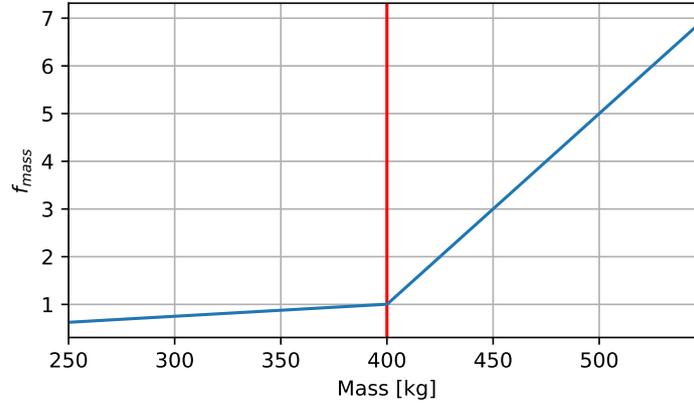
(a) Objective function for the mass of the MAV.
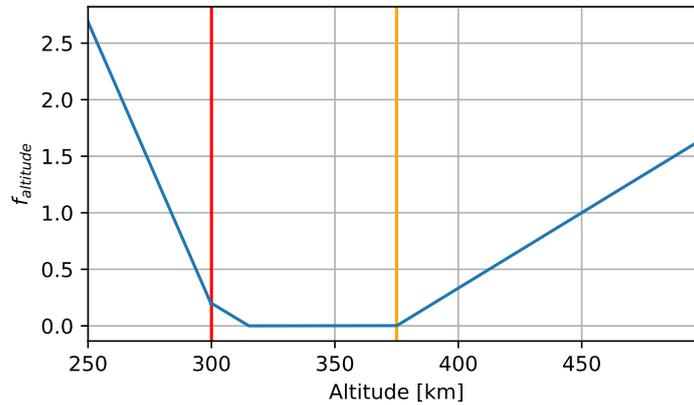
(b) Objective function for the final orbital altitude.

Figure 3.15: Representation of the functions mapping the simulation results to the objective space.

An additional objective to minimise the cost of the MAV could have been added. However, only the SRM will have their size changed, and they will thus be the main elements influencing the cost in this study. In addition, the mass of the SRM can be taken as a direct proxy for the cost [44]. While the mass can be modelled precisely, the cost cannot, and taking both of them as objectives would only duplicate the objective of minimising the mass of the MAV.

3.3.3. Design space exploration

As mentioned by Mooij and Dirkx [86], the design variables can take values within their respective design space, also called the search space. It is important that the design space is known before starting the optimisation. More importantly, non-relevant sections of the design space should be excluded, to allow for a more efficient optimisation process. This can be done via a design space exploration (DSE). The methodology followed to study the design space is to split the design variable in five categories. These categories are the following:

- The SRM geometry of the first stage.
- The SRM geometry of the second stage.
- The initial lift-off and stage separation angles.
- The TVC angles.
- All design variables at the same time.

Different methods have then been used to vary the design variable for each of the categories individually. Also, to capture the impact that design variables would have on each other, variation on the design variables all at the same time, across categories, has also been studied. For each simulation run for given design variables, the MAV state and mass over time has been resampled to 2000 points and saved. Also, data such as the value of the design variables, the value of the objective functions, and the value of relevant parameters such as the burn time of each motor, have all been saved in a database. This allows to browse the results by running requests on the database, selecting sets of design variables or results as needed.

One at a time analysis

This procedure has only been applied to study the design space of the SRM-related design variables. As discussed by Hamby [87], it consists in keeping all of the design variables constant, and varying only one of the design variable at a time. The baseline values are the one used to generate the simulation benchmark that were discussed in Section 3.2.4.

This process is applied for all of the different type of SRM considered: spherical, tubular, rod and tube, multi-fin, and anchor.

Monte Carlo

A Monte Carlo has been applied to all categories of design variables individually, and together. With it, each design variable is randomly chosen within its range using a uniform distribution. This means that, to cover the design space to a satisfying extent, a high number of samples are required. Also, this methods allows to find out what kind of probabilistic distribution, if any, exists in the outputs, in light with the probabilistic distribution of the inputs.

Sobol sequence

As introduced by Sobol [88], the variance-based sensitivity analysis, also called the Sobol method, is a probabilistic model. A Sobol sequence is meant to cover a multi-dimensional space more uniformly, leaving less room for large spaces without any design variable. This means that, in theory, less samples are required with this method than with a Monte Carlo.

Still, it was decided to use a combination of both a Sobol sequence and a Monte Carlo to explore the design space. This is to ensure a good coverage thanks to the Sobol sequence, and still allow for a more random exploration to be carried through Monte Carlo. As such, 6000 samples have been taken from a Monte Carlo, and 8192 from a Sobol sequence, for each category of design variables, and for each type of SRM for the first stage. In total, this means that 283 840 propellant burn and ascent simulations have been run for the design space exploration.

Results analysis

From the results of the one at a time analysis, the cross-sectional geometry of each resulting SRM has then been plotted, so that the most extreme cases could be visually inspected. From this, the range in which the design variables can vary has been manually tuned, ensuring that no combination of design variables would result in unfeasible SRM design, for instance containing unmanufacturable features.

The results of both the Monte Carlo analysis and the Sobol sequence have been analysed in a similar manner. First of all, the altitude over time of all of the ascent simulations run has been plotted in one figure for each of the design variable category. While the resulting plot is not the most readable, it allows to see whether simulations cover the trajectory space to a satisfying extent.

Then, plots have been made of the objective scores as a function of the design variable values. The goal of this plot is to then see if some part of the design variables range always results in very high objectives score. If that is the case, the design variable range can then be narrowed down, to keep only design variables that results in low objective scores.

Last but not least, heat maps have been made based on the correlation between the design variables values and the objectives scores. This allows to reflect on which design variables impacts the simulation the most. If some design variables are found to not affect simulation results or objective scores, they can be excluded from the optimisation.

3.3.4. Initial population

The initial population of the optimisation could be initialised at random, by letting Pagmo randomly select design variables. However, this means that the initial population would most likely be far from optimum, and it would take the first generations of the optimisation to even starts finding good solutions.

Instead, it has been decided to start from a population sampled amongst the best solutions from the design space exploration. The goal through this is to start with a population that is already close to optimum, but still contains solutions that differs from each other, to keep a mixed population. As such, optimum design variables from the design space exploration are selected from different categories, as shown in Table 3.7.

... of the initial population	is taken from the design space exploration of...
2/7	the initial lift-off and stage separation angles
1/7	the TVC angles
3/7	the SRM geometry
1/7	all design variables at the same time

Table 3.7: Initial population composition.

3.3.5. Optimiser

In this case, an optimiser is the algorithm that varies the design variables of a set of points, run the simulations to compute the different objective scores and, based on the scores of the entire population, compute the next design variables to be evaluated, making the entire set of points, also referred to as population, evolve. Each iteration is called a generation. This section explores the different optimisers that are implemented in Pagmo, how one has been selected, and how it has been tuned to offer the most efficient optimisation for the given research.

Available algorithms

The following optimisation algorithms are implemented in Pygmo for multiple objectives problems [89]:

- Non-dominated Sorting Genetic Algorithm (NSGA2): this algorithm is a variation of the genetic algorithm. In it, different sets of design variables constitute so called population members. They are encoded as bit-strings. A new generation is made by using a specific crossover of population members and bit mutations, and then using non-dominated sorting and the distance between the population members [90].

- Multi-objective Evolutionary Algorithm with Decomposition: this algorithm decomposes the population to find optimum solutions in different directions, and combining them, using a Differential Evolution operator [91].

- Multi-objective Hypervolume-based Ant Colony Optimisation (MHACO): this algorithm maps the individuals in the population to a hypervolume, in order to archive the different solutions. Then, it uses non-dominated fronts to rank them and generate new population members [92].

- Non-dominated Sorting Particle Swarm Optimisation (NSPSO): this algorithm treats the individuals in the population as particles, and iteratively moves them closer to a local optimum. The variant implemented contains three different methods that help ensures the diversity of the Pareto front [93].

- Improved Harmony Search: this algorithm is essentially a genetic algorithm, with the improvement that each of the individual tunes itself to the other ones over time, to progressively arrive at an harmonised population [94].

To make the optimisation process more CPU-efficient, it is required that batch fitness evaluation is supported by the optimiser. This means that, instead of running simulation serially, having to wait for each simulation to end before starting a new one, the optimiser will run the simulations in batches, running simulations in parallel in different threads. From the optimisers listed above, only the NSGA2, MHACO, and NSPSO optimisers allow for both multi-objective optimisation and batch fitness evaluation.

Optimiser selection

To select the best optimiser given the problem at hand, the three different optimisers have been used to run separate optimisations. In this process, the default parameters of each optimiser are kept, as implemented in Pagmo. Then, two sets of plots have been made for each optimiser.

The first set of plots shows the value that each design variable has at each generation. To be able to compare the design variables values to each other, their value has been scaled to fir their allowed range. These plots offer a visual aid to investigate how well the optimiser keeps variation in the design variables, which is important for the earlier generations, to ensure that the optimiser is not stuck in a local optimum.

The second set of plots shows the objective score of each individual in the population, also as a function of the generation. The allows to investigate if and how the optimiser is improving the objective scores, and thus converging to optimums.

Optimiser tuning

Once an optimiser has been selected, it is now time to tune its different parameters. This is done varying the parameters one-at-a-time, in decreasing order of expected influence. The parameter that is estimated to have the highest influence on the optimisation is thus first varied between different values, while the other are kept to their default values. The effect that the given parameter has on the optimisation is then analysed, and the best value for the given parameter is chosen. Then, the next parameter is varied, taking the default values except for the first parameter that was now tuned. This process is repeated until all parameters have been tuned.

It is important to note that the optimiser has been tuned only with the optimisation that uses the multi-fin SRM for the first stage, while optimisations have later also been run for the tubular, rod and fin, and anchor SRM, using the same optimiser. This is done to prevent having to run the same time consuming process four times. Still, it is estimated that the multi-fin SRM optimisation is representative of the optimisation for the other SRMs, as it uses the second highest number of design variables, one shy of the anchor SRM.

3.3.6. Optimum identification

After a multi-objective optimisation has been carried, many optimums can be considered best depending on which of the objectives is prioritised. To present the optimum solutions, Pareto fronts have been generated. As described by Kumar et al. [95], a Pareto front is the collection of points that dominate the solution space. That is, there is no solution that scores higher than a Pareto optimal solution without scoring lower on a different objective. This is illustrated in Figure 3.16.
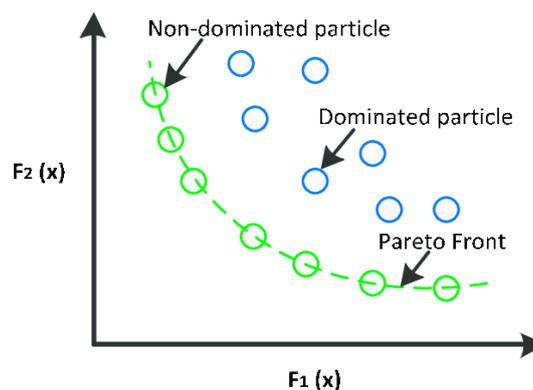
Figure 3.16: Pareto front illustration, with objectives F_1 and F_2 [95].

In the case of this research, Pareto fronts have been generated by plotting the sum of the objective of the periapsis and apoapsis altitude, versus the objective in mass. Presenting the Pareto fronts in this way offers two advantages. First, this allows to make 2D plots. Second, it allows to see how low the mass of the MAV can get while staying within the required altitude range.

3.3.7. Optimum refinement

The last step in the optimisation process is to refine the Pareto fronts of optimal solutions to try and find even better solutions. To that extent, a Sobol sequence has been used to vary the design variable of the Pareto optimum points, to potentially find even better solutions.

This process is run by first varying the design variables corresponding to the Pareto optimal points by multiplying them by factors between 0.999 and 1.001, selected by the Sobol sequence. For this, 64 design variable variations are investigated for each Pareto optimal point. Then, optimums are refined by varying the design variables not by a given factor, but by a random value within 0.1% of the design variable range. This is once again done using a Sobol sequence and taking 64 variations per point. Last but not least, the point that leads to the absolute minimum mass while adhering to the requirement on the target altitude is taken. Then, the design variables of this points are varied by up to 0.25% of the design variable range. In this case, a Sobol sequence is used with 1024 samples.

Results

This chapter presents all of the different results that have been generated through this research. First, the results that come from setting up the different models are shown in Section 4.1. Follows the discussion of the simulation results in Section 4.2, discussing the basic feasibility of the MAV ascent, and the challenges of the more complex simulation to reach a satisfactory accuracy. Then, the Section 4.3 compile the results from the optimisation setup, and the optimisation results themselves. Last but not least, Sections 4.4 and 4.5 present the results of the validation and sensitivity analysis that have been run to guarantee the credibility of the results.

4.1. Models setup

The first results are the ones that emerge from setting up the different models. These results contain the aerodynamic coefficients of the MAV from Missile DATCOM and from the DSMC, and the parameters of the SRM propellant used for both stages.

4.1.1. Missile DATCOM

First, the aerodynamic coefficients of the first section of the MAV have been computed using Missile DATCOM. To do so, the geometry of the MAV needs to be known. To this effect, a CAD model has been made based on the preliminary design discussed in Section 2.2.2. A drawing of this CAD model can be seen in Figure 4.1.

This CAD model was used to first estimate the reference area of the MAV, which is taken to be of $S_{ref} = 0.144$ m^2. What is important here is not the exactitude of the value, but to ensure that the same one is used in Missile DATCOM and in the ascent simulation, for the aerodynamics coefficients to be consistent.

Also from this CAD, the coordinated of 15 points have been sampled along the line that, once revolved, would form the MAV. This sampling is done by trial and error so that, once processed by Missile DATCOM, these points represent the MAV as expected.

The sampled points are represented in blue in Figure 4.2. In the same figure, the orange points represent the reconstructed geometry as computed by Missile DATCOM. Comparing Figures 4.1 and 4.2, it is judged that Missile DATCOM satisfactorily understands the geometry of the MAV.
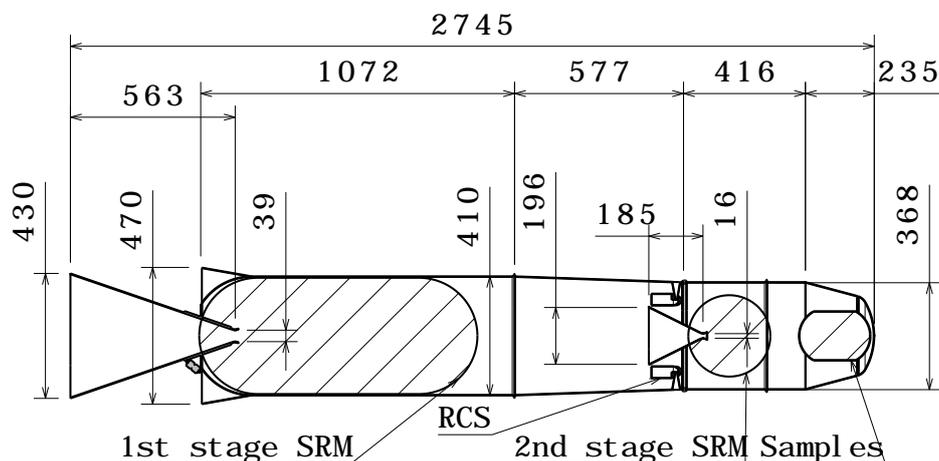
Figure 4.1: Dimensions of the MAV based on a CAD generated using CATIA.
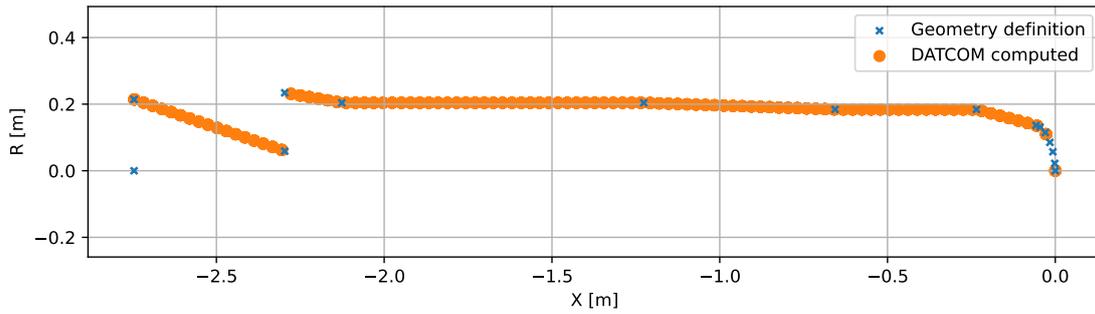
Figure 4.2: Geometry of the MAV in Missile DATCOM.

Running Missile DATCOM, and processing the output, the aerodynamic coefficients of the first section are extracted as a function of both the Mach number and the angle of attack. Due to the geometry of the vehicle, and not varying the sideslip angle in Missile DATCOM, the side force coefficients are all zero. During the ascent simulation, a 2D interpolation will then be performed by Tudat to get the lift and drag coefficient as a function of the angle of attack and Mach number.

The drag and lift coefficients are represented in Figures 4.3 and 4.4 respectively. As seen, the drag coefficient does not significantly vary with the angle of attack, since the angle of attack is only varied between the low values of -4 deg to 4 deg. The drag coefficient also shows a significant increase in the transonic regime, with a peak of about $C_D = 1.4$ at Mach number 1.2. Also, as one would expect from the radial symmetry of the MAV, the lift coefficient corresponding to the angle of attack α is equal to the lift coefficient of the opposite angle $-\alpha$.
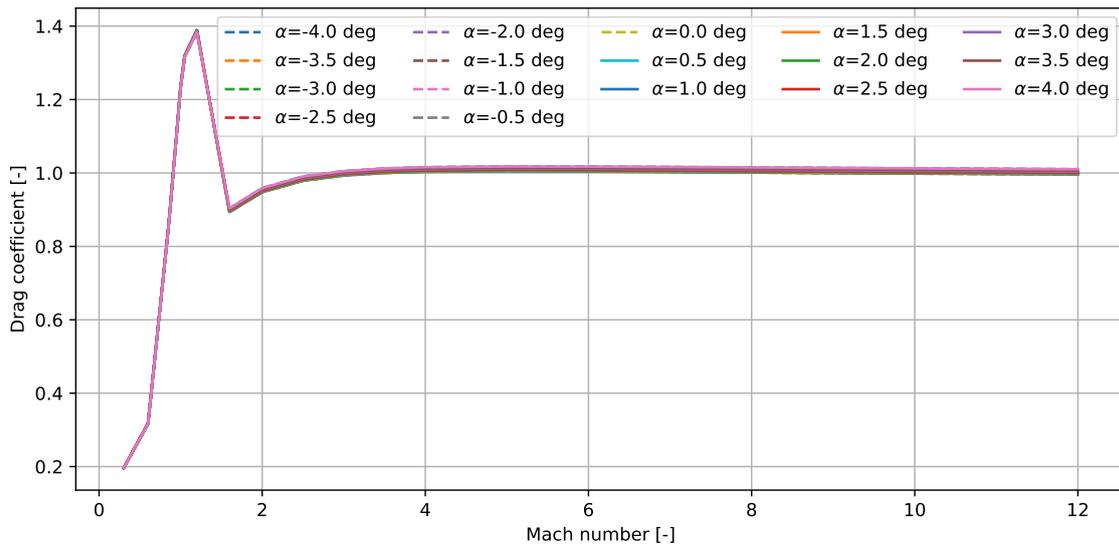
Figure 4.3: Drag coefficient of the MAV as a function of Mach number for different angles of attack, from Missile DATCOM.
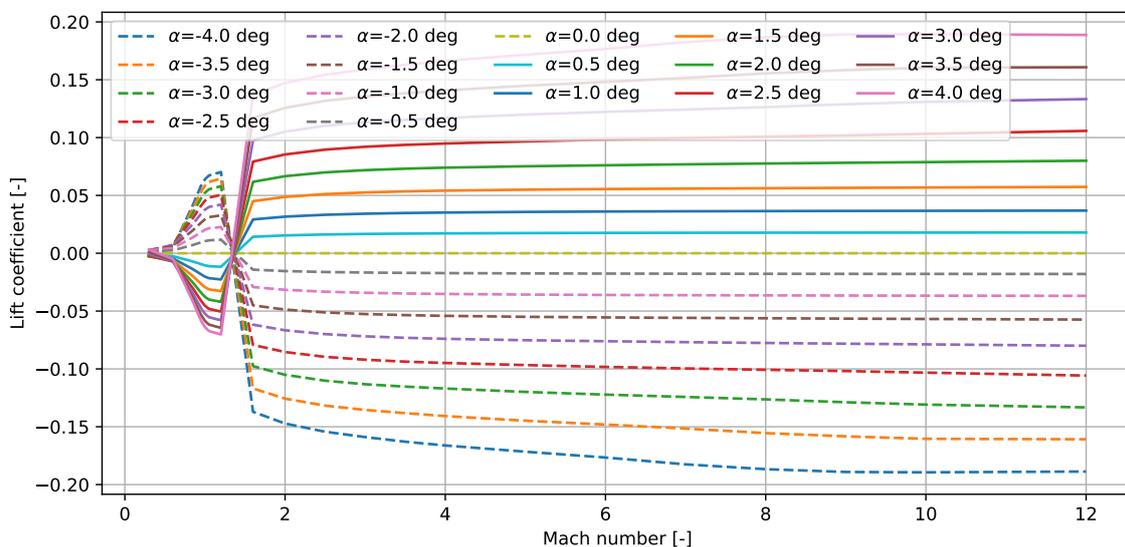
Figure 4.4: Lift coefficient of the MAV as a function of Mach number for different angles of attack, from Missile DATCOM.

4.1.2. DSMC

The drag coefficient of the second stage is obtained as a function of altitude from DSMC. To do so, the atmospheric conditions must be known for each altitude for which DSMC will be run. These are compiled in Table 4.1.

Altitude [km]	Velocity [m/s]	Density [kg/m³]	Temperature [K]	Pressure [Pa]	Mixture [mol/mol] CO2, N2, Ar, CO, O, O2
100	3.50334E+03	1.82031E-07	1.28643E+02	4.58579E-03	93.270%, 2.445%, 2.513%, 0.965%, 0.607%, 0.200%
125	3.49086E+03	4.29462E-09	1.33166E+02	1.07168E-04	81.674%, 6.398%, 4.972%, 3.851%, 2.509%, 0.596%
150	3.47851E+03	1.94534E-10	1.72404E+02	7.27184E-06	65.826%, 11.923%, 4.910%, 7.796%, 8.533%, 1.012%
175	3.46629E+03	2.15307E-11	1.78405E+02	9.83263E-07	44.654%, 16.782%, 3.754%, 10.801%, 22.822%, 1.186%
200	3.45420E+03	3.60822E-12	1.79196E+02	2.05882E-07	24.723%, 17.401%, 2.297%, 11.148%, 43.375%, 1.055%
225	3.44223E+03	8.64996E-13	1.79330E+02	6.32226E-08	11.796%, 14.477%, 1.210%, 9.397%, 62.345%, 0.775%
250	3.43039E+03	2.74321E-13	1.79369E+02	2.59061E-08	5.102%, 10.644%, 0.582%, 7.033%, 76.132%, 0.507%
275	3.41867E+03	1.01443E-13	1.79380E+02	1.29177E-08	1.900%, 7.190%, 0.249%, 4.862%, 85.498%, 0.301%
300	3.40707E+03	4.26962E-14	1.79382E+02	7.73174E-09	0.648%, 4.581%, 0.098%, 3.174%, 91.333%, 0.166%
350	3.38421E+03	1.10285E-14	1.79383E+02	4.31813E-09	0.074%, 1.758%, 0.015%, 1.271%, 96.835%, 0.047%
400	3.36181E+03	4.70352E-15	1.79383E+02	3.26268E-09	0.012%, 0.679%, 0.003%, 0.501%, 98.792%, 0.014%
450	3.33985E+03	2.97560E-15	1.79382E+02	2.72788E-09	0.004%, 0.280%, 0.001%, 0.203%, 99.507%, 0.005%
500	3.31831E+03	2.29431E-15	2.29431E-15	2.35800E-09	0.003%, 0.132%, 0.001%, 0.088%, 99.775%, 0.002%

Table 4.1: Atmospheric conditions as a function of altitude, used as DSMC inputs.

While the DSMC is run to get the drag coefficient as a function of altitude, it is also an opportunity to investigate how the Martian atmosphere behaves around the second stage.

The velocity of the flow in the x-direction is shown in Figures 4.5 and 4.6 for altitude of 100 km and 500 km respectively. These figures show slices of the 3D simulation, visualised using the ParaView software [96]. This shows that, at low altitude, the particles that hit the MAV are reflected back according to the cosine law, but then collide with the incoming flow and get back to the MAV, creating a sort of boundary layer. At higher altitudes, the incoming is less dense, hence is stops the reflected flow less. This is seen in the more focused beam of reflected particles.

This phenomenon is also caused by the higher concentration in atomic Oxygen in the Martian atmosphere at higher altitudes. As discussed in Section 3.1.1, this means that the accommodation coefficient is higher, making the particle reflection more specular.
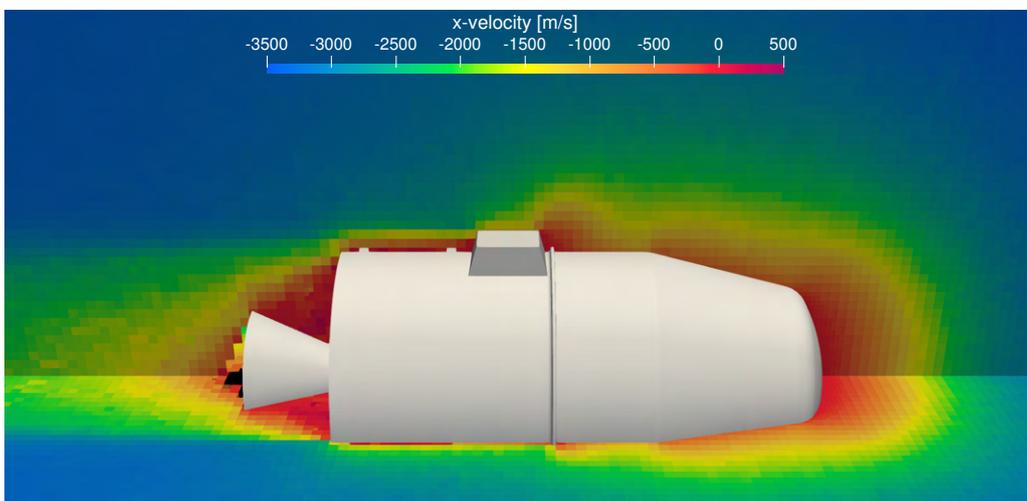
Figure 4.5: Velocity of the atmosphere around the MAV at an altitude of 100 km, from the DSMC.
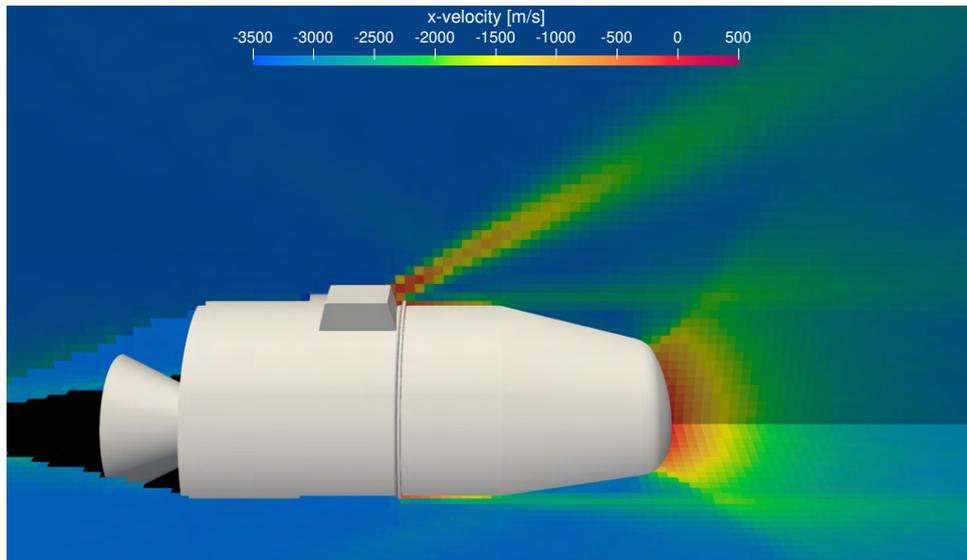
Figure 4.6: Velocity of the atmosphere around the MAV at an altitude of 500 km, from the DSMC.

The mass density of the flow around the second stage is then shown in Figures 4.7 and 4.8, at altitudes of 150 km and 500 km respectively. These figures show that, at 150 km, there is a shockwave in front of the vehicle, as seen by the bubble of higher density in front of the nose. At higher altitudes, the shockwave starts to separate, and the atmosphere is freer to flow around the vehicle.
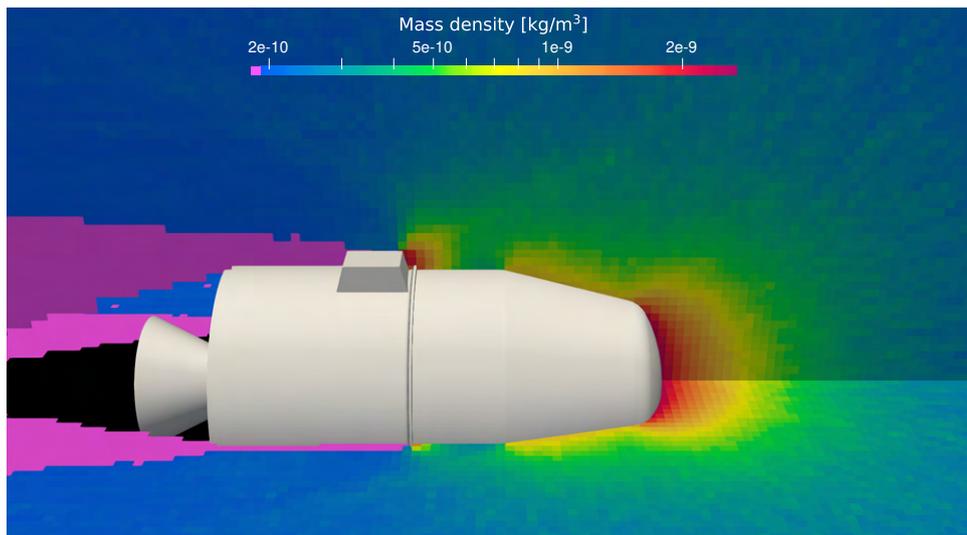
Figure 4.7: Mass density of the atmosphere around the MAV at an altitude of 150 km, from the DSMC.
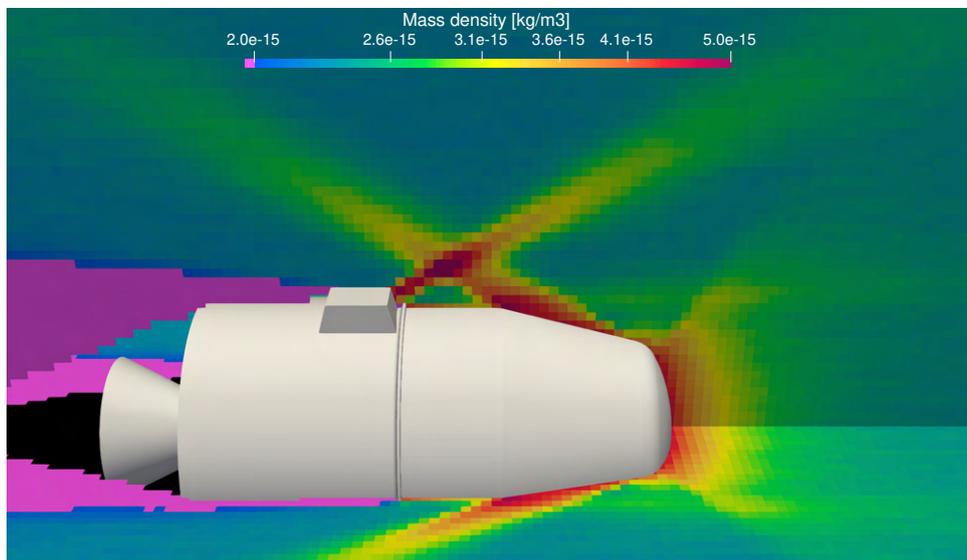
Figure 4.8: Mass density of the atmosphere around the MAV at an altitude of 500 km, from the DSMC.

Finally, the drag coefficient as a function of altitude is shown in Figure 4.9. This coefficient is computed using once again a reference surface area of $S_{\text{ref}} = 0.144$ m^2. This shows that the drag coefficient peaks at around 130 km, before decreasing and then plateauing after around 250 km. This can be explained by the atmospheric conditions also varying less with the higher altitude.

Moreover, a quadratic spline has been fitted through the drag coefficient as a function of altitude, represented in orange. During the simulation, Tudat will then use this spline to interpolate the drag coefficient for any altitude.
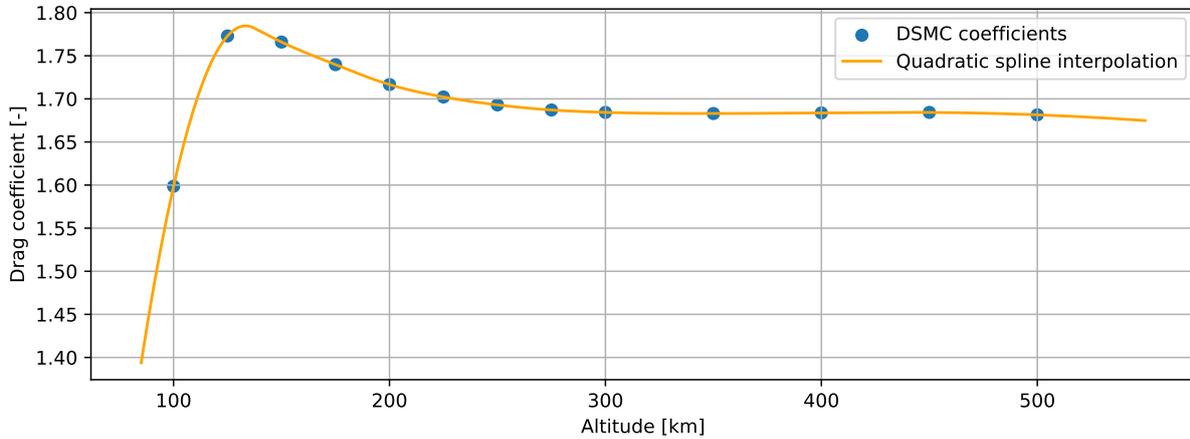
Figure 4.9: Drag coefficient for the MAV as a function of altitude from DSMC.

4.1.3. SRM propellant

The different parameters required to run the SRM burn simulation are compiled in Table 4.2. It is worth noting that, while an effort is made to get values as close as possible to the ones of the TP-H-3544 propellant, the exact values are not known. However, the code base that supports this research has been made to be easily adapted to updated parameters values.

Table 4.2 shows both the proxy values that have been found in literature, as well as the values used in the simulation, making an informed decision based on the sources. The molar mass and specific heat ratio have been found not in literature, but using the CEA and RPA-C software, and comparing results.

In case of the CEA software, a flame temperature of 3645 K was used, and a chamber pressure range of 10 bar to 75 bar. In case of the RPA-C software, a chamber pressure of 35 MPa was used. In both cases, the propellant recipe was set to 70% AP, 16% Al, and 14% HTPB.

In addition to the parameters compiled in Table 4.2, a combustion efficiency of 0.93 and a thrust efficiency of 0.95 have been used, as conservative estimates inspired by values from the reader of Zanbergen, B. [40].

Parameter	Source	Value from source	Value used for simulation
T_c [K]	[19]	3645	3645
ρ_p [kg/m^3]	[58]	1860	1854.5
	[20, 59]	>1749	
n [-]	[60, 61]	0.31	0.31
a [mm/s/MPa^n]	[60, 61]	4.202	4.202
M [g/mol]	CEA	22.75 to 24.528	24.14
	RPA-C	26.2646	
γ [-]	CEA	1.1148 to 1.194	1.125
	RPA-C	1.1575	

Table 4.2: Propellant parameters used for the SRM burn simulation.

4.2. Simulation

Trough baseline simulations and their setup, various results have also been gathered. First, the semi analytical ascent model is used to ensure the feasibility of the MAV to reach orbit within the mass margin. Then, a benchmark simulation is setup, that is used to tune the integrator, environment, and select a propagator to reach a pre-defined positional and velocity accuracy.

4.2.1. Semi analytical ascent

Results from the semi analytical ascent model that was discussed in Section 3.2.1 are presented in this section. First of all, the parameters that have been used to run this simulation are listed in Table 4.3. In addition, a value of $S_{ref} = 0.144$ [m^2] is used for the MAV reference surface area, and a constant drag coefficient of $C_D = 1.1$, as average from the Missile DATCOM results. These are extracted from the work of JPL discussed in Section 2.2.2.

Parameter	Value		
	Stage 1 burn	Coasting	Stage 2 burn
F_T [N]	9854	0	6937
c_{eff} [m/s]	293 · 9.81 = 2874.33	N.A.	282 · 9.81 = 2766.42
M_P [kg]	216	N.A.	54
M_0 [kg]	385	385-216=169	99
δ [deg]	30	N.A.	180

Table 4.3: Parameters used to run the semi analytical ascent simulation.

Table 4.4 then showcases the ΔV budget of the MAV ascent, as computed using the semi analytical simulation. In this table, the results from using a vertical ascent and a gravity turn are both presented. This showcases that, as one would expect, gravity losses are lower when using a gravity turn, at the expense of an increase in drag loss. The coasting and second stage circularisation phases are then based on the final state after the gravity turn. This ΔV budget also shows that, during the short time of the second stage burn, drag losses are insignificant. However, it is expected that drag losses will become non-zero as the MAV stays in orbit for a longer period of time.

Ascent phase	Final velocity	Initial velocity	ΔV from thrust	ΔV from gravity loss	ΔV from drag loss
First stage vertical ascent	2200.6	100	2366.5	-234.4	-31.5
First stage gravity turn ($\delta = 30$ deg)	2236.8	100	2366.5	-179.3	-50.5
Coasting (almost ballistic flight)	1557.9	2236.8	0	-678.0	-0.8
Second stage circularisation	3740.6	1557.9	2181.0	1.6	0

Table 4.4: ΔV budget based on the semi analytical ascent simulation. All values are in m/s.

Furthermore, this semi analytical simulation allows to plot the altitude and velocity of the MAV over time. As shown in Figure 4.10, the final altitude of the MAV after the circularisation burn is just shy of 400 km, and the final velocity is of 3740 m/s. As previously shown in Equation 3.1, orbital velocity on Mars at an altitude of 300 km is of 3407 m/s. Since the orbital velocity decreases with altitude, this means that the MAV has enough velocity once at 400 km to stay in orbit, according to this semi analytical simulation.
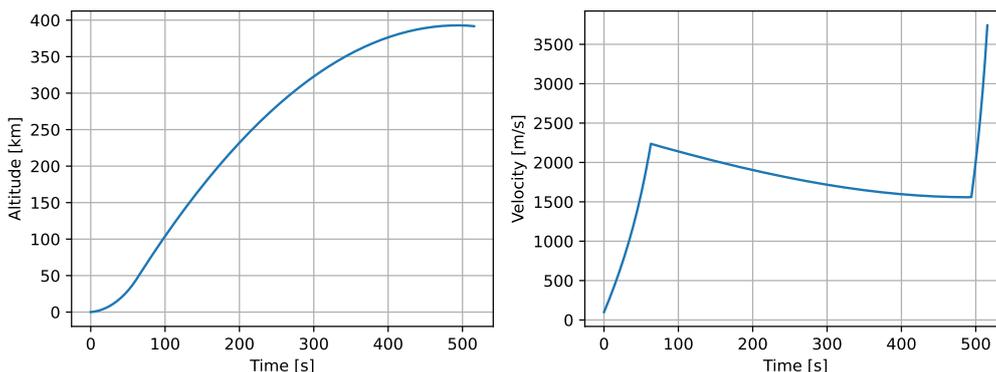
Figure 4.10: Altitude and velocity of the MAV over time using the semi analytical ascent simulation and a gravity turn with $\delta = 30$ deg.

Finally, from the thrust and effective velocity, the burn time of both motors has been computed. It is of $t_b = 63$ s for the first stage, and $t_b = 21.5$ s for the second stage. Also, the coasting phase between stage 1 burnout and apogee lasts 431 s, or 7 minutes and 11 seconds.

In conclusion, this semi analytical MAV ascent simulation shows the apparent feasibility of the MAV reaching orbit for its given mass and thrust, and it serves as a baseline to compare velocity and burn times from the more detailed simulations later on.

4.2.2. Benchmark

To generate the benchmark, baseline values are required for the SRM geometry and ascent parameters. These values have been tuned manually, based on the ascent trajectory that results from running the simulation with them. Since the integrator and environment have not been tuned yet, a setup is then guessed, as to tune these baseline values. This setup is to use a variable step size integrator with RKF7(8) coefficients and a tolerance of 1E-18. The environment is setup to comprise the classic exponential atmosphere model, the SRM thrust, and SH up to D/O 4 for Mars gravitational field. The manually tuned baseline parameters are compiled in Table 4.5, in case the first stage SRM is of the multi-fin type. The baseline values that have been manually tuned for the different first stage SRM types are presented and discussed in Appendix A.1. Running the SRM burn and ascent simulation with these parameters for the multi-fin first stage SRM results in periapsis and apoapsis altitudes of 300 km and 505 km respectively, with a MAV GLOM of 409 kg, including 90 kg for the second stage. The TVC deflection angles were set at random, rather than at zero, as to capture the effect of TVC on the simulation later on during the benchmark generation. The SRM parameters were tuned to reach orbit, while keeping close to the thrust level discussed in Section 2.2.2.

Design variable	Baseline value
Launch angle [deg]	57.5
Stage separation angle [deg]	90
TVC in y-direction, 5 nodes [deg]	0, 2.86, 5.73, 0, 2.85
TVC in z-direction, 5 nodes [deg]	0, -2.86, 0, 2.86, 2.86
Spherical SRM outer radius fraction $R_{o,2,\text{frac}}$ [-]	0.6875
Spherical SRM inner radius fraction $R_{i,2,\text{frac}}$ [-]	0.5555
First stage SRM length L [m]	1.05
First stage SRM outer radius R_o [m]	0.24
Multi-fin SRM inner radius fraction $R_{i,1,\text{frac}}$ [-]	0.729
Multi-fin SRM number of fins N_f [-]	20
Multi-fin SRM length fraction $L_{f,\text{frac}}$ [-]	0.2857
Multi-fin SRM width fraction $w_{f,\text{frac}}$ [-]	0.5093

Table 4.5: Baseline values manually tuned for the benchmark, using a multi-fin first stage SRM.

The first step in the benchmark generation has been to find the highest accuracy possible for the thrust simulation of both stages. For the first stage, the error in mass and thrust as a function of the number of function evaluations can be seen in Figure 4.11. As expected, increasing the number of function evaluations, and thus decreasing the time step, the final error decreases.

However, just after 1E+08 function evaluations, as marked by the vertical green line, the error starts to increase again. This is the point that has been identified as where the computer numerical accuracy is reached. As such, the highest accuracy possible for the thrust simulation of the first stage contains an estimated error of about 8E-05 N and 1E-06 kg.

The same effect can be seen in Figure 4.12, for the simulation of the second stage thrust. One can also see that the errors behave more linearly over the number of function evaluations for the second stage thrust than for the second stage thrust. This is because the first stage SRM uses a multi-fin geometry, that causes an important jump in thrust when the fins burnout. This jump in thrust causes the error in thrust and in mass to be more erratic.
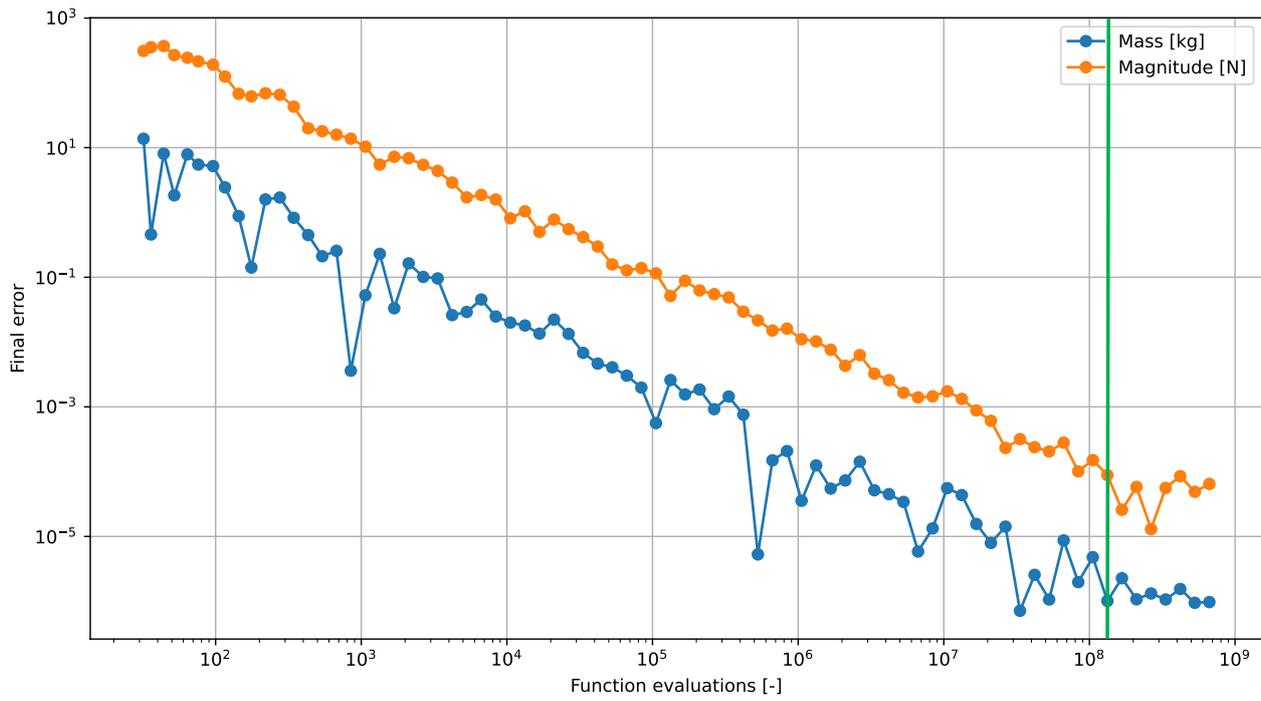
Figure 4.11: Error in thrust and mass versus function evaluations for the stage 1 thrust integration.
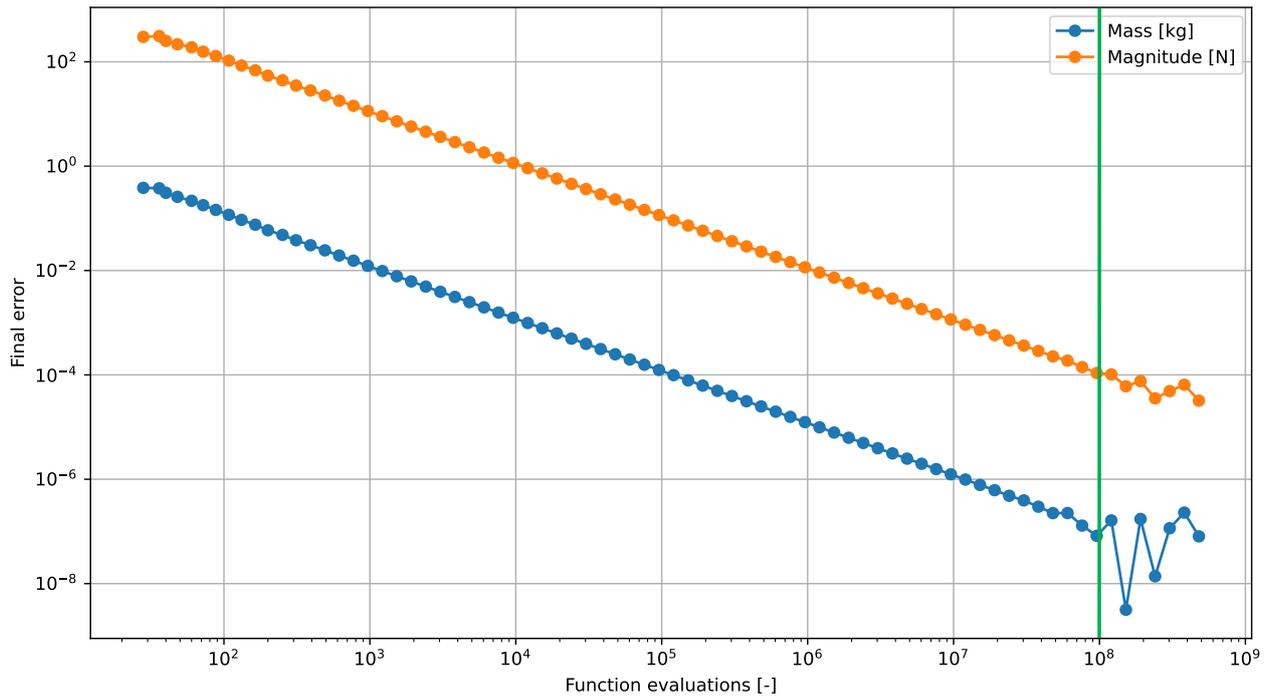
Figure 4.12: Error in thrust and mass versus function evaluations for the stage 2 thrust integration.

Table 4.6 presents a summary of the best accuracy feasible for the thrust simulation from the SRM of both stages. In this table, the last columns list the relationship between the error in magnitude and in mass. Indeed, as can be seen in Figures 4.11 and 4.12, the errors can be seen to decrease with the same slope, but scaled by some factor. This factor is thus reported in Table 4.6, and will be of use later in the study of the feasibility of this benchmark.

Stage	Best time step [s]	Number of function evaluations	Magnitude error [N]	Mass error [kg]	Ratio magnitude / mass error [-]
1	7.9036E-07	1.3289E+08	8.8375E-05	1.0117E-06	90
2	9.9500E-07	9.5764E+07	1.0849E-04	8.2331E-08	930

Table 4.6: Summary of the best achievable accuracy for each SRM propellant burn.

With the benchmark for the thrust simulation of both stages completed, the benchmark has been generated for each phase of the ascent. The results of this process can be seen in Figures 4.13 to 4.14 for the first stage, and in Appendix A.2 for the second stage. These show that, as for the thrust simulation, there is a constant factor linking the error in position and in velocity. Furthermore, the best feasible accuracy is reached with roughly the same number of function evaluations for each phase.
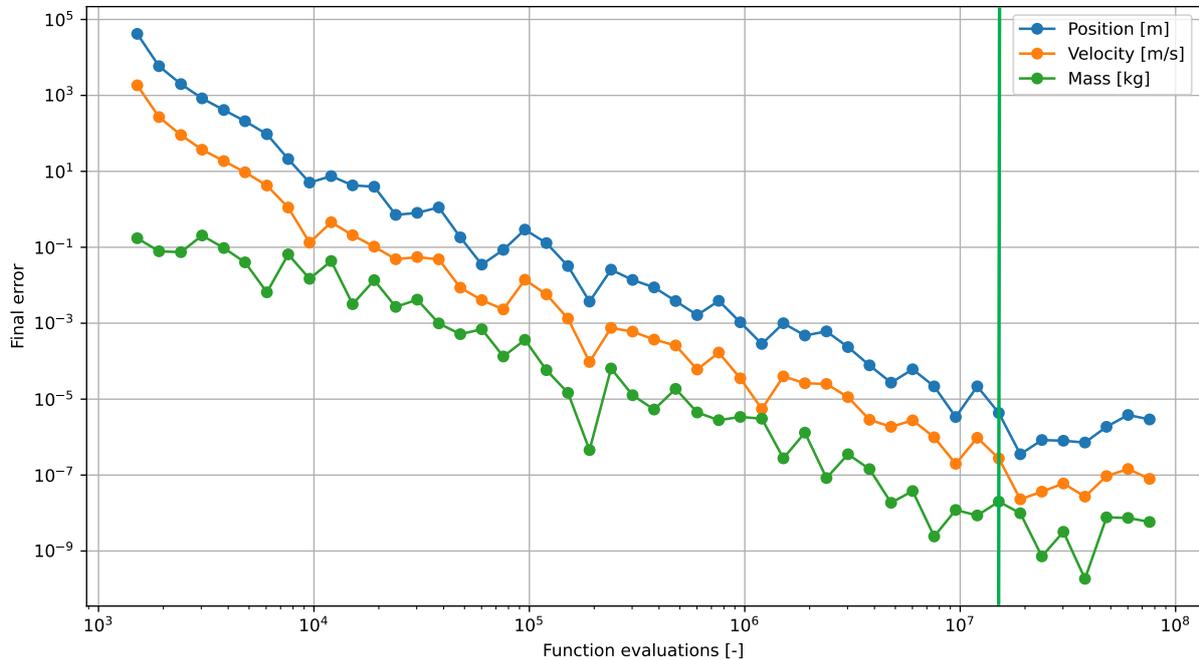
Figure 4.13: Error in position, velocity, and mass versus function evaluations for the stage 1 powered ascent integration.
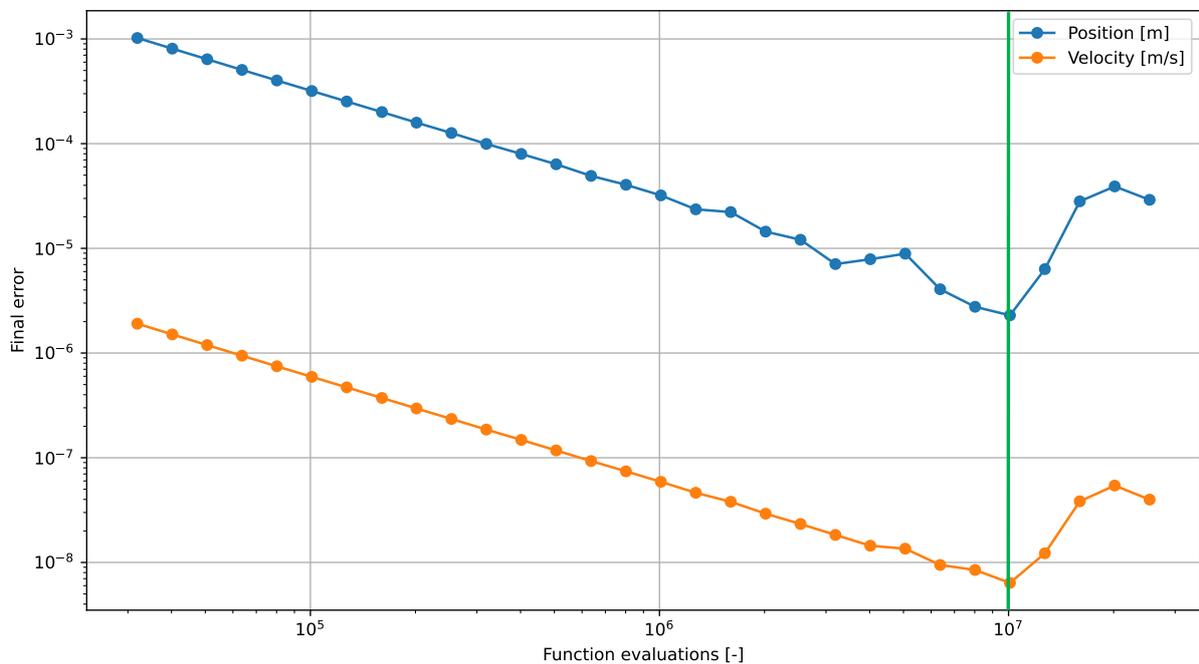
Figure 4.14: Error in position and velocity versus function evaluations for the stage 1 unpowered ascent integration.

Table 4.7 compiles the best achievable accuracy for each phase of the ascent simulation. One point of interest is that the feasible accuracy for the unpowered phases of the ascent is reached with a higher time step than for powered phases. This is because the unpowered simulations typically run for considerably higher simulated times than the powered ones. This means that any rounding error has more time steps to accumulate.

Stage	Thrust	Best time step [s]	Number of function evaluations	Position error [m]	Velocity error [m/s]	Ratio position / velocity error [-]
1	Yes	4.9868E-05	1.9019E+07	3.5362E-07	2.2993E-08	10
1	No	1.9853E-03	1.0093E+07	2.2928E-06	6.3764E-09	285
2	Yes	3.9612E-04	2.1651E+06	1.5594E-06	1.4792E-07	10
2	No	6.2780E-03	5.1570E+07	1.4626E-03	1.3215E-06	1000

Table 4.7: Summary of the best achievable accuracy for each section of the ascent.

In splitting the benchmark in different sections of the ascent, the error that would be present in the earlier phase of the ascent simulation do not affect the later phases. This is not realistic, and could lead to the benchmark, when run as one single simulation, to be significantly less than the benchmark when run as separate simulations.

For the benchmark, the target accuracy is set to be 100 times lower than the simulation accuracy of 5 km and 5 m/s. This means that the benchmark target accuracy is of 50 m and 5 cm/s. Having this factor of 100 thus allows to detect errors when tuning the simulation with a higher level of certainty. This also allows to analyse the error in position and velocity of the simulation below the target, which may put light to possible integrator stability errors.

This effect has been studied by faking an error in position and velocity at the beginning of the second stage unpowered simulation, and looking what is the maximum error that still allows to reach the required benchmark accuracy of 50 m and 5 cm/s. Variations in position and velocity are made at the same time, using the ratio of their errors as analysed in the benchmark, and reported in Table 4.7.

The results of this process is shown in Figure 4.15. From it, one can see that, to reach a final accuracy of 50 m and 5 cm/s, the initial error cannot be higher than 1 m and 1 mm/s. The same procedure is then run for the second stage powered ascent simulation, with a required accuracy this time of 1 m and 1 mm/s. Repeating this process until the first stage powered ascent results in a maximum allowed error of 5.6E-04 m and 1.9E-06 m/s at the end of first stage powered ascent. These required accuracies for each ascent phases are summarised in Table 4.8. Similar plots to the one of Figure 4.15 are presented in Appendix A.2 for the other ascent sections.

In this table, the errors from the highest accuracy benchmark simulations are presented on the same rows. This allows to check whether the benchmark can offer the required accuracy for each ascent phase. As the minimum feasible final errors are lower than the maximum allowed final errors, the benchmark can indeed propagate errors lower than 50 m and 5 cm/s.
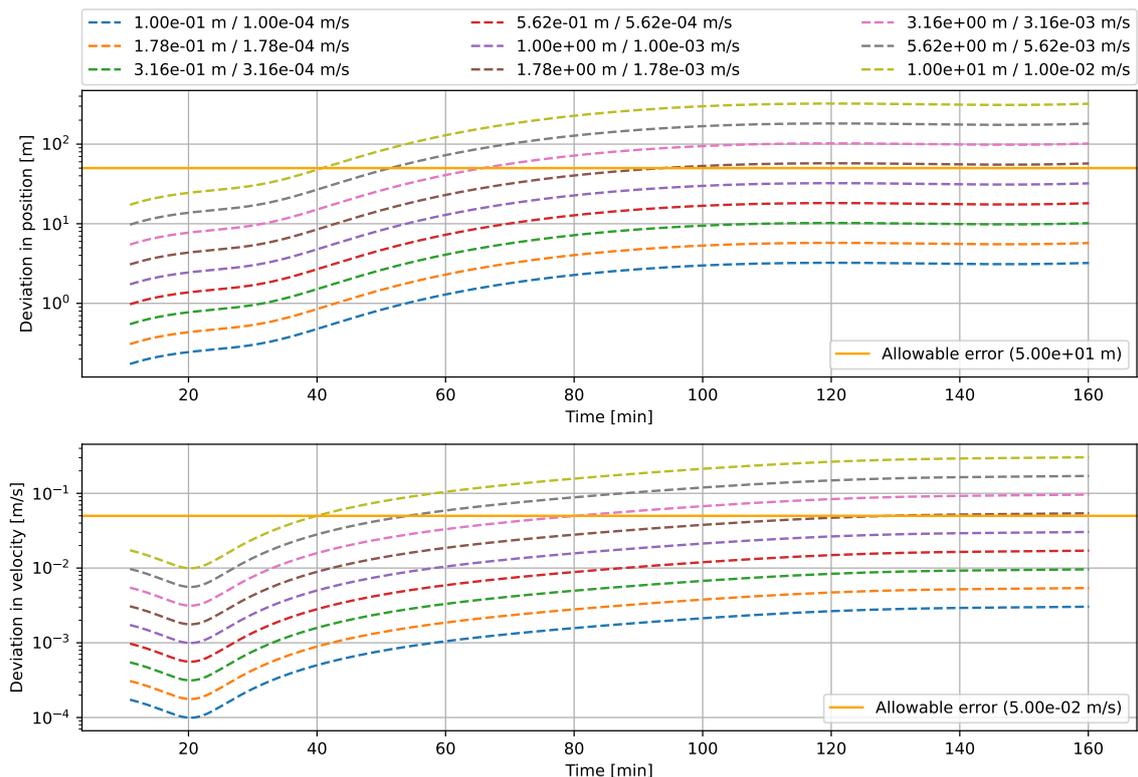
Figure 4.15: Effect of variations in initial state on the unpowered second stage ascent.

Stage	Thrust	Maximum allowed final error		Minimum feasible final error	
		Position [m]	Velocity [m/s]	Position [m]	Velocity [m/s]
2	No	5.0E+01	5.0E-02	1.4626E-03	1.3215E-06
2	Yes	1.0E+00	1.0E-03	1.5594E-06	1.4792E07
1	No	5.6E-03	5.6E-04	2.2928E-06	6.3764E-09
1	Yes	5.6E-04	1.9E-06	3.5362E-07	2.2993E-08

Table 4.8: Summary of the feasible and required benchmark ascent integration accuracy for each phase, for a final accuracy of 50 m and 5 cm/s.

However, ensuring that errors in position and velocity do not increase over the benchmark accuracy requirement is just half the effort. It is also important to ensure that the thrust simulation is accurate enough to not introduce errors that will grow.

4.2.3. Integrator tuning

To tune the thrust RK4 integrator time step, the study from Section 4.2.2 is run again. However, this time, the target accuracy is of 5 km and 5 m/s, being the accuracy desired for the simulations during the optimisation. The results of this are compiled in Table 4.9, with plots of the error in position and velocity as a function of the initial state error shown this time shown in Appendix A.3. Unsurprisingly, the values of this table are the same as the ones of Table 4.8, reduced by a factor of 100, as is the required accuracy.

Stage	Thrust	Maximum allowed final error	
		Position [m]	Velocity [m/s]
2	No	5.0E+03	5.0E-00
2	Yes	1.0E+02	1.0E-01
1	No	5.6E-01	5.6E-02
1	Yes	5.6E-02	1.9E-04

Table 4.9: Summary of the required ascent integration accuracy for each phase, for a final accuracy of 5 km and 5 m/s.

From Table 4.9, the required accuracy for the powered phases of the ascent are known. A constant error in thrust and mass flow is then simulated for the SRM burn simulation, as to investigate what is the maximum allowed error, given the accuracy requirement for the powered ascent phases. For the second stage SRM burn simulation, Figure 4.16 shows that a maximum error of 3.2 N and 3.4 g/s is acceptable.
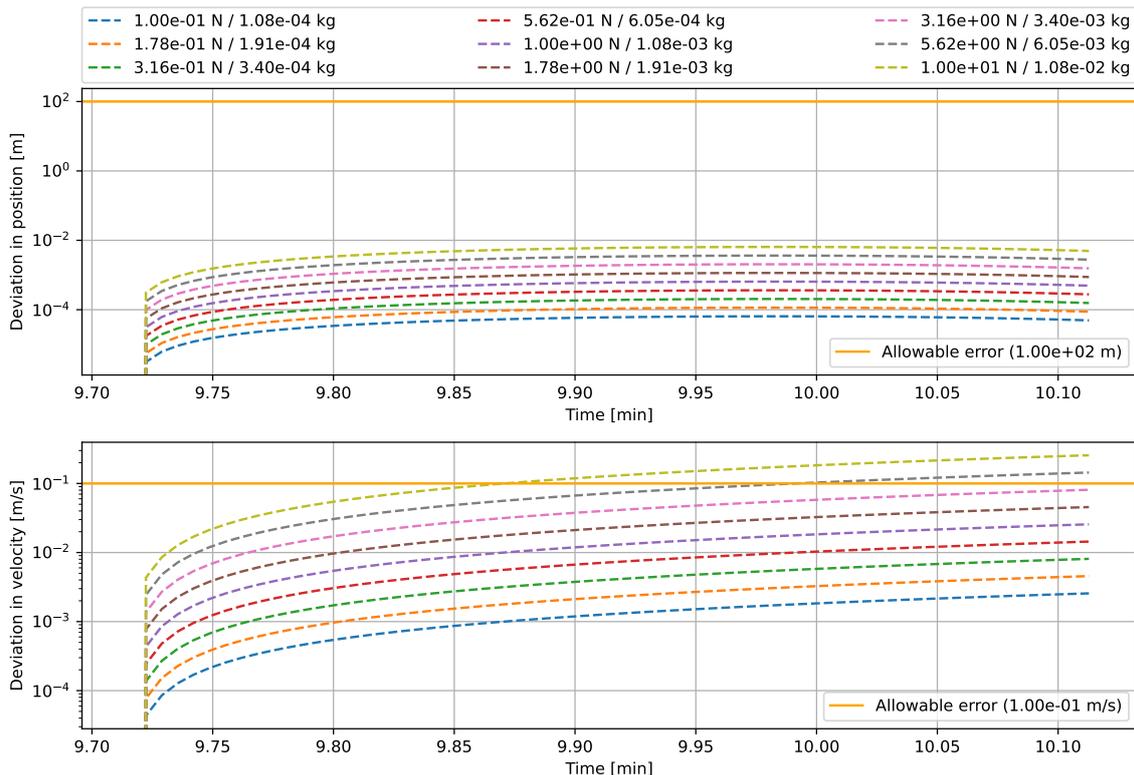
Figure 4.16: Effect of variations in thrust on the powered second stage ascent.

The required SRM burn simulation accuracy is summarised in Table 4.10 for both stages. The time step that corresponds to these errors is also reported in the same table, as well as the CPU time required to run the SRM burn simulation that achieves such accuracy. These time steps represent the tuned RK4 integrator values for the SRM burn simulation, which will be used during the optimisation simulations. Keeping some margin, it is then recommended to use a time step of 2E-06 s for the first stage burn simulation, and of 1.5E-02 s for the second stage burn simulation.

For the first stage SRM burn simulation, the accuracy requirement is such that is takes more than 9 min to run on an Intel© Xeon© E5-2683 CPU cadenced at 2 GHz. This is unacceptable, since this will bottleneck the entire optimisation process. As such, an effort has been made to first optimise the code, and ensure that constant values are computed only once. This however did not change the required CPU time significantly. Instead, the SRM burn simulation has been re-written in C++, where it was previously implemented in Python. Doing so, and interfacing the C++ burn simulation with the Python ascent simulation, the CPU time is reduced by a factor of more than 36, down to 15 s.

Stage	Maximum allowed error		Corresponding time step [s]	Function evaluations [-]	CPU time [s]
	Thrust [N]	Mass flow [kg/s]			
2	3.2E+00	3.4E-03	2.7E-02	3.4E+03	0.075
1	3.2E-04	3.5E-06	3.2E-06	3.3E+07	550 → 15

Table 4.10: Summary of the feasible and required benchmark SRM burn integration accuracy.

Then, the integrator used during the ascent simulation itself also needs tuning. As such, a fixed step Runge-Kutta integrator is used, varying the time step and the integration coefficients, and comparing the results with the benchmark results to get an estimate of the error for each combination. This results in Figure 4.17 for the error in position. From this figure, it can be seen that the explicit mid-point integration method is the one that reached the required accuracy represented by the horizontal orange line, with the least function evaluations. However, this number of function evaluations is still of about 3E+06. The error in velocity is shown in Appendix A.3.
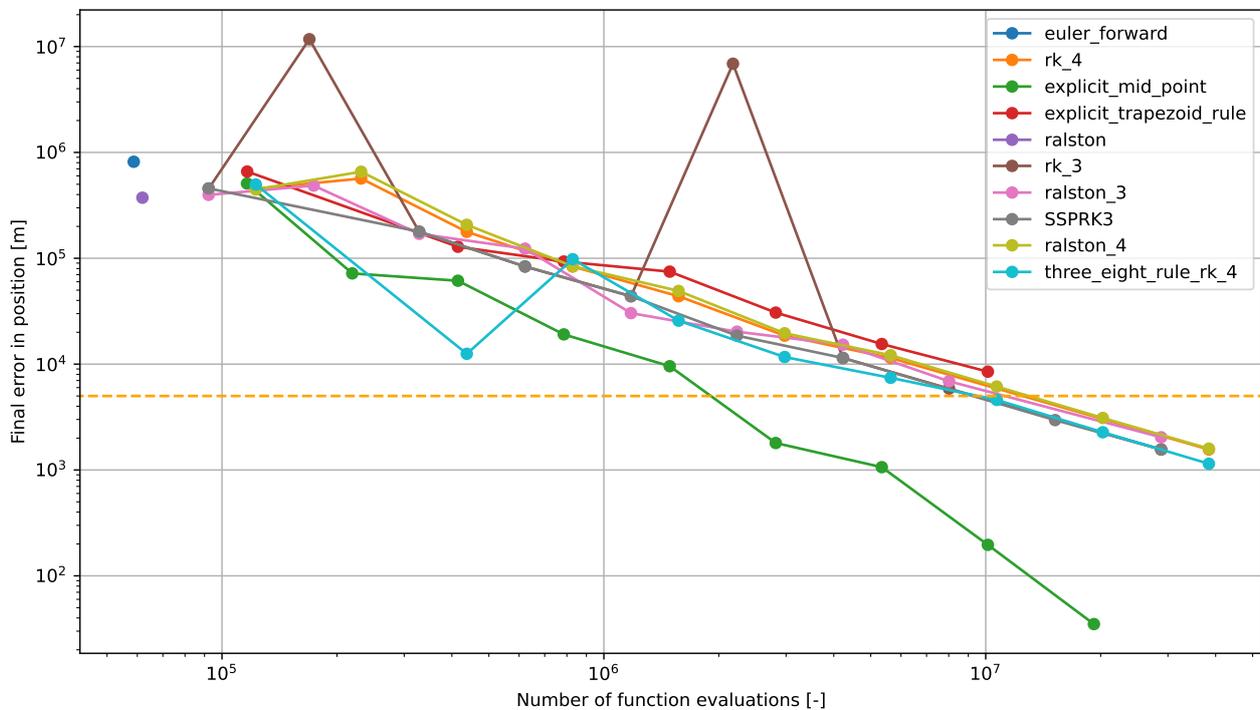
Figure 4.17: Error in position versus number of function evaluations for various fixed step integrators.

It has then been decided to run the same study, but with variable step integrators. This makes more sense given the problem at hand, since a lower time step is required when thrust is used. Using a variable step, the integrator is expected to automatically adjust its time step based on whether thrust is active or not. This results in Figure 4.18, showing the error in position versus the number of function evaluations for the different integration coefficients. This figure was generated varying not the time

step, but the time step control tolerance. A similar plot showing the error in velocity is shown in Appendix A.3.

As seen, the RKF4(5) variable step method reaches an accuracy of 3.6 km (and 3.3 m/s) with just shy of 2E+03 function evaluations, 1500 times less than for fixed step integration. This corresponds to a step size control tolerance of 1E-07.
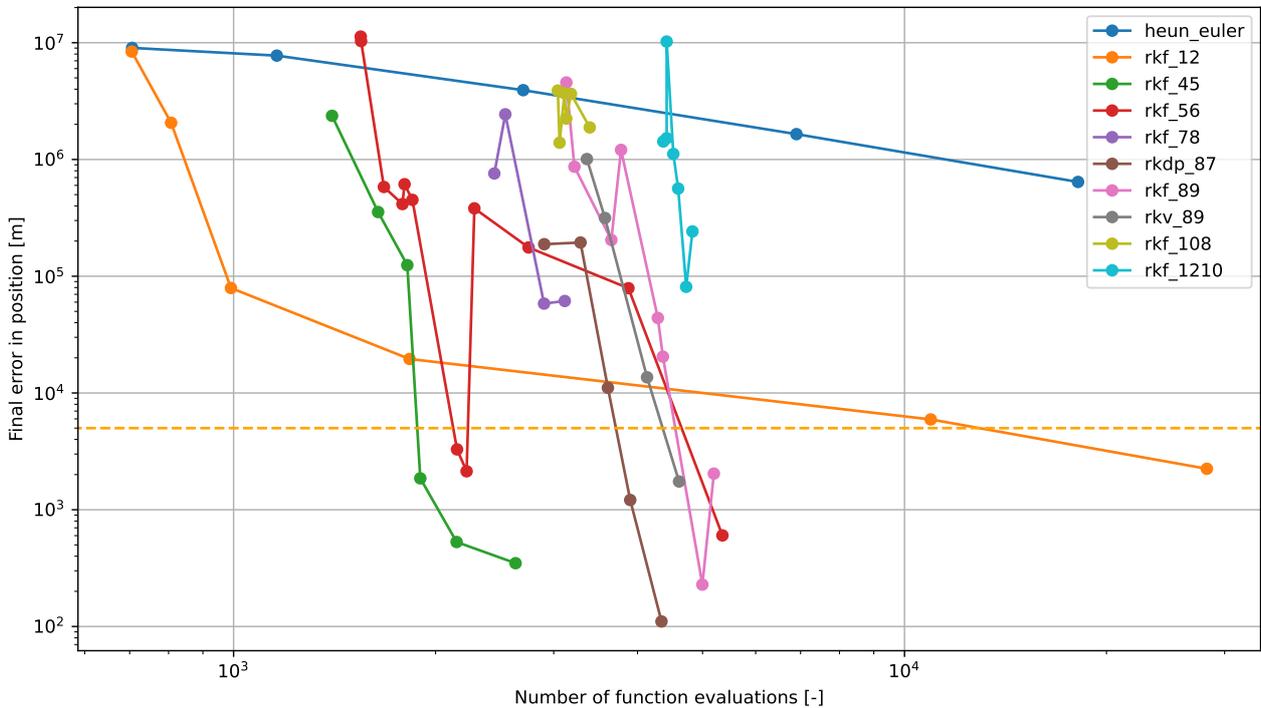
Figure 4.18: Error in position versus number of function evaluations for various variable step integrators.

The time step that is computed by the step size control algorithm over time from the RKF4(5) method is shown in Figure 4.19. This confirms that the time step automatically gets reduced when thrust is activated or deactivated. It is worth noting that, during use of thrust, the time step is automatically allowed to grow higher, since there are less changes in dynamics. During the unpowered phases, the time step progressively grows. These phases are coasting and an orbital propagation once the MAV is in orbit. It thus makes sense that the time step is allowed to be higher during these phases.
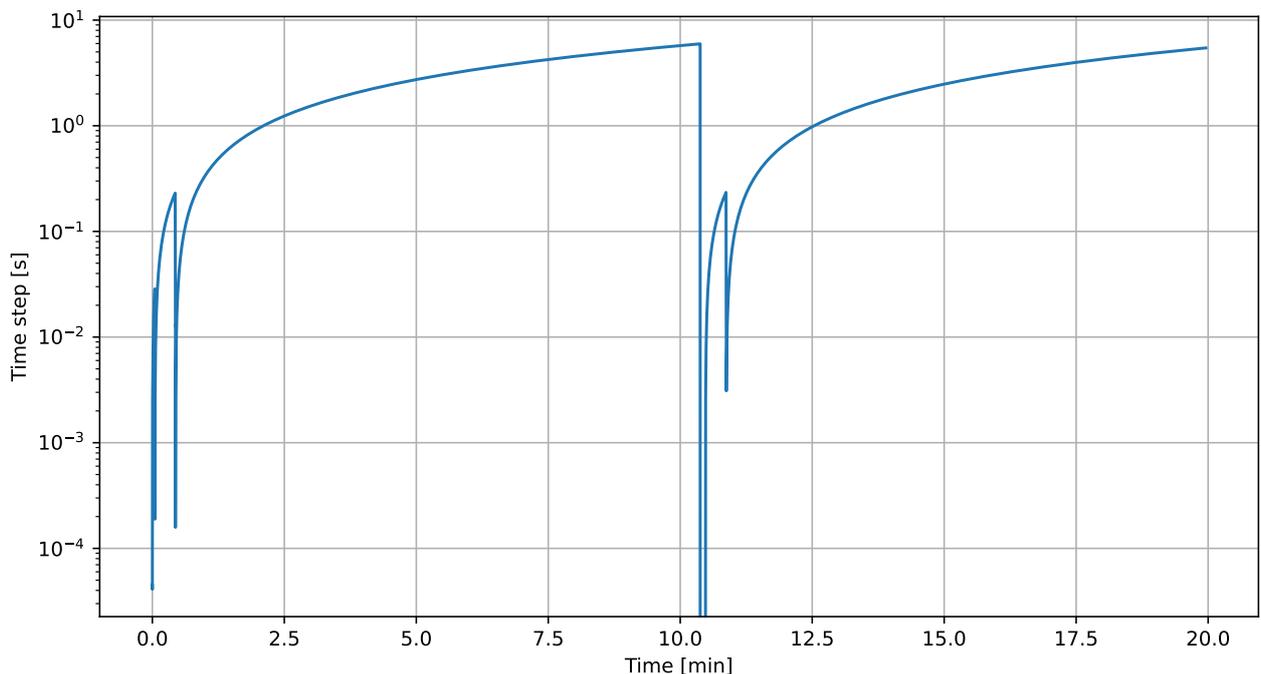
Figure 4.19: Evolution of the time steps from the variable step size control for the benchmark simulation.

In conclusion, the following integrator setup is to be used for the optimisation simulations, to ensure an accuracy of 5 km and 5 m/s:

- First stage thrust simulation:
  - RK4 integrator
  - Fixed time step of 2E-06 s
- Second stage thrust simulation:
  - RK4 integrator
  - Fixed time step of 1.5E-02 s
- For the ascent simulation:
  - RKF4(5) integrator
  - Step size to vary between 3E-05 s and 60 s
  - Step size control tolerances of 1E-07
  - A maximum and minimum factor increase of 1.01 and 0.01 respectively
  - A safety factor of 0.8

If a higher accuracy is required, it is recommended to lower the time step by a factor of 10 for the thrust simulations, and the step size control by a factor of 10^5 for the ascent simulation.

4.2.4. Environment tuning

With the integrator tuned, the environment tuning is now performed, as to reach the required accuracy using the simplest models possible. The required accuracy is now decreased, to take the integration error into account. As such, the environment tuning is performed to reach a accuracy in position of 1.4 km and in velocity of 1.7 m/s. Each plot made in the environment tuning showing the error in position is also presented in Appendix A.4 showing the error in velocity.

Mars gravitational acceleration

First of all, different models have been tried to represent the gravitational acceleration caused by Mars on the MAV. For each attempt, the error in position and velocity is computed by comparing results with the use of SH up to D/O 14, which is assumed to be of higher accuracy. This results in the errors shown in Figure 4.20, and summarised in Table 4.11. From these, it is decided to use SH up to D/O 6 for the optimisation ascent simulations.
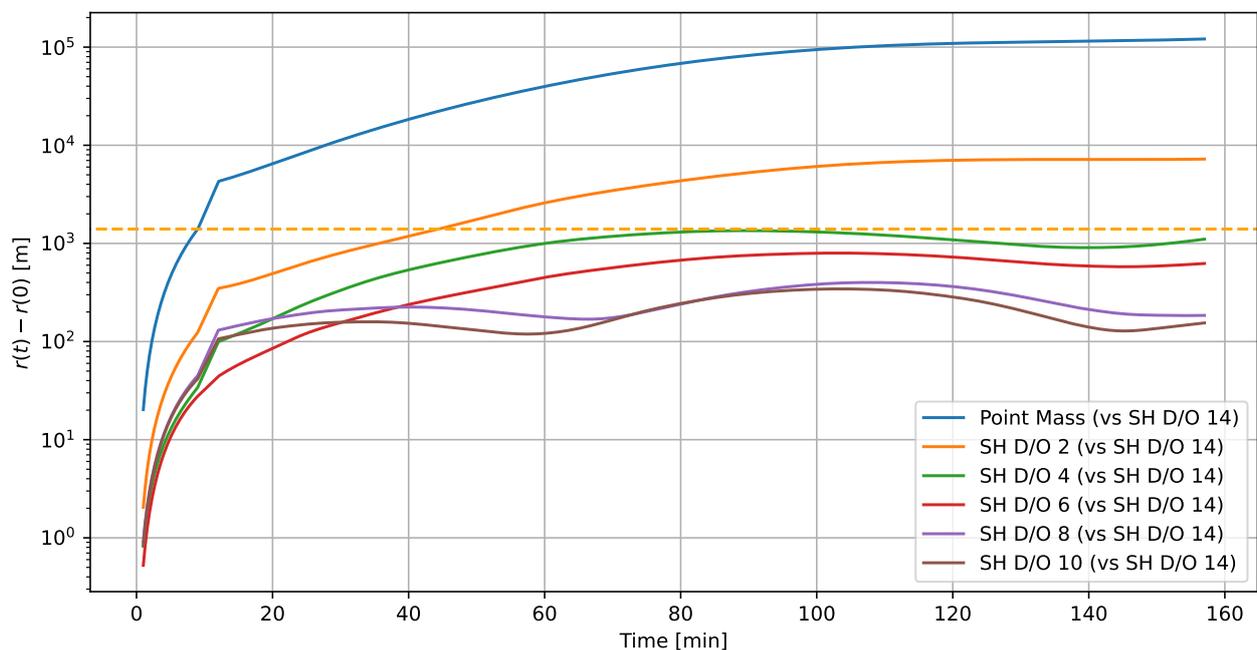
Figure 4.20: MAV relative position over time for different Mars gravitational field models.

Model	Final position difference [m]	Final velocity difference [m/s]
Point Mass	1.21E+05	1.10E+02
SH D/O 2	7.23E+03	6.88E+00
SH D/O 4	5.74E+03	5.61E+00
SH D/O 6	1.10E+03	1.16E+00
SH D/O 8	1.85E+02	1.30E-01
SH D/O 10	1.55E+02	1.33E-01

Table 4.11: State error due to different models for Mars gravitational field, compared to the use of SH up to D/O 14.

Other gravitational accelerations

The gravitational acceleration of other bodies than Mars has also been considered. First, Table 4.12 the distance from Mars of various bodies in the Solar system, as well as their gravitational parameters. From these, the acceleration that these bodies have on another body located on Mars is computed. From the order of magnitude of these accelerations, the Sun will for instance have a significantly higher effect than the Earth or Venus, as one would expect.

Body	Distance from Mars [m]	Distance from Mars [AU]	Gravitational parameter [m³/s²]	Gravitational acceleration on Mars [m/s²]
Sun	2.08e+11	1.39e+00	1.33e+20	3.06e-03
Jupiter	5.90e+11	3.94e+00	1.27e+17	3.64e-07
Venus	3.16e+11	2.11e+00	3.25e+14	3.26e-09
Earth	2.77e+11	1.85e+00	3.99e+14	5.21e-09
Saturn	1.24e+12	8.28e+00	3.79e+16	2.47e-08
Phobos	5.99e+06	4.00e-05	7.07e+05	1.97e-08

Table 4.12: Gravitational acceleration of various bodies on Mars on January 1st 2031.

The ascent simulation has then been run with each the gravitational acceleration of each of these bodies modelled as point masses. Phobos is not included in this study, for which the position is not available at the given dates in the SPICE ephemeris. However, from Table 4.12, it is known that the effect of Saturn and Phobos should be of similar magnitude. This results in Figure 4.21 and Table 4.13, showing that most of these accelerations are irrelevant for the MAV ascent for the target accuracy. It is thus decided to only use the gravitational acceleration of the Sun, as this is the most relevant one, leading to an error of 110 m and 9.97 cm/s if omitted.
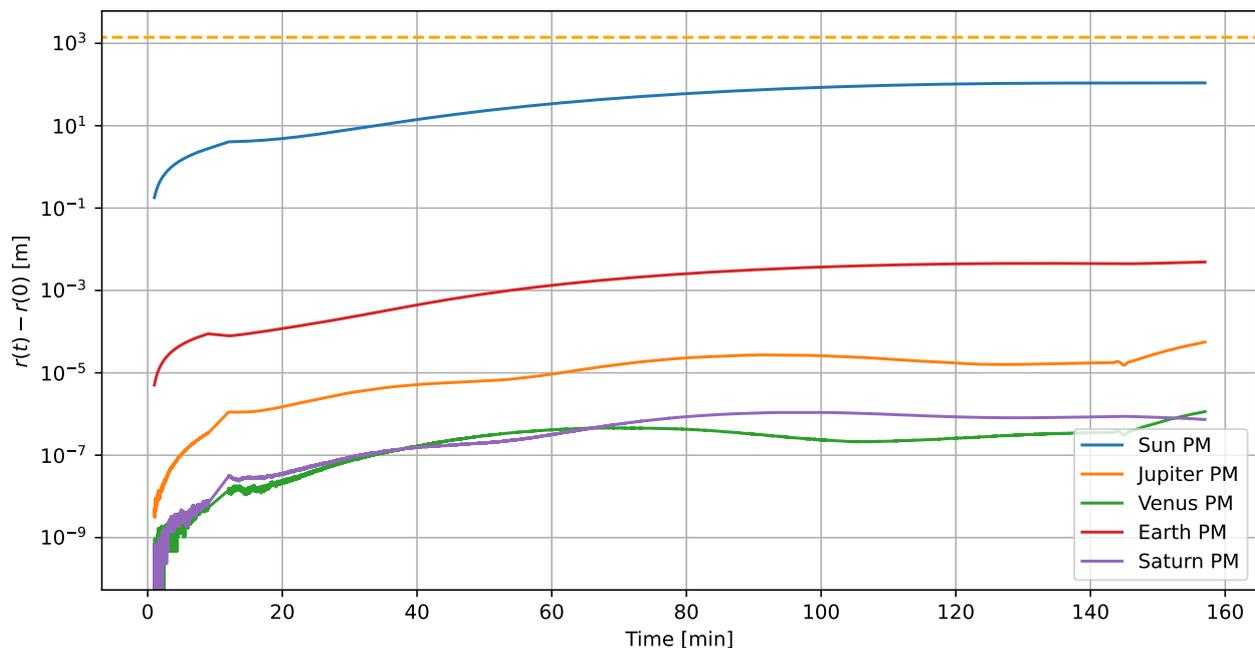
Figure 4.21: MAV relative position over time for different point mass contributions to the environment.

Model	Final position difference [m]	Final velocity difference [m/s]
Sun PM	1.10E+02	9.97E-02
Jupiter PM	5.59E-05	3.77E-08
Venus PM	1.14E-06	7.83E-10
Earth PM	4.92E-03	4.36E-06
Saturn PM	7.38E-07	7.73E-10

Table 4.13: State error due to different bodies PM contributions to the environment.

Miscellaneous accelerations

The accelerations caused by the Solar radiation pressure on the MAV and by Schwarzschild relativistic correction are then considered. Following the same process, Figure 4.22 and Table 4.14 have been generated. This shows that none of these accelerations are required to reach the target accuracy, by a fair margin.
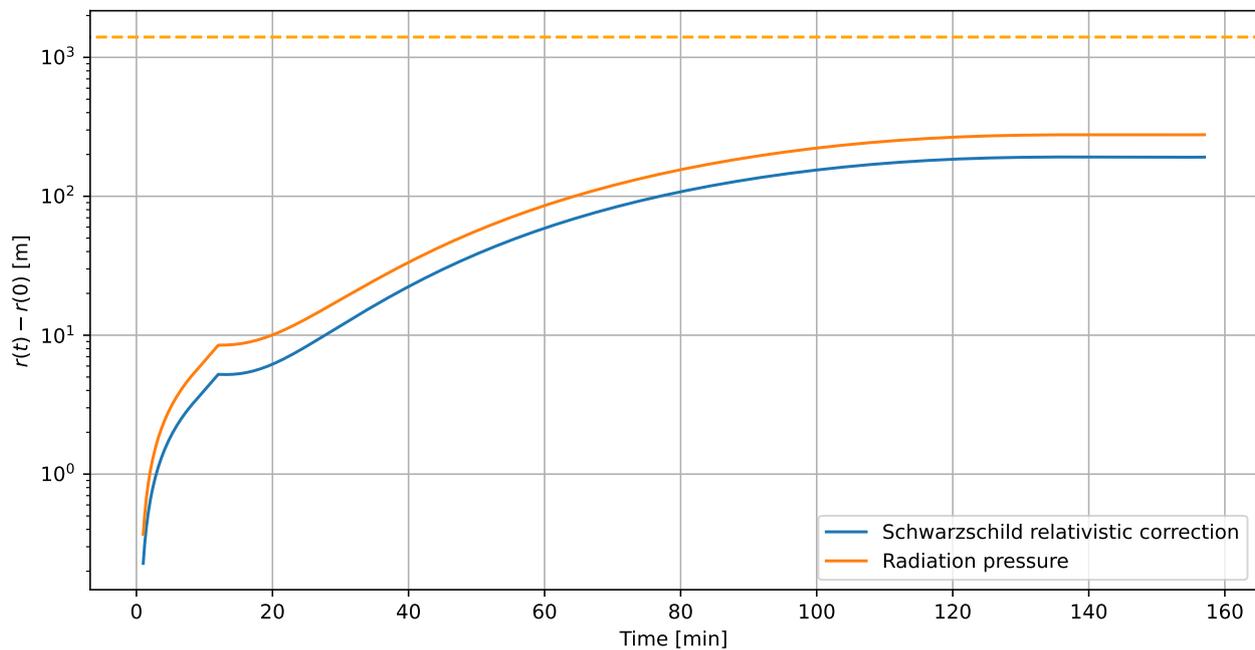
Figure 4.22: MAV relative position over time for different miscellaneous acceleration models.

Model	Final position difference [m]	Final velocity difference [m/s]
Schwarzschild relativistic correction	1.91E+02	1.75E-01
Radiation pressure	2.77E+02	2.54E-01

Table 4.14: State error due to miscellaneous contributions to the environment.

Mars atmospheric drag

Last but not least, the different models for the Martian atmosphere have been compared, which impacts drag. First, the atmospheric density that the MAV encounters during its ascent is plotted as a function of altitude in Figure 4.23 for the different atmosphere models. This clearly shows that the two-step exponential atmosphere and MCD 5.3 models keep relatively close up to roughly 150 km, in thicker and thus more important parts of the atmosphere. However, the classic exponential atmosphere model significantly deviates starting already at around 50 km.
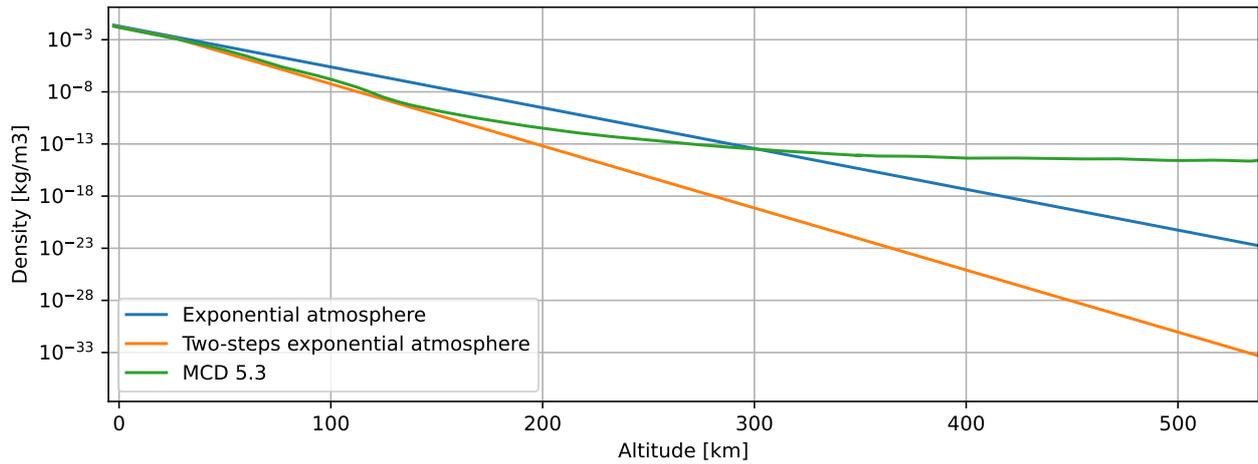
Figure 4.23: Density profile of the different Mars atmosphere models.

Then, the position of the MAV over time is plotted in Figure 4.24, for the use of the different atmosphere models. This shows that, indeed, the highest deviation is reached when using the exponential atmosphere model. However, while towards the end of the simulation, the different deviations appear to correct themselves to some extent, after increasing at around 120 min at periapsis. Also, it is decided to use the two-step exponential atmosphere model for the optimisation, due to the low speed of querying the MCD 5.3 during the simulation, and the fact that the smoother two-step exponential atmosphere models cause less sudden variations in the atmosphere, allowing the variable step size integrator to keep lower time steps. After the optimisation, selected results will be run again with the MCD 5.3 atmosphere model, to see if the results are still within the target accuracy.
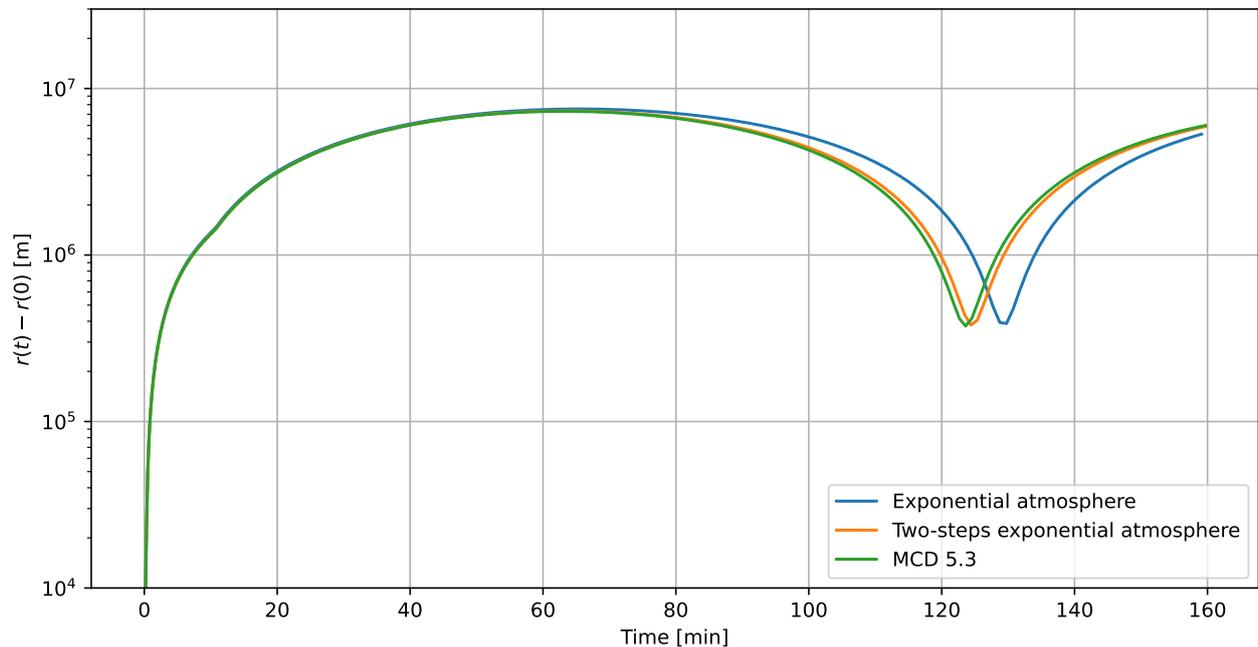
Figure 4.24: MAV relative position over time for different Mars atmosphere models.

In conclusion, the following environment models are to be used for the optimisation ascent simulation:

- Mars gravitational field modelled as Spherical Harmonics up to Degree and Order (SH up to D/O) 6.
- The Sun gravitational field modelled as a point mass.
- The Martian atmosphere modelled as a two-step exponential atmosphere.

In light of these results, in case simulations are run that require a higher accuracy, it is recommended to increase the SH D/O from 6 to 8, to include the radiation pressure from the Sun on the MAV modelled as a cannonball, and the use the MCD to get the atmosphere density.

4.2.5. Propagator selection

With the integrator and environment tuned, the last step is to confirm that the propagator does not introduce errors due to for instance singularities. For the benchmark generation, a Cowell propagator was used. The error in position of the MAV for all propagators compared to Cowell is plotted in Figure 4.25. As seen, the deviations with the Cowell propagator are minor. This seems to indicate that no particular singularity is reached during the simulation.
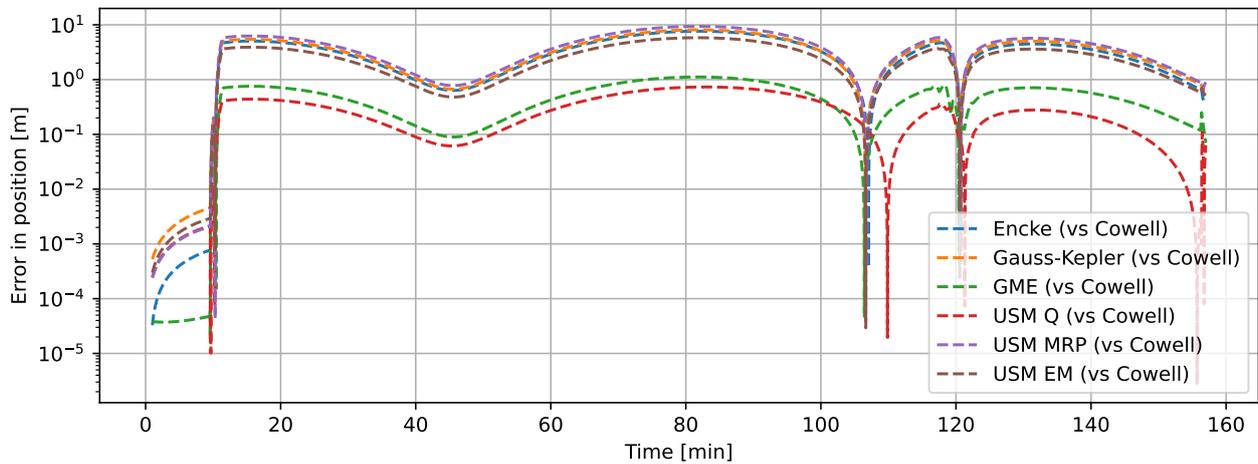
Figure 4.25: MAV relative position over time for different propagators.

Figures 4.26 to 4.27 show different parameters computed by all the propagator during the MAV ascent simulation. First, the eccentricity history shows that, while the initial eccentricity is of 0, the high thrust allows it to decrease to lower values almost immediately. Similarly, the angular momentum, while low at first, increases almost instantly. This means that, while these are potential singularities, these parameters reach non-critical values in a few time steps, preventing any of the propagator to introduce significant errors. Moreover, the inclination of the MAV never reaches 0 deg nor 180 deg, preventing any singularity to arise.

This explains why no significant differences are found when selecting a distinct propagator. As such, it is decided to keep the Cowell propagator, since the integrator and environment were tuned to it. This prevents having to re-run the integrator and environment tuning to ensure that no difference would have been introduced during a change of propagator.
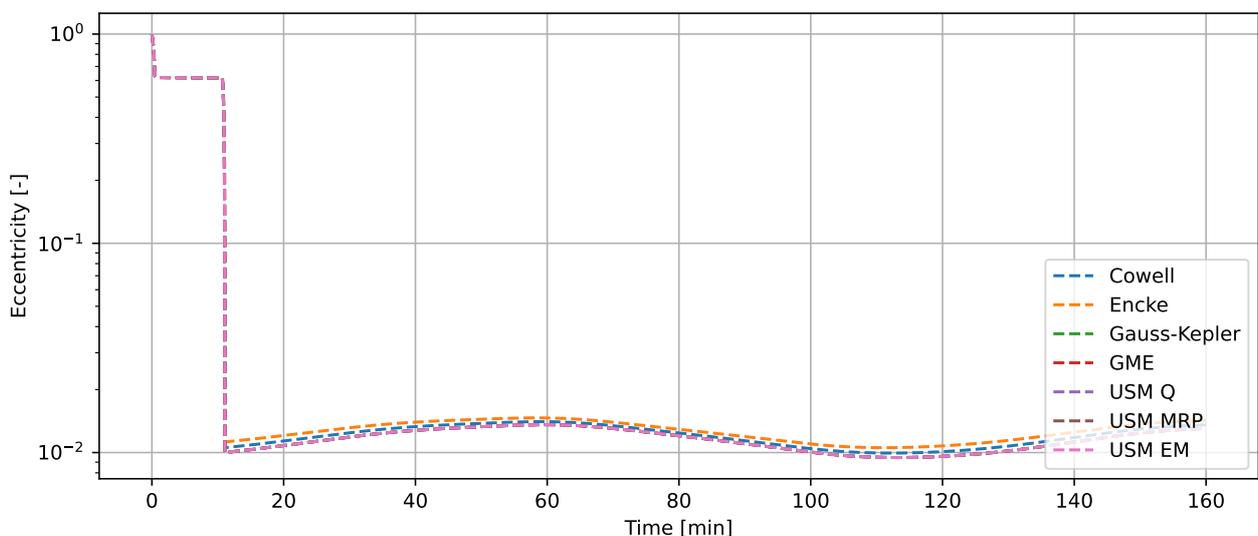
Figure 4.26: MAV orbital eccentricity over time for different propagators.
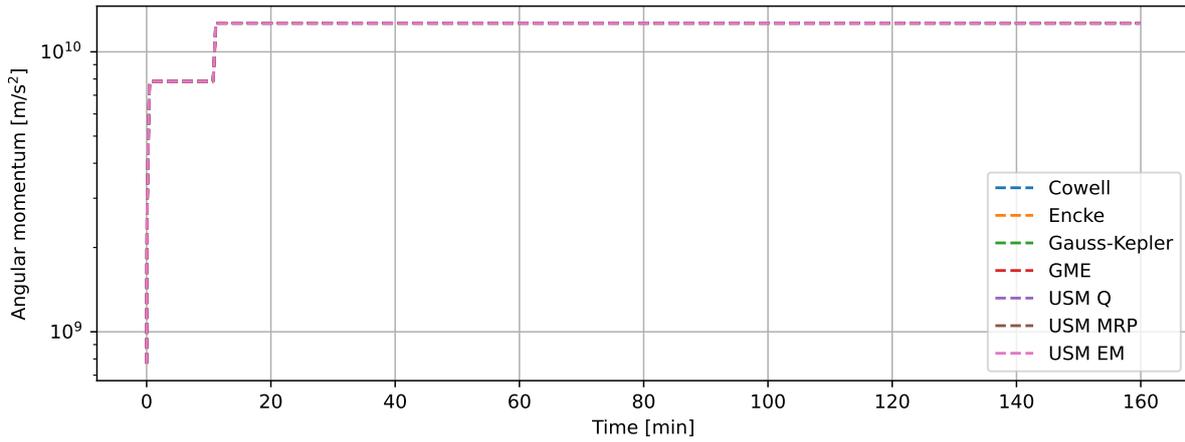
Figure 4.27: MAV momentum over time for different propagators.
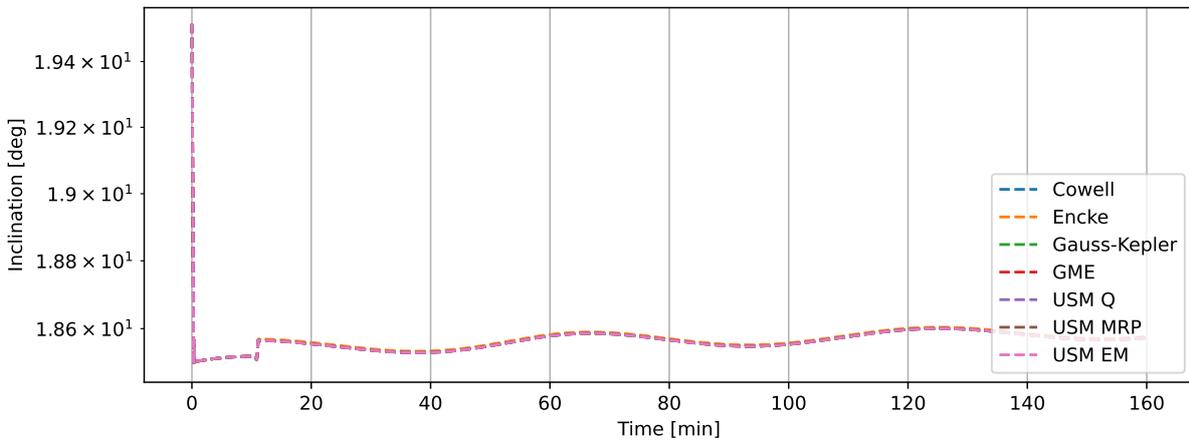
Figure 4.28: MAV orbital inclination over time for different propagators.

Baseline simulation

In tuning of the baseline simulation parameters, integrator, environment, and propagator, had to be guessed as a starting point. Since all of these have now been tuned, it is possible to plot the position and mass of the MAV over time, for both the initial guessed setup, and the tuned one. These are shown in Figure 4.29. From it, a clear difference can be seen in the altitude history, indicating that the initially guessed setup was introducing a significant error.
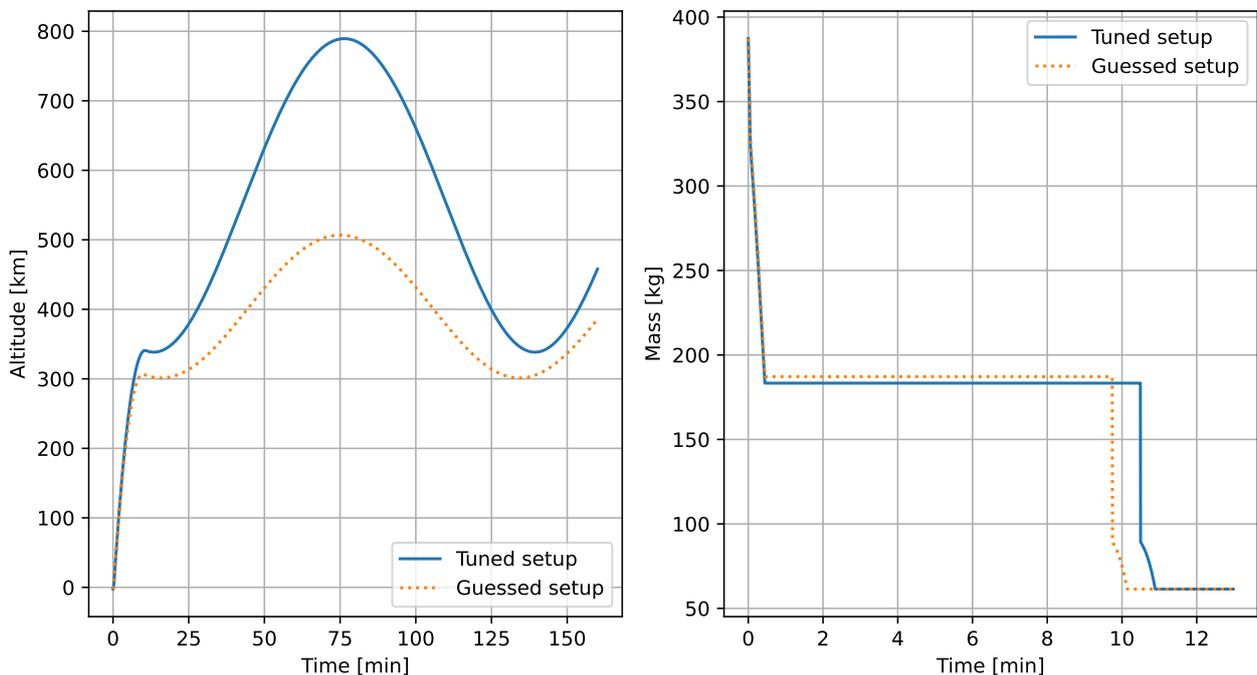
Figure 4.29: Altitude and mass of the MAV over time, for the benchmark and the guessed simulation setup.

4.3. Optimisation

Most of the results come from the optimisation that is run to find the configuration that allows the MAV to reach orbit with the lowest GLOM possible. In the optimisation context, the first results come from the design space exploration, studying the relation between the design variables and their effect on the objective space. Then, an example of how the initial optimisation population is generated is shown. Follows the results of the optimisation algorithm selection and tuning. Lastly, the optimisation results themselves are presented and discussed.

4.3.1. Design space exploration

The first part of the design space exploration was carried out to restrict the range in which the design variables that define the SRM geometry can vary. As such, the design variables of each of the SRM type have been varied one at a time, with the exception of the outer radius and length of the motors. This is because the motor length is not related to the other design variables, and its range will thrust not change. Similarly, the inner geometry of the motor is typically specified as a function of the outer radius, as discussed in Section 3.1.3. Varying the outer radius will therefore scale the entire motor dimensions, and the outer radius does not need to have its allowed range tuned.

   First, the inner radius of the tubular SRM is varied, as shown in Figure 4.30. This figure was generated by trial and error to find the best range of inner radius for the SRM. It was thus decided to set the inner radius fraction range to [0.2, 0.9], as to keep the smallest inner empty tube big enough to guide the hot gas out, and the biggest outer propellant shell thick enough to provide a significant thrust impulse.
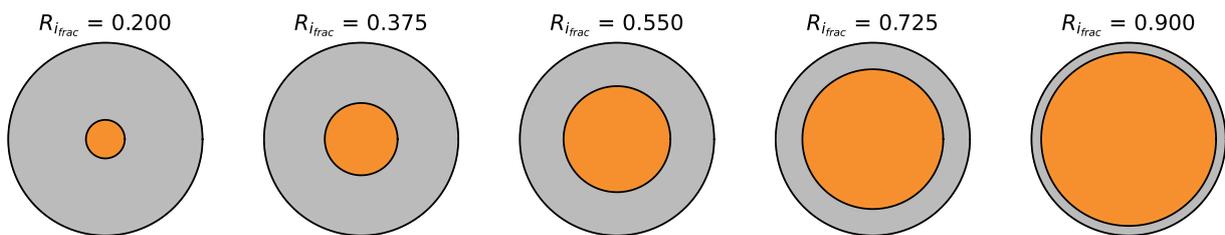
Figure 4.30: Cross-sectional geometry of tubular SRM from the design space exploration.

Then, the same process was carried out for the rod and tube SRM, as shown in Figure 4.31. The intermediate radius fraction and the inner radius fraction range are both set to [0.2, 0.9]. This is to allow for hot gases to escape through the propellant rod and tubes, and ensure a sufficient outer propellant tube thickness.
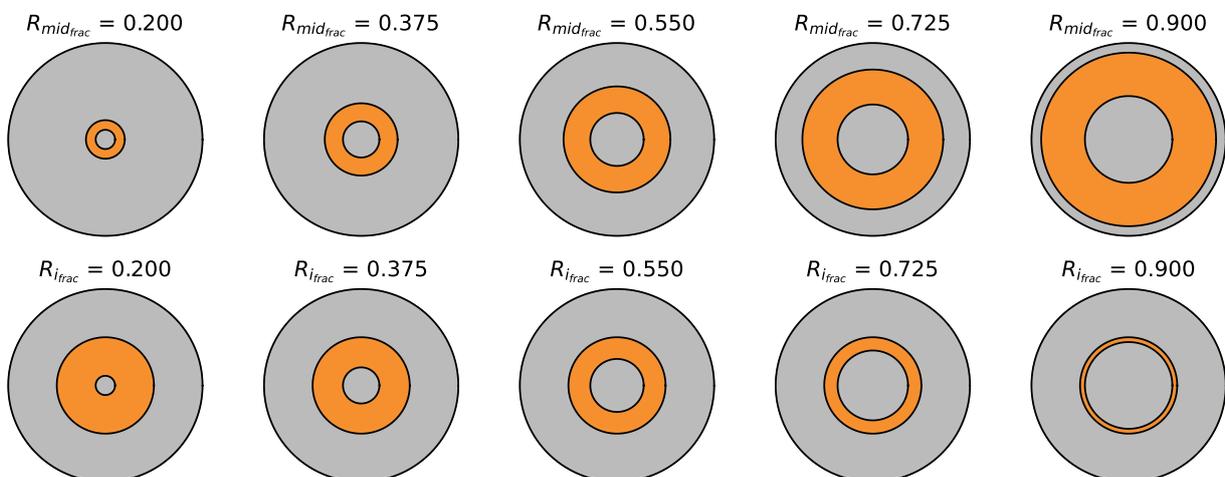
Figure 4.31: Cross-sectional geometry of rod and tube SRM from the design space exploration.

The multi-fin SRM geometry is more complex, containing more design variables. As shown in Figure 4.32, the inner radius fraction range is once again set to [0.2, 0.9]. Then, the number of fins is

allowed to vary between 3 and 15. This is to ensure a radial symmetry, as having only two thick fins could potentially make the vehicle unbalanced. The upper bound of 15 fins is set to ensure that there are not a high number of thus very thin and fragile fins. Then, the fin length fraction range is set to [0.25, 0.75], to avoid having fins so short that they would burn immediately, or so long and slender that they would break or interfere with each other. Lastly, the fin width fraction range is set to [0.35, 0.9], to prevent them from interfering with each other, or from being too thin and prevent breaking during ascent from Earth, during Mars atmospheric entry, or during Mars ascent.
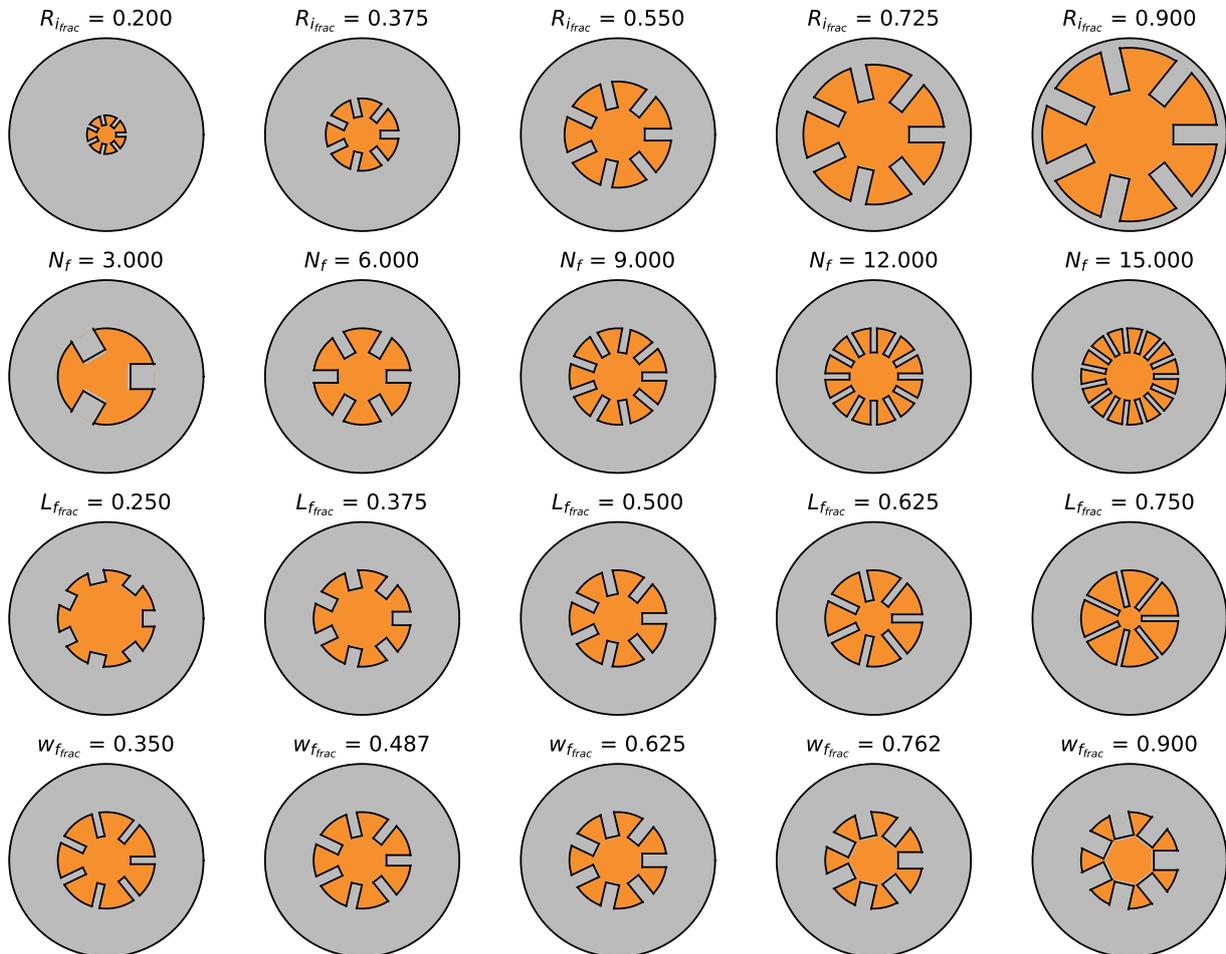
Figure 4.32: Cross-sectional geometry of multi-fin SRM from the design space exploration.

The spherical SRM geometry also contains only one design variable, the inner radius fraction. It was decided to allow this variable to vary between 0.15 and 0.8. Because of the spherical nature of the motor, it was decided to prevent the inner empty sphere and exhaust tube from being too small, thus condensing all of the hot gases into a confined space.
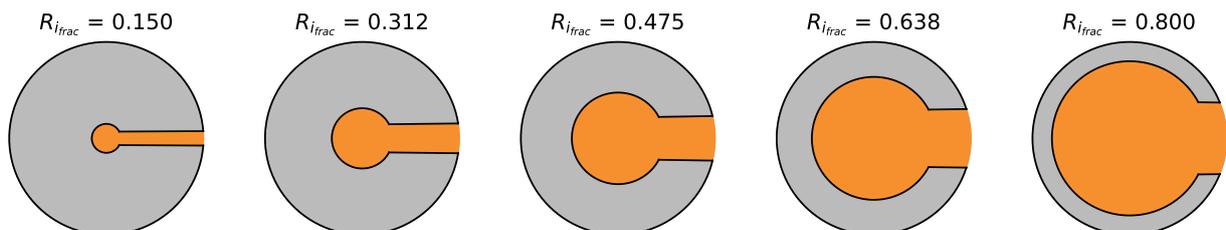
Figure 4.33: Cross-sectional geometry of spherical SRM from the design space exploration.

Lastly, the anchor geometry is the most complex. The range for each of the design variables of this geometry are represented in Figure 4.34. The inner radius fraction $R_{i_{frac}}$, spacing fraction w_{frac}, and spoke width fraction $\delta_{s_{frac}}$ range are all set to ensure that, at their minimum, the exhaust gas

have a large enough volume to flow through. In their maximum, the geometric features made by the propellant must be thick enough to ensure their solidity. Then, the fillet radius fraction $r_{f_{frac}}$ is allowed to vary between what essentially results in squared or rounded fins. Lastly, the minimum number of anchors N_a is set to 2, as given in the constraint of Equation 3.39. The maximum number of spokes is set to 6, which has been found to be the critical point after which the constraint of Equation 3.41 is hardly satisfied.
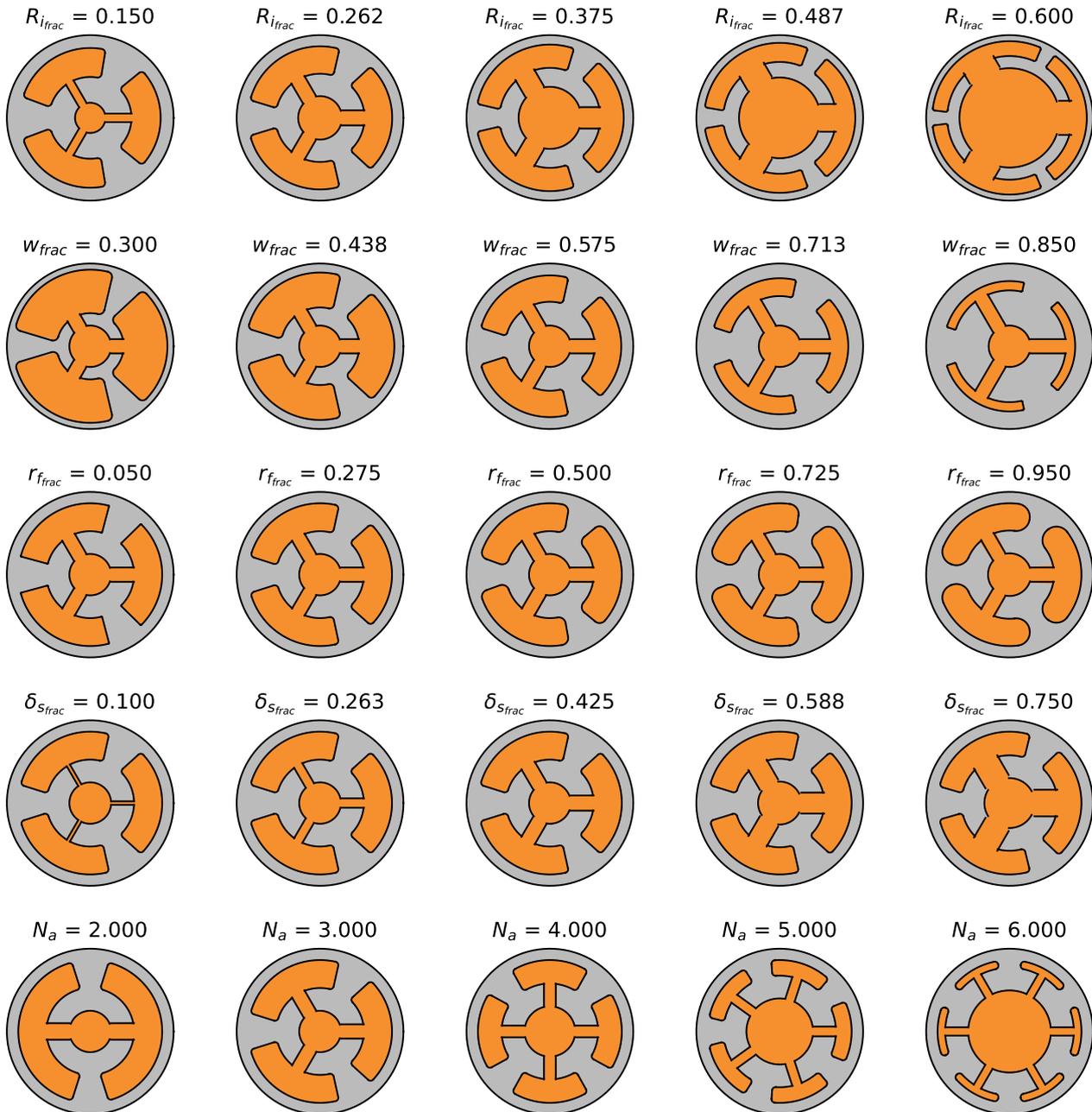
Figure 4.34: Cross-sectional geometry of anchor SRM from the design space exploration.

All of the design variables related to the geometry of the SRM of both stages have then be varied using a combination of the MC and Sobol sequence methods. The plot of Figure 4.35 shows that, amongst the 20 thousand simulations that were run from this process, only about 30 were successful in having their final altitude between 200 km and 500 km. This hints at how sensitive the simulation is to the performance of the SRM, as expected.
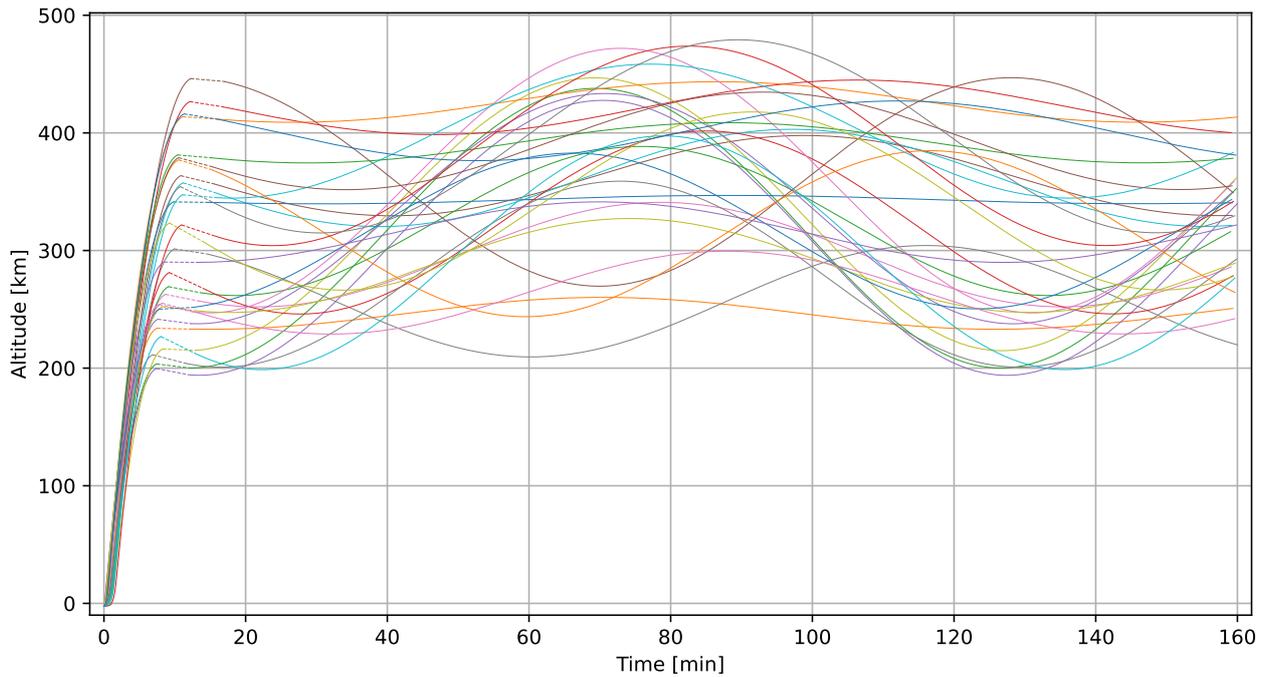
Figure 4.35: Flight path of the MAV from the design space exploration, varying the SRM geometries.

From these 20 thousand simulations run by varying the SRM geometries, the correlations between the design variables and the objective score in mass, apoapsis altitude, and periapsis altitude, have also been investigated. These correlations are shown in Figure 4.36, and show that the outer radius of both stage SRM, as well as the length of the first stage SRM, have the highest correlation with the score in mass. This makes sense since these variables directly dictate on the overall volume, and thus mass, of the SRM. The inner radius of the first stage SRM also appears to be correlated with the mass score. This is most likely due to this parameter, when increased, drastically decreasing the first stage SRM mass.

A correlation between the outer radius and final orbit altitude can also be seen in Figure 4.36. This can be explained as the first stage SRM is supposed to contain most of the thrust impulse, and the outer radius most dictates for how long the motor will burn.

Lastly, the mass score can be seen plotted as a function of the value of the different design variables in Figure 4.37. No clear relation can be extracted from this figure, and the allowed range for the SRM design variables has thus not been tuned further.
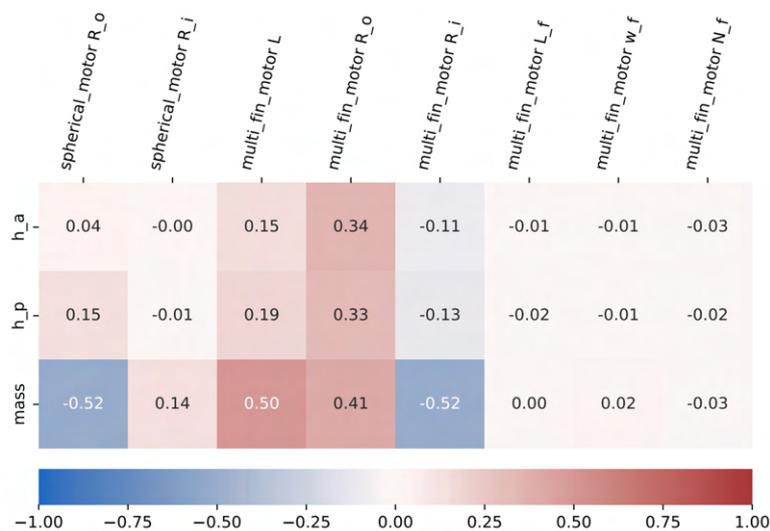
Figure 4.36: Heatmap of the correlation between the SRM geometry and the MAV mass and orbital altitudes.
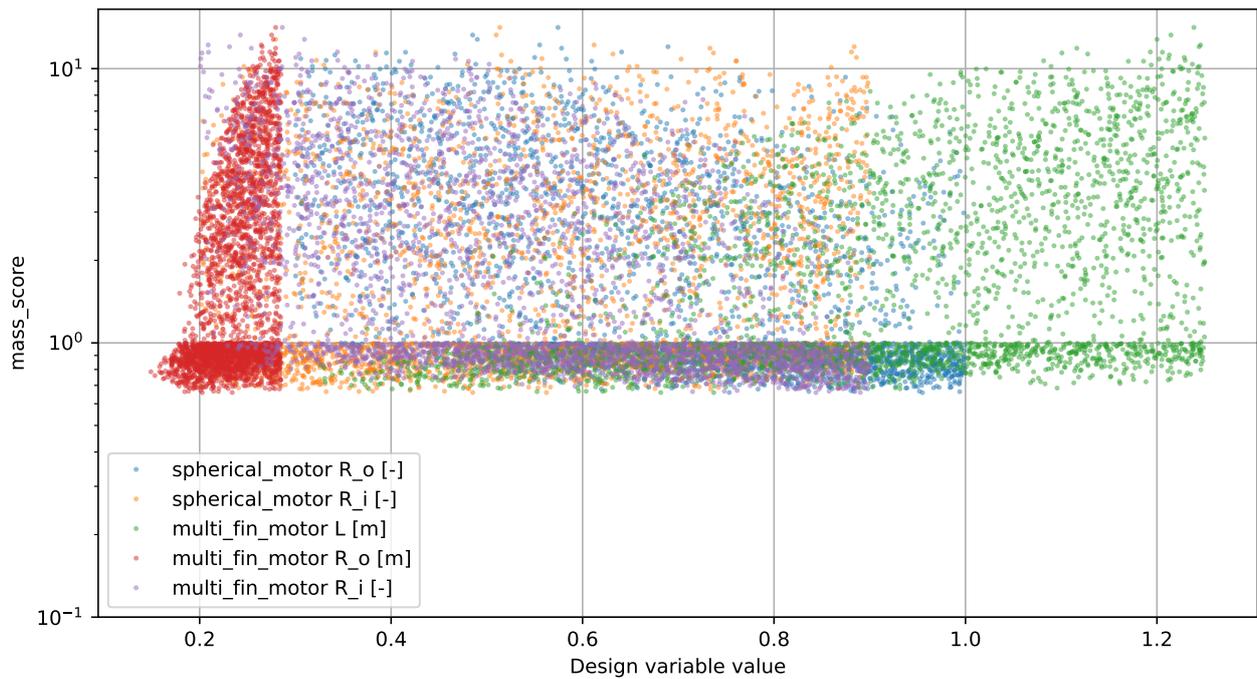
Figure 4.37: Plot of the MAV mass scores as a function of the SRM geometry.

With the design space exploration of the SRM geometry parametrisation complete, the design space of the variables related to the ascent itself can be explored. These design variables are the initial lift-off and stage separation angles, and the TVC angles.

First, all design variables have been varied at the same time, including the SRM design variables. Doing so, and taking 20 thousand random samples, reduces the number of ascents that results in altitudes in the 200 km to 500 km range. This is shown in Figure 4.38. This is because, varying all design variables at the same time, the imperfections in each of them add to deviations from the baseline. No more analysis has been made from this, due to the number of variations and inherent chaos in the generated 20 thousand simulation dataset. This serves to show that, simply varying everything, does not result in optimums.
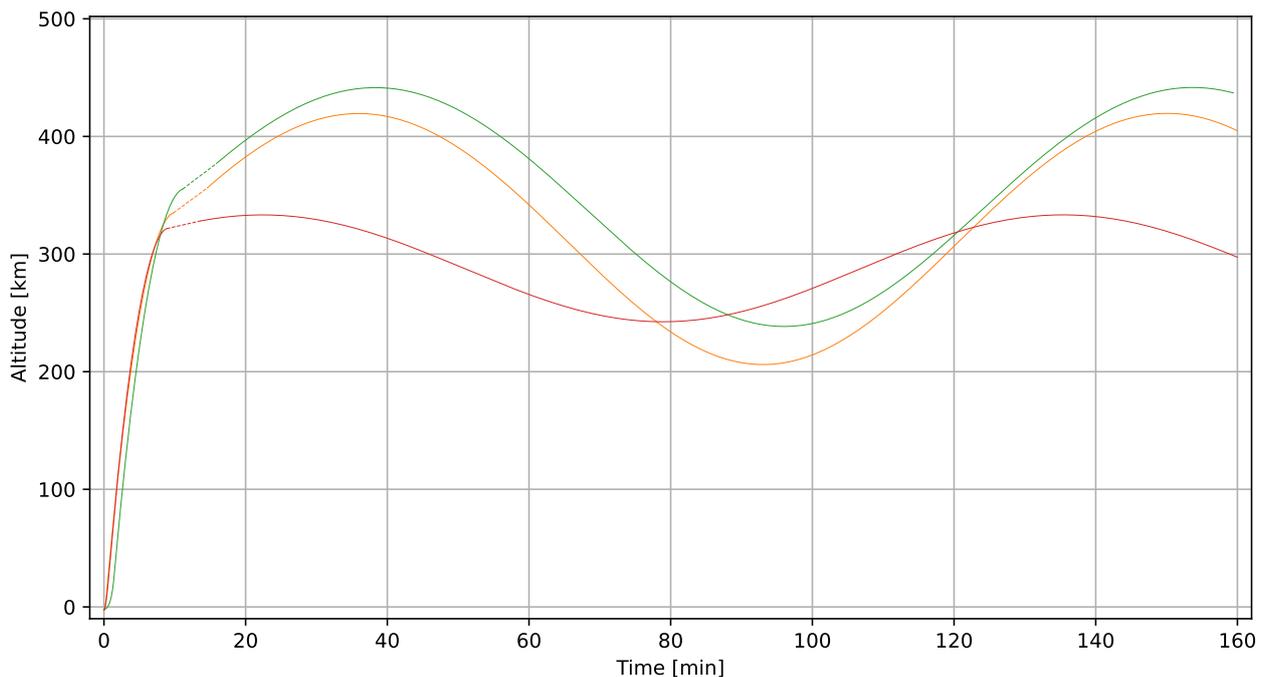
Figure 4.38: Flight path of the MAV from the design space exploration, varying all design variables.

The initial lift-off and stage separation angles design variables have then been varied together, keeping others at baseline values. From Figure 4.39, one can see that varying only these two variables

results in more ascents that end in the 200 km to 500 km range. This is because the thrust and TVC used are similar, hence variations in how the MAV is launched and stage separation occurs leaves less room to stray away from the baseline.
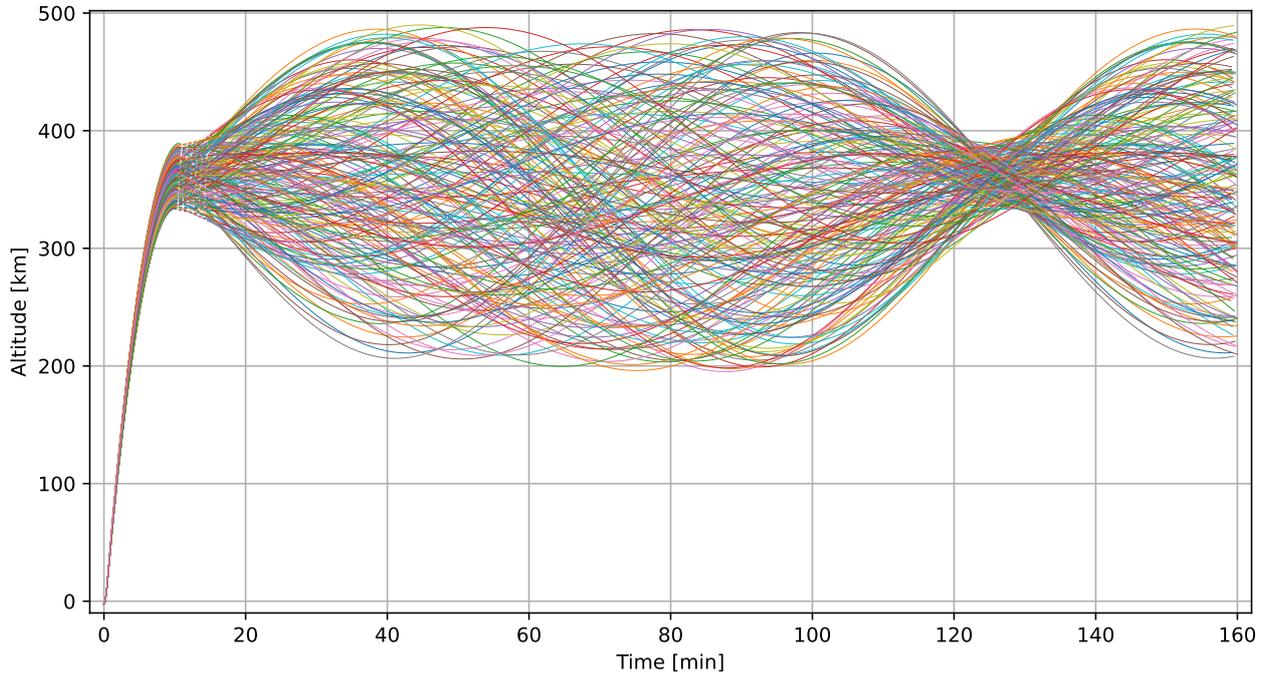
Figure 4.39: Flight path of the MAV from the design space exploration, varying the initial angles.

The correlation between the initial angles design variables and the periapsis and apoapsis objective score has then been analysed, as shown in the heatmap of Figure 4.40. In this figure, 'angle_1' is the launch angle and 'angle_2' is the stage separation angle. From it, one can clearly see that there is a strong correlation between the lift-off angle and the periapsis altitude. This makes sense, as launching the MAV from a more vertical position, and keeping the motors the same, will result in a higher periapsis altitude, and the other way around.
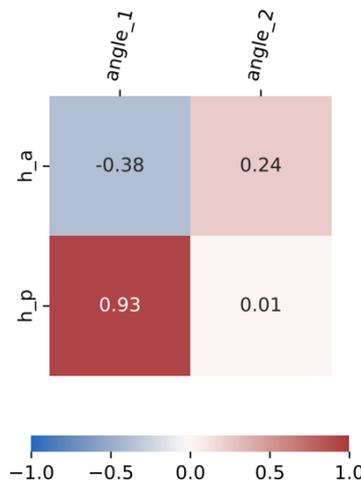
Figure 4.40: Heatmap of the correlation between the initial angles and the orbital altitudes.

Also related to the initial angles design variables, Figure 4.41 has been made, showing the altitude objective scores as a function of these design variables. Looking at the stage separation angle 'angle_2', a value of for instance 85 deg can result in good objective scores of 0.1, but also of bad scores of 100. However, taking a value of for instance 70 deg will never results in a score lower than 20. It is thus decided to remove any angle that results exclusively in scores above 20, as shown by the horizontal green line. This means that the lift-off angle should be kept between 53 deg and 60 deg, and the stage separation angle between 75 deg and 104 deg. Margins are added on top of that,

hence it is decided to use the following ranges: [47.5 deg, 60 deg] for the launch angle and [70 deg, 110 deg] for the stage separation angle.
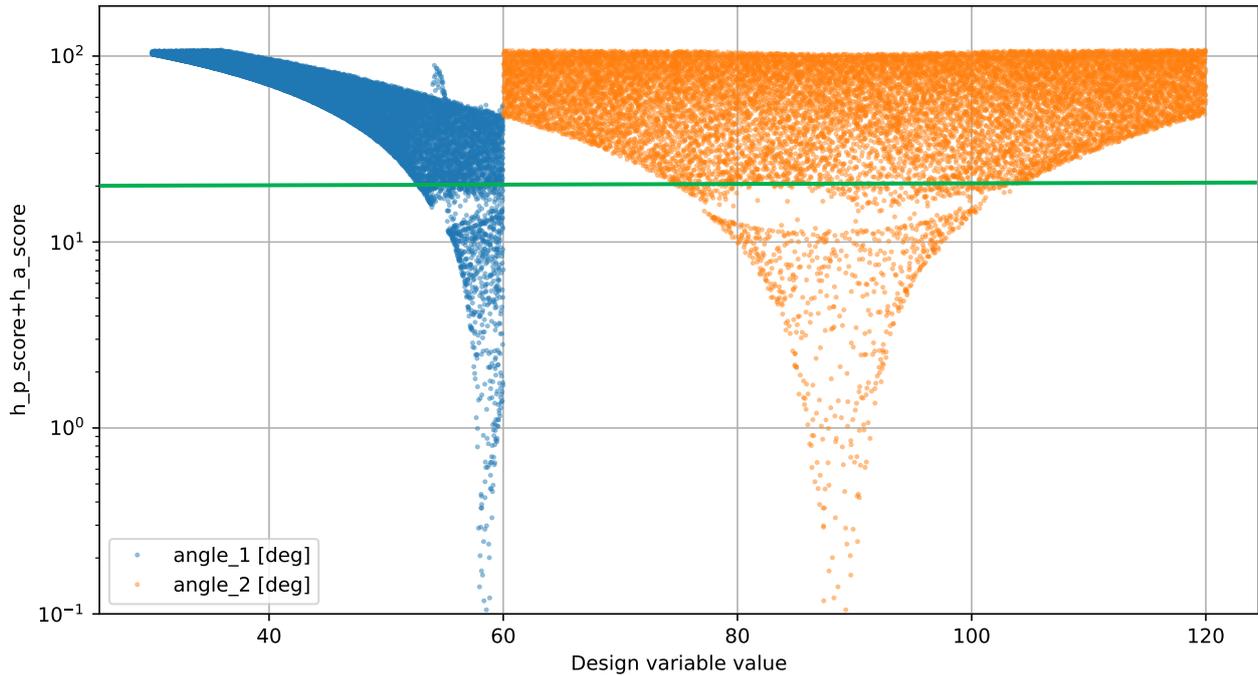
Figure 4.41: Plot of the orbital altitudes scores as a function of the initial angles.

Last but not least, the design space of the TVC angles design variables has been explored. To this effect, Figure 4.42 has been made. From the 20 thousand samples, 9700 of them resulted in altitudes within the 200 km to 500 km range. This shows that TVC is mostly used to fine-tune the ascent, since results stay close to the baseline.
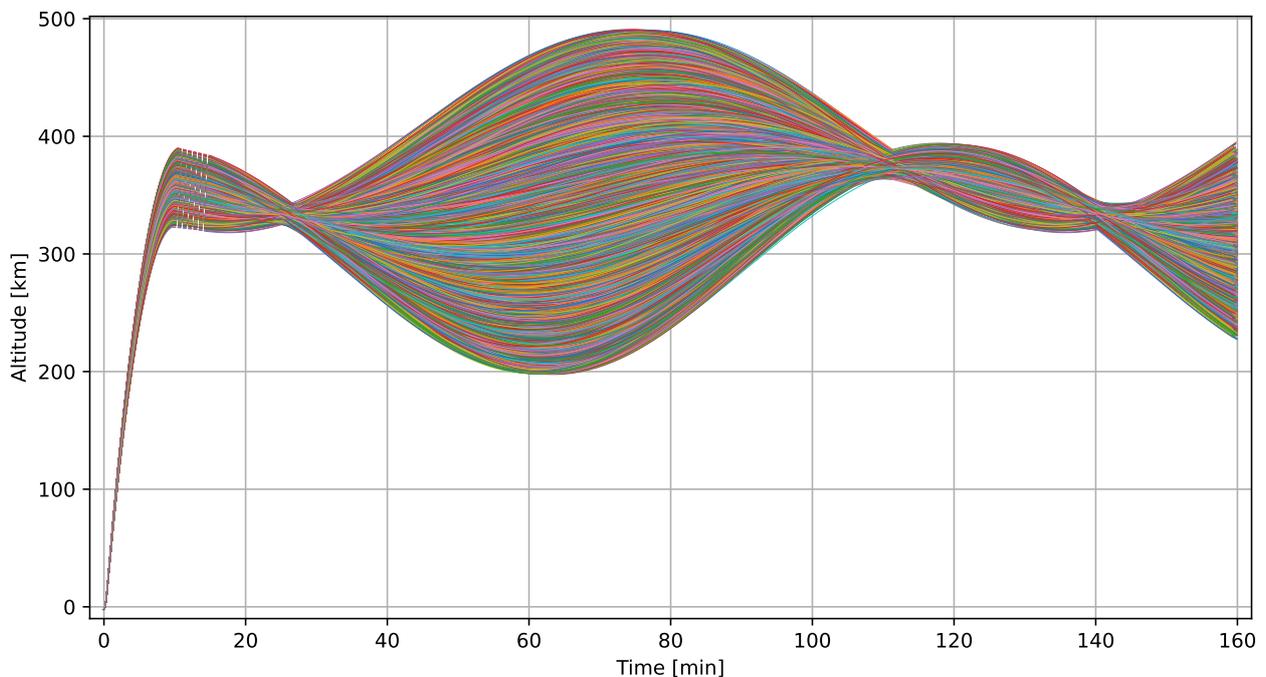
Figure 4.42: Flight path of the MAV from the design space exploration, varying the TVC angles.

The correlation between the TVC angles design variables and the periapsis and apoapsis objective score, as well as the final orbit inclination, is presented in Figure 4.43. From this figure, the use of TVC in the y-direction is shown to mostly affect the inclination of the final orbit, while TVC in the z-direction mostly affects the final orbital altitudes. Since the inclination is not taken as an objective, it is decided

to remove the design variables of the TVC in the y-direction. This leaves less variable to work with for the optimiser later on, which is assumed results in a more efficient optimisation.
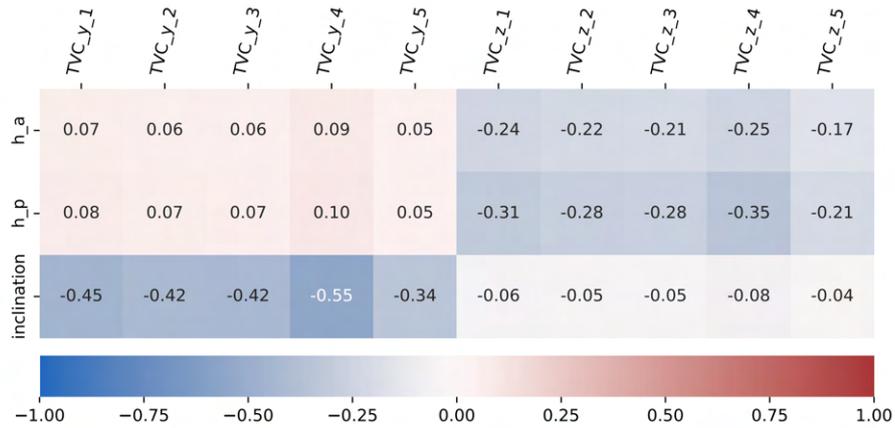
Figure 4.43: Heatmap of the correlation between the TVC angles and the orbital altitudes.

Finally, Figure 4.44 shows the altitude objective scores as a function of the TVC angles in the z-direction. From the chaos of the plot, one can see that no relation can be deducted. This result is used to keep this design variable range unchanged.
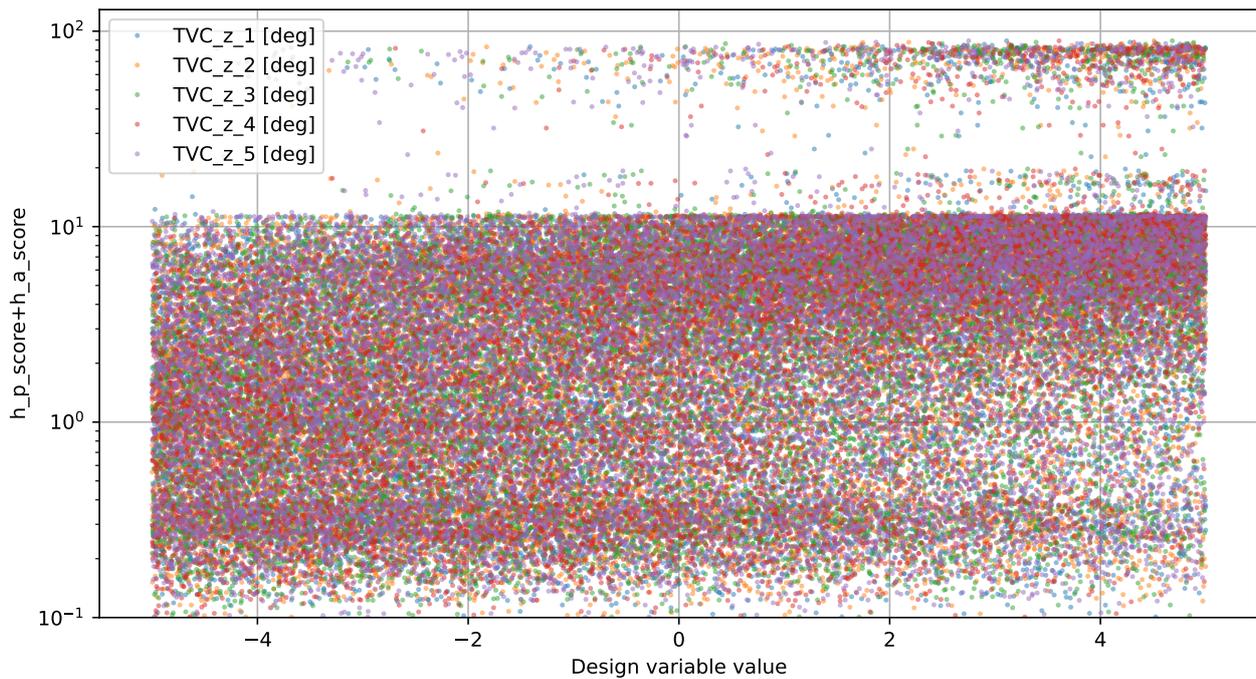
Figure 4.44: Plot of the orbital altitudes scores as a function of the TVC angles.

In conclusion, the TVC design variables in the y-direction have been removed, and the remaining design variables can vary in the range summarised in Table 4.15. As a reminder, there are 5 TVC angles in the z-direction that are equispaced in time during the first stage burn, the spherical SRM geometry is for the second stage, and one optimisation is run per SRM type for the first stage.

Parameter	Minimum value	Maximum value
Launch angle [deg]	47.5	60
Stage separation angle [deg]	70	110
TVC angle in the z-direction [deg] (x5)	-5	5
First stage SRM outer radius R_o [m]	0.1	0.285
First stage SRM length L [m]	0.3	1.25
Spherical SRM outer radius fraction $R_{o,2,\text{frac}}$ [-]	0.3	1.0
Spherical SRM inner radius fraction $R_{i,2,\text{frac}}$ [-]	0.2	0.9
Tubular SRM inner radius fraction $R_{i,1,\text{frac}}$ [-]	0.2	0.9
Rod and tube SRM intermediate radius fraction $R_{mid,1,\text{frac}}$ [-]	0.2	0.9
Rod and tube SRM inner radius fraction $R_{i,1,\text{frac}}$ [-]	0.2	0.9
Multi-fin SRM inner radius fraction $R_{i,1,\text{frac}}$ [-]	0.2	0.9
Multi-fin SRM number of fins N_f [-]	3	15
Multi-fin SRM length fraction $L_{f,\text{frac}}$ [-]	0.25	0.75
Multi-fin SRM width fraction $w_{f,\text{frac}}$ [-]	0.35	0.9
Anchor SRM inner radius fraction $R_{i,1,\text{frac}}$ [-]	0.15	0.6
Anchor SRM number of anchors N_a [-]	2	6
Anchor SRM spacing fraction w_{frac} [-]	0.3	0.85
Anchor SRM fillet radius fraction $r_{f,\text{frac}}$ [-]	0.05	0.95
Anchor SRM spoke width fraction $\delta_{s,\text{frac}}$ [-]	0.1	0.75

Table 4.15: Tuned range for the design variables.

4.3.2. Initial population

The initial population is taken from the results of the design space exploration. Following the fraction of design variables to take from each part of the design space exploration that was given in Section 3.3.4, an initial population of for instance 72 individuals looks as in Figure 4.45. This appears to cover the trajectory space well, which is helped by the fact that the population is taken from the different design space explorations run.
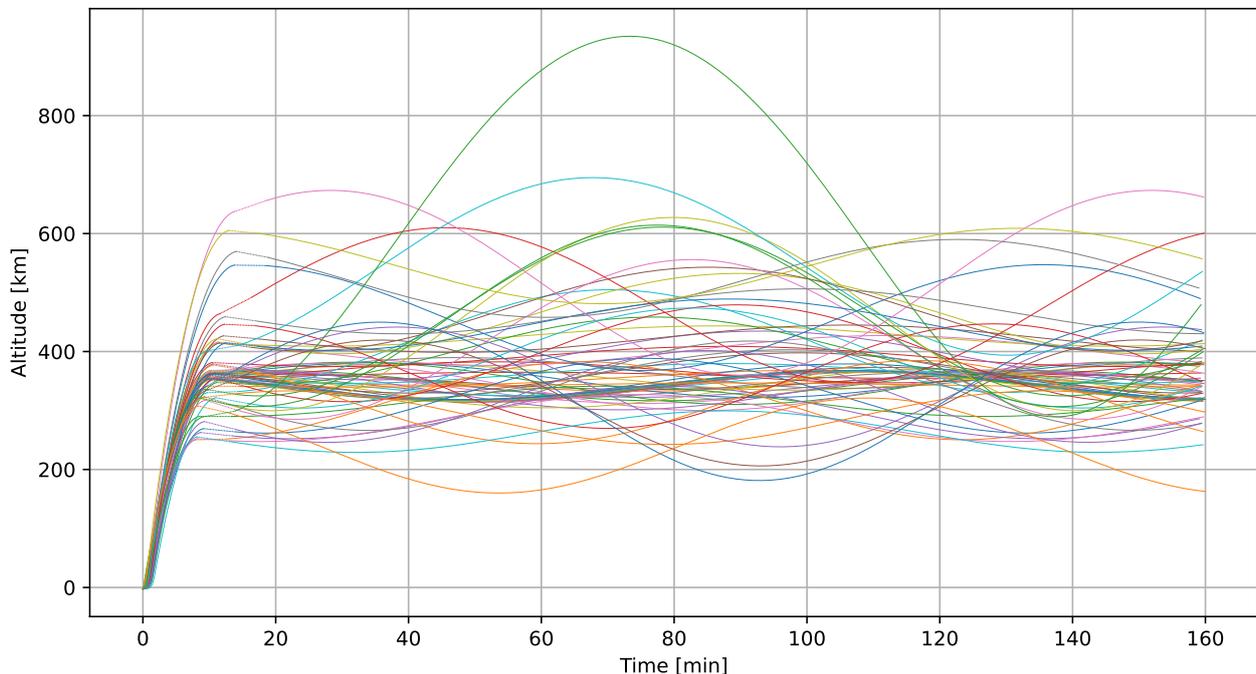
Figure 4.45: Initial optimisation population composed of 72 individuals taken from the design space exploration.

4.3.3. Optimiser selection

With the design space explored, and the initial population setup, an optimisation algorithm can be selected. First, the NSGA2 optimiser was tried. An optimisation has then been run with this algorithm, with a population of 72 individuals, and for 30 generations. The value that the optimiser used for the design variables of each individual, at each generation, have been saved. Plotting for instance the initial angles of the entire population as a function of the generation number results in Figure 4.46. From it, one can see that there is very little variation in the values taken by the optimiser for the initial

angle design variables. The same can be said when generating similar plots for the other design variables. It is thus decided to not use this optimiser, as it does not leave sufficient variation in the population.
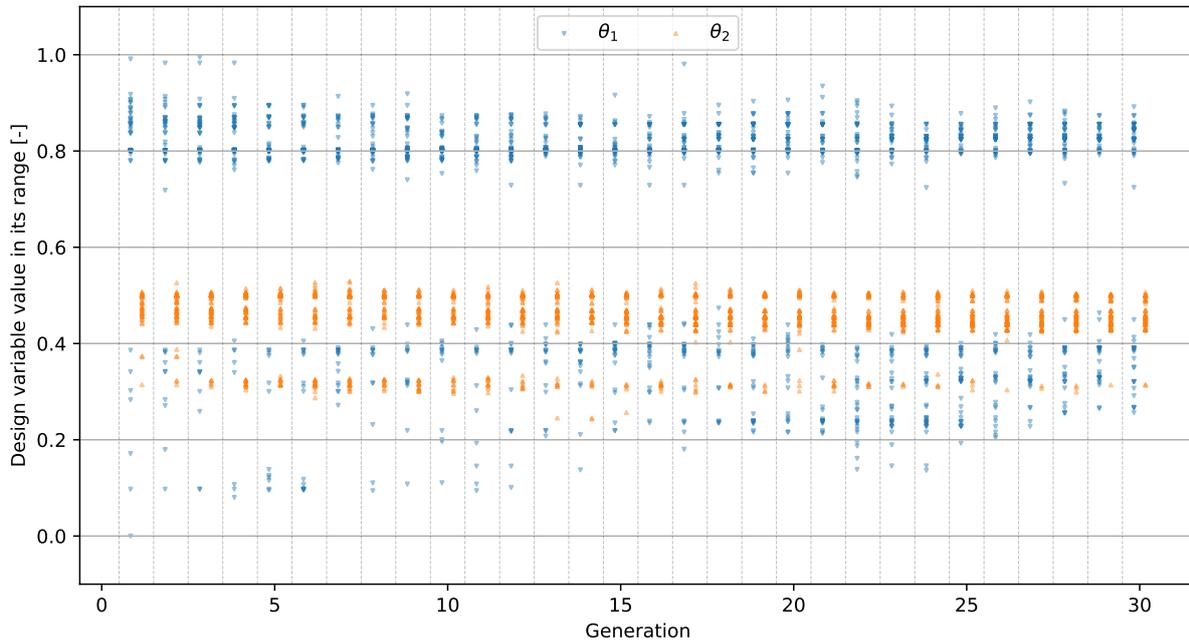
Figure 4.46: Evolution of the initial angles design variables during an optimisation with the NSGA2 optimiser.

Then, the same optimisation has been run, but changing the optimiser to the MHACO algorithm. Generating the same plot as before in Figure 4.47 shows a higher spread in design variables values. The convergence of the optimisation objectives is then investigated. This is done by plotting the scores of each population individual as a function of the generation number, as in Figure 4.48. In this plot, bigger dots represent the minimum objective score of a generation across the entire population. From this figure, it can be observed that, while the apoapsis altitude objective stays low, the behaviour of the two other objective is erratic. While the spread in design variables is satisfactory with the MHACO optimiser, the convergence of the objectives is not.
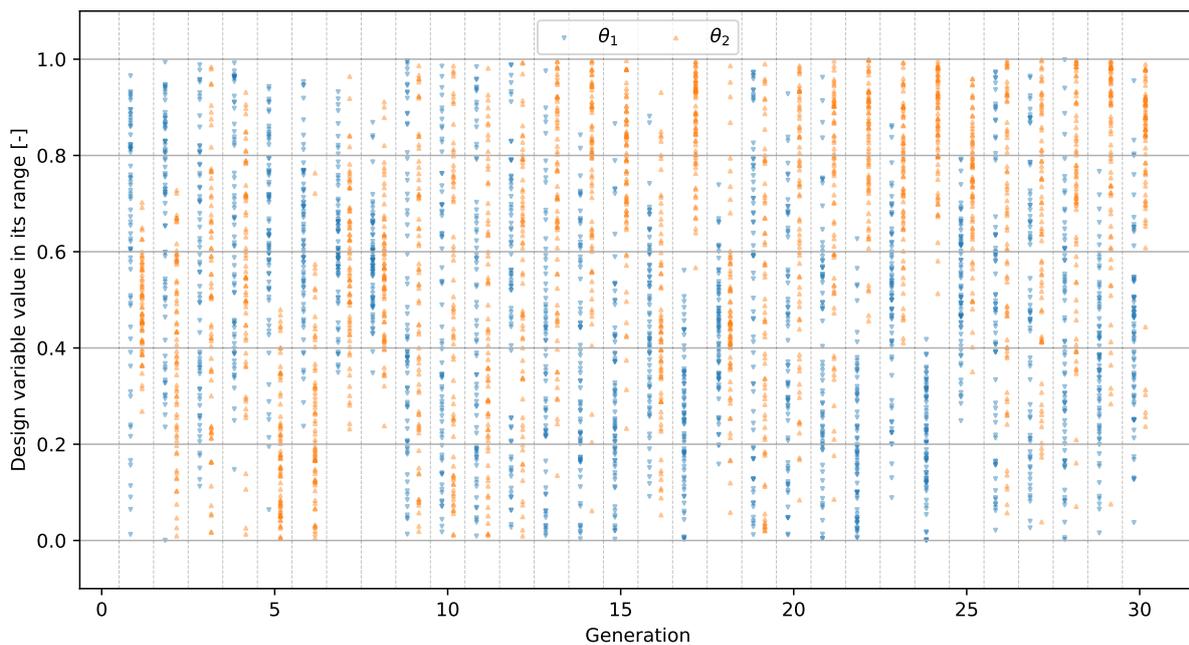
Figure 4.47: Evolution of the initial angles design variables during an optimisation with the MHACO optimiser.
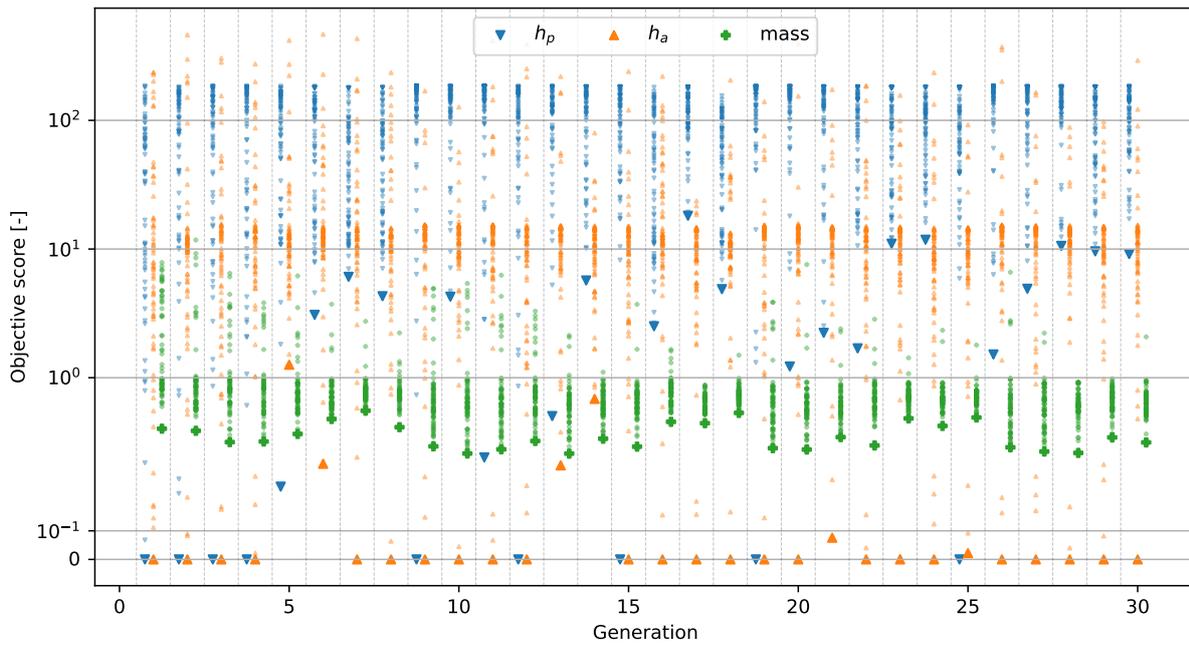
Figure 4.48: Evolution of the different objectives during an optimisation with the MHACO optimiser.

The last optimiser that can handle both multiple objectives and batch fitness evaluation is the NSPSO algorithm. Figures 4.49 and 4.50 show the spread in design variables and objective score convergence for this algorithm. From these plots, it is judged that both of these aspects are satisfactory, especially when compared to results from the NSGA2 and MHACO optimisers. It is thus decided to move forward and use the NSPSO optimiser for the optimisation.
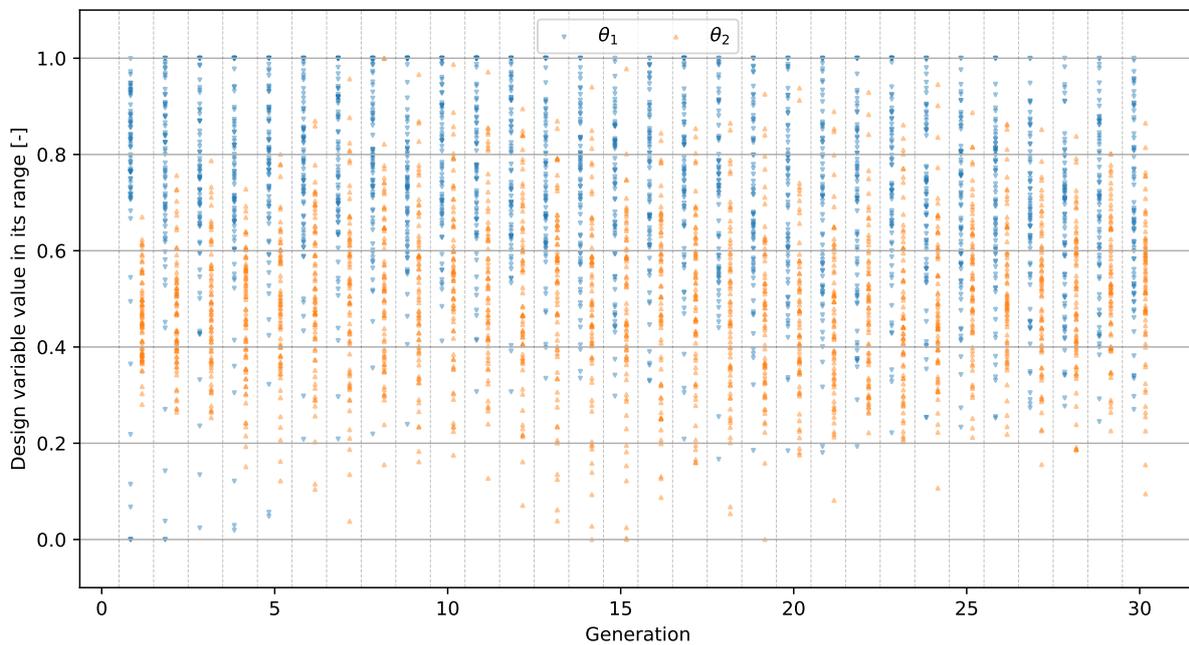
Figure 4.49: Evolution of the initial angles design variables during an optimisation with the NSPSO optimiser.
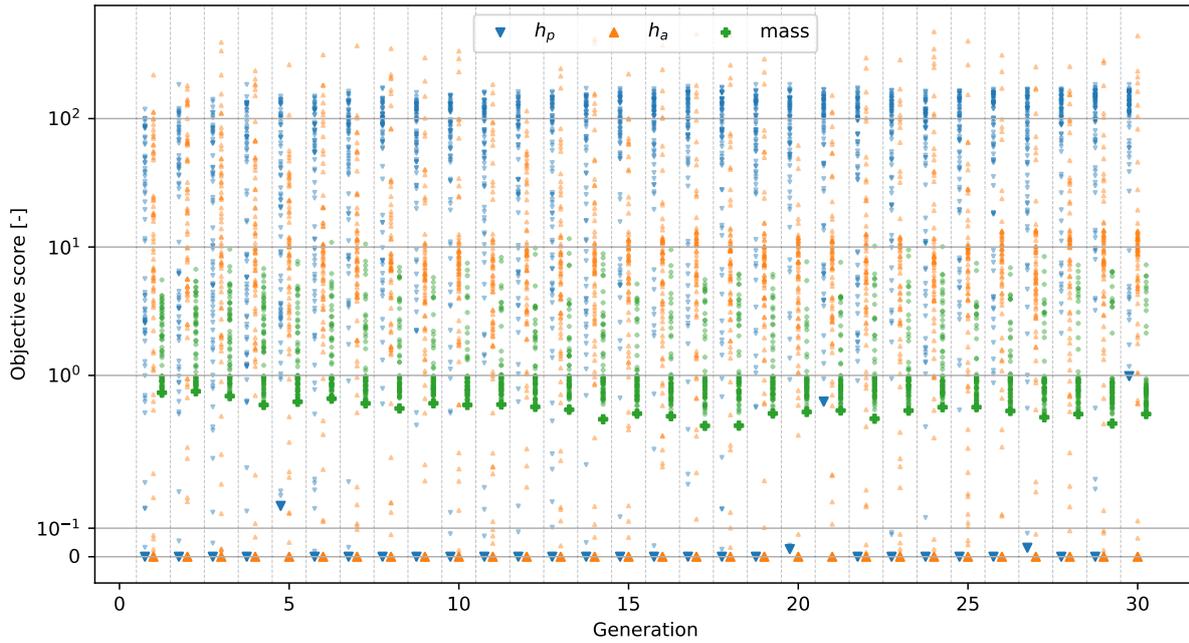
Figure 4.50: Evolution of the different objectives during an optimisation with the NSPSO optimiser.

In total, to select the optimiser, one optimisation has been run for each of the 3 optimisers, with a population of 72 individuals, for 30 generations. This means that, for the optimiser selection, 6480 SRM burn and ascent simulations have been run.

4.3.4. Optimiser tuning

Now that the NSPSO optimiser has been selected, its 7 different parameters can be tuned. Each of its parameter has been varied manually by taking between 3 and 5 values, and running a full optimisation each time, with 30 generations and a population of 72. In total, 31 optimisations have been run in this tuning process, requiring 66 960 SRM burn and ascent simulations to be carried.

For each optimisation, three distinct plots are made to investigate the minimum and mean objective scores, the standard deviation of the design variables, and the resulting Pareto fronts. This also means that, in total, 93 plots have been generated and manually analysed. Since this takes too much space for this report, an example of each plot can be found in Figures 4.51 to 4.53.
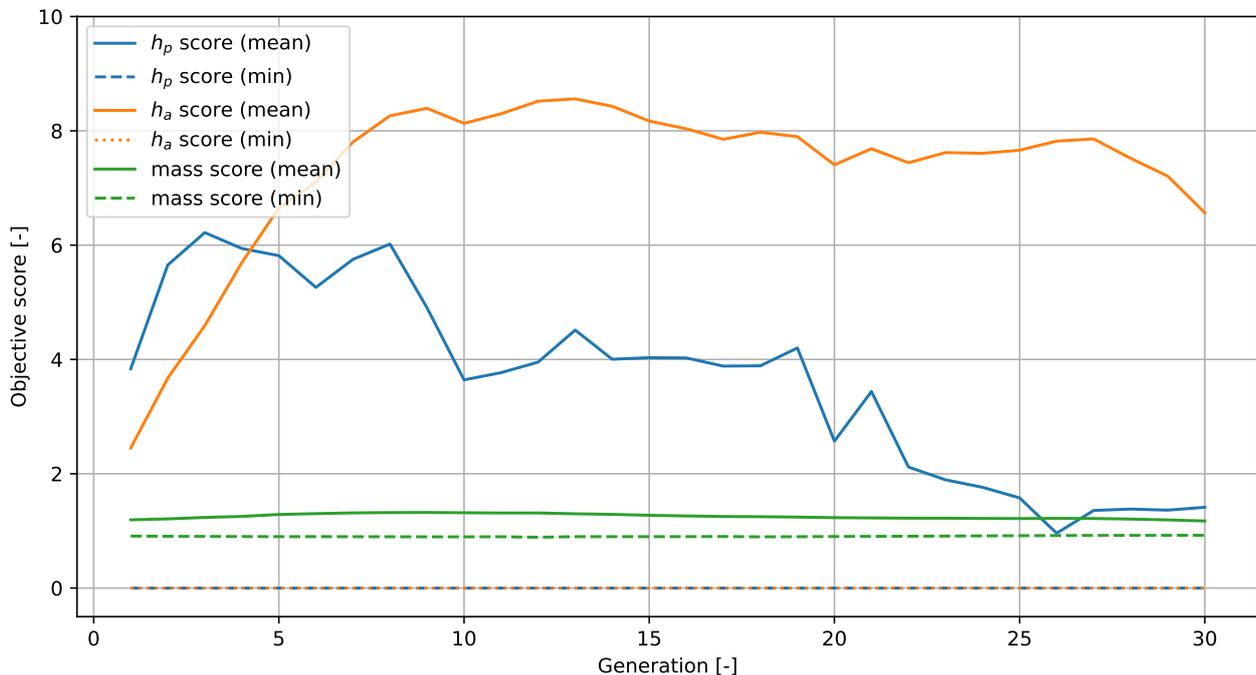
Figure 4.51: Example of minimum and mean objective scores for given optimiser parameters.
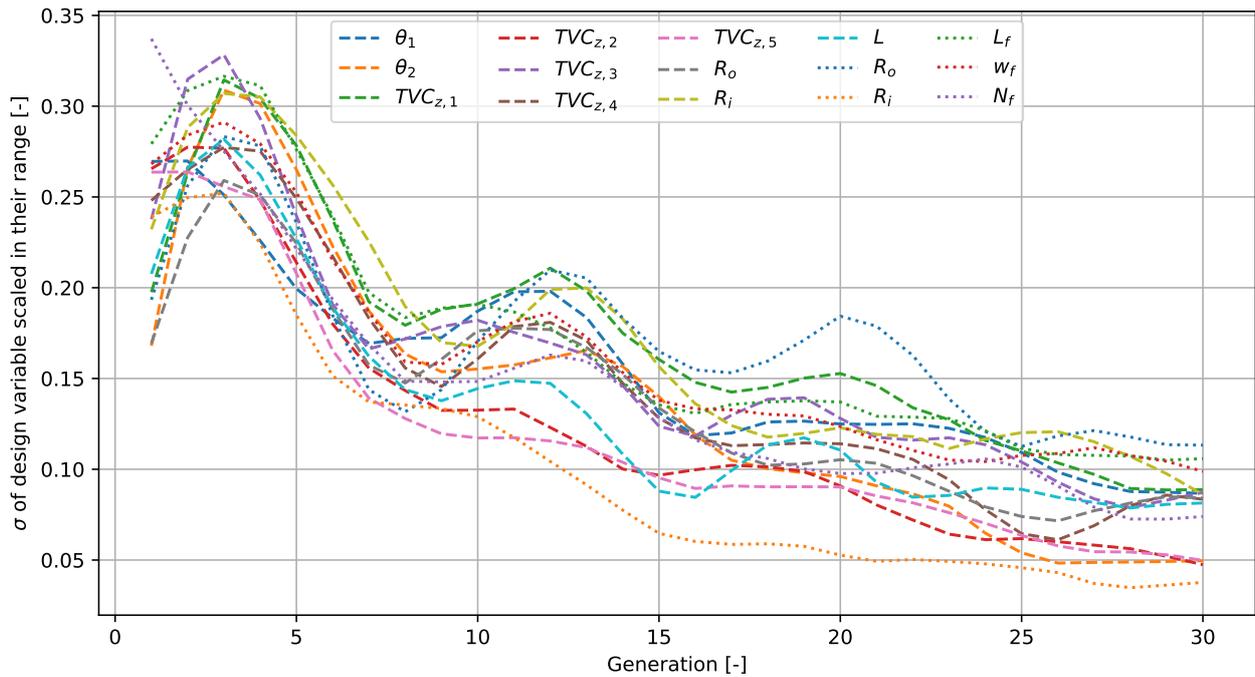
Figure 4.52: Example of plot showing the standard deviation of the design variables for given optimiser parameters.
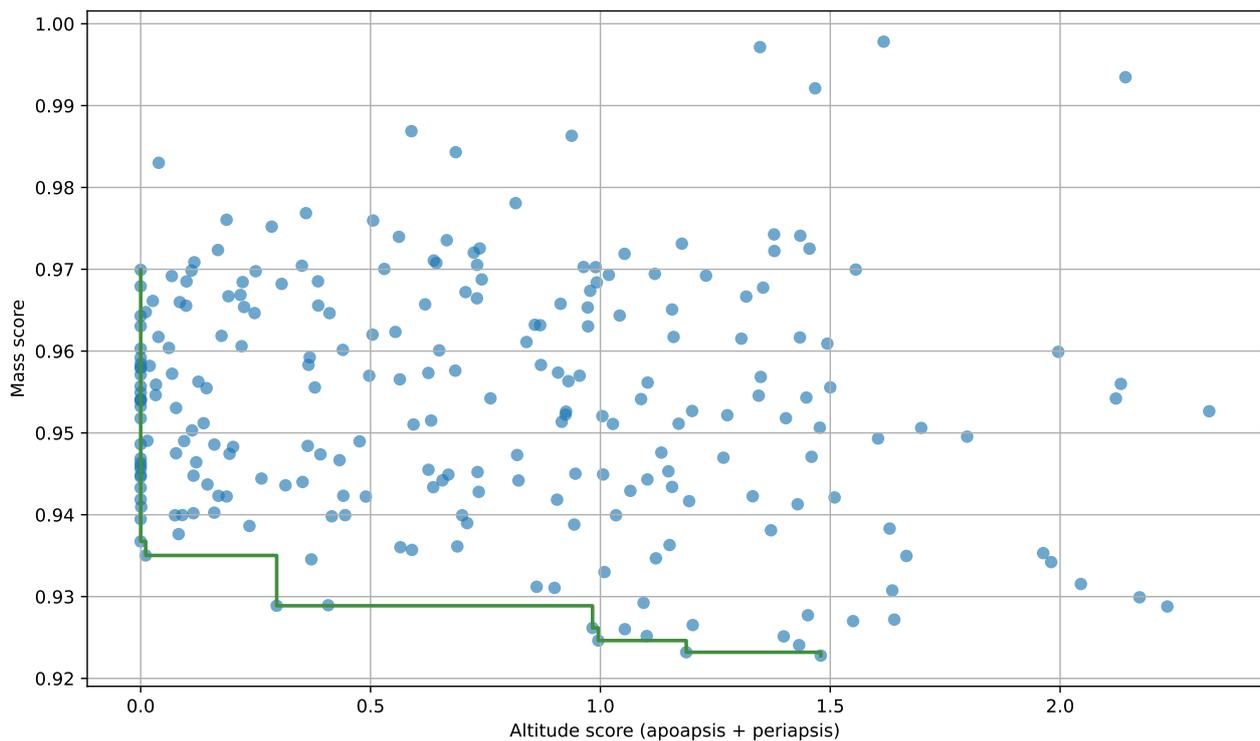
Figure 4.53: Example of Pareto plot for given optimiser parameters.

Figure 4.51 is used to analyse the convergence of the optimisation towards optimums. Then, Figure 4.52 is made by scaling the design variables by their allowed range, and computing the standard deviation in the population at each generation. From this plot, the convergence of the design variables can be analysed. Having the design variable converging is wished, to attain an optimum, but this convergence should not be too rapid, at the expense of missing part of the design space. Lastly, Figure 4.53 shows the objective scores of the entire optimisation population across all generations. This completes Figure 4.51 in the analysis of how well a given optimisation converges to a front of optimums, and allows to analyse the spread of the population in the objective space.

The parameters have thus been tuned one at a time. Starting with the default values specified in Pagmo, a given parameter is tuned, then its best value selected. It is then fixed to this tuned value,

and the next parameter is in turn tuned. This process is repeated for all the parameters. These parameters have been tuned in the order that was estimated to be of decreasing influence on the optimisation, and trying to keep the ones that appear related tuned after another. The parameters, their meaning, default values, allowed range, observed effect on the optimisation, and their tuned values, are all compiled in Table 4.16.

Parameter	Explanation	Allowed range	Default value	Observed effect	Tuned value
Diversity mechanism	How Pareto front diversity is ensured.	[Crowding distance, Niche count, Max min]	Crowding distance	Niche count manages to keep the h_p score below 10 the longest. Crowding distance is the worst at keeping design variables variation.	Niche count
Leader selection range	How far from optimums are leaders selected from the front.	<100	2	Value of 5 is best at keeping DVs varied, and making objectives still converge.	5
omega	Inertial particle weight.	]0, 1[	0.6	The higher the value, the slower the design variables convergence. A value of 0.8 allows for rough objective convergence, to be mitigated with other parameters.	0.8
v_coeff	Maximum velocity of a particle.	]0, 1]	0.5	The lowest the value, the less chaotic the particles movement. Low value of 0.05 allows for design variables and objectives to converge slowly, making more possibilities be explored, and the Pareto front more unified.	0.05
chi	How much the velocity is scaled by the attraction to other particles.	>0	0.5	Lower values clearly make variations in design variables decrease more linearly and slower. Best convergence compromise between 0.1 and 0.5.	0.25
c1	Magnitude of the attraction of a particle to its previous best.	>0	0.01	This parameter does not affect convergence significantly.	0.01
c2	Magnitude of the attraction of a particle to one of the leaders.	>0	0.5	The design space is not significantly affected by this parameter. A lower value leaves more varied particles in the objective space.	0.25

Table 4.16: NSPSO parameters tuning.

4.3.5. Optimums

With the optimiser selected and tuned, the actual optimisations have been run, one for each first stage SRM type: tubular, rod and tube, multi-fin, and anchor. For these optimisations, the number of particles has been increased from 72 to 136, and the number of generations has been increased from 30 to 100. Also, the optimisation has been run with 5 different seeds for the random number generators: 42, 13, 123, 846, and 579. This means that, in total, 272 thousand ascent and propellant burn simulations have been run for the different optimisations.

An example of the objective score behaviour across generations can be seen in Figure 4.54. This shows that some jumps are present in the minimum periapsis altitudes scores, while the minimum mass score decreases very slowly. This is due to the design variables converging to a smaller set of values after some time, leading to potentially worse scores when trying different design variables.

The variation in design variables is shown over the generation number in Figure 4.55. As discussed above, there is a clear converge of the design variables towards a given set of values, since the standard deviation between them gets below 3.5% after about 65 generations.
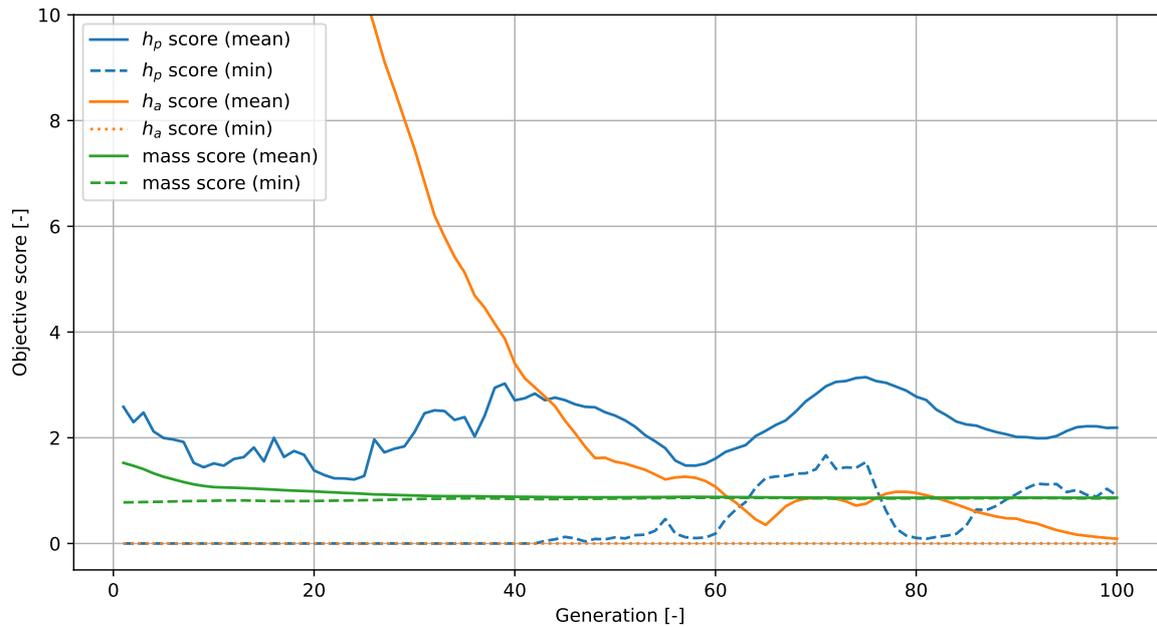
Figure 4.54: Minimum and mean objective scores for the MAV optimisation, with a seed of 123, using an anchor SRM for the first stage.
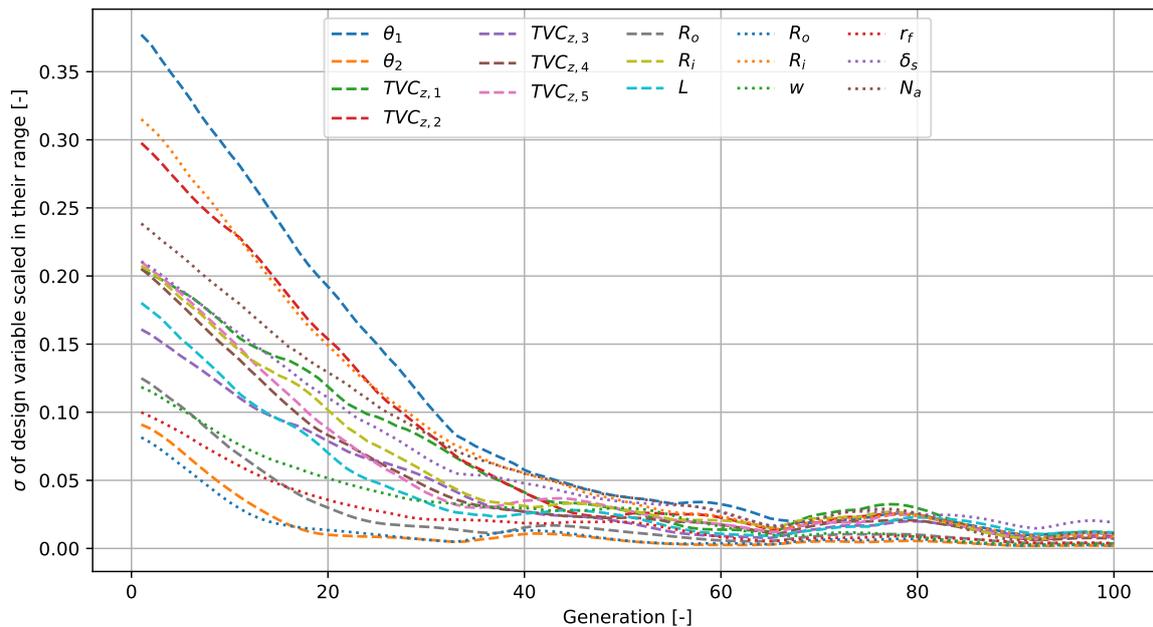
Figure 4.55: Standard deviation of the design variables for the MAV optimisation, with a seed of 123, using an anchor SRM for the first stage.

The best results from each design space exploration, optimisation, and refinement, are then compared. These are the simulations that results in the correct orbital altitude range, for the least mass, for a given first stage SRM type. These results are compiled in Table 4.17. From it, it can be seen than the MAV that uses an anchor cross-sectional geometry for its first stage SRM results in the lightest MAV, at a GLOM of 355.5 kg, 44.5 kg lower than the maximum allowed mass of 40 kg. The other SRM types for the first stage results in the GLOM of the MAV to be between 370 kg and 375.6 kg, still 7.5 % lower than the maximum allowed mass.

First stage SRM type	Anchor	Tubular	Rod and tube	Multi-fin
Launch mass [kg]	355.48	373.24	375.56	370.19
Second stage mass [kg]	101.79	103.86	100.28	94.232
Apoapsis altitude [km]	350.78	323.11	331.29	347.67
Periapsis altitude [km]	333.13	315.68	316.26	315.58

Table 4.17: Comparison of the optimisation results from the different first stage SRM types.

Then, the objective scores of altitude and mass are compared in Figure 4.56, combining the results of the optimisations from all 5 seeds for the optimisation that uses the anchor SRM for the first stage, resulting in the lowest GLOM. This figure shows that various simulations result in the target altitude being reached, represented by an altitude score of 0. At the same time, since the mass score is below 1, the target mass is also reached. Even better, the target altitudes are reached for mass scores of about 0.883, corresponding to a MAV initial mass of 372 kg, 28 kg below the requirement.
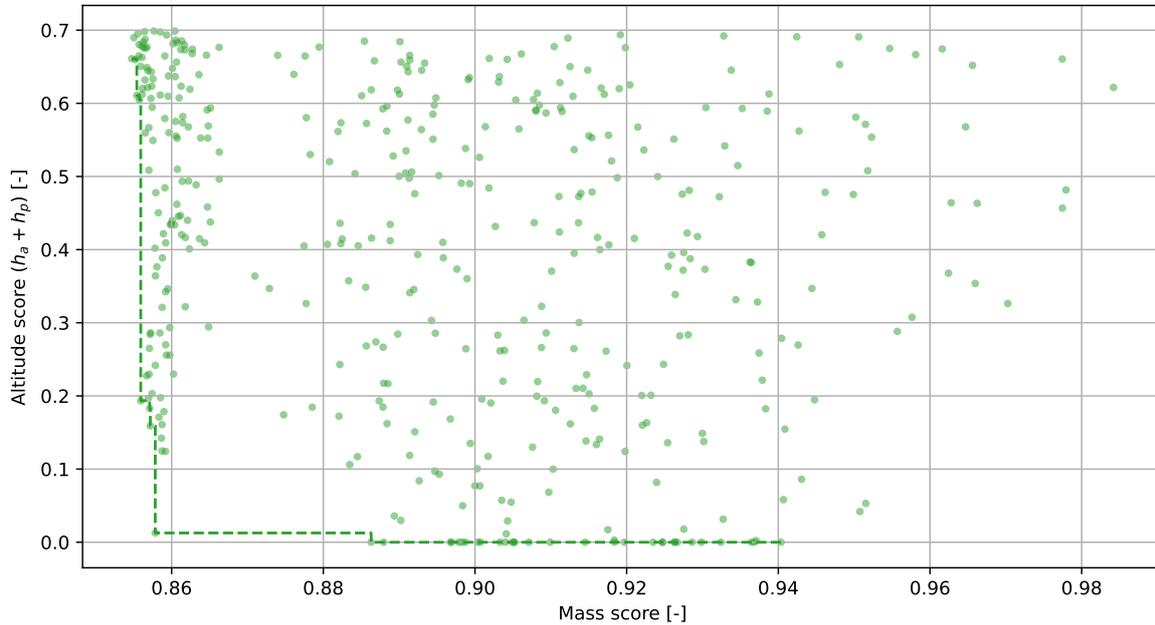
Figure 4.56: Pareto plot for the MAV optimisation, compiling results from seeds of 42, 13, 123, 846, and 579, using an anchor first stage SRM.

The Pareto front refinement process presented in Section 3.3.7 has then been run, in an attempt to find more Pareto optimum points. The results of the different front refinements are shown on Figure 4.57. In the same figure, the optimisation results are also presented, as well as the results from the design space exploration. Indeed, while these results are not from the optimisation itself, all of these simulations that have been run are valid and interesting results nonetheless. This figure then compiles the results of 131 thousand SRM burn and ascent simulations, cropped to only show altitude scores of a maximum of 0.7, and mass scores below 1. The different plots combined in Figure 4.57 are shown in Appendix A.5. This appendix also presents the combined Pareto front from the optimisations that use the tubular, rod and tube, and multi-fin SRM for the first stage.
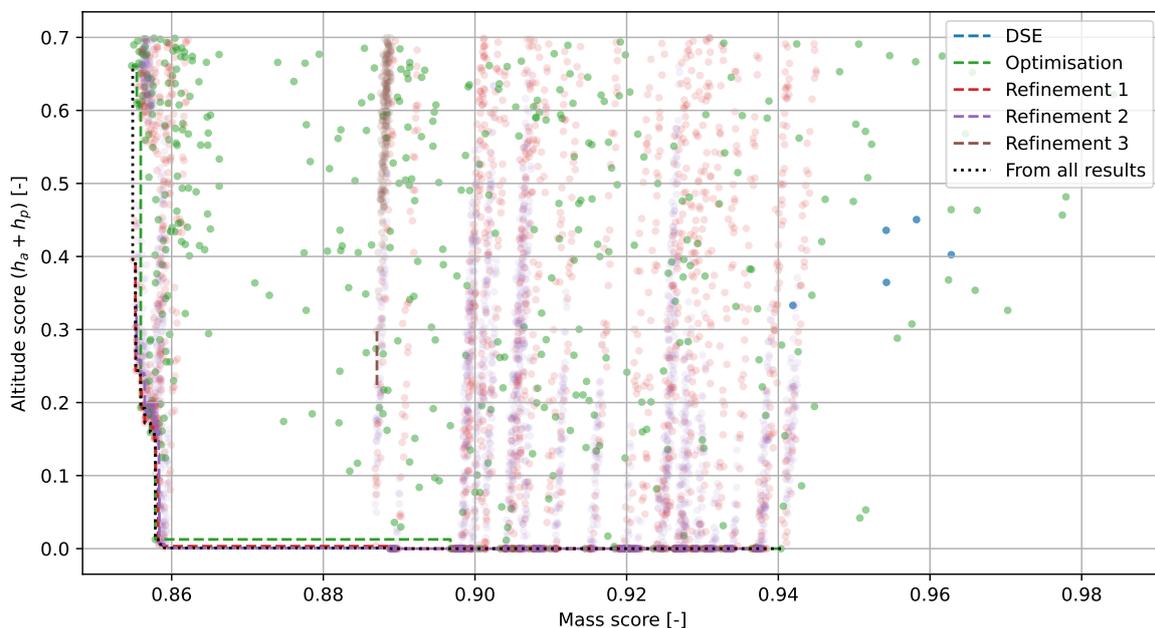
Figure 4.57: Pareto plot compiling all results from the simulation database, using an anchor first stage SRM.

From Figure 4.57, one can see that the different front refinements that have been run did manage to improve the Pareto front, both by making it more uniform, and by finding new Pareto points. Thanks to this refinement, new solutions have been found that manages to reach the target orbit with a lower MAV GLOM. Looking at Figure 4.57, it is estimated that, while some slightly better solutions could still be found, the benefit of running more simulations is not worth the effort, given the uniformity of the Pareto front.

The best solution from the refined Pareto front is with a MAV GLOM of 343.14 kg, the second stage accounting for 102.7 kg. This is 12.3 kg lower than before the refinement. This solution results in an apoapsis altitude of 339.4 km, and a periapsis altitude of 314.1 km, well above the hard lower altitude bound of 300 km. Furthermore, the burn time of the first stage is of 17.8 s, while the second stage burn time is of 25.2 s.

The design variables that result in this optimum MAV ascent can be found in Table 4.18. The angle at which the two-stages separate is found to be optimum at 89.13 deg, which makes sense since this is close to 90 deg, which results in the MAV circularising at its apogee, being the most efficient use of thrust. The difference of 1 deg allows for the second stage to make up for the loss in altitude caused by firing exactly at apogee, meaning that the SRM burn ends after the vehicle has already fallen back to Mars for roughly half the burn time. This optimum also shows that TVC in the z-direction is indeed used, with deflection of up to 2 deg during the first stage burn. However, since the first stage burn is of only 18 s, the number of nodes should be decreased, a separate TVC optimisation be made, or TVC removed altogether from the optimisation and be kept for control as to limit deviations between the real and the simulated trajectories. Appendix A.5 shows similar tables to Table 4.18, showing the optimum design variables in case different first stage SRM types are used.

While these results are for the solution that has the absolute lowest mass while still achieving the target orbit, the standard deviation between the 100 best solutions has been computed, as indicated in the rightmost column of Table 4.18. This standard deviation is indicated as a percentage of the optimum value, and it is a measure of the spread of the design variables used to achieve the target orbit with the least mass possible, at maximum 1.7 kg above the absolute optimum.

From the standard deviations, it can be seen that, for instance, different values can be used for TVC, without impacting how optimum the solution is. The same can be said for the anchor SRM fillet radius. This hints that both of these values are of lower importance. Other standard deviations are much lower. There is close to no room to change the values of for instance the stage separation angle, or the number of anchors as to improve the solution.

Design variable	Baseline value	Optimum value	100 optimums σ
Launch angle [deg]	57.5	50.79	2.37%
Stage separation angle [deg]	90	89.13	0.09%
TVC in z-direction, 5 nodes [deg]	0, 2.86, 0, -2.86, -2.86	0.6277, 1.0052, 0.1329, -1.8895, -1.7226	238.57%, 74.66%, 54.22%, 57.29%, 34.49%
Spherical SRM outer radius fraction $R_{o,2,\text{frac}}$ [-]	0.76	0.7573	1.13%
Spherical SRM inner radius fraction $R_{i,2,\text{frac}}$ [-]	0.4816	0.5388	5.83%
First stage SRM length L [m]	1.15	0.9813	3.32%
First stage SRM outer radius R_o [m]	0.26	0.2438	1.18%
Anchor SRM inner radius fraction $R_{i,1,\text{frac}}$ [-]	0.6346	0.4981	2.24%
Anchor SRM number of anchors N_a [-]	6	4	0%
Anchor SRM spacing fraction w_{frac} [-]	0.7895	0.7632	8.06%
Anchor SRM fillet radius fraction $r_{f,\text{frac}}$ [-]	0.5	0.4547	10.98%
Anchor SRM spoke width fraction $\delta_{s,\text{frac}}$ [-]	0.091	0.5105	4.67%

Table 4.18: Optimum design variables, compared to baseline values, and their standard deviation amongst 100 optimums, using an anchor first stage SRM.

The ascent simulation of this optimum that achieves the target accuracy with the lowest mass can then be seen plotted in Figure 4.58. Most notably, the 2D trajectory of the MAV in Figure 4.58e clearly shows that the launch starts more horizontally than what would be considered a classic launch on Earth. This works well on Mars thanks to the lowed atmospheric density and, in this case, the high thrust magnitude.
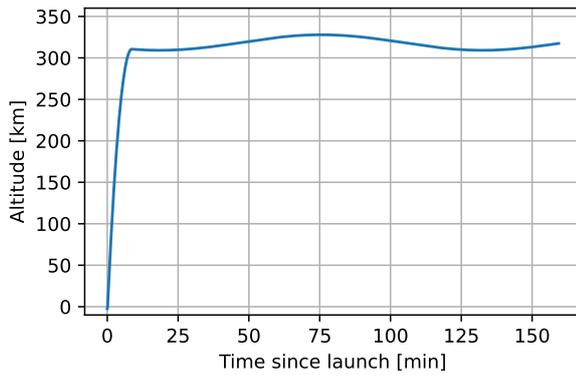
(a) MAV altitude history.
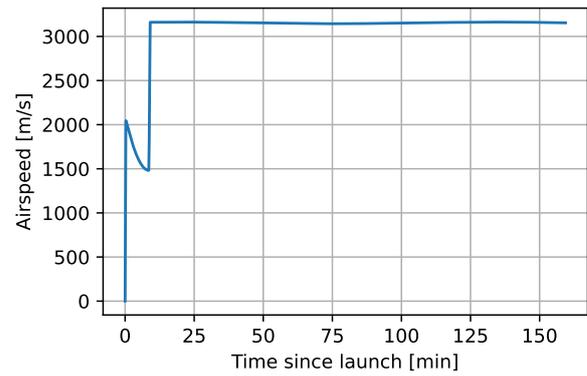
(b) MAV airspeed history.
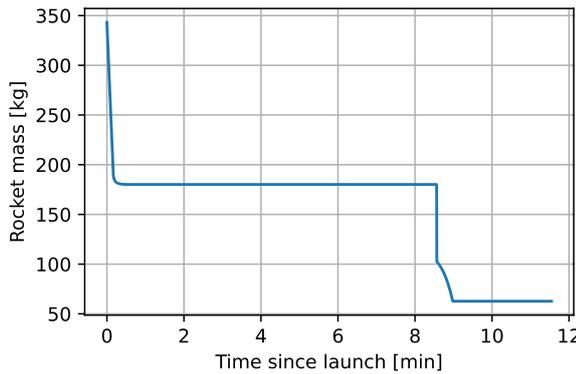
(c) MAV mass history up to 3min after stage separation
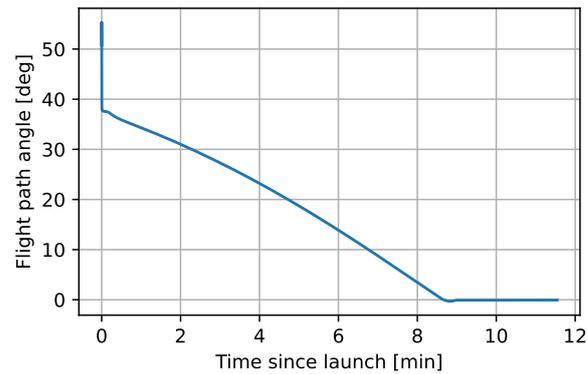
(d) MAV flight path history up to 3min after stage separation.
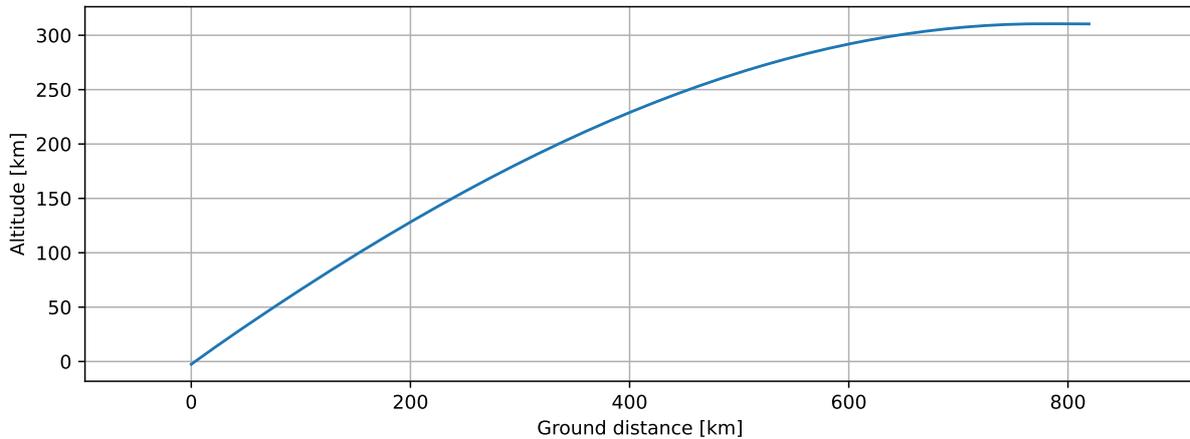
(e) MAV 2D flight path history up to second stage burnout. Both axes use the same scale.

Figure 4.58: Ascent simulation of the MAV for the best optimum found, using an anchor first stage SRM.

Lastly, the optimum SRM geometry of both stages can be seen in Figures 4.59a and 4.59b, visually representing the values related to the SRM geometry from Table 4.18. Computing the SRM geometry for both stages from the design variables, the first stage SRM has a length of $L = 0.9813$ m, an outer radius of $R_{o,1} = 24.381$ cm, an inner tube radius of $R_{i,1} = 12.146$ cm, and $N_a = 4$ anchors with a separation of $w = 3.113$ cm, a fillet radius of $r_f = 0.659$ cm, and a spoke width of $\delta_s = 8.768$ cm. The second stage spherical SRM has an outer radius of $R_{o,2} = 18.463$ cm and an inner radius of $R_{i,2} = 9.947$ cm. The thrust profile that corresponds to both these SRM can be seen in Figures 4.59c and 4.59d. This showcases the high initial thrust of the anchor SRM, followed by a lower and decreasing thrust once the inside of the SRM has burned out. The thrust profile of the second stage spherical SRM only increases over time, due to its spherical nature. The cross-sectional geometry of the optimum SRM in case the first stage SRM is not an anchor SRM are shown in Appendix A.5.

(a) First stage SRM.

(b) Second stage SRM.

(c) First stage thrust profile.

(d) Second stage thrust profile.
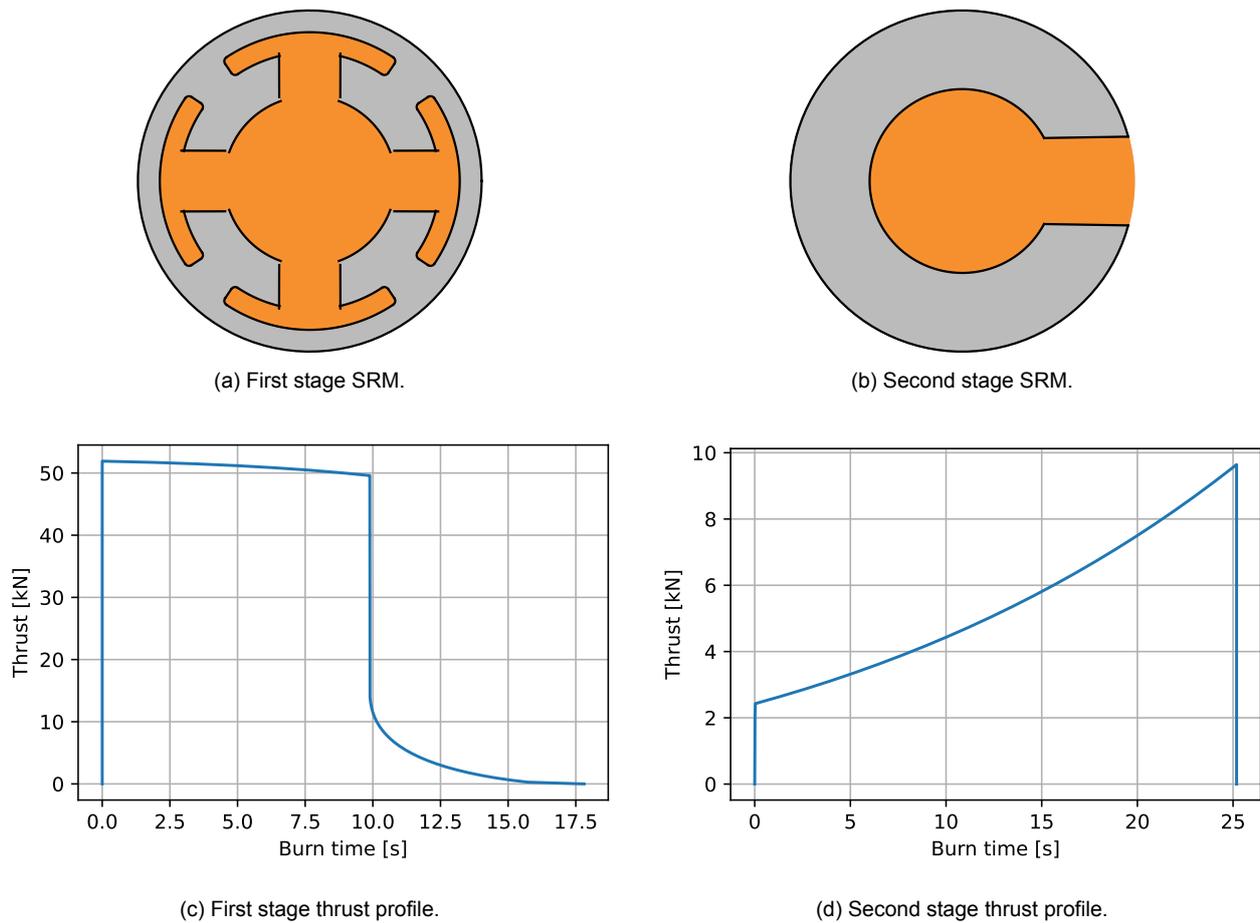
Figure 4.59: Cross-sectional geometry and thrust profile of the SRM that results in the optimum MAV ascent, using an anchor first stage SRM.

From the database of 634 thousand solutions generated through this research, 18 thousand resulted in a MAV mass within the requirement of 400 kg, and reaching the target orbit. This means that different researches can be done without having to run a new optimisation or even simulations, by first looking at the database of solutions to get a first good sense of what can be achieved.

While the optimisation process was run with the objective of minimising the launch mass, the priority could shift to for instance the minimisation of the second stage burn time. The minimum achievable second stage burn time while still fitting the orbital altitude target is of 5.22 s, with a periapsis altitude of 328.7 km and an apoapsis altitude of 360.4 km. In this case, the MAV would use a multi-fin first stage SRM, have a GLOM of 389 kg, and the first stage SRM would have a significantly higher burn time of 34.7 s. This solution is interesting as the second stage is uncontrolled, potentially leading to high orbital deviations if the burn time is too high. Also, this solution still manages a MAV mass 11 kg lower than limited.

In conclusion, the optimisation shows that not only the MAV ascent on Mars is feasible given the required mass and target orbit, but that there is even a possibility of reducing the mass by 57 kg, allowing for instance to bring another payload on Mars besides the MAV, in the same lander. This could consist in more science instruments, or for instance more helicopters to bring samples back to the MAV. Furthermore, these results indicate that a higher payload mass would be feasible, given the 57 kg margin. However, the exact amount of extra payload mass that could be added is not known. New optimisations would be required to tune the MAV SRM geometry and ascent to a new payload mass.

4.4. Validation

In this section, the different part of this research that have been validated are discussed. These are the validity of the DSMC results, the validation of the thrust models, and the study of whether or not

the required simulation accuracy is reached. Other parts of this research that have not been validated are assumed to be validated already before use. These are for instance the SPARTA, CEA, RPA-C, and Tudat software, which were validated through numerous unit tests and experiments.

4.4.1. DSMC

From the atmospheric conditions of Table 4.1 and the length of the MAV second stage, one can compute the Mach and Knudsen numbers as a function of altitude, as reported in Table 4.19. Importantly, this shows that the Knudsen between 0.1 and 10 at an altitude of 100 km, then it is always above 10. This means that the flow is transitional at 100 km, and free-molecular at higher altitudes, indicating that using the DSMC method is suited to study the aerodynamics of the MAV in the Martian atmosphere.

Altitude [km]	Mach number [-]	Knudsen number [-]	Altitude [km]	Mach number [-]	Knudsen number [-]
100	19.5229	1.05289E+00	275	10.4315	3.62925E+06
125	18.7224	4.27472E+01	300	10.1257	1.17448E+07
150	15.7385	8.99116E+02	350	9.8292	7.51163E+07
175	14.3184	7.96352E+03	400	9.6898	2.35742E+08
200	12.9100	5.05252E+04	450	9.6002	4.27082E+08
225	11.7434	2.47954E+05	500	9.5286	5.86192E+08
250	10.9463	9.96530E+05			

Table 4.19: Mach and Knudsen numbers as a function of altitude.

Two plots have been made to ensure that the constraints of Equations 3.4 and 3.6 are met. First, Figure 4.60 shows the Knudsen number of the grid size for the DSMC at an altitude of 100 km, which is the Knudsen number computed using the grid size as reference length. This shows that this grid size is too high at the closest of the MAV nose, being of about 3 where it should be at least 5. This is caused by the high number of particles accumulating in front of the MAV, increasing the mean free path. Running the DSMC on a server, 211 Go of memory were being used after the different refinements of the grid. Decreasing the grid size further would then require a different computer to run the simulation on. Moreover, this effect is only present at an altitude of 100 km. At higher altitudes, the lower velocity of the MAV and lower atmospheric density result in higher Knudsen numbers.

Figure 4.61 then shows the number of PPC for the DSMC at an altitude of 250 km. This shows that, except in the wake at the back of the MAV, there is at least 10 PPC in the simulation volume. The same has been noted for all the altitudes. This number is achieved thanks to a process of trial and error, increasing the number of PPC at simulation start and at each refinement until a satisfactory number of PPC is reached.
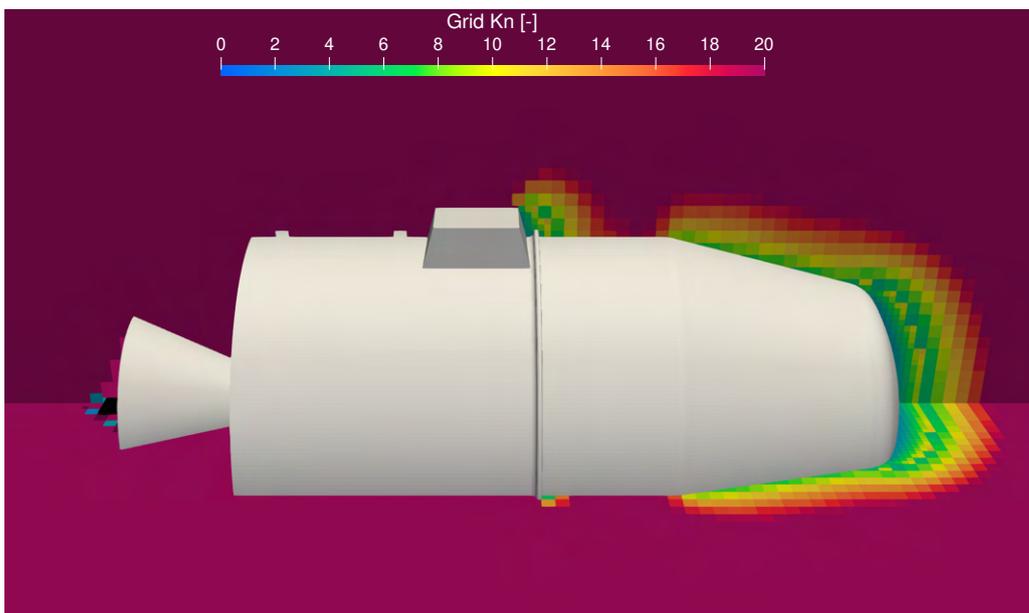
Figure 4.60: Grid size Knudsen number from the DSMC method at an altitude of 100 km.
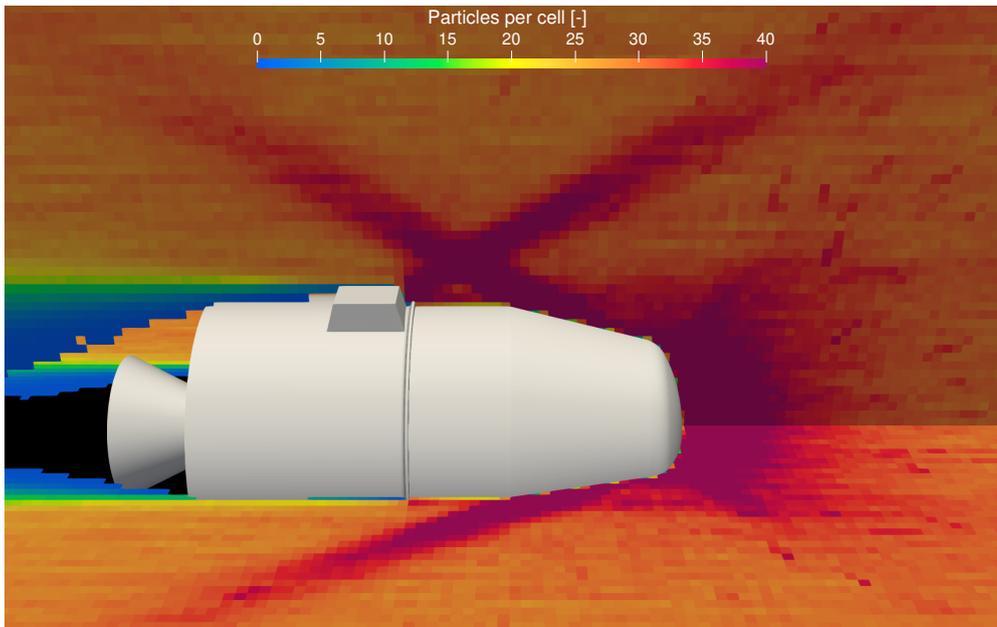
Figure 4.61: Number of simulated particles per simulation grid cell from the DSMC, at an altitude of 250 km.

Because the DSMC method is suited for the MAV in the Martian atmosphere, and it was run by fulfilling the constraints on the grid Knudsen number and the number of PPC, it is judged that the drag coefficients that were obtained with this method are valid for use in the simulations that were run in this research.

4.4.2. Thrust

Validation of the thrust models has been made by comparing results to the ones from the openMotor [97] software. This software has been validated by comparing SRM burn simulations with results from solid rocket motor burns, with model rocket motors of different cross-sectional geometry. In particular, the tubular and rod and tube geometries are implemented and validated in openMotor.

For a tubular geometry, a length of $L = 13.97$ cm, an outer radius of $R_o = 4.153$ cm, and an inner radius of $R_i = 2.8$ cm has been used. Also, the following propellant and motor parameters are used: throat area $A_t = 1.533$ cm^2, area ratio $\epsilon = 6.25$, propellant burn exponent $n = 0.3273$, propellant burn coefficient $a = 3.5952 \cdot 10^{-5}$ m/s/MPa^n, chamber temperature $T_c = 2800$ K, ambient pressure $p_a = 101325$ Pa, propellant density $\rho_p = 1670$ kg/m^3, propellant molar mass $M = 0.02367$, and specific heat ratio $\gamma = 1.21$.

This corresponds to a model tubular SRM that was used for validation by the openMotor developers. A comparison of the implemented simulated SRM burn and the openMotor results is then shown in Figure 4.62, comparing both the thrust and the propellant mass over the burn time. From this comparison, the difference in thrust is of a maximum of 20 N, which is of 3.7%. The final propellant mass is, as expected the same, at 0 kg. However, the openMotor burn time is of 0.02 s more, which is 0.3% higher than the implemented simulation.
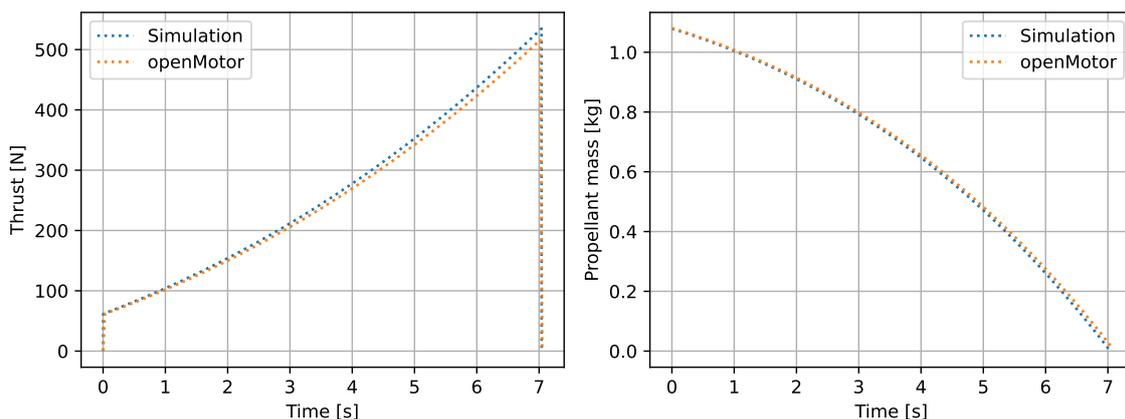
Figure 4.62: Verification of the thrust computation from the tubular SRM geometry.

For the rod and tube geometry, a length of $L = 91.44$ cm, an outer radius of $R_o = 18.415$ cm, an intermediate radius of $R_{\mathrm{mid}} = 10.73$ cm, and an inner radius of $R_i = 7.68$ cm has been used. For the propellant and motor parameters, the same as for the tubular geometry are used, expect for the followings: throat area $A_t = 13.795$ cm^2, area ratio $\epsilon = 8.29$, propellant burn coefficient $a = 3.517 \cdot 10^{-5}$ m/s/MPa^n, chamber temperature $T_c = 3500$ K, and propellant density $\rho_p = 1680$ kg/m^3.

A comparison of the thrust and propellant mass over the burn time of the results from openMotor and the simulation are then shown in Figure 4.63. This shows that the simulation gives a thrust at a maximum of 400 N below the one from openMotor, which is of 3.1%. The burn time is 0.28 s lower with openMotor, which is a difference of 0.4%.
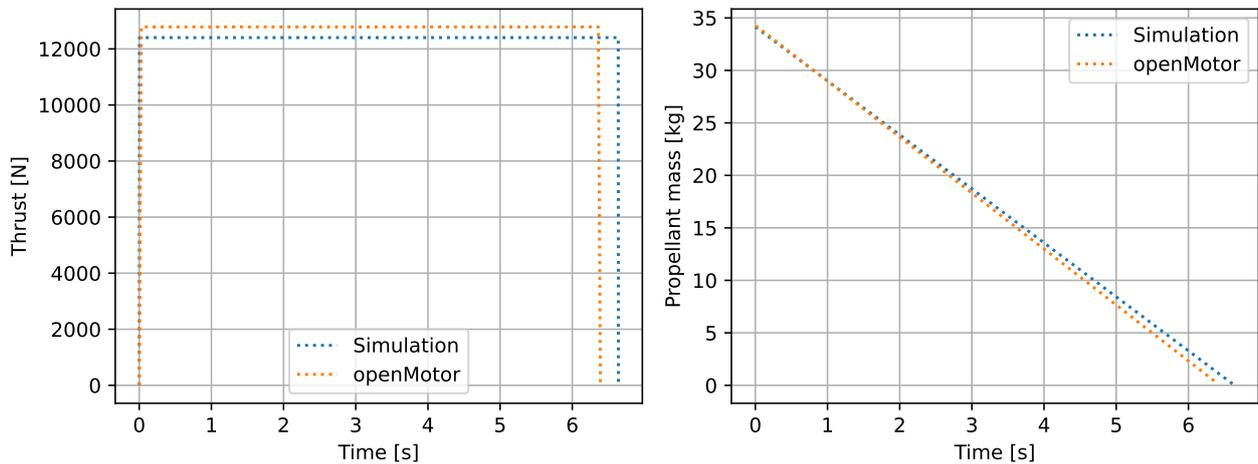
Figure 4.63: Verification of the thrust computation from the rod and tube SRM geometry.

Both of these comparisons with openMotor are taken as an indication that the implementation of the thrust in this research is valid for the computation of the thrust profile as a function of the tubular and rod and tube geometries, and as a function of the propellant parameters.

Unfortunately, the openMotor software does not allow to simulate an anchor, multi-fin, or spherical SRM. This means that the thrust and mass over time from these models is validated for given propellant parameters, but not for their geometry. Because the multi-fin and spherical SRM both use relatively straightforward equations, as given in Section 3.1.3, it was decided that manually verifying the equations is satisfactory. However, the anchor model is significantly more complex.

This anchor model has been validated by comparing results with the peer-reviewed article that established its geometry parametrisation [65]. In particular, the burning perimeter over the burn distance has been validated. In Figure 4.64a, one can see the total burning perimeter over the burn distance from the peer-reviewed article at the top of the figure, with the burning perimeter of the 7 distinct segments of the anchor geometry at the bottom of the figure. These segments were shown in Figure 3.10.

The same geometry as the one from the article is then used, and the burn simulation is run, saving the burning perimeter of the anchor model as well as the burned distance. In Figure 4.64b, these parameters are then plotted against each other. This then shows that the implemented anchor SRM geometry model corresponds with the one of the peer-reviewed article exactly. A superposition of both figures, at the same axis scale, is shown in Appendix A.6. One may notice that inches are used as a distance unit in the peer-reviewed article, while meters are used in the implementation of this research. This however does not cause any difference in how the perimeter and overall SRM burns over time, since the geometry equations are all linear.

(a) Burning perimeter over burned distance from the peer-reviewed article [65].

(b) Burning perimeter over burned distance from the implemented simulation.
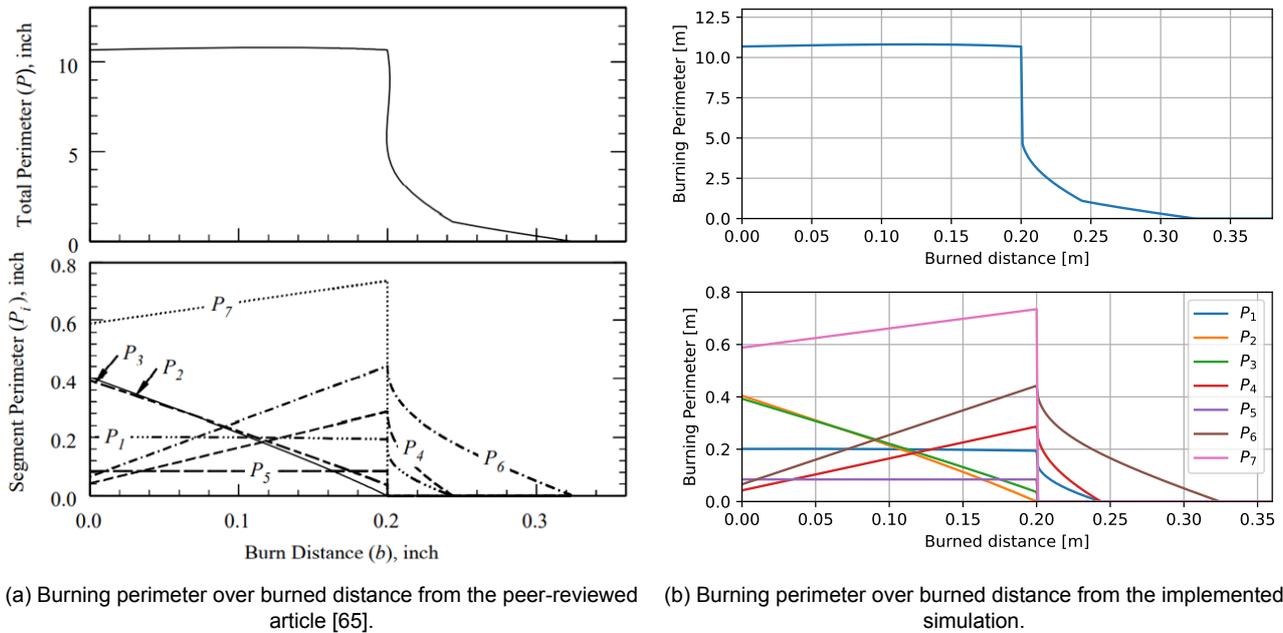
Figure 4.64: Verification of the burning perimeter over burned distance for the anchor SRM geometry.

4.4.3. Ascent simulation accuracy

In Sections 4.2.3 and 4.2.4, an effort is made to tune the integrator and environment used for the optimisation simulation, as to ensure an accuracy of 5 km and 5 m/s with the lowest CPU time possible. From the optimisation, the 200 solutions that achieve the target orbit with the lowest GLOM have been sampled. Then, their design variables have been used to re-run the SRM burn and ascent simulations.

However, this time, a time step 10 times lower has been used for the SRM burn simulation. For the ascent simulation, the same integrator is used, but with a time step control tolerance that is 10^5 times lower. For the simulated Martian environment, the SH used to represent the gravitational field of Mars have been increased from a D/O 6 to 8. The effect of the radiation pressure from the Sun is added modelled as a cannonball.

The difference in position between the simulations run during the optimisation and the ones run with a higher accuracy is shown in Figure 4.65. This shows that, at a maximum, the final difference in position is computed to be of a maximum of 4.39 km, with a mean difference of 1.07 km. The same plot is shown in Appendix A.6, showing that the maximum difference in final position is of 36.25 m/s, with a mean velocity difference of 11.4 m/s. This shows that the integrator and environment tuning was successful at reaching the target positional accuracy of 5 km. The target accuracy in velocity of 5 m/s is not reached. However, the latter was set as a way to keep at the same time the position accuracy under control, since the velocity influences the position. Also, the final position is used in the optimisation, since the final periapsis and apoapsis altitudes are scored as objectives. The final velocity of the MAV is not used in the optimisation.
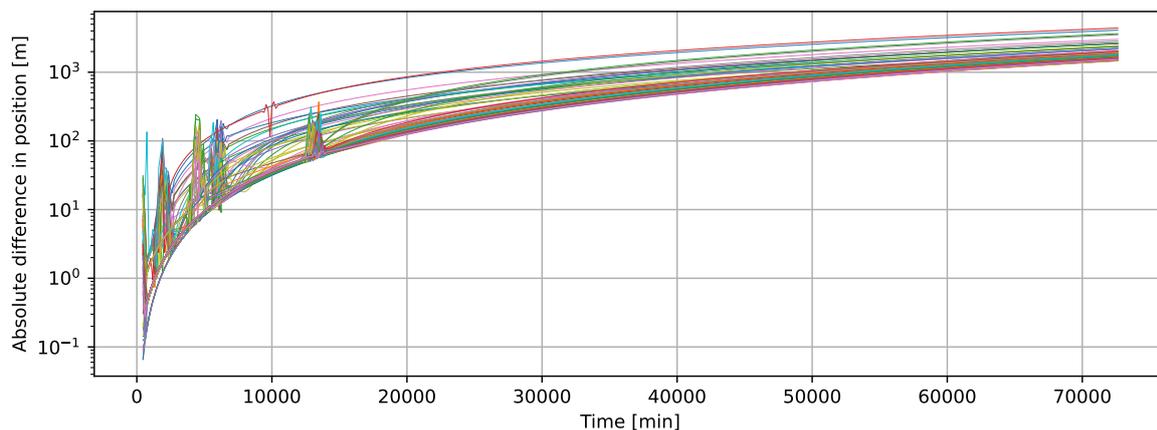
Figure 4.65: Difference in position over time of selected simulations run with a higher accuracy integrators and environment.

4.5. Sensitivity analysis

In this section, the robustness of different aspect of the research is studied. First, the decay of the final MAV orbit is studied. Follows with the sensitivity analysis of the MAV trajectory to initial conditions and to the environment. Finally, the sensitivity of the MAV trajectory to variations in the design of the MAV and in its SRM is studied.

The goal of the sensitivity analysis is mainly to ensure that, would any deviations from the simulation be present, the resulting final orbit would still allow for the ERO to safely rendezvous with the MAV. This means that the inclination should be close to the target one of 20 deg, and the rendezvous altitude should be at a minimum of 300 km.

4.5.1. Orbital decay

Once the ascent of the MAV is complete, the second stage of the vehicle stays in orbit. Due to perturbations in the gravitational field, radiation pressure from the Sun, and drag of the Martian atmosphere, this orbit changes over time. First, the decay of the final orbit in the optimum case has been studied. This optimum final orbit has a periapsis of 314.4 km and apoapsis of 339.4 km. This orbit has been numerically propagated with the integrator and environment settings that were recommended for the highest accuracy. This propagation is terminated once the periapsis gets below 300 km, corresponding to the moment at which the ERO cannot safely rendezvous with the samples anymore, as discussed in Section 2.2.1.

This results in the periapsis and apoapsis history shown in Figure 4.66. As seen in this plot, there is a global trend in both of these altitudes to decay towards Mars. Also, waves can be seen, showing that, periodically, the apoapsis decays while the periapsis increases. This essentially shows a sinusoidal variation of the orbital eccentricity over time. From Figure 4.66, the altitude of the MAV second stage becomes too low after 84 days. This thus leaves 84 days for the ERO to match the orbit of the MAV second stage and catch the orbiting samples.
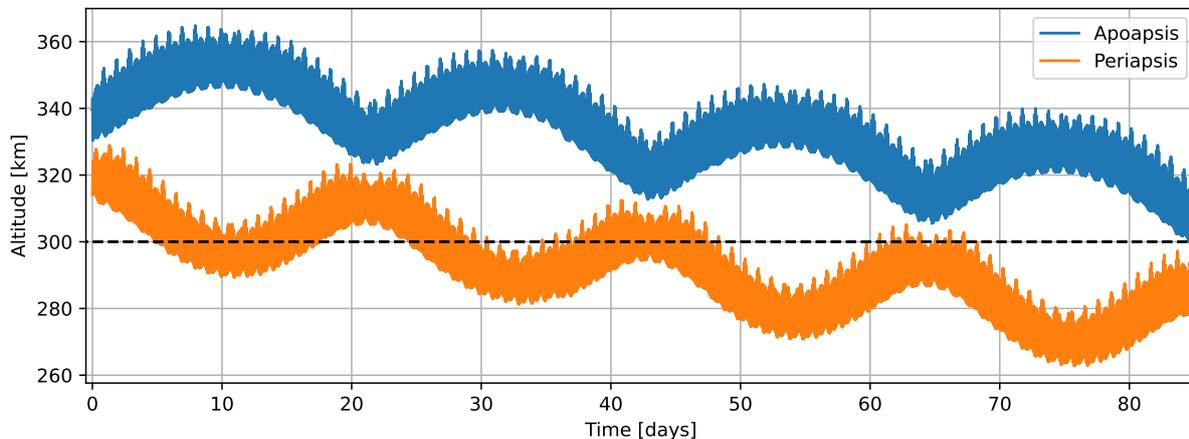
Figure 4.66: Apoapsis and periapsis of the MAV second stage over time, starting from $h_p = 314.1$ km and $h_a = 339.4$ km.

However, during the decay of the orbit, the orbital inclination also changes, mostly due to the non-homogeneous gravitational field of Mars. It is planned for the ERO to catch the samples in an orbit inclined at 20 deg, which is also the target orbital inclination of the MAV. Figure 4.67 then shows the change in inclination of the MAV over time during orbit decay. This shows that, while the inclination varies from the target one, there are 5 more times at which the inclination becomes the target one again. Including the initial orbit right after launch, this thus leaves a total of 6 opportunities for the ERO to catch the samples in the target orbit, with 5 repeats, the latest being 78 days after launch.
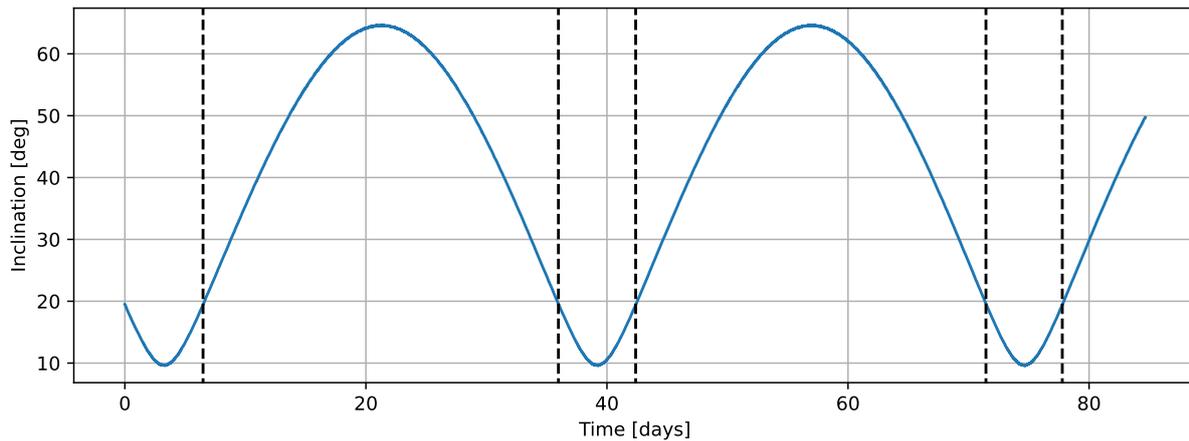
Figure 4.67: Inclination of the MAV second stage over time, starting from $h_p = 314.1$ km and $h_a = 339.4$ km.

Lastly, the same study has been made for different initial altitudes. Propagating circular orbits of different altitudes until decay resulted in Figure 4.68, showing the decay time and number of inclinations repeats as a function of the initial periapsis altitude. As one could expect, increasing the initial orbital altitude results in a higher orbital lifetime, and a higher number of inclinations repeats, leaving more opportunities for the ERO to catch the samples. Also, while this plot has been generated for circular orbits only, increasing the initial apoapsis altitude while keeping the periapsis identical will only increase both the orbital lifetime and the number of repeats.
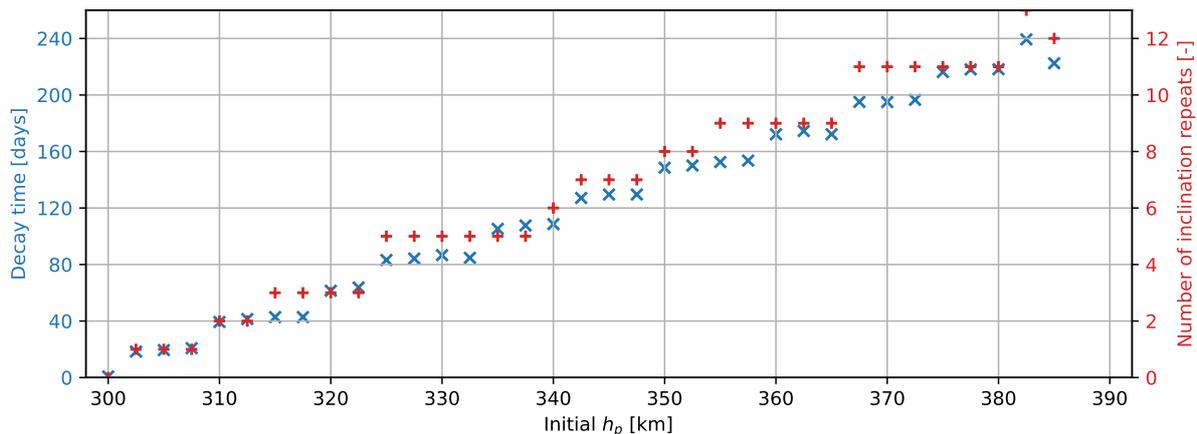
Figure 4.68: Number of inclinations repeats and orbit lifetime as a function of the initial periapsis altitude.

4.5.2. Initial conditions

Next, the initial ascent conditions have been varied to study what impacts these have one the final orbit of the MAV. These initial conditions are first the altitude, latitude, and longitude of the launch site. Then, the possibility of a delay between the apogee and stage separation is investigated.

Variations in launch and stage separation angle, while important, are not part of the sensitivity analysis. These parameters, being design variables in the optimisation, were already analysed during the design space exploration, in Section 4.3.1.

To define the bounds in which the altitude, latitude, and longitude can vary, the topography of the Jezero crater, and of the Perseverance landing ellipse, are used. This is because the MAV launch site is in the same area of the Jezero crater as Perseverance, and the landing ellipse is assumed to be of similar size. As shown in the topographic map of Appendix A.7.1, the landing ellipse covers a latitude range of 6.6 km and a longitude range on 7.7 km. In that ellipse, altitudes range from -2.45 km to -2.65 km, the nominal launch altitude of the MAV being of -2.55 km.

Altitude

The launch altitude of the MAV has first been varied using a normal distribution, with the mean altitude at -2550 m, and a standard deviation of 50 m, as to cover initial altitudes ranging from roughly -2650 m

to -2450 m. Taking 10 thousand samples from this distribution, simulations have then been run for each initial altitude, and the resulting final orbit compared to the nominal one, using a launch altitude of -2550 m.

The result of this process is shown in Figure 4.69, which presents the error in launch altitude, the final error in periapsis, apoapsis, and inclination, all plotted against each other. The diagonal of this figure shows the density distribution of each parameter, showing in this case the distribution in launch altitudes and in resulting orbital error. The lower triangle of this plot then shows individual dots of each simulation for different initial altitudes. These dots not only show the relation between the initial altitude and the final orbital error, but also of the relation of the orbital errors with one another. Lastly, the upper triangle of this plot shows the same data as the dots, but as contour plots, to show the distribution of the final orbital error space.
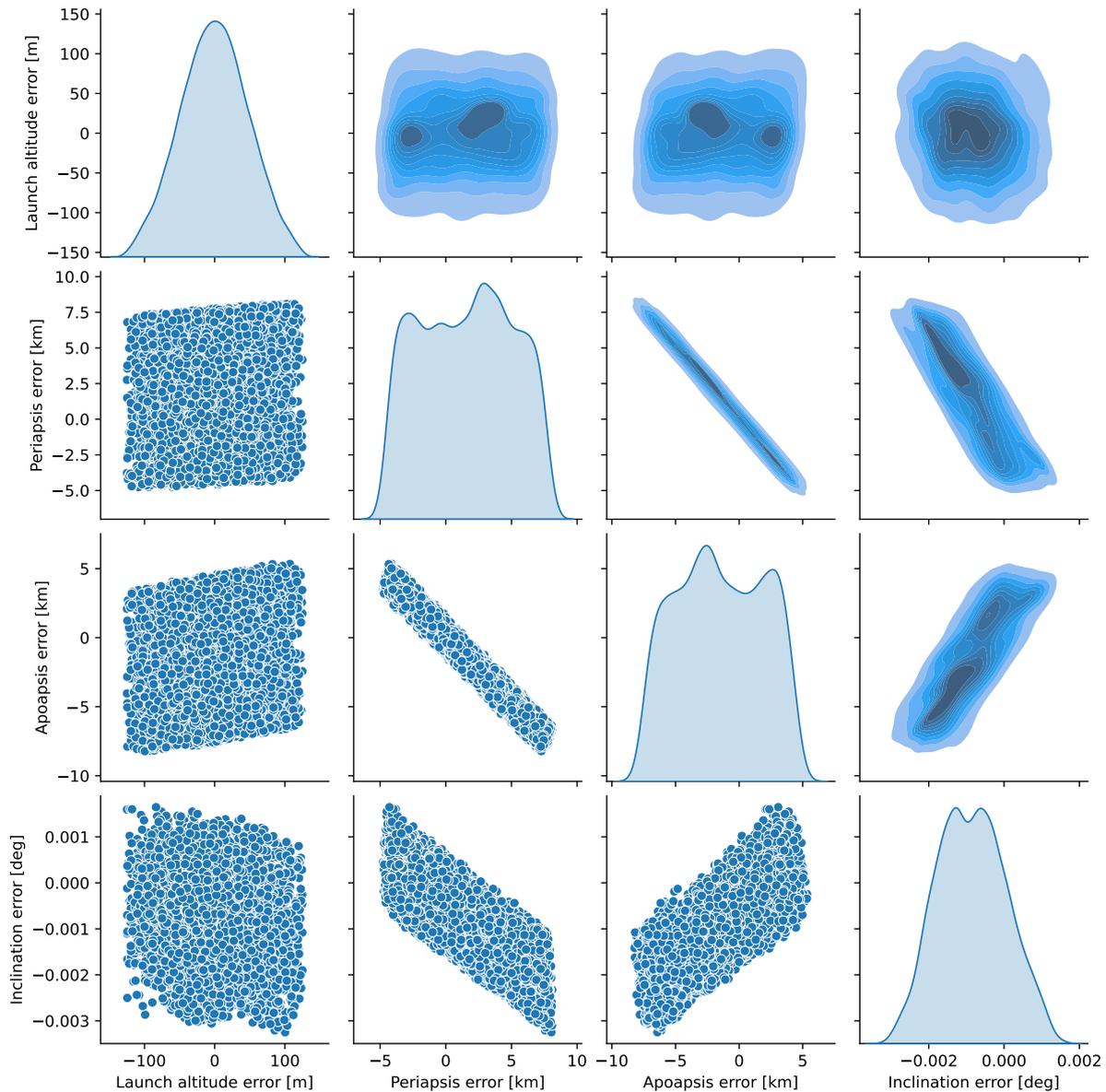
Figure 4.69: Initial altitude, periapsis, apoapsis, and inclination errors, all plotted against each other.

Figure 4.69 first shows the normal distribution of the error in launch altitude on the top left. Then, on the bottom right, a similar distribution can be seen, showing that the final inclination error also behaves as a normal distribution, but with the highest error being of a magnitude of 0.0025 deg, which is deemed insignificant.

The same figure then shows a correlation between the periapsis error and the apoapsis error. This is interpreted as follows: an increase in periapsis is due to a decrease in apoapsis, and vice-versa. This effect can also be seen in the periapsis and apoapsis error distributions: they both have the same shape, but are mirrored around an axis at roughly $x = 2.5$ km.

Most importantly, the highest periapsis decrease caused by the error in altitude is of 5 km. Since the periapsis has a nominal altitude of 314 km, this means that the ERO can still rendezvous and catch the samples when the periapsis is decreased by 5 km, to 309 km.

Also, a high error in altitude of 100 m can only be present if the initial position of the MAV is completely unknown after landing. In such scenario, while the MAV can still reach a satisfactory orbit, it is unlikely that the MSR mission can continue. Instead, if the initial altitude is known to be different from -2550 m at the position where the MAV landed, a simulation can simply be re-run with the new initial altitude, so that the launch and separation angles, as well as the TVC angles, can be adapted if needed.

Latitude and longitude

The same process has been repeated to study the final orbital error due to an initial error in latitude and longitude, using a normal distribution with a mean of 18.5 deg and a standard deviation of 0.11 deg for the latitude, and a mean of 77.5 deg and a standard deviation of 0.13 deg for the longitude. Once converted to distances, this normal distribution then covers the entire landing ellipse of Perseverance.

Running once again 10 thousand simulations results in the plot presented in Appendix A.7.1 as Figure A.27. This shows that, once again, there is the same correlation between the error in periapsis and in apoapsis. This time, the maximum decrease in periapsis is higher, at 8 km. This puts the periapsis altitude at 306 km, which is getting critical, but still acceptable for rendezvous. The maximum absolute inclination error is also increased to 0.18 deg. This is however still deemed satisfactory, as the ERO can most likely afford to do a manoeuver to adapt its orbital inclination by such a small amount.

Stage separation

Last but not least, a possible delay in the stage separation has been studied, as this would essentially cause the MAV to start its second stage burn after the apoapsis has been passed. As such, a gamma distribution has been used, with a shape parameter of 1, and a scale parameter of 2, meaning that most of the separation delays are in the range from 0 s to 7.5 s.

As presented in Appendix A.7.1, in Figure A.28, this results in a very clear correlation between the stage separation delay and all the orbital errors. These orbital errors even follow the same distribution than the stage separation delay, showing that these can easily be predicted once the stage separation delay is known. However, since the second stage is uncontrolled, it is impossible to adapt the second stage burn to counteract any arising errors.

While a stage separation delay of 7.5 s results in a significant maximum decrease in periapsis of 20 km, the effect on the periapsis is opposite, meaning that the apoapsis is increased by 20 km. This is due to the stage separation occurring very shortly after apogee. Such a decrease in periapsis means that the ERO can no longer rendezvous with the MAV at its periapsis. However, rendezvous is still possible at apoapsis, especially since its altitude is increased at the same time. It is also worth noting that an unexpected stage separation of more than a few second is deemed unlikely. It is expected that either up to 2 s of unplanned delay could occur, or that stage separation would not occur at all, resulting in mission failure. In case a 2 s delay occurs, the loss in periapsis altitude is of a more manageable 8 km. Furthermore, these results show that, if a lower than nominal periapsis is expected from the first stage burn, the stage separation can be delayed as to increase the periapsis altitude.

4.5.3. Environment

The effect of two key variations in the environment on the final MAV orbit have been studied. These are variations in the atmospheric density of Mars, and the addition of winds. First, the atmospheric density has been varied between 0.65 to 1.3 times the nominal density computed from the two-step exponential atmosphere model. These factors of 0.65 and 1.3 have been set according to variations in atmospheric density measured and predicted at the Perseverance landing site, as taken from the online interface of the MCD [48].

Running 10 thousand simulation with this variation in density results in Figure A.29. From this figure, one can first see on the top left that the density was, as planned, scaled by factors ranging between 0.65 and 1.3. Then, it can be observed that these variations in atmospheric density cause

the periapsis to decrease by as much as 50 km, increasing at the same time the apoapsis by 50 km. This means that, either rendezvous with the ERO is not feasible anymore, or it must be carried at apoapsis, immediately after launch. This seems sub-optimal. At such, because of the very high influence of atmospheric density on the final orbit, it is recommended to launch the MAV when it is predicted that the atmosphere is of average density. This can be done by ensuring that no solar storm is expected. The ascent simulation could also be re-run just before launch to adapt the launch, TVC, and stage separation angles if needed, as to lower deviations in the final orbit. It is for instance possible to change the separation time and angle to increase the periapsis altitude, at the cost of the apoapsis altitude.

The second environmental effect that has been studied is the addition of winds. These have been taken from the MCD [46, 47], as predicted for the entire year 2031. These winds are then coupled with the ascent simulation as a function of time, altitude, longitude, and latitude. In this process, simulations have been run with the predicted winds for each Solar longitude increment of 1 deg, and time of day increments of 1 hr. This results in 8616 simulations run, with winds at lift-off of up to 15 m/s. The error in periapsis, apoapsis, and inclination is then plotted in Figure A.30, showing that a decrease in periapsis of up to 5 km can be expected when including winds. This is manageable, and the ERO can still rendezvous to catch the samples. Still, it is recommended to re-run the ascent simulation before launch to adapt its control. To do so, the wind and preferably density profile of the atmosphere shall be known. Thanks to the savings in MAV mass, a meteorological balloon could for instance be added to the MAV launch platform, and used to sound the atmosphere before launch.

Possible variations in the gravitational field of Mars have not been studied in more details for this sensitivity analysis. Instead, findings from the environment tuning in Section 4.2.4 have been used. During the selection of a numerical gravitational model for Mars, it was found that switching from SH of D/O 6 to 8 caused an absolute difference in final position of up to 1 km. It is then expected that, would the SH model of Mars be slightly incorrect, the difference caused by this error would be at maximum in the order of 1 km. This does not significantly affect the final orbit of the MAV, and has thus not been studied further.

4.5.4. Vehicle design
Lastly, the effect of three key variations in vehicle design on the final MAV orbit have been considered. These are thrust misalignment, SRM sizing errors, and payload mass variations. In addition to these, the study of variation in drag coefficient was considered, but deemed a duplicate of the study of variation in atmospheric density. Indeed, the drag, as computed in Equation 3.72, varies linearly with both the atmospheric density ρ and the drag coefficient C_D. This means that scaling the drag coefficient by factors between 0.65 and 1.3 will give the exact same results as the analysis of sensitivity to the atmospheric density.

First, thrust misalignment was thus studied. It was assumed that misalignment of more than half a degree would be detected during manufacturing, and corrected. As such, 10 thousand simulations have been run, varying the thrust vector by angles taken from a normal distribution, along both of the y and z body axes. The results from this analysis are shown in Figure A.31. This time, more significant deviations in the orbital inclination are found, of up to 0.5 deg. This confirms the findings of the variation of TVC in the design space exploration in Section 4.3.1: varying thrust in the y-direction has a direct impact on the orbital inclination. This figure also shows that decreases in periapsis of up to 25 km can be expected when thrust is misaligned of up to 0.5 deg. While this comes with a raise in apoapsis of up to 60 km, this is deemed sub-optimal for ERO rendezvous. It is then recommended to ensure thrust misalignment of less than 0.1 deg during manufacturing, to keep deviations in periapsis and apoapsis to a few kilometres.

Then, deviations between the optimum SRM geometry and the actual MAV SRM geometry have been studied. This was done by varying all inner dimensions of the optimum anchor geometry by up to 1 mm, as well as the inner radius of the second stage spherical SRM. All dimensions are varied at the same time, to keep the number of simulations run to 10 thousand, instead of running different combinations with a lower less representative number of samples. Also, this then presents the most critical case. Indeed, if some SRM features were to be increased in size and other decreased, the effect on the trajectory could self-mitigate. Such deviations in geometry could be introduced for instance when the solid rocket propellant is put in its mould for curing. As shown in Figure A.32, such

deviations in the SRM geometry results in significant difference in periapsis and apoapsis altitudes, of up to 100 km. This is unacceptable, and is likely to result in mission failure. This is because a deviation in the SRM geometry will not only affect the thrust that the MAV uses to ascent, but also the mass of the vehicle. It is then recommended to ensure that deviations between the design geometry and the manufactured motor is in the order of 0.1 mm. This is most likely manageable for a manufacturer such a Northrop Grumman [61].

Variations in payload mass of up to 0.75 kg have been studied. These would be caused by either adding more samples in the payload container, or by launching the MAV lacking some of the samples. Figure A.33 shows the result of this sensitivity analysis. From it, it can be seen that an increase in payload mass of up to 0.75 kg will decrease the periapsis by of to 15 km. This becomes at the limit of the altitude at which the ERO is to rendezvous with the MAV, meaning that it is recommended to adhere to the planned payload mass, with deviations of up to 0.3 kg that are deemed tolerable.

Finally, it is important to realize that, while some deviations to the simulation results in acceptable difference in final periapsis in the order of a few kilometres, combining these deviations together is likely to result in error in final orbit in the range of 10 km to 100 km, which is then no longer acceptable. It is thus recommended to adhere to the nominal values as closely as possible, and to re-run propellant burn and ascent simulations before launch, with all conditions known with a higher accuracy.

5

Conclusion

This chapter concludes the research that has been carried out. To this extent, the research questions that were established in Section 2.3.1 are first answered in Section 5.1. Then, a discussion on the overall feasibility of the Mars Ascent Vehicle is made in Section 5.2, in light of the findings throughout this research.

5.1. Research answers

In this section, the different research questions and sub-questions are answered. These answers come directly from the findings in Chapter 4.

Sub-question 1.1: *What cross-sectional geometry and length of the first stage solid motor gives the best orbital accuracy at lowest mass?*

Answer: The optimum first stage SRM was found to be of a length of 0.9813 m, with an anchor cross-sectional geometry, an inner tube radius of $R_{i,1} = 12.146$ cm, and $N_a = 4$ anchors with a separation of $w = 3.113$ cm, a fillet radius of $r_f = 0.659$ cm, and a spoke width of $\delta_s = 8.768$ cm.

Sub-question 1.2: *What inner and outer diameter of the second stage spherical solid motor gives the best orbital accuracy at lowest mass?*

Answer: The optimum second stage spherical SRM dimensions were found to be an outer radius of 18.463 cm and inner radius of 9.947 cm.

Sub-question 1.3: *How should thrust vectoring control be used through ascent to reach the target orbit with the highest accuracy?*

Answer: TVC was found to be irrelevant in the y-direction as it only affects the orbital inclination. In the z-direction, TVC was found to be optimum by deflecting the nozzle by the following angles equispaced over time during first stage burn: 0.6277 deg, 1.0052 deg, 0.1329 deg, -1.8895 deg, and -1.7226 deg. However, due to the short burn time of the motor, it is recommended to not use TVC values directly from the optimisation, but to use TVC as a way to control the MAV ascent to minimise deviations from the target trajectory.

Sub-question 1.4: *What is the minimum burn time of both stages combined that allows to reach the target orbit?*

Answer: The burn time of of the motors for the MAV that reaches the target orbit with the least mass is of 17.8 s for the first stage, and 25.2 s for the second stage. Looking at other results in the database that was generated, the target orbit can also be reached for a higher mass of 370.7 kg and burn times of 24.91 s and 7.25 s for the first and second stage respectively, which is the minimum feasible combined burn time, at 32.16 s. Reducing the burn time reduces the time during which the MAV risks failure due to motor failure. Lastly, the target orbit can be reached with a longer burn time for the first stage motor of 34.7 s, and a much lower burn time for the second stage of 5.2 s. The lower burn time of the second stage could be taken as an advantage since this second stage is not guided. Leaving most of the burn time to the first stage, when control is possible, could then be favoured.

Sub-question 1.5: *What is the minimum Mars Ascent Vehicle Gross Lift-Off Mass that allows to reach the target orbit?*

Answer: The minimum MAV lift-off mass that allows to reach the target orbit is of 343.14 kg, including the second stage that has a mass of 102.7 kg. This is achieved using the SRM geometry described

in **Sub-questions 1.1 and 1.2**.

Sub-question 1.6: *What are the optimum launch and stage separation angles to reach the target orbit with minimum Gross Lift-Off Mass?*
Answer: To reach the target orbit with the MAV GLOM of 343.14 kg, the launch angle shall be of 50.78 deg from the vertical, or 39.22 deg from the local horizon. This means that the MAV starts its ascent closer to the horizontal than a classic Earth ascent, allowing it to gain horizontal velocity early on. The optimum stage separation angle is of 89.13 deg from the vertical, or 0.87 deg from the local horizon. This means that the second stage is used to mostly circularise the MAV orbit. The launch angle is to be controlled by the VECTOR system, and the stage separation angle by RCS of the first stage.

The answers to the research **Sub-questions 1.1 to 1.6** then combine and give an answer to the research **Sub-question 1**: *What vehicle configuration and control allows to reach the target orbit with the highest accuracy and lowest mass?* More details about the answers to these questions are given in Section 4.3.5.

Sub-question 2.1: *How does the initial launch angle impact the final propagated state?*
Answer: As discussed in Section 4.3.1, the launch angle has a high impact on the periapsis altitude. This is explained as, keeping the SRM geometry and TVC angles fixed, the launch angle will directly dictate what apogee altitude is reached. The second stage makes most efficient use of thrust when it is close to the horizontal, since this circularises the orbit. During the burn at the apogee, if the vehicle is horizontal, the apoapsis will be the most impacted.

Sub-question 2.2: *What choice of integrator, propagator, and environment leads to the best combination of speed and accuracy for the numerical simulation of the propellant burn and ascent trajectory?*
Answer: In light of the altitude objective score from Equation 3.113, an accuracy requirement of 5 km and 5 m/s was set for the simulations that are run during the optimisation. For the thrust simulation, the RK4 integrator that was setup reaches this accuracy requirement using a time step of 15 ms for the second stage burn simulation, and of 0.002 ms for the first stage burn simulation. For the ascent simulation, a RKF4(5) integrator should be used, with a variable step size control algorithm that uses a tolerance of 1E-07. The environment used during the ascent simulation should contain a two-step exponential atmosphere model, spherical harmonics up to degree and order 6 for the gravitational field of Mars, and the effect of the Sun gravity modelled as a point mass. All of these, combined with a Cowell propagator, allow to run the ascent simulation using only less than 2000 function evaluations, taking less than 3 s on a Intel© Xeon© E5-2683 CPU cadenced at 2 GHz.

Sub-question 2.3: *How do deviations in the environment affect the simulated states of the Mars Ascent Vehicle?*
Answer: In Section 4.2.4, through environment tuning, it was found that the choice of a model for the gravitational field of Mars has the highest impact on the MAV ascent. This can be expected as it is the second highest acceleration on the vehicle after its thrust. Spherical Harmonics up to a degree and order of 6 are used to ensure an orbital accuracy of 5 km and 5 m/s. The environment model that second impacts the ascent simulation the most is the atmosphere of Mars. Three models have been studied. For the required speed, a two-step exponential atmosphere model is used. However, since the climate of Mars can be predicted up to some extent, it is recommended to use the Mars Climate Database to get the atmospheric density as a function of 3D position and time, when the exact lift-off time is known, to re-run the ascent simulation. This recommendation is made again in the sensitivity analysis run in Section 4.5, as both the atmospheric density and the possible winds have been found to greatly impact the final orbit reached.

Sub-question 2.4: *How do deviations in the initial state affect the final propagated state and final orbit?*
Answer: If initial deviations are present in the altitude, latitude, or longitude of the MAV at launch, the final periapsis gets decreased by a maximum of 5 km, which leaves the orbit high enough for the

ERO to safely rendezvous. Similarly, a delay in stage separation results in the second stage burn starting after apogee, and increases the periapsis altitude, at the expense of a decrease in apoapsis altitude. These effects are discussed in more length in Section 4.5.2.

Sub-question 2.5: *How stable is the final orbit of the second stage at the minimum target altitude?*
Answer: The final orbit of the second stage starts at a periapsis of 314 km and an apoapsis of 339 km. It decays below the limit of 300 km after 84 days, after which it is no longer safe for the ERO to rendezvous. Also, after the initial orbital injection, there are 5 more times at which the orbital inclination repeats itself to the rendezvous one, with the last repeat being 78 days after lift-off. A more extensive discussion on the final orbit stability is made in Section 4.5.1.

The combination of the answers to the research **Sub-questions 2.1 to 2.5** combine to answer the research **Sub-question 2**: *How do key simulation variations impact the overall propagated states of the Mars Ascent Vehicle and the final payload orbit?* Finally, with **Sub-question 1 and 2** answered, the **Main question** is also answered: *What is the optimum trajectory and solid propellant geometry for the Mars Ascent Vehicle to bring samples from the Martian surface to a defined Martian orbit?*

5.2. Feasibility

With this research done, the feasibility of using a two-stage solid propellant rocket bring samples from the surface of Mars to orbit can be assessed. Given the effort that was put in making the propellant burn and ascent simulations accurate, it is judged that numerical results can be used to extrapolate on the real-world feasibility of the MAV. This section also reflects on whether the numerical results would lead to a feasible design and mission for the MAV in real-world conditions.

First of all, the required MAV size and mass that were decided by JPL can be fulfilled by a fair margin of 57 kg. This means that the research discussed in this report also served as verification of the requirements from JPL. With the 57 kg that are not taken by the MAV in the configuration found in this research, additional scientific payload or engineering equipment such as extra helicopters could be added to the MAV lander. This also allows for some growth in the MAV design itself, which is always welcome in the design of systems such as a rocket.

Then, the target altitude range of 300 km to 375 km also appears to be realistic. On one hand, the MAV can reach this altitude range within the required MAV size and mass. On the other hand, this orbital altitude range leaves 78 days for the ERO to catch the samples in the orbit found to be optimum before crossing the 300 km altitude, with 6 different opportunities at which the orbital inclination of the orbiting samples repeat itself.

This research also showcases the usefulness of the VECTOR launch system, that allows the MAV to be launched from a initial angle between 30 deg and 60 deg. This launch angle range allows to control in what target orbit the MAV is going. Moreover, it was found that the full range of motion of the VECTOR launch system is not required, since launching below 47.5 deg from the vertical does not allow for the MAV to reach orbit with a low mass.

From the sensitivity analysis that was run, it was found that the final orbit that is reached by the MAV is mostly stable. With most unplanned variation in the initial state, environment, and vehicle design, the periapsis altitude decreases between 5 km to 10 km, while the apoapsis increases by close to the same distance. This means that it is possible to correct the ascent trough the modification of the launch, TVC, and stage separation angles, to counteract deviations that were not taken into account in the simulations run in this research. As such, it is recommended to study the addition of a weather balloon to be released from the MAV launch platform, to get the most accurate information about the atmosphere before the launch, and adapt the trajectory if needed.

Last but not least, the use of a two-stage rocket, that uses solid rocket motors is confirmed to allow for the target orbit to be reached. While a hybrid rocket would offer thrust control during the ascent, it was shown that the thrust profile of both stages can individually be tuned beforehand, by editing the cross-sectional geometry of the solid propellant.

<div align="right">6</div>

Recommendations

This section compiles all the recommendations that are made, would this research be pursued further. The recommendations are grouped into two categories: possible improvements to the setup used, and investigations that could be made based on where this research ended. While these recommendations are made specifically for this research, they are also applicable to similar research that are still to begin.

6.1. Setup improvements

From the experience gained in this research, various recommendation are made to make the setup used more efficient computationally, but also at finding optimum results. From the optimisation, it was first found that, after running it for 75 generations, the entire population was using a very similar set of design variables. From this lack of variation, the remainder of the optimisation could not continue to find better results. To save CPU time, and potentially allow for more random seeds to be tried, it is then recommended to use 75 generations instead of 100.

Then, it is recommended to implement different sets of design variables. For instance, more design variables could be used, such as the throat radius of both stages, or the stage separation altitude or time, instead of always separating exactly at apogee. At the opposite, less design variables could also be used, removing for instance the TVC, keeping it only for control. Some of the design variables could also be fixed to some values, such as part of the SRM geometry. Using more design variables allow for more fine tuning from the optimiser, but using less design variables also allows to run a more efficient optimisation, since the effect that variables have on the objectives become clearer. There is thus a fine balance to be found

Lastly, more objectives could be added to the optimisation, such as minimising the acceleration, the burn time, the chamber pressure, or the dynamic pressure at first stage burnout to keep RCS authority. However, it may be needed to run separate optimisation for some objectives, as combining all of them in a 7 objective optimisation may simply results in design variables being tried at pseudo-random, and no objective convergence to be attainable.

6.2. Investigations

Different elements can be investigated, starting from the codebase that was generated for this research. First of all, a database of 634 thousand solutions was generated for this research, with 18 thousand of them allowing the MAV to reach orbit within the required altitude and mass requirements. In this database, each solution contains the design variables used to run a simulation, the MAV launch mass, the final periapsis and apoapsis altitude, the final orbital inclination, and the burn times of both SRM. Thanks to the size of this dataset, Machine Learning or a similar technique can be used to fit a model to the data, and then use the model to predict the outcome of a new untried solution. This would allow to run optimisations at a higher speed, and the run the actual simulation for selected ones, to see if the outcome is the same as the one that was predicted.

Then, while the ascent model that was setup propagates three degrees of freedom, it can be made more complex by propagating 6 degrees of freedom, adding the simulation of the rotational dynamics of the vehicle. This would allow to implement TVC and RCS more accurately, and to study the orbital decay including effects such as torque due to the imperfect gravitational field of Mars. This would also allow to push further the study of the achievable numerical accuracy.

Finally, the addition of a model spin and de-spin thrusters can be added, which make the second stage of the MAV spin on itself and stabilises it during the second stage burn. The inclusion of this model would allow to study the feasibility of rotating the vehicle around an axis that is at an angle with its centerline. This would reduce the effectiveness of the second stage burn, which could be used to tune the second stage thrust profile and reduce orbital dispersion.

Bibliography

[1] Yaghoubi, D. and Schnell, A. Mars Ascent Vehicle Solid Propulsion Configuration. In *2020 IEEE Aerospace Conference*, pages 1–11. IEEE, 2020. `doi:10.1109/AERO47225.2020.9172654`.

[2] Dassault Systèmes. Design Engineering, CATIA, V5R21, 2011. `https://www.3ds.com/products-services/catia` (accessed 23 August 2021).

[3] Muirhead, B.K. and Nicholas, A.K. and Umland, J. and Sutherland, O. and Vijendran, S. Mars Sample Return Campaign Concept Status. *Acta Astronautica*, 176:131–138, 2020. `doi:10.1016/j.actaastro.2020.06.026`.

[4] National Aeronautics and Space Administration, Zurbuchen, T. H. Summary of NASA Responses to Mars Sample Return Independent Review Board Recommendations, 10 November 2020. `https://www.nasa.gov/sites/default/files/atoms/files/nasa_esa_mars_sample_return_final_report_small.pdf` (accessed 18 June 2022).

[5] Jet Propulsion Laboratory. NASA Will Inspire World When It Returns Mars Samples to Earth in 2033, 27 July 2022. `https://www.jpl.nasa.gov/news/nasa-will-inspire-world-when-it-returns-mars-samples-to-earth-in-2033` (accessed 27 July 2022).

[6] Farley, K.A. and Williford, K.H. and Stack, K.M. and Bhartia, R. and Chen, S. and de la Torre, M. and Hand, K. and Goreva, Y. and Herd, C.D.K. and Hueso, R. et al. Mars 2020 Mission Overview. *Space Science Reviews*, 216:1–41, 2020. `doi:10.1007/s11214-020-00762-y`.

[7] NASA/JPL-Caltech. Mars 2020 Sample Tube 3D print files, March 2020. `https://nasa3d.arc.nasa.gov/detail/Mars-2020-Sample-Tube-3D-print-files` (accessed 30 May 2021).

[8] Clery, D. for Science. Scrapping original plan, Mars mission turns to existing rover to help bring samples home, 27 July 2022. `doi:10.1126/science.ade1501`.

[9] Merlo, A. and Larranaga, J. and Falkner, P. Sample Fetching Rover (SFR) for MSR. In *Proceedings of the 12th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA 2013)*, 2013.

[10] Sutherland, O. and Vijendran, S. and Huesing, J. and Geelen, K. and Feili, D. and Sanchez-Perez, J.M. and Nicholas, A. and Lock, R. Mars Sample Return-Earth Return Orbiter: ESA's next Interplanetary Electric Propulsion Mission Concept. In *36th International Electric Propulsion Conference, Vienna, Austria*, 2019.

[11] Younse, P. and Chiu, C.Y. and Cameron, J. and Dolci, M. and Elliot, E. and Ishigo, A. and Kogan, D. and Marteau, E. and Mayo, J. and Munger, J. and Ohta, P. and Strahle, J.W. and Willson, R. and Olds, E. and Malyan, V. Concept for an on-orbit capture and orient module for potential mars sample return. In *2020 IEEE Aerospace Conference*, pages 1–22, 2020. `doi:10.1109/AERO47225.2020.9172814`.

[12] Corliss, J., NASA Langley Research Center. Mars Sample Return and Earth Entry Vehicle, June 25, 2020. 17th International Planetary Probe Workshop (IPPW 2020), `https://ntrs.nasa.gov/citations/20205003699` (accessed 3 June 2021).

[13] Kminek, G. and Conley, C. and Hipkin, V. and Yano, H. COSPAR's planetary protection policy. Space Res. Today 200, 12–25 (2017). `https://cosparhq.cnes.fr/assets/uploads/2019/12/PPPolicyDecember-2017.pdf` (accessed 5 June 2021).

[14] Jackman, A. and Ma, P. and Bishop, K. and Frame, K. Mars Ascent Vehicle-Payload?, Spacecraft?, Launch Vehicle? - A Systems Approach to MAV. *International Astronautical Congress*,

October 2019. `https://ntrs.nasa.gov/citations/20190032302` (accessed 1 August 2021).

[15] Prince, A.S. and Kibbey, T.P. and Schnell, A. and Yaghoubi, D. and Karp, A.C. and Nakazono, B. A Design for a Two Stage Solid Mars Ascent Vehicle. In *AIAA Propulsion and Energy 2019 Forum*, page 4149, 2019. `doi:10.2514/6.2019-4149`.

[16] McCollum, L.T. and Schnell, A. and Yaghoubi, D. and Bean, Q. and McCauley, R. and Prince, A. Development Concepts for Mars Ascent Vehicle (MAV) Solid and Hybrid Vehicle Systems. In *2019 IEEE Aerospace Conference*, pages 1–10. IEEE, 2019. `doi:10.1109/AERO.2019.8741965`.

[17] Yaghoubi, D. Returning Geological Samples from Mars. *NASA Marshall Space Flight Center*, March 2019. `https://ntrs.nasa.gov/citations/20190002128` (accessed 3 August 2021).

[18] Yaghoubi, D. Emerging Technologies for Mars Exploration: Mars Ascent Vehicle. *AIAA Next Gen Technical Symposium*, September 2020. `https://ntrs.nasa.gov/citations/20205006215` (accessed 3 August 2021).

[19] Bean, Q. and McCollum, L.T. and McCauley, R. and Prince, A. and Yaghoubi, D. and Schnell, A. Mars Ascent Vehicle (MAV) Propulsion Subsystems Design. In *2019 IEEE Aerospace Conference*, pages 1–9, 2019. `doi:10.1109/AERO.2019.8742205`.

[20] Dankanich, J. and Rousseau, J. and Williams, J. Mars Ascent Vehicle-Propellant Aging. *George C. Marshall Space Flight Center Research and Technology Report 2014*, 2015. `https://ntrs.nasa.gov/citations/20160008023` (accessed 4 May 2022).

[21] Yaghoubi, D. and Ma, P. Integrated Design Results for the MSR DAC-0.0 Mars Ascent Vehicle. In *2021 IEEE Aerospace Conference (50100)*, pages 1–17. IEEE, 2021. `doi:10.1109/AERO50100.2021.9438481`.

[22] Petrović, S.D. Mars Sample Return Ascent Trajectory Analysis using Taylor Series Integration. *MSc thesis, Delft University of Technology*, March 2017. supervised by Dr. Ir. E. Mooij and Ir. R. Noomen.

[23] Desai, P. and Braun, R. and Engelund, W. and Cheatwood, F. and Kangas, J. Mars Ascent Vehicle Flight Analysis. In *7th AIAA/ASME Joint Thermophysics and Heat Transfer Conference*, page 2850, 1998. `doi:10.2514/6.1998-2850`.

[24] Whitehead, J. Mars Ascent Propulsion Trades with Trajectory Analysis. In *40th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, page 4069, 2004. `doi:10.2514/6.2004-4069`.

[25] Di Sotto, E. and Bastante, J. C. and Drai, R. System and GNC concept for RendezVous into elliptical Orbit for Mars Sample Return mission. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, page 6852, 2007. `doi:10.2514/6.2007-6852`.

[26] Benito, J. and Johnson, B. Trajectory optimization for a Mars ascent vehicle. In *AIAA/AAS Astrodynamics Specialist Conference*, page 5441, 2016. `doi:10.2514/6.2016-5441`.

[27] Hetterich, R. Using Advanced Design Methods to Optimize the Mars Ascent Vehicle First Stage Solid Rocket Motor. In *AIAA NextGen Symposium*, September 2019. M19-7616, `https://ntrs.nasa.gov/citations/20190032381` (accessed 17 July 2021).

[28] Hoak, D.E. and Anderson, R. and Goss, C.R. The USAF stability and control DATCOM, Air Force Wright Aeronautical Laboratories. *Wright Patterson Air Force Base, Ohio*, 1978.

[29] Dr. Williams, D.R. NASA Goddard Space Flight Center. Mars Fact Sheet, November 2020. `https://nssdc.gsfc.nasa.gov/planetary/factsheet/marsfact.html` (accessed 6 June 2021).

[30] Air Force, United States. *US standard atmosphere, 1976*, volume 76-1562. National Oceanic

and Atmospheric Administration, 1976.

[31] Kennard, E. H. *Kinetic theory of gases*, volume 483. McGraw-hill New York, 1938. `doi:10.1038/142494a0`.

[32] Boyd, I. D. and Schwartzentruber, T. E. *Nonequilibrium Gas Dynamics and Molecular Simulation*. Cambridge University Press, Cambridge Aerospace Series, April 2017. `doi:10.1017/9781139683494`.

[33] Koura, K. and Matsumoto, H. Variable soft sphere molecular model for air species. *Physics of Fluids A: Fluid Dynamics*, 4(5):1083–1085, 1992. `doi:10.1063/1.858262`.

[34] Parodi, P. Analysis and Simulation of an Intake for Air-Breathing Electric Propulsion Systems. *MSc Thesis, Università di Pisa*, July 2019. `doi:10.13140/RG.2.2.12966.34889`.

[35] Bird, G. A. *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. Molecular Gas Dynamics and the Direct Simulation of Gas Flows. Clarendon Press, 1994. ISBN 9780198561958.

[36] Pilinski, M. D. and Argrow, B. M. and Palo, S. E. Semiempirical Model for Satellite Energy-Accommodation Coefficients. *Journal of Spacecraft and Rockets*, 47(6):951–956, 2010. `doi:10.2514/1.49330`.

[37] Karniadakis, G. and Beskok, A. and Aluru, N. *Microflows and nanoflows: fundamentals and simulation*, volume 29. Springer Science & Business Media, 2006. `doi:10.1007/0-387-28676-4`.

[38] Plimpton, S. J. and Moore, S. G. and Borner, A. and Stagg, A. K. and Koehler, T. P. and Torczynski, J. R. and Gallis, M. A. Direct simulation Monte Carlo on petaflop supercomputers and beyond. *Physics of Fluids*, 31(8):086101, 2019. `doi:10.1063/1.5108534`.

[39] González-Galindo, F. and López-Valverde, M. A. and Forget, F. and García-Comas, M. and Millour, E. and Montabone, L. Variability of the Martian thermosphere during eight Martian years as simulated by a ground-to-exosphere global circulation model. *Journal of Geophysical Research: Planets*, 120(11):2020–2035, 2015. `doi:10.1002/2015JE004925`.

[40] Zandbergen, B.T.C. Thermal Rocket Propulsion. *Delft University of Technology, Faculty of Aerospace Engineering*, 2.08, August 2020.

[41] Acton, C. and Bachman, N. and Semenov, B. and Wright, E. A look towards the future in the handling of space science mission geometry. *Planetary and Space Science*, 150:9–12, 2018. `doi:10.1016/j.pss.2017.02.013`.

[42] Acton, C.H. Ancillary data services of NASA's Navigation and Ancillary Information Facility. *Planetary and Space Science*, 44(1):65–70, 1996. `doi:10.1016/0032-0633(95)00107-7`.

[43] Dirkx, D. and Mooij, E. Acceleration Modelling, AE4866, Propagation and Optimization. Astrodynamics and Space Missions, Delft University of Technology, 24 February 2020.

[44] Wertz, J.R. and Everett, D.F. and Puschell, J.J. *Space mission engineering: the new SMAD*. Space technology library, Microcosm Press, 2011. EAN 9781881883159.

[45] Gallais, P. *Models of Atmosphere*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. `doi:10.1007/978-3-540-73647-9_3`.

[46] Forget, F. and Hourdin, F. and Fournier, R. and Hourdin, C. and Talagrand, O. and Collins, M. and Lewis, S.R. and Read, P.L. and Huot, J. Improved general circulation models of the Martian atmosphere from the surface to above 80 km. *Journal of Geophysical Research: Planets*, 104(E10):24155–24176, October 1999. `doi:10.1029/1999JE001025`.

[47] Millour, E. and Forget, F. and Lopez-Valverde, M. and Lefevre, F. and Gonzalez-Galindo, F. and Lewis, S. and Chaufray, J. and Montabone, L. and Zakharov, V. and Spiga, A. and et al. Exploring the interannual variablitity of the Martian atmosphere with the Mars Climate Database v5. 3. *42nd COSPAR Scientific Assembly*, 42:C4–3, 2018.

[48] Spiga, A. and LMD/OU/IAA/ESA/CNES. Mars Climate Database: The Web Interface, v5.3. `http://www-mars.lmd.jussieu.fr/mcd_python` (accessed 6 June 2021).

[49] Delft University of Technology. Tabulated Atmosphere Settings - TU Delft Astrodynamic Toolbox documentation, 2018. `https://tudat.tudelft.nl/tutorials/tudatFeatures/environmentSetup/tabulatedAtmosphere.html` (accessed 6 June 2021).

[50] Justus, C.G. and Johnson, D.L. Mars Global Reference Atmospheric Model 2001 Version (Mars-GRAM 2001): Users Guide. *Marshall Space Flight Center Engineering Directorate*, April 2001. `https://ntrs.nasa.gov/citations/20010056680` (accessed 22 August 2021).

[51] Mooij, E. AE4870B Re-entry Systems: Lecture Notes. *Astrodynamics and Space Missions, Delft University of Technology*, 2019-2020.

[52] Wertz, J.R. *Orbit & Constellation Design & Management*. Microcosm Press and Springer, ISBN:978-0-7923-7148-9, 2001.

[53] Takahashi, Y. and Scheeres, D.J. Small body surface gravity fields via spherical harmonic expansions. *Celestial Mechanics and Dynamical Astronomy*, 119(2):169–206, 2014. `doi:10.1007/s10569-014-9552-9`.

[54] Wieczorek, M. and Meschede, M. SHTools: Tools for Working with Spherical Harmonics. *Geochemistry, Geophysics, Geosystems*, 19, 05 2018. `doi:10.1029/2018GC007529`.

[55] Dirkx, D. and Cowan, K. Acceleration Models, Formulation, Introduction and Non-Gravitational, AE4868 Numerical Astrodynamics. Lecture notes, Delft University of Technology, 2020.

[56] Montenbruck, O. and Gill, E. *Satellite orbits: models, methods and applications*. Springer Science & Business Media, 2012. `https://www.researchgate.net/publication/312122488_Satellite_Orbits_Models_Methods_and_Applications` (accessed 12 December 2021).

[57] G. Petit and B. Luzum. *IERS conventions (2010)*, volume 36. (IERS Technical Note 36) Frankfurt am Main: Verlag des Bundesamts für Kartographie und Geodäsie, ISBN:3-89888-989-6, January 2010.

[58] Sutton, G.P. and Biblarz, O. *Rocket propulsion elements*. John Wiley & Sons, 2001. ISBN 0-471-32642-9.

[59] Willer, Rodney L and Chi, Minn-Shong and Gleeson, Robert and Hill, John C. Diaminoglyoxime and diaminofurazan in propellants based on ammonium perchlorate, December 1991. US Patent 5,071,495.

[60] Terzic, J. and Zecevic, B. and Baskarad, M. and Catovic, A. and Serdarevic-Kadic, S. Prediction of internal ballistic parameters of solid propellant rocket motors. *Problemy Mechatroniki: uzbrojenie, lotnictwo, inżynieria bezpieczeństwa*, 2:7–26, 2011.

[61] Northrop Grumman. Propulsion Products Catalog, April 2016. `https://www.northropgrumman.com/wp-content/uploads/NG-Propulsion-Products-Catalog.pdf` (accessed 29 August 2021).

[62] NASA Glenn Research Center. Chemical Equilibrium Applications (CEA), Reference Number LEW-17687-1. `https://software.nasa.gov/software/LEW-17687-1` (accessed 19 March 2022).

[63] RP Software+Engineering UG, Germany. Rocket Propulsion Analysis Combusion (RPA-C), 2022, v2.x. `https://www.rocket-propulsion.com/RPA-C/features.htm` (accessed 19 March 2022).

[64] Schafer, J.I. Solid Rocket Propulsion. *Space Technology, John Wiley and Sons, New York, NY*, 1969.

[65] Umbel, M.R. An exact geometric analysis of the generalized anchor grain configuration. In *44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, page 4697, 2008.

`doi:10.2514/6.2008-4697`.

[66] Wittenberg, H. and Ambrosius, B.A.C. and Naeije, M.C. and Mooij, E. *Rocket Motion Lecture Notes AE4-870A*. Faculty of Aerospace Engineering, Delft University of Technology, Course Reader, September 2014.

[67] Delft University of Technology. Tudat Space: Documentation, 2020. `https://tudat-space.readthedocs.io/en/latest/index.html` (accessed 17 June 2021).

[68] Tudat-team/tudatpy, Delft University of Technology. A Python platform to perform astrodynamics and space research, 2021. `https://github.com/tudat-team/tudatpy` (accessed 17 June 2021).

[69] Delft University of Technology. Tudat Space: Simulation, 2020. `https://tudat-space.readthedocs.io/en/latest/_src_api/simulation.html` (accessed 17 June 2021).

[70] Delft University of Technology. Tudat Space: Translational Dynamics, 2020. `https://tudat-space.readthedocs.io/en/latest/_src_api/propagation_setup/dynamics_types/translational.html` (accessed 17 June 2021).

[71] Delft University of Technology. Tudat Space: Rotational Dynamics, 2020. `https://tudat-space.readthedocs.io/en/latest/_src_api/propagation_setup/dynamics_types/rotational.html` (accessed 17 June 2021).

[72] Urrutxua Cereijo, H. *High fidelity models for near-earth object dynamics*. PhD thesis, Universidad Politécnica de Madrid, April 2015.

[73] Pollock, K.R. An Analysis Of Orbital Propagators for Low Earth Orbit Rendezvous. Technical report, Naval Postgraduate School Monterey CA, 1994.

[74] Dirkx, D. and Mooij, E. Formulation of Equations of Motion, AE4866, Propagation and Optimization. Astrodynamics and Space Missions, Delft University of Technology, 13 February 2019.

[75] Rodríguez, J.L.C. Study of analytical solutions for low-thrust trajectories. Technical report, Escuela Técnica Superior de Ingenieros Aeronáuticos Universidad Politécnica de Madrid, March 2017. `https://github.com/astrojuanlu/pfc-uc3m` (accessed 20 June 2021).

[76] G.R. Hintz. Survey of orbit element sets. *Journal of guidance, control, and dynamics*, 31(3):785–790, 2008. `doi:10.2514/1.32237`.

[77] Vittaldev, V. and Mooij, E. and Naeije, M. Unified state model theory and application in astrodynamics. *Celestial Mechanics and Dynamical Astronomy*, 112(3):253–282, 2012. `doi:10.1007/s10569-011-9396-5`.

[78] Hahn, G.D. A modified Euler method for dynamic analyses. *International Journal for Numerical Methods in Engineering*, 32(5):943–955, October 1991. `doi:10.1002/nme.1620320502`.

[79] Noomen, R. Space Mission Design: Integrators. Mission Geometry and Orbit Design course lecture, Delft University of Technology, Faculty of Aerospace Engineering, V4.13, June 2020.

[80] Butcher, J.C. Coefficients for the study of Runge-Kutta integration processes. *Journal of the Australian Mathematical Society*, 3(2):185–201, 1963. `doi:10.1017/S1446788700027932`.

[81] Fehlberg, E. *Low-order classical Runge-Kutta formulas with stepsize control and their application to some heat transfer problems*, volume 315. NASA Marshall Space Flight Center, July 1969. `doi:10.1007/BF02241732`.

[82] Dormand, J.R. and Prince, P.J. A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26, March 1980. `doi:10.1016/0771-050X(80)90013-3`.

[83] Biscani, F. and Izzo, D. A parallel global multiobjective framework for optimization: pagmo. *Journal of Open Source Software*, 5(53):2338, September 2020. `doi:10.21105/joss.02338`.

[84] Noël, L. and Langelaar, M. and van Keulen, F. Engineering Optimization Concepts and Applica-

tions. Lecture ME46060, Precision and Microsystems Engineering, Delft University of Technology, 2020.

[85] Papalambros, P.Y. and Wilde, D.J. *Principles of optimal design: modeling and computation*. Cambridge university press, May 2017. ISBN:9781107132672.

[86] Dirkx, D. and Mooij, E. Optimisation Basics, AE4866, Propagation and Optimization. Astrodynamics and Space Missions, Delft University of Technology, 2020.

[87] Hamby, D.M. A comparison of sensitivity analysis techniques. *Health physics*, 68(2):195–204, 1995. `doi:10.1097/00004032-199502000-00005`.

[88] Sobol, I.M. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*, 55(1):271–280, 2001. The Second IMACS Seminar on Monte Carlo Methods. `doi:10.1016/S0378-4754(00)00270-6`.

[89] Pagmo development team. Capabilities, Pygmo 2.18.0 documentation, 2021. `https://esa.github.io/pygmo2/overview.html` (accessed 22 August 2021).

[90] Deb, K. and Pratap, A. and Agarwal, S. and Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002. `doi:10.1109/4235.996017`.

[91] Zhang, Q. and Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731, 2007. `doi:10.1109/TEVC.2007.892759`.

[92] Acciarini, G. and Izzo, D. and Mooij, E. MHACO: a Multi-Objective Hypervolume-Based Ant Colony Optimizer for Space Trajectory Optimization. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2020. `doi:10.1109/CEC48606.2020.9185694`.

[93] Li, X. A non-dominated sorting particle swarm optimizer for multiobjective optimization. In *Genetic and evolutionary computation conference*, pages 37–48. Springer, 2003. `doi:10.1007/3-540-45105-6_4`.

[94] Mahdavi, M. and Fesanghary, M. and Damangir, E. An improved harmony search algorithm for solving optimization problems. *Applied mathematics and computation*, 188(2):1567–1579, 2007. `doi:10.1016/j.amc.2006.11.033`.

[95] Kumar, M. and Nallagownden, P. and Elamvazuthi, I. Advanced Pareto Front Non-Dominated Sorting Multi-Objective Particle Swarm Optimization for Optimal Placement and Sizing of Distributed Generation. *Energies*, 9:982, 11 2016. `doi:10.3390/en9120982`.

[96] Ahrens, J. and Geveci, B. and Law, C. Paraview: An end-user tool for large data visualization. *The visualization handbook*, 717(8), 2005. `doi:10.1016/B978-012387582-2/50038-1`.

[97] Reilley, A. and Sojourner, A. and Quigley, J. and Garcia, C. and Lengkeek, P. and Rivera, S. and Coman, B. and Heisey, A. Open Motor, v0.4.0, 2019. `https://github.com/reilleya/openMotor` (accessed 29 August 2021).

[98] ESA/DLR/FU Berlin, CC BY-SA 3.0 IGO. New topographic map of Jezero Crater - Mars 2020's future home, 2 October 2020. `https://www.esa.int/ESA_Multimedia/Images/2020/09/New_topographic_map_of_Jezero_Crater_Mars_2020_s_future_home` (accessed 13 July 2021).

A
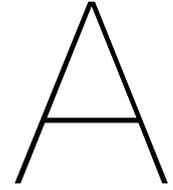
Appendices

This appendix compiles the different resources and results that are secondary in this research. First, Appendix A.1 presents the design variables that were manually tuned to constitute baseline for the design space exploration. Then, Appendix A.2 compiles the plots generated to support the generation of the simulation benchmark, that were not shown in the body of the report, since they are secondary. Appendices A.3 and A.4 then show the secondary figures generated during the tuning of the integrator and environment respectively. Follows Appendix A.5 with the results from all the optimisations that were run, for all first stage SRM types. Appendix A.6 then shows the secondary plots made from the validation. Lastly, Appendix A.7 shows all the figures related to the sensitivity analysis.

A.1. Baseline design variables

In Section 4.2.2, the baseline design variables are given in case a multi-fin SRM is used for the first stage, which results in an orbit with a periapsis at 300 km and an apoapsis at 505 km. The launch, separation, and TVC angles have been taken identical as the ones shown in Table 4.5 for all first stage SRM types.

In case a tubular SRM is taken for the first stage, the baseline design variables have been manually tuned to the values of Table A.1. This baseline results in a MAV GLOM of 445 kg, 101 kg of which being the second stage, and leaves the MAV in an orbit with a periapsis of 453 km and an apoapsis of 464 km. When using the rod and tube type for the first stage SRM, the manually tuned design variables are the ones shown in Table A.2. Running the simulations with these variables result in a final orbit with a periapsis altitude of 414 km and an apoapsis altitude of 513 km. In this case, the MAV GLOM is 486 kg, and the second stage mass is 109 kg. Lastly, in case an anchor SRM is used for the first stage, the baseline design variables are the ones shown in Table A.3. This results in a MAV GLOM of 415 kg, of which 109 kg is the second stage, and the final MAV orbit has a periapsis altitude of 295 km and an apoapsis altitude of 672 km.

Design variable	Baseline value
Spherical SRM outer radius fraction $R_{o,2,\text{frac}}$ [-]	0.76
Spherical SRM inner radius fraction $R_{i,2,\text{frac}}$ [-]	0.632
First stage SRM length L [m]	1.15
First stage SRM outer radius R_o [m]	0.25
Tubular SRM inner radius fraction $R_{i,1,\text{frac}}$ [-]	0.64

Table A.1: Baseline design variables, using a tubular first stage SRM.

Design variable	Baseline value
Spherical SRM outer radius fraction $R_{o,2,\text{frac}}$ [-]	0.76
Spherical SRM inner radius fraction $R_{i,2,\text{frac}}$ [-]	0.482
First stage SRM length L [m]	1.05
First stage SRM outer radius R_o [m]	0.25
Rod and tube SRM intermediate radius fraction $R_{mid,\text{frac}}$ [-]	0.56
Rod and tube SRM inner radius fraction $R_{i,1,\text{frac}}$ [-]	0.357

Table A.2: Baseline design variables, using a rod and tube first stage SRM.

107

Design variable	Baseline value
Spherical SRM outer radius fraction $R_{o,2,\text{frac}}$ [-]	0.76
Spherical SRM inner radius fraction $R_{i,2,\text{frac}}$ [-]	0.4816
First stage SRM length L [m]	1.15
First stage SRM outer radius R_o [m]	0.26
Anchor SRM inner radius fraction $R_{i,1,\text{frac}}$ [-]	0.6346
Anchor SRM number of anchors N_a [-]	6
Anchor SRM spacing fraction w_{frac} [-]	0.7895
Anchor SRM fillet radius fraction $r_{f,\text{frac}}$ [-]	0.5
Anchor SRM spoke width fraction $\delta_{s,\text{frac}}$ [-]	0.091

Table A.3: Baseline design variables, using an anchor first stage SRM.

A.2. Simulation benchmark

Figures A.1 and A.2 show the final error in position, velocity, and mass for the benchmark ascent as a function of the number of function evaluations. The vertical green line shows the number of function evaluations above which the time step used becomes too small and the numerical rounding error accumulates. Then, Figures A.3 and A.4 show the allowable initial errors in position and velocity for the benchmark ascent of the second stage with and without thrust respectively. These allowable initial errors are studied to ensure that the final benchmark accuracy is of 50 m and 5 cm/s.
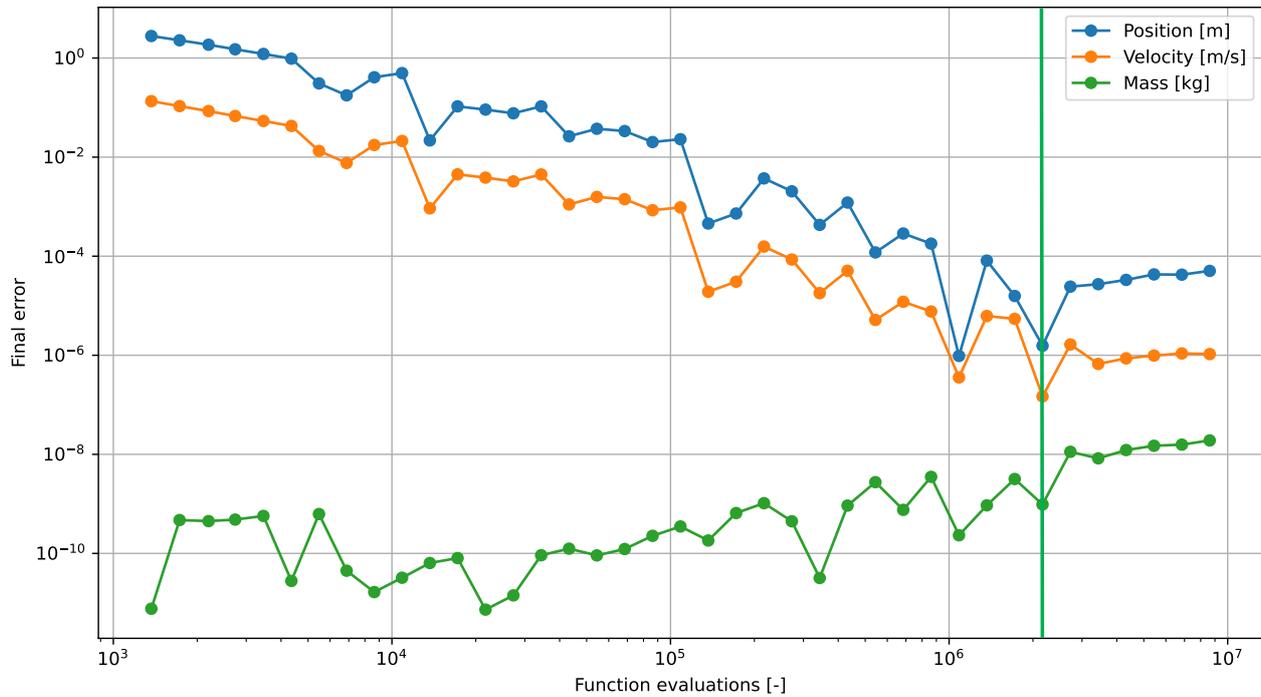
Figure A.1: Error in position, velocity, and mass versus function evaluations for the stage 2 powered ascent integration.
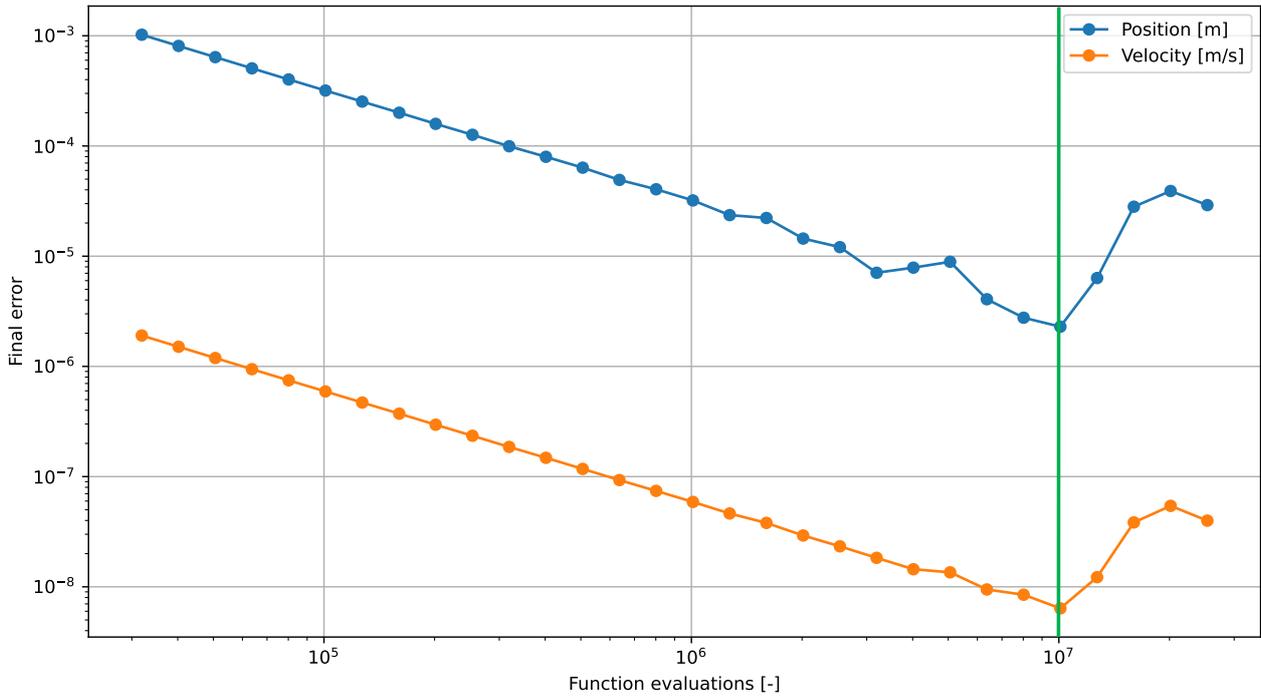
Figure A.2: Error in thrust and mass versus function evaluations for the stage 2 unpowered ascent integration.
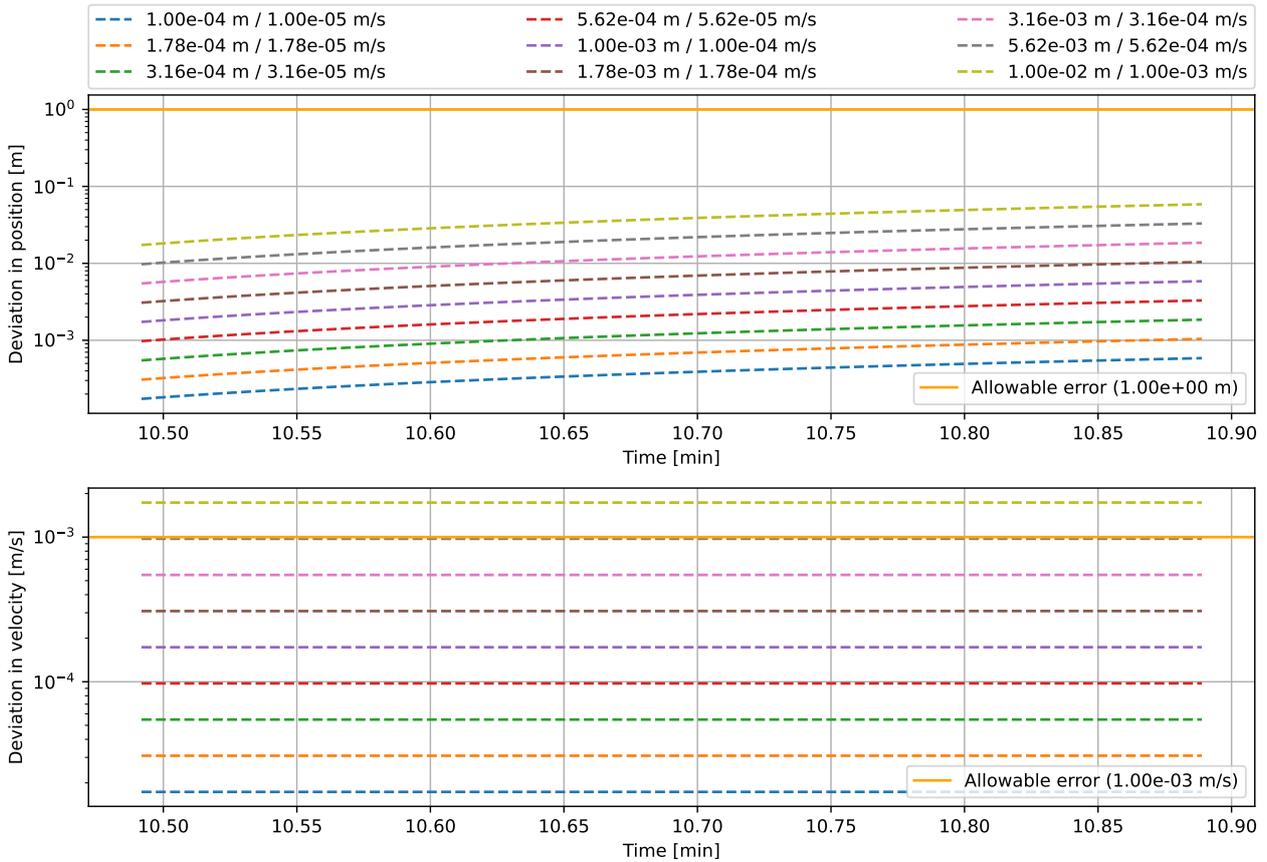
Figure A.3: Effect of variations in initial state on the powered second stage ascent.
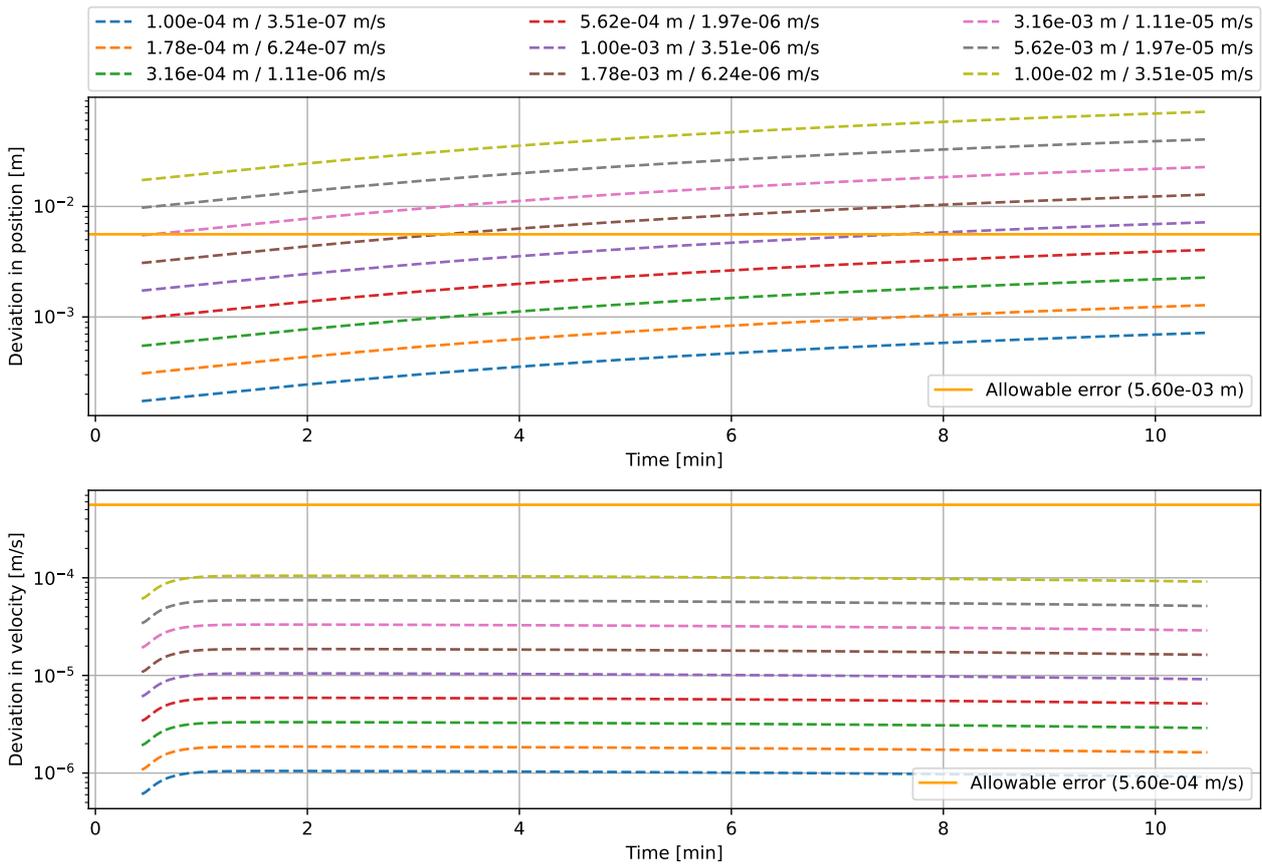
Figure A.4: Effect of variations in initial state on the unpowered first stage ascent.

A.3. Integrator tuning

This section shows the different figures used to generate Table 4.9. These figures then present the maximum allowable initial error in position and velocity for each phase of the ascent, to have an accuracy at the end of the ascent of 5 km and 5 m/s, which is the accuracy required from the integrator. Figure A.5 then presents the growing error in position and velocity for the ascent of the second stage without thrust. Figure A.6 presents the same data for the ascent of the second stage with thrust. Figures A.7 and **??** then present the growing error for the ascent of the first stage without thrust and with thrust respectively.

This section then also presents Figures A.9 and A.10. In these, the final error in velocity versus the number of function evaluations is shown for different integration scheme, used in fixed and variable step integration respectively. The horizontal orange line represents the required accuracy in velocity.
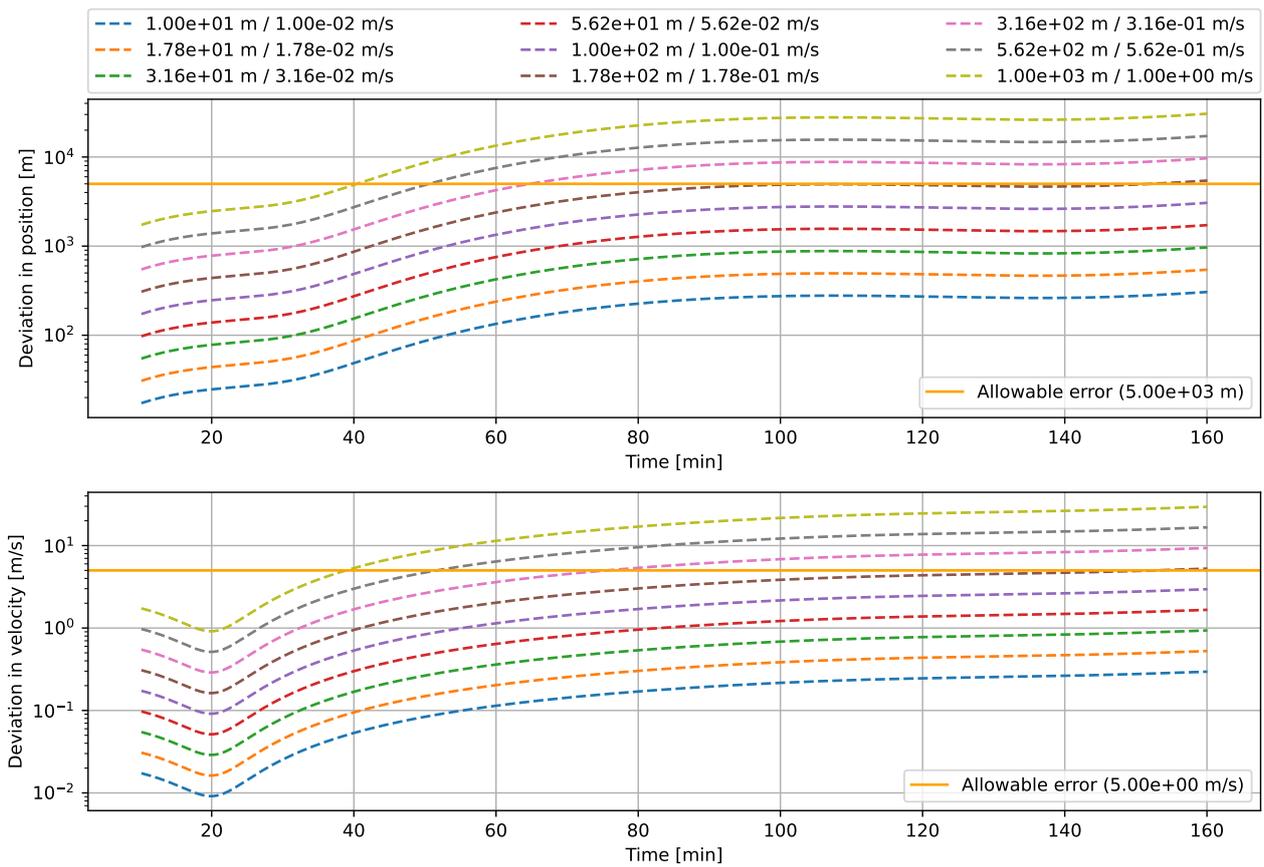
Figure A.5: Effect of variations in initial state on the powered second stage ascent.
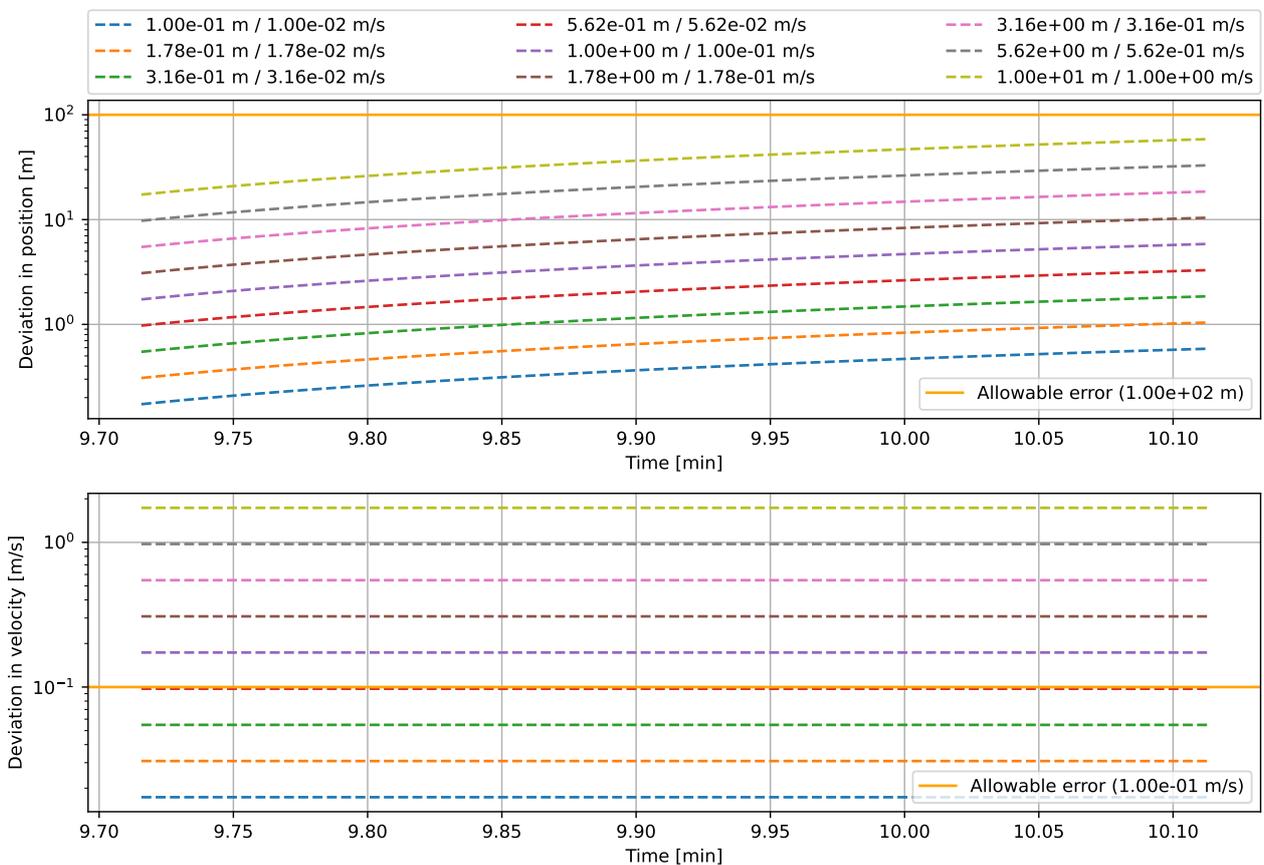
Figure A.6: Effect of variations in initial state on the powered second stage ascent.
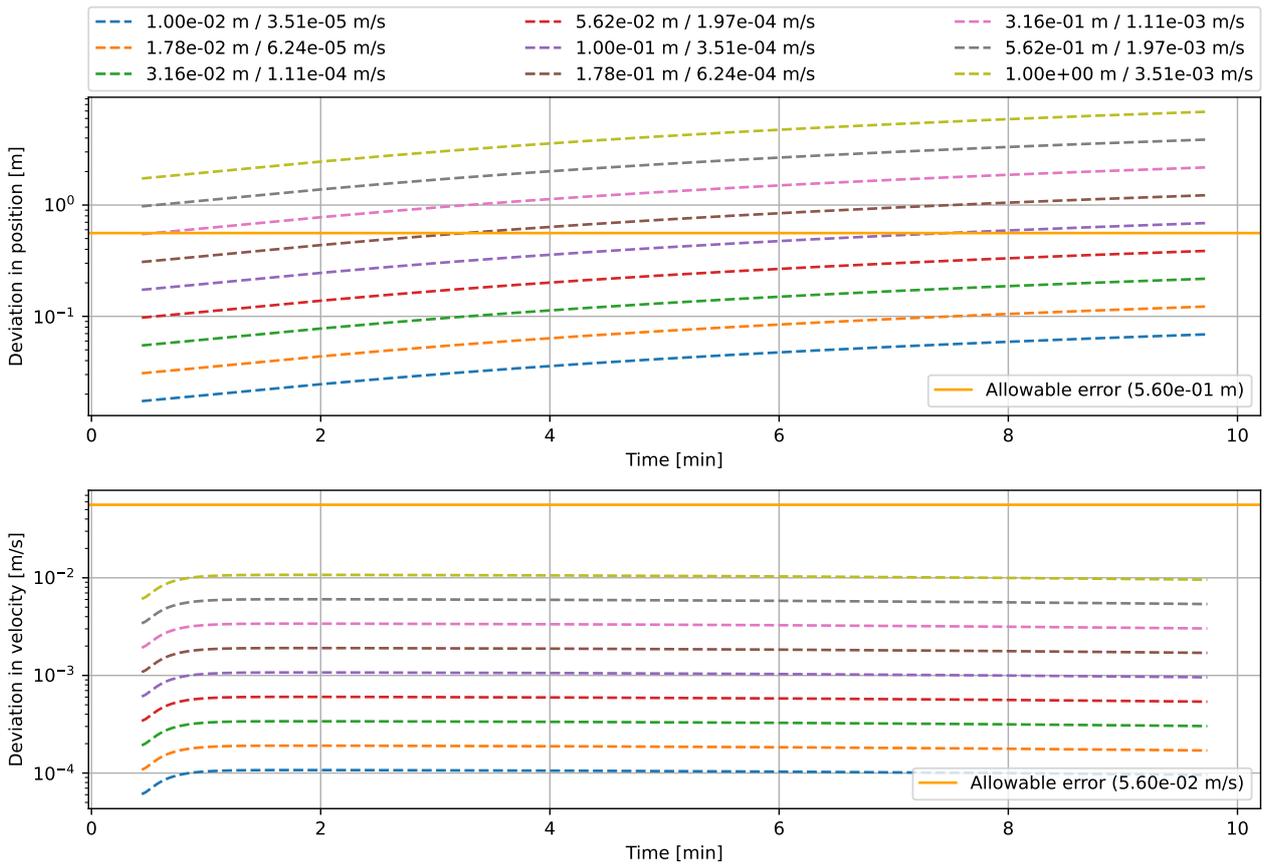
Figure A.7: Effect of variations in initial state on the powered second stage ascent.
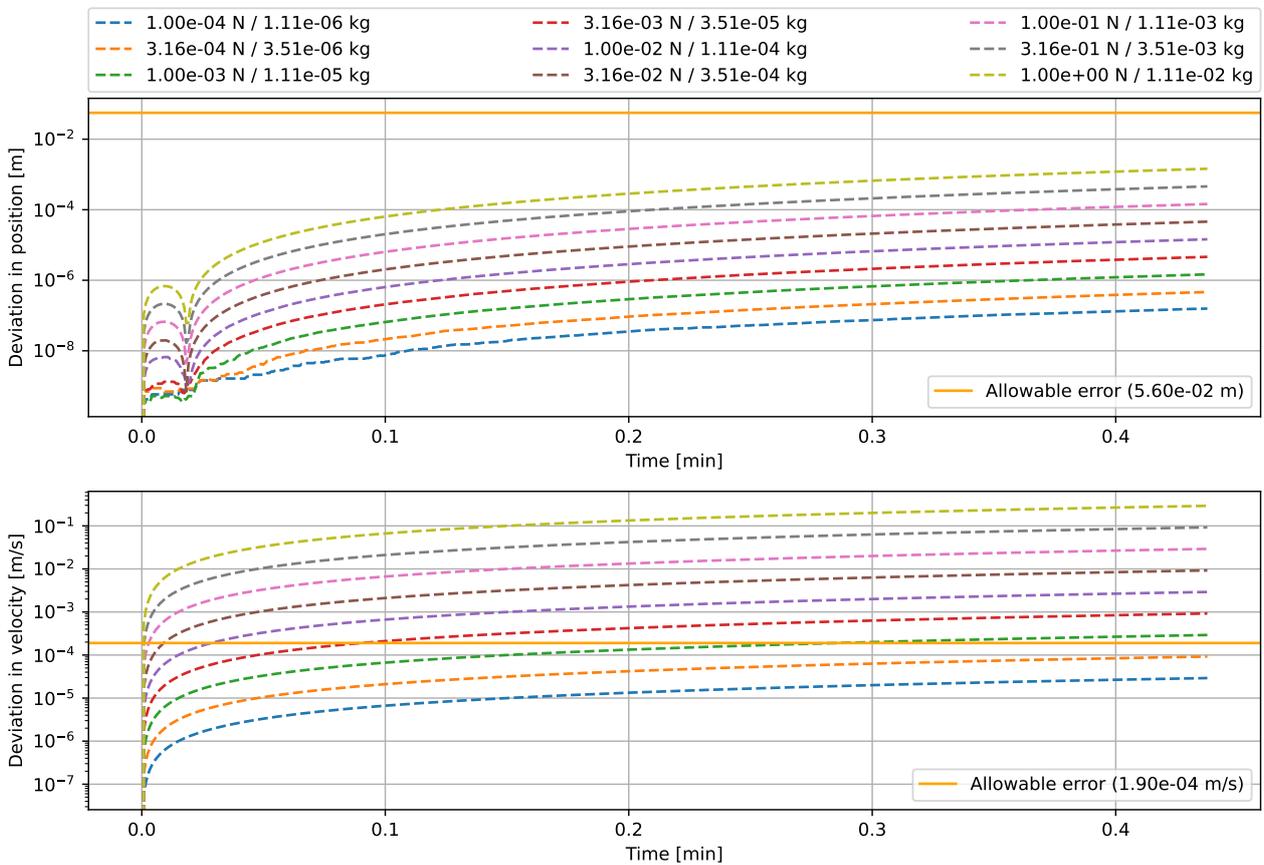
Figure A.8: Effect of variations in thrust on the powered first stage ascent.
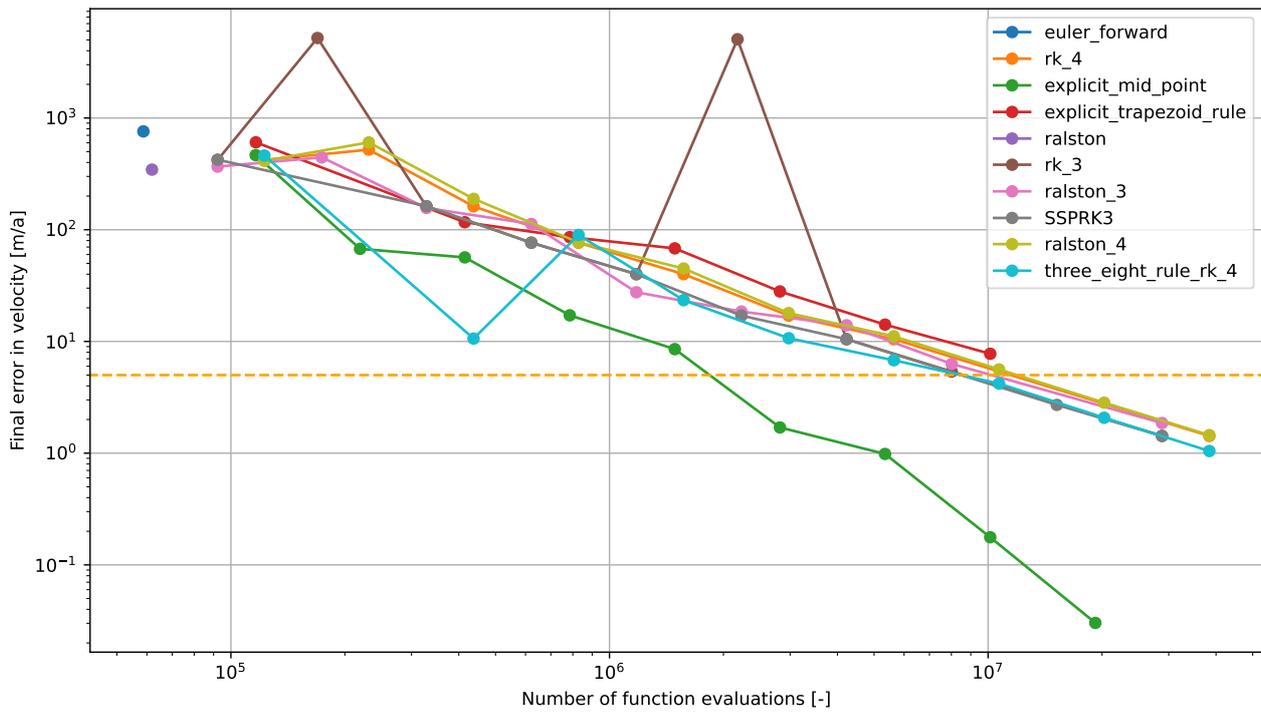
Figure A.9: Error in velocity versus number of function evaluations for various fixed step integrators.
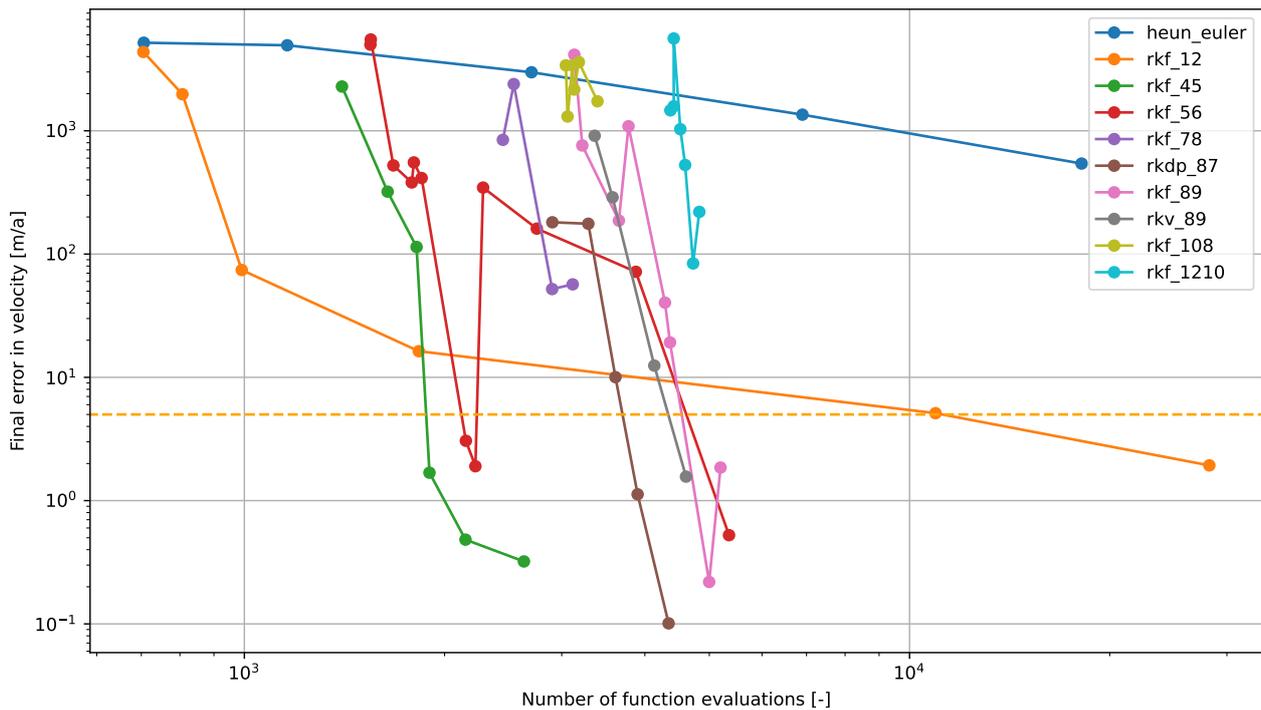
Figure A.10: Error in velocity versus number of function evaluations for various variable step integrators.

A.4. Environment tuning

This section presents all the figures generated when tuning the environment used, but each time showing the error in velocity instead of the error in position is presented in Section 4.2.4. First, the velocity history of the MAV ascent is shown in Figure A.11 for the different models used to represent the gravitational field of Mars. Figure A.12 then shows the difference in velocity when the effect of the mass different bodies is added, modelled as point masses. Follows Figure A.13 that shows the effect of adding the Schwarzschild relativistic correction and the Solar radiation pressure on the velocity history. Lastly, Figure A.14 shows the velocity history for different models for the Martian atmosphere.
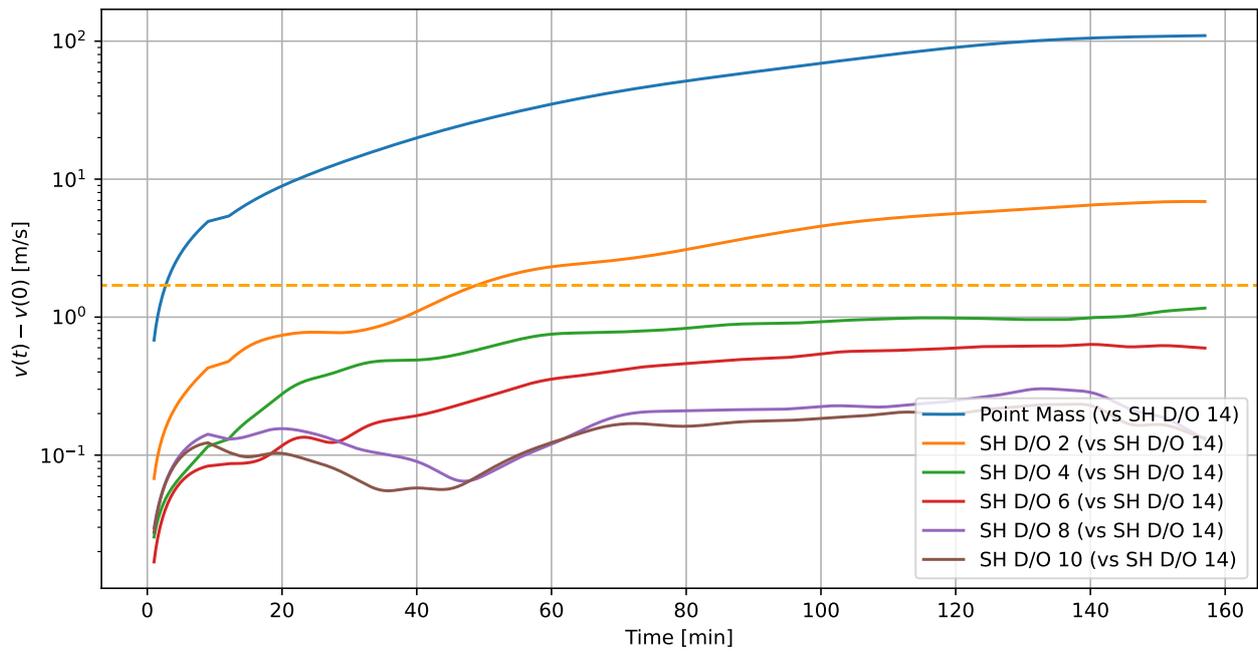
Figure A.11: MAV relative velocity over time for different Mars gravitational field models.
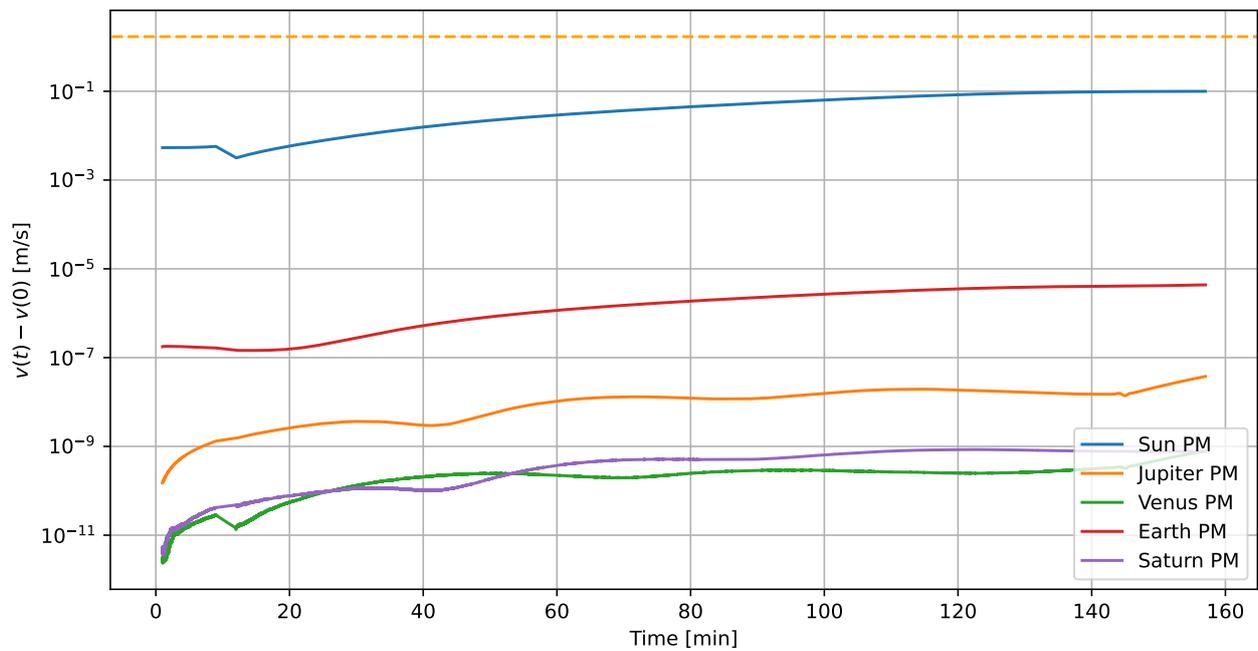
Figure A.12: MAV relative velocity over time for different point mass contributions to the environment.
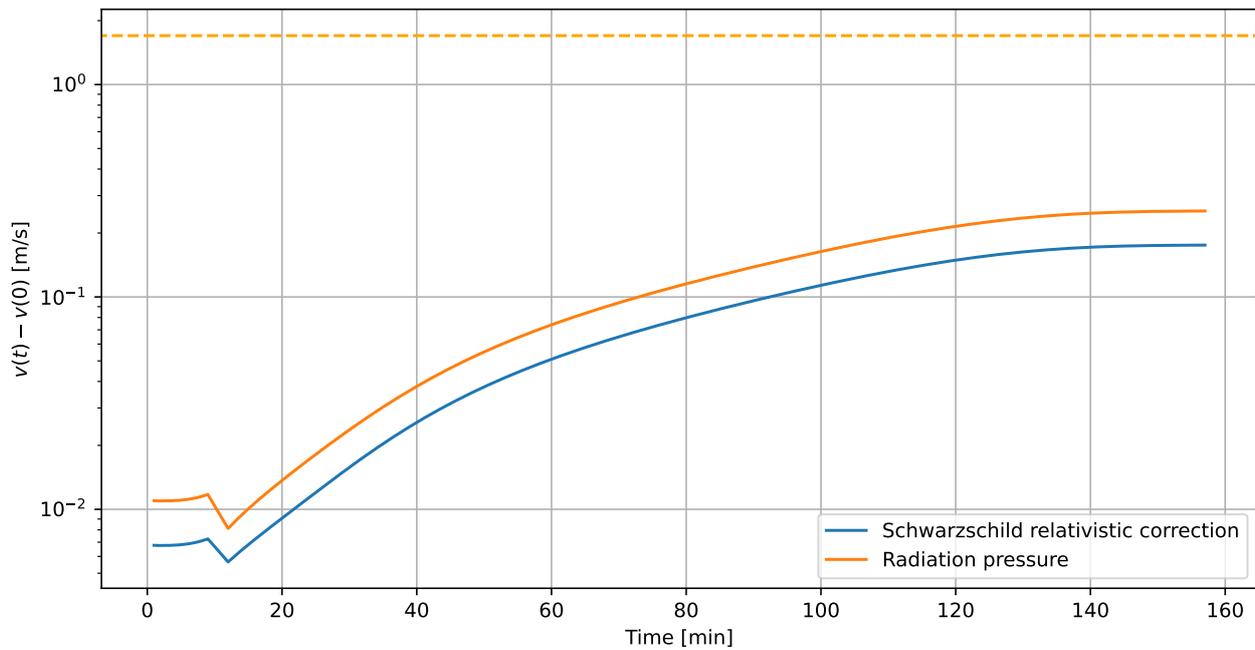
Figure A.13: MAV relative velocity over time for different miscellaneous acceleration models.
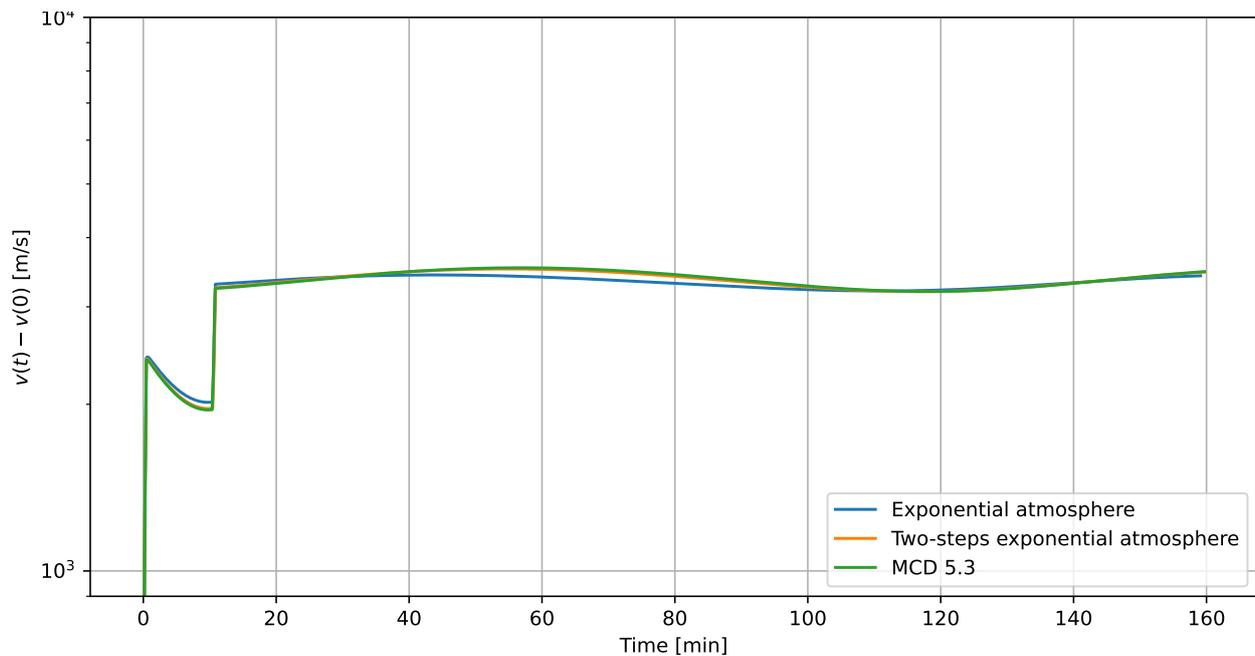
Figure A.14: MAV relative velocity over time for different Mars atmosphere models.

A.5. Optimisation results

In this section, the individual Pareto fronts made for the optimisation that uses the anchor first stage SRM geometry are first shown in Figures A.15 to A.17. These fronts were presented combined in Figure 4.57. Then, the combined Pareto fronts for the optimisations that use the multi-fin, tubular, and rod and tube first stage SRM geometries are shown in Figures A.18 to A.20 respectively. Last but not least, this section presents on the optimum that were found for the optimisations that use different first stage SRM geometries than the anchor geometry, the latter being found to be optimum. These optimum design variables are presented in Tables A.4 to A.6, with visual representations of the optimum cross-sectional SRM geometries in Figures A.21 to A.23.
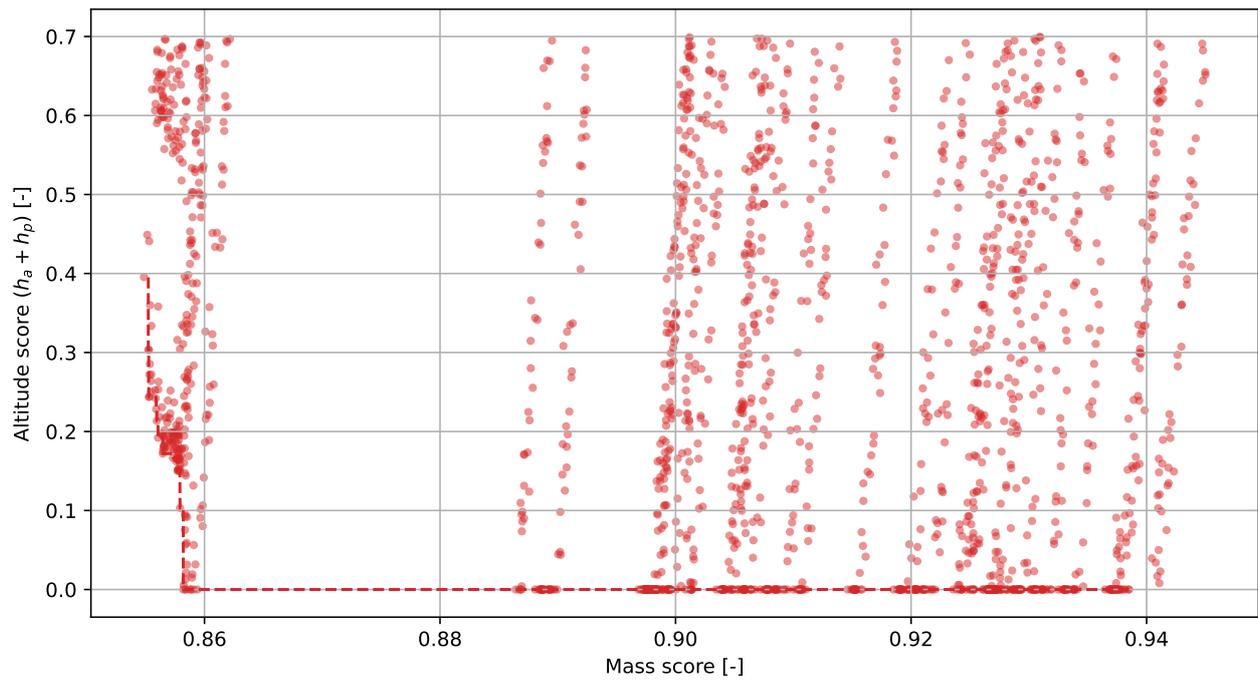
Figure A.15: Pareto plot compiling results from the Pareto front refinement using design variables scaling at random in [0.999, 1.001], using an anchor first stage SRM.
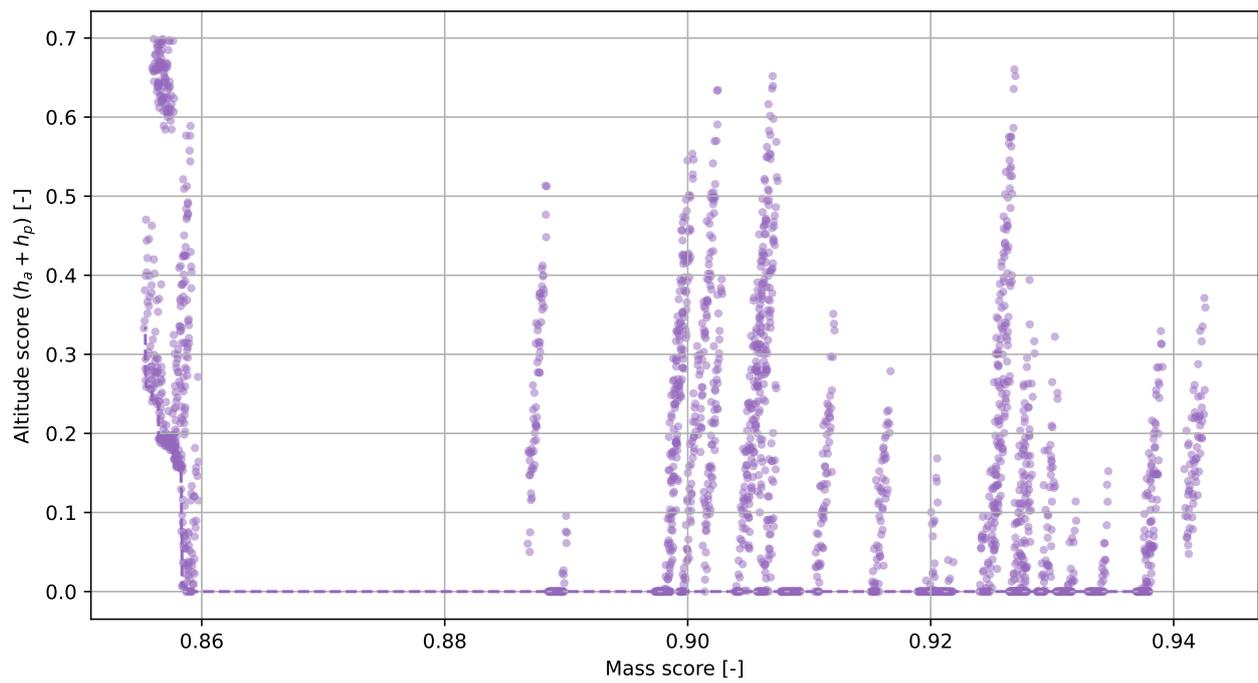
Figure A.16: Pareto plot compiling results from the Pareto front refinement using design variables varied at random by up to 0.1% their allowed range, using an anchor first stage SRM.
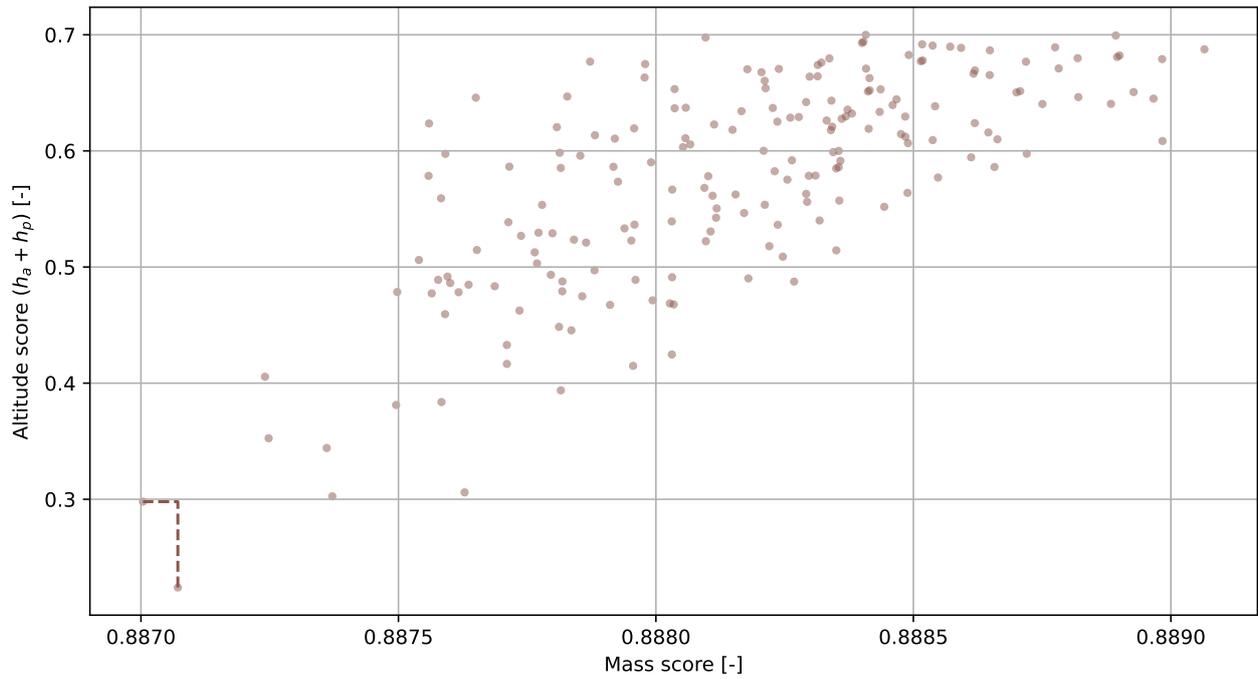
Figure A.17: Pareto plot compiling results from the Pareto front refinement based on the absolute optimum, using design variables varied at random by up to 0.25% their allowed range, using an anchor first stage SRM.
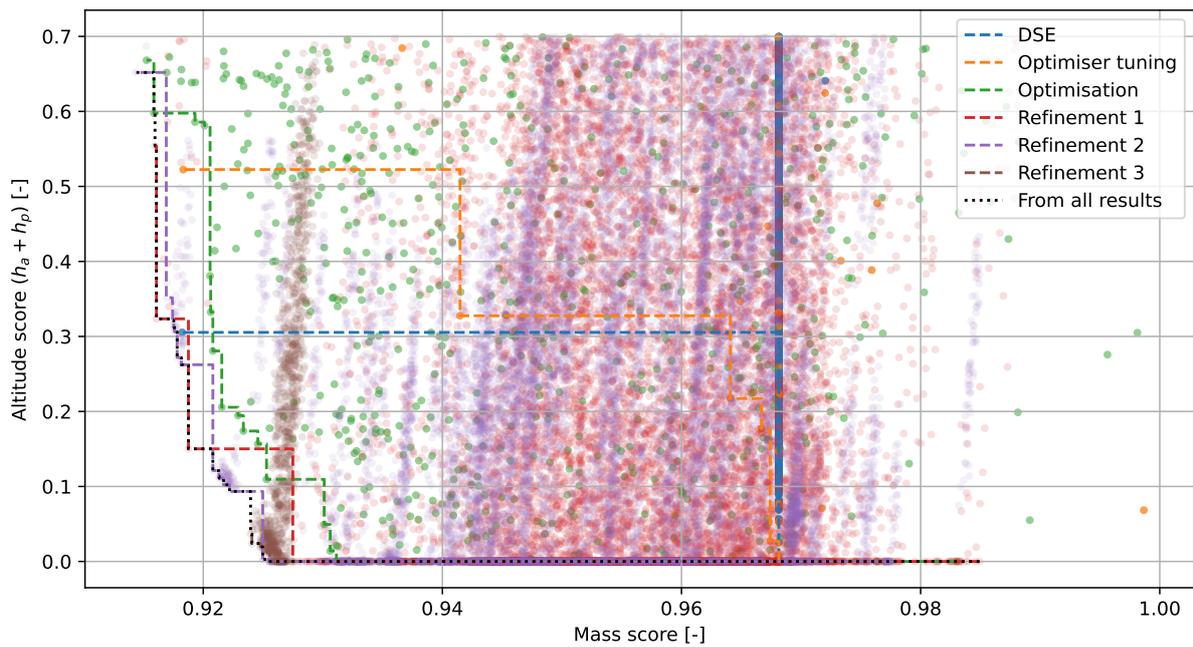
Figure A.18: Pareto plot compiling all results from the simulation database, using a multi-fin first stage SRM.
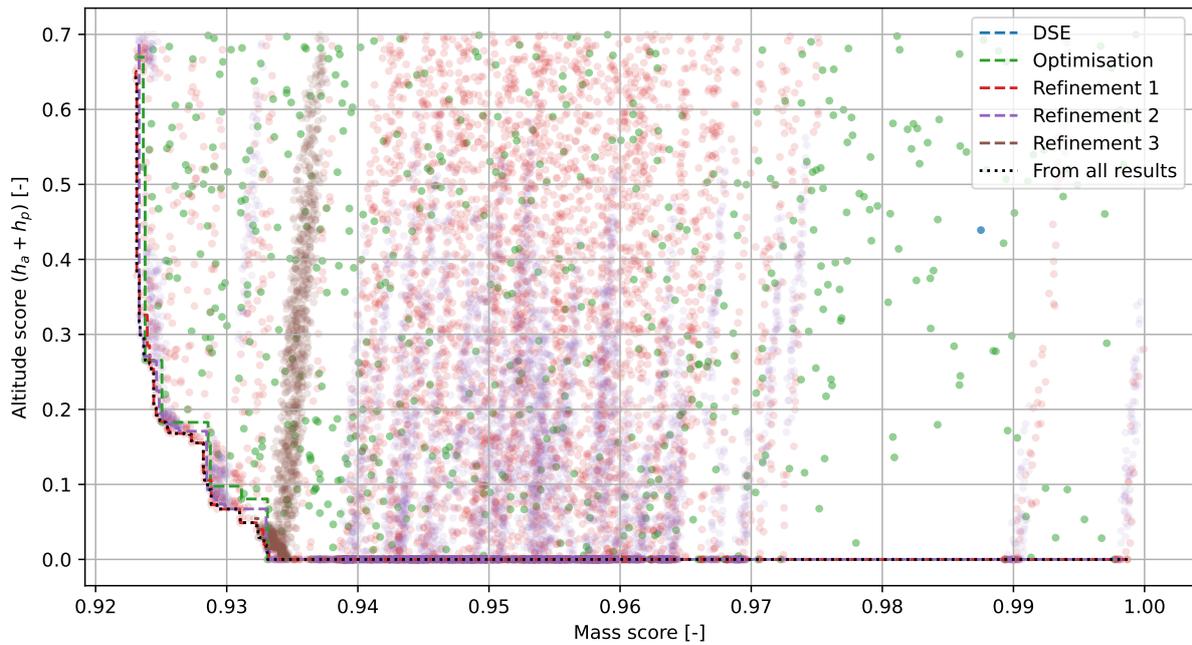
Figure A.19: Pareto plot compiling all results from the simulation database, using a tubular first stage SRM.
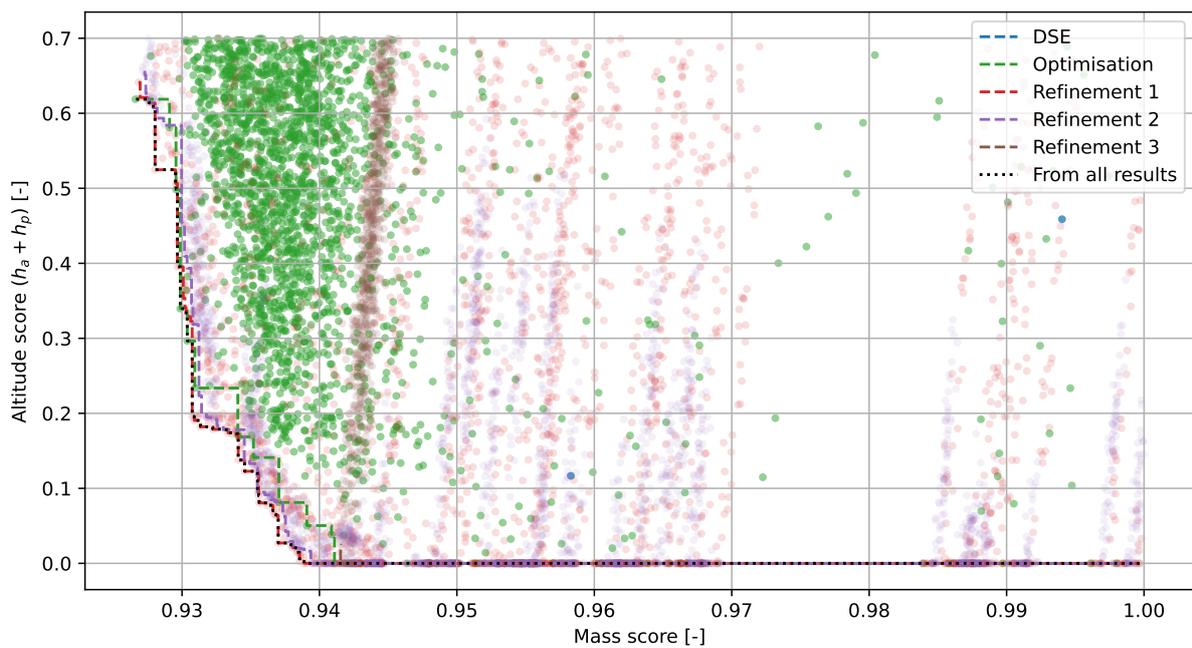
Figure A.20: Pareto plot compiling all results from the simulation database, using a rod and tube first stage SRM.

Design variable	Baseline value	Optimum value
Launch angle [deg]	57.5	57.39
Stage separation angle [deg]	90	89.457
TVC in z-direction, 5 nodes [deg]	0, 2.86, 0, -2.86, -2.86	0.1509, 2.699, 0.1146, -2.822, -2.806
Spherical SRM outer radius fraction $R_{o,2,\text{frac}}$ [-]	0.6875	0.7774
Spherical SRM inner radius fraction $R_{i,2,\text{frac}}$ [-]	0.5555	0.7786
First stage SRM length L [m]	1.05	1.0987
First stage SRM outer radius R_o [m]	0.24	0.26
Multi-fin SRM inner radius fraction $R_{i,1,\text{frac}}$ [-]	0.729	0.8492
Multi-fin SRM number of fins N_f [-]	20	9
Multi-fin SRM length fraction $L_{f,\text{frac}}$ [-]	0.2857	0.4925
Multi-fin SRM width fraction $w_{f,\text{frac}}$ [-]	0.5093	0.4261

Table A.4: Optimum design variables, compared to baseline values, using a multi-fin first stage SRM.

(a) First stage SRM.   (b) Second stage SRM.
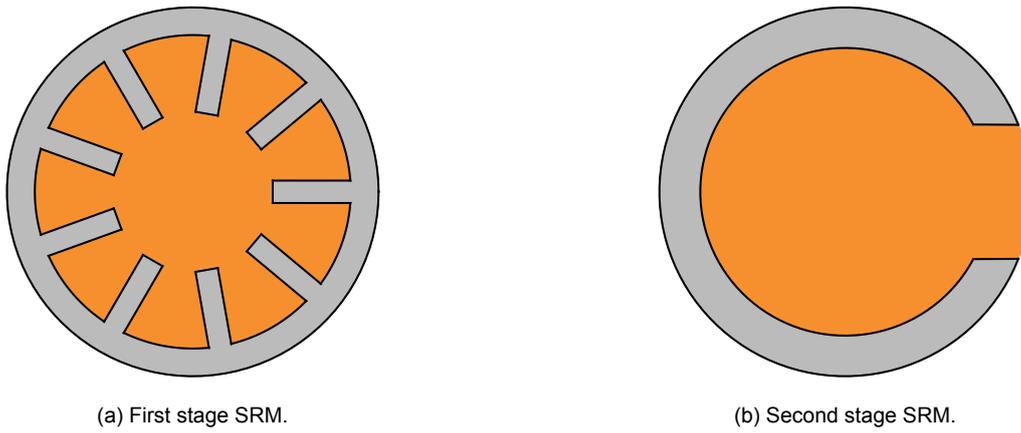
Figure A.21: Cross-sectional geometry of the SRM that results in the optimum MAV ascent, using a multi-fin first stage SRM.

Design variable	Baseline value	Optimum value
Launch angle [deg]	57.5	51.118
Stage separation angle [deg]	90	88.475
TVC in z-direction, 5 nodes [deg]	0, 2.86, 0, -2.86, -2.86	-0.296, 1.9631, -0.3228, -2.0261, -2.8118
Spherical SRM outer radius fraction $R_{o,2,\text{frac}}$ [-]	0.76	0.7858
Spherical SRM inner radius fraction $R_{i,2,\text{frac}}$ [-]	0.632	0.5582
First stage SRM length L [m]	1.15	1.1075
First stage SRM outer radius R_o [m]	0.25	0.2388
Tubular SRM inner radius fraction $R_{i,1,\text{frac}}$ [-]	0.64	0.7193

Table A.5: Optimum design variables, compared to baseline values, using a tubular first stage SRM.

(a) First stage SRM.   (b) Second stage SRM.
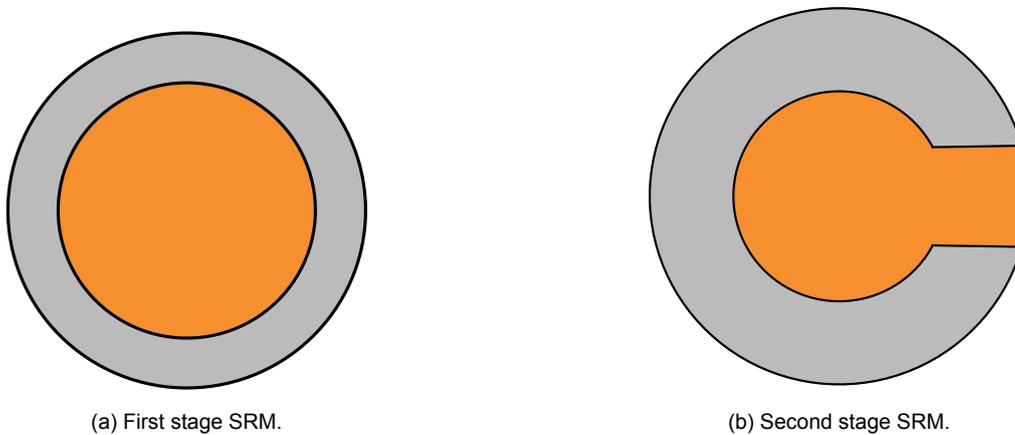
Figure A.22: Cross-sectional geometry of the SRM that results in the optimum MAV ascent, using a tubular first stage SRM.

Design variable	Baseline value	Optimum value
Launch angle [deg]	57.5	53.817
Stage separation angle [deg]	90	89.42
TVC in z-direction, 5 nodes [deg]	0, 2.86, 0, -2.86, -2.86	-0.2825, 2.7195, 0.1805, -2.5896, -2.6568
Spherical SRM outer radius fraction $R_{o,2,\text{frac}}$ [-]	0.76	0.8408
Spherical SRM inner radius fraction $R_{i,2,\text{frac}}$ [-]	0.482	0.6709
First stage SRM length L [m]	1.05	1.0261
First stage SRM outer radius R_o [m]	0.25	0.2298
Rod and tube SRM intermediate radius fraction $R_{mid,\text{frac}}$ [-]	0.56	0.701
Rod and tube SRM inner radius fraction $R_{i,1,\text{frac}}$ [-]	0.357	0.3681

Table A.6: Optimum design variables, compared to baseline values, using a rod and tube first stage SRM.

(a) First stage SRM.   (b) Second stage SRM.
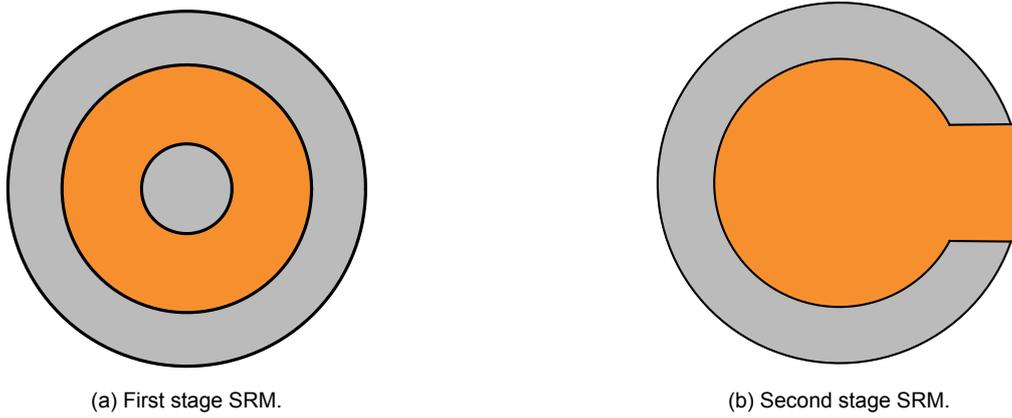
Figure A.23: Cross-sectional geometry of the SRM that results in the optimum MAV ascent, using a rod and tube first stage SRM.

A.6. Validation

In this appendix, Figure A.24 first presents a juxtaposition of the validation and simulated burning perimeter for the anchor SRM geometry. These are shown separately in Figure 4.64. Then, Figure A.25 presents the error in velocity over time for 200 simulations that were re-run with a higher accuracy integrator and environment. A similar plot is shown in Figure 4.65 for the error in position.
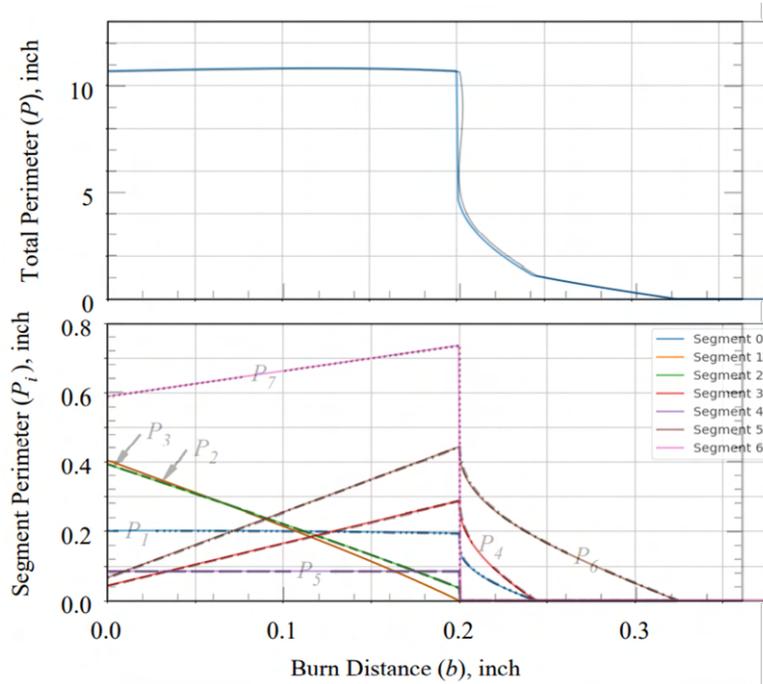
Figure A.24: Superposition of the peer-reviewed figure with the computed burning perimeter segments for the anchor SRM geometry.
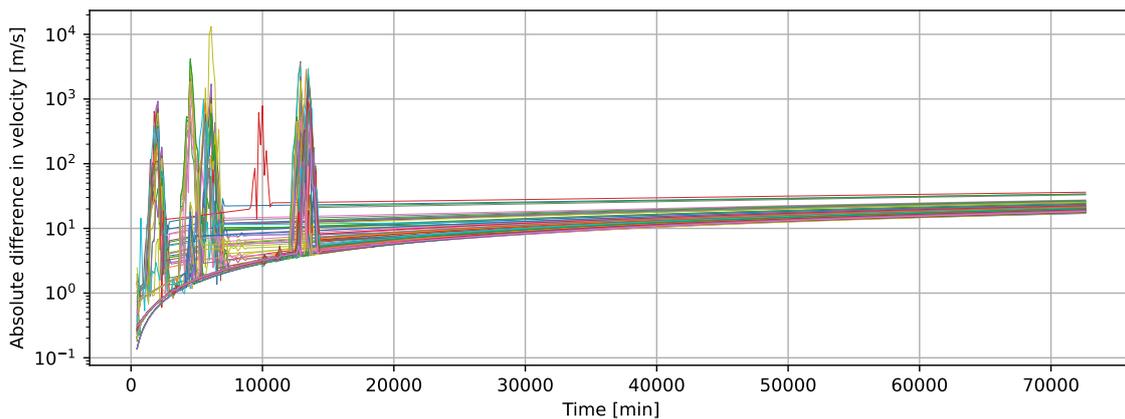
Figure A.25: Difference in velocity over time of selected simulations run with a higher accuracy integrator and environment.

A.7. Sensitivity analysis

In this appendix, the results of three different parts of the sensitivity analysis are shown graphically. These are the sensitivity analysis of the results to initial errors in Appendix A.7.1, to environment error in Appendix A.7.2, and to vehicle design errors in Appendix A.7.3.

A.7.1. Initial errors

Figure A.26 shows a topographic map of the Jezero crater, taken by the Mars express mission. This shows that the landing ellipse of Perseverance covers a latitude of about 6.6 km and a longitude of about 7.7 km. The altitudes range between -2.45 km and -2.65 km. Figure A.27 then shows the error in periapsis, apoapsis, and inclination of the final orbit, when variation in initial latitude and longitude are made. Lastly, Figure A.28 shows the same errors in case there is a delay between apogee and stage separation.
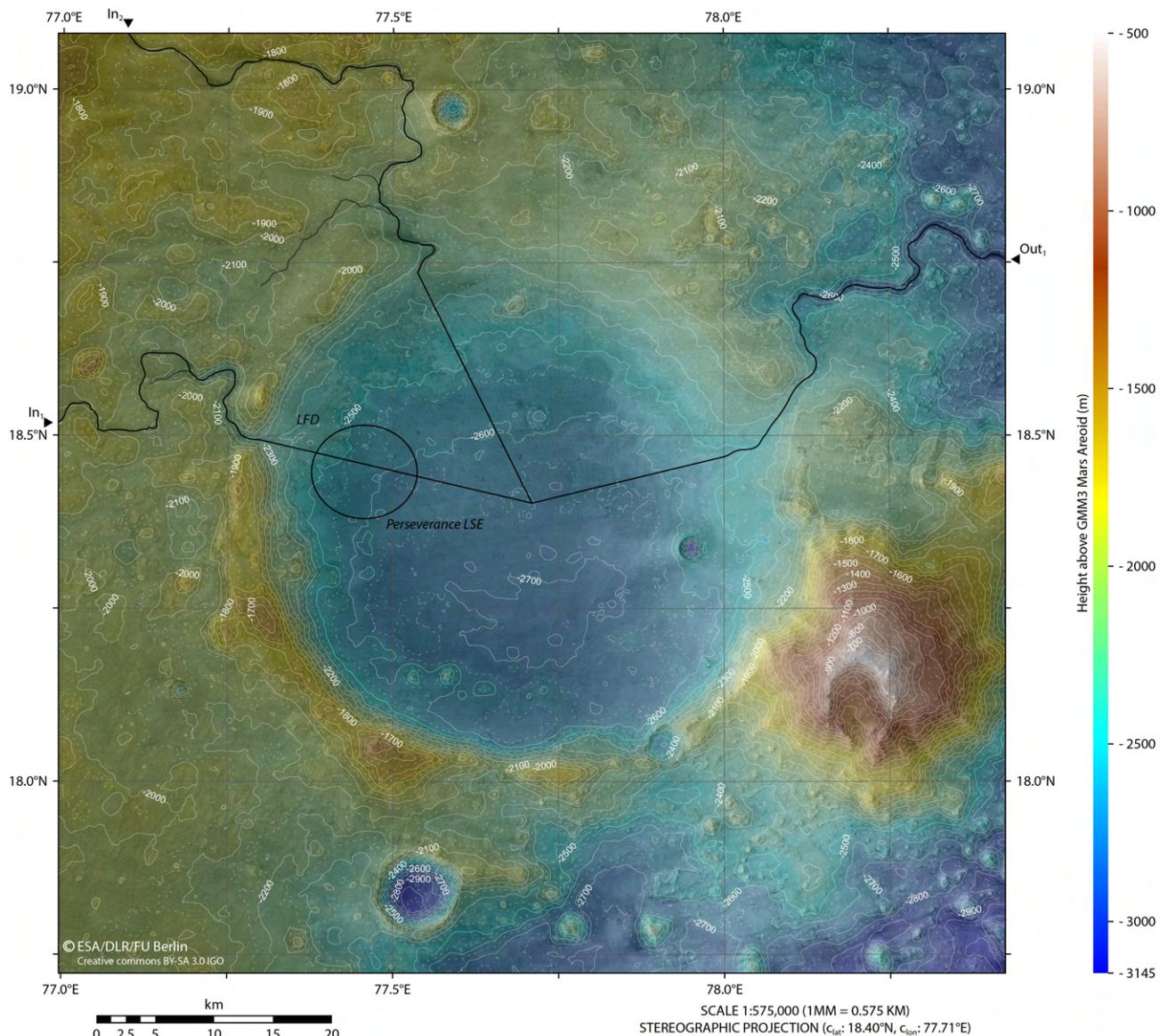
Figure A.26: Topographic map of the Jezero crater [98].
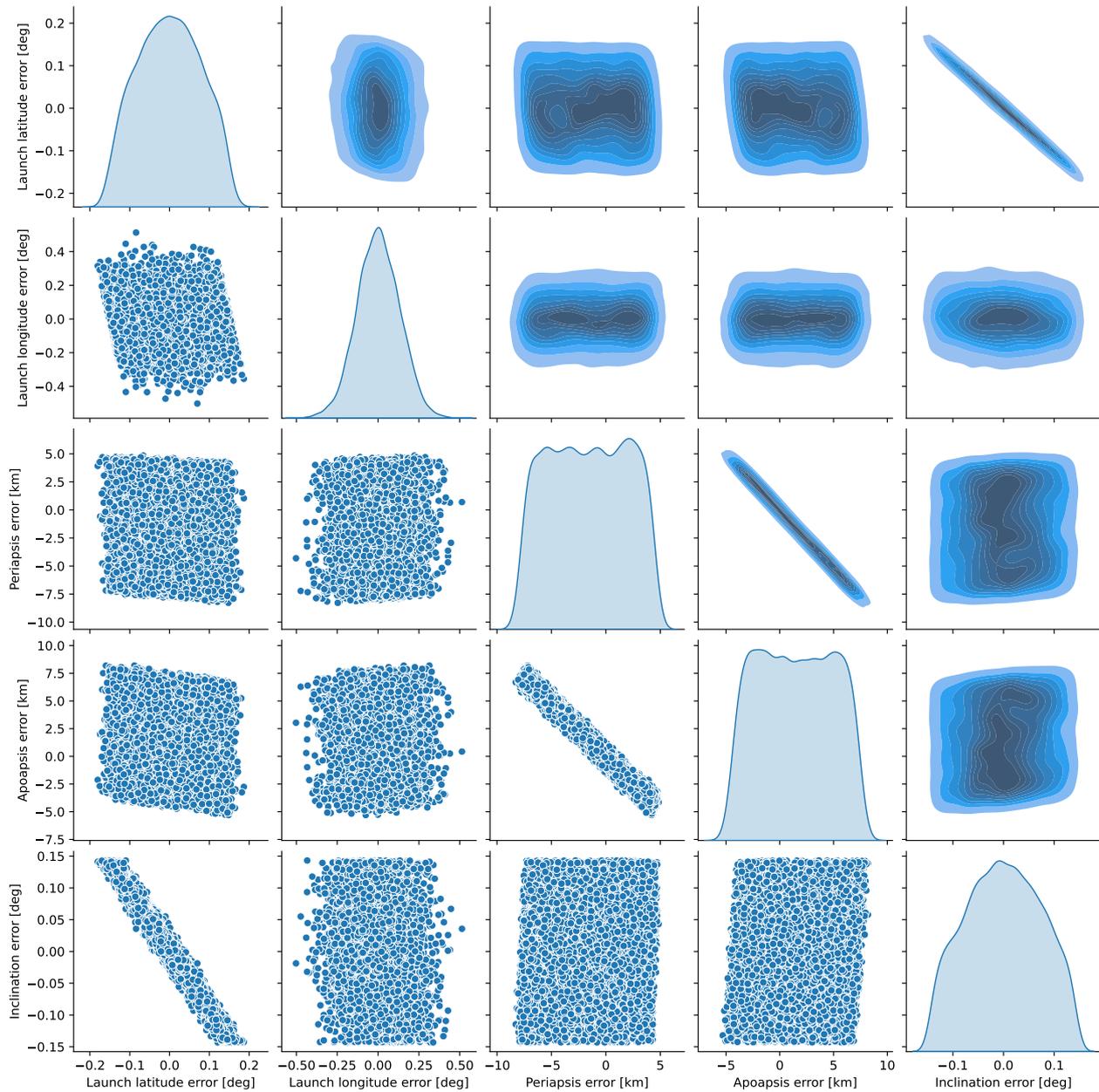
Figure A.27: Initial latitude, initial longitude, periapsis, apoapsis, and inclination errors, all plotted against each other.
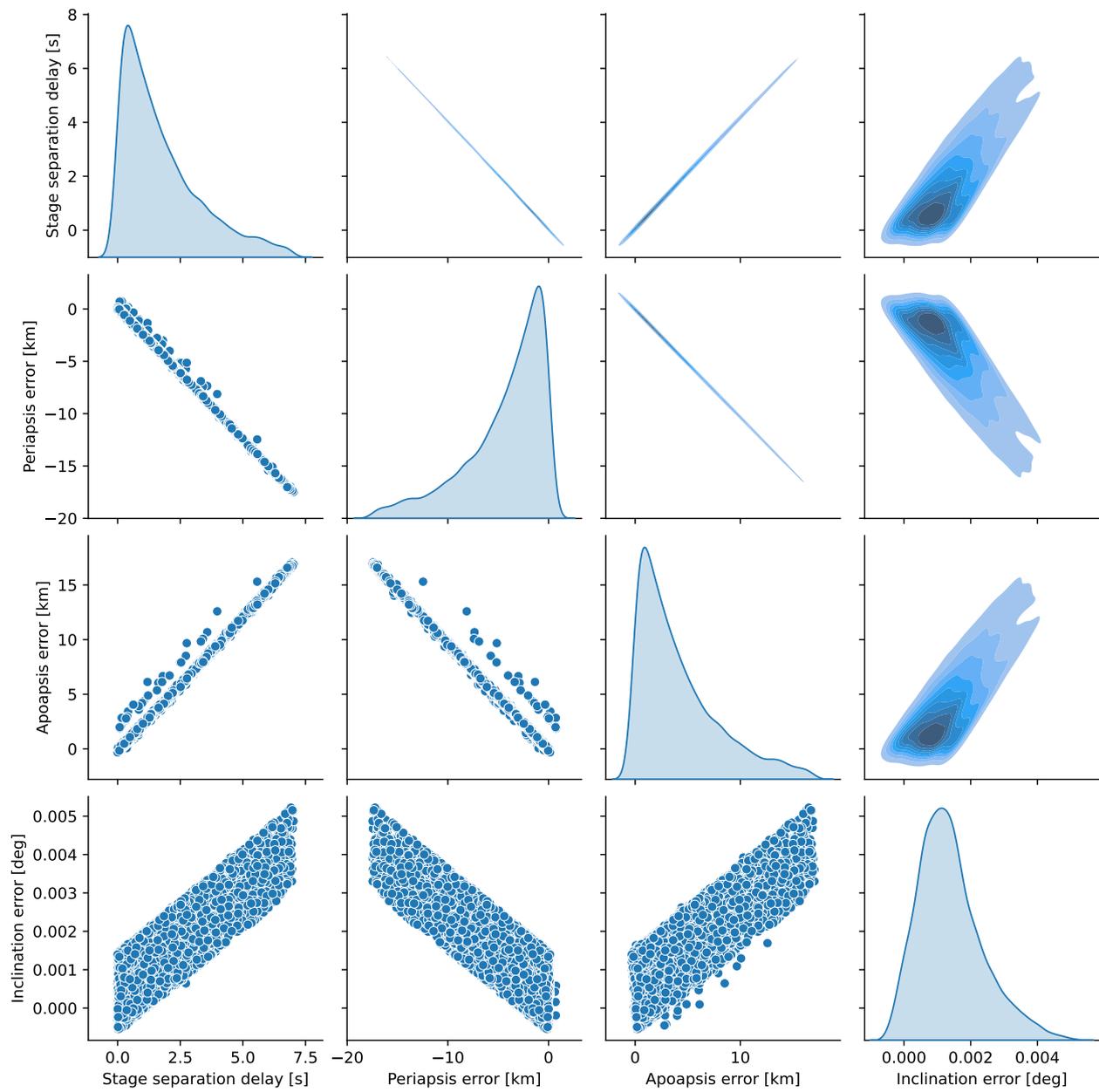
Figure A.28: Stage separation delay, periapsis, apoapsis, and inclination errors, all plotted against each other.

A.7.2. Environment

This section presents the plots made to support the analysis of the sensitivity of the final MAV orbit to deviations in the environment. First, Figure A.29 shows the orbital errors compared together with variations in atmospheric density. Then, Figure A.30 shows the same orbital errors compared with each other, in case winds are added to the ascent simulation.
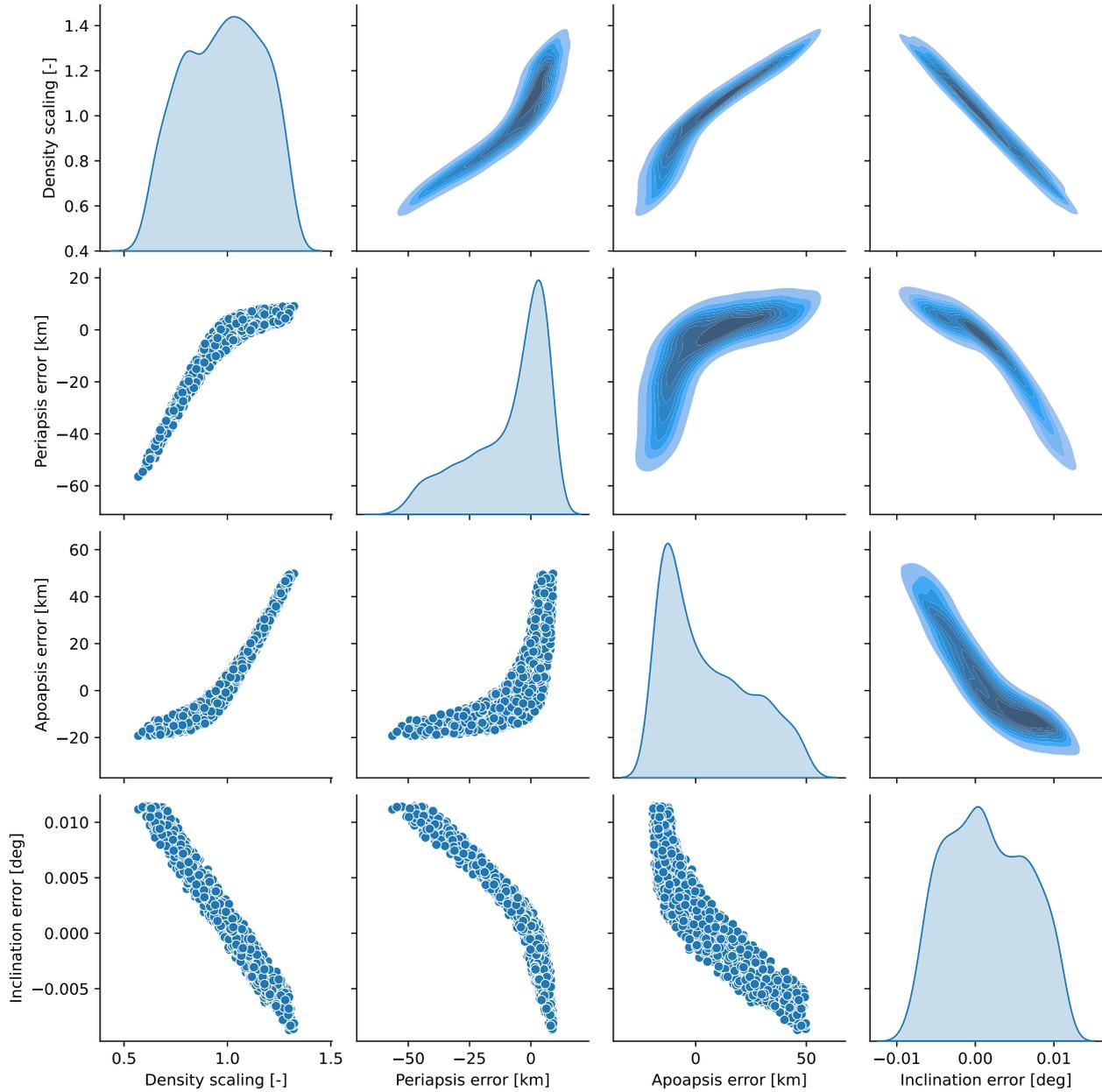
Figure A.29: Variation in atmospheric density, periapsis, apoapsis, and inclination errors, all plotted against each other.
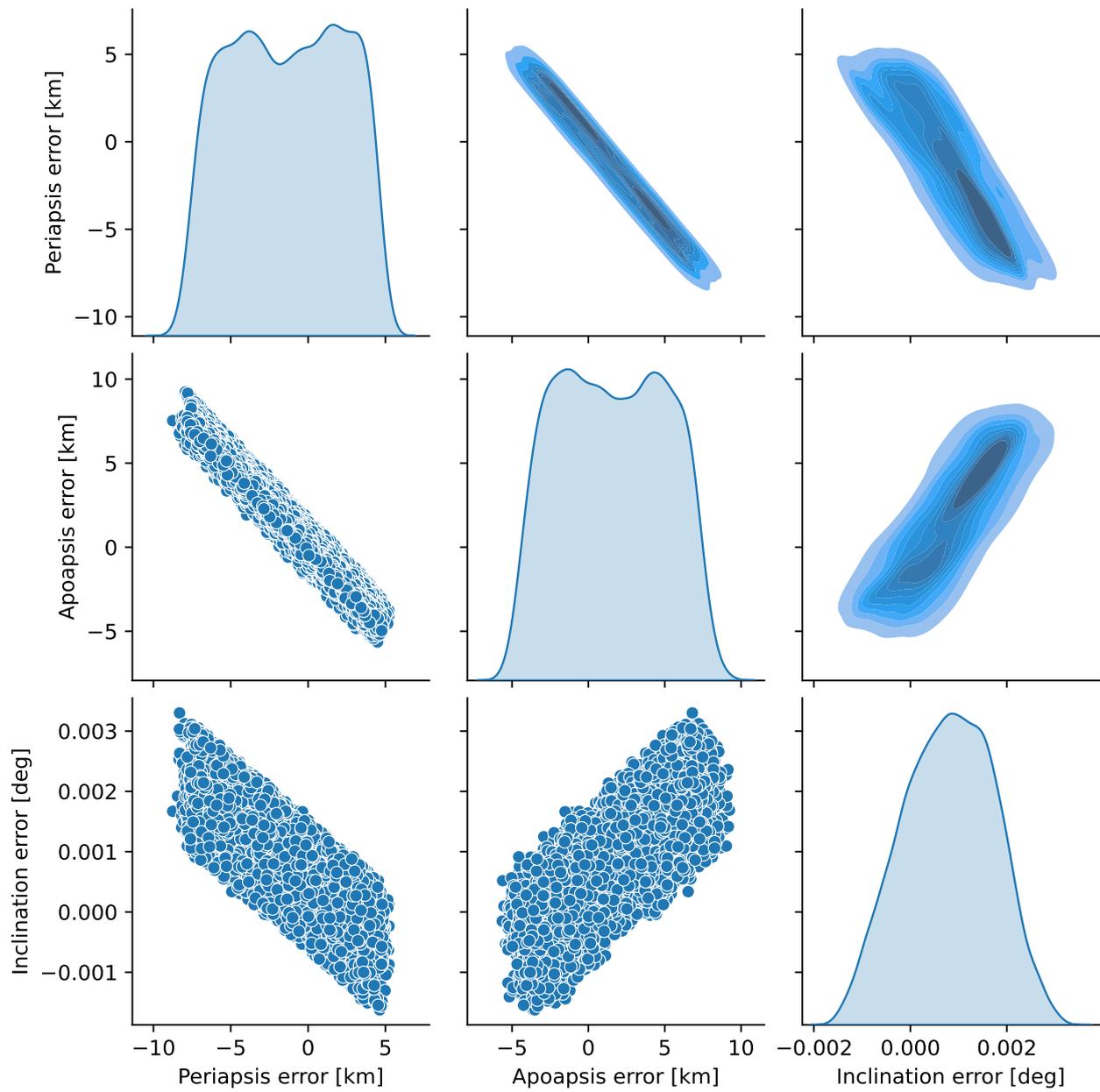
Figure A.30: Periapsis, apoapsis, and inclination errors, all plotted against each other, from the addition of winds from the MCD.

A.7.3. Vehicle design

Lastly, this section shows the relation between the different final errors in periapsis, apoapsis, and inclination, when errors in the design of the vehicle are made. These plots are made in case thrust is misaligned, the SRM are incorrectly sized, and the payload mass is varied, in Figures A.31 to A.33 respectively.
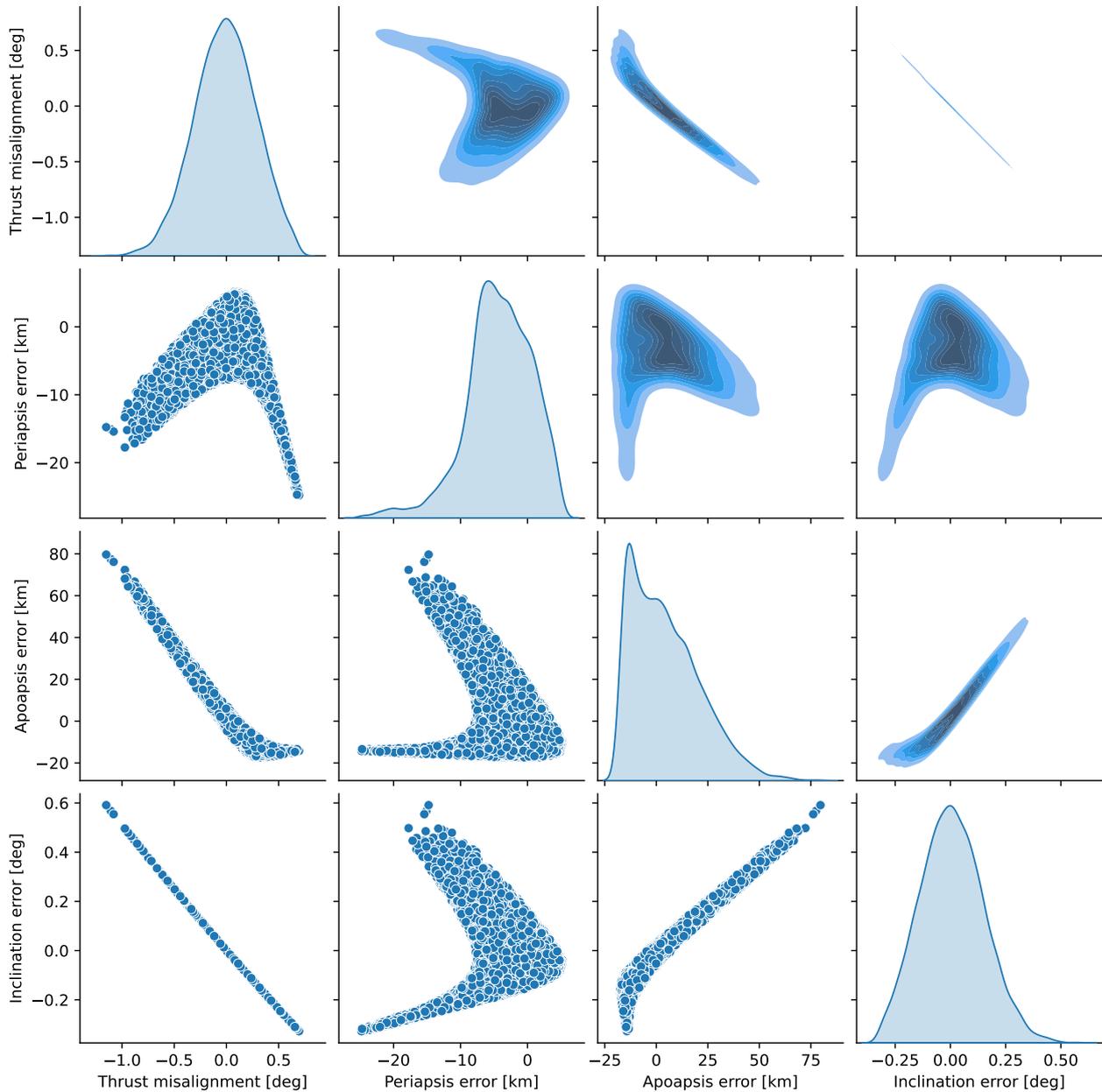
Figure A.31: Thrust misalignment, periapsis, apoapsis, and inclination errors, all plotted against each other.
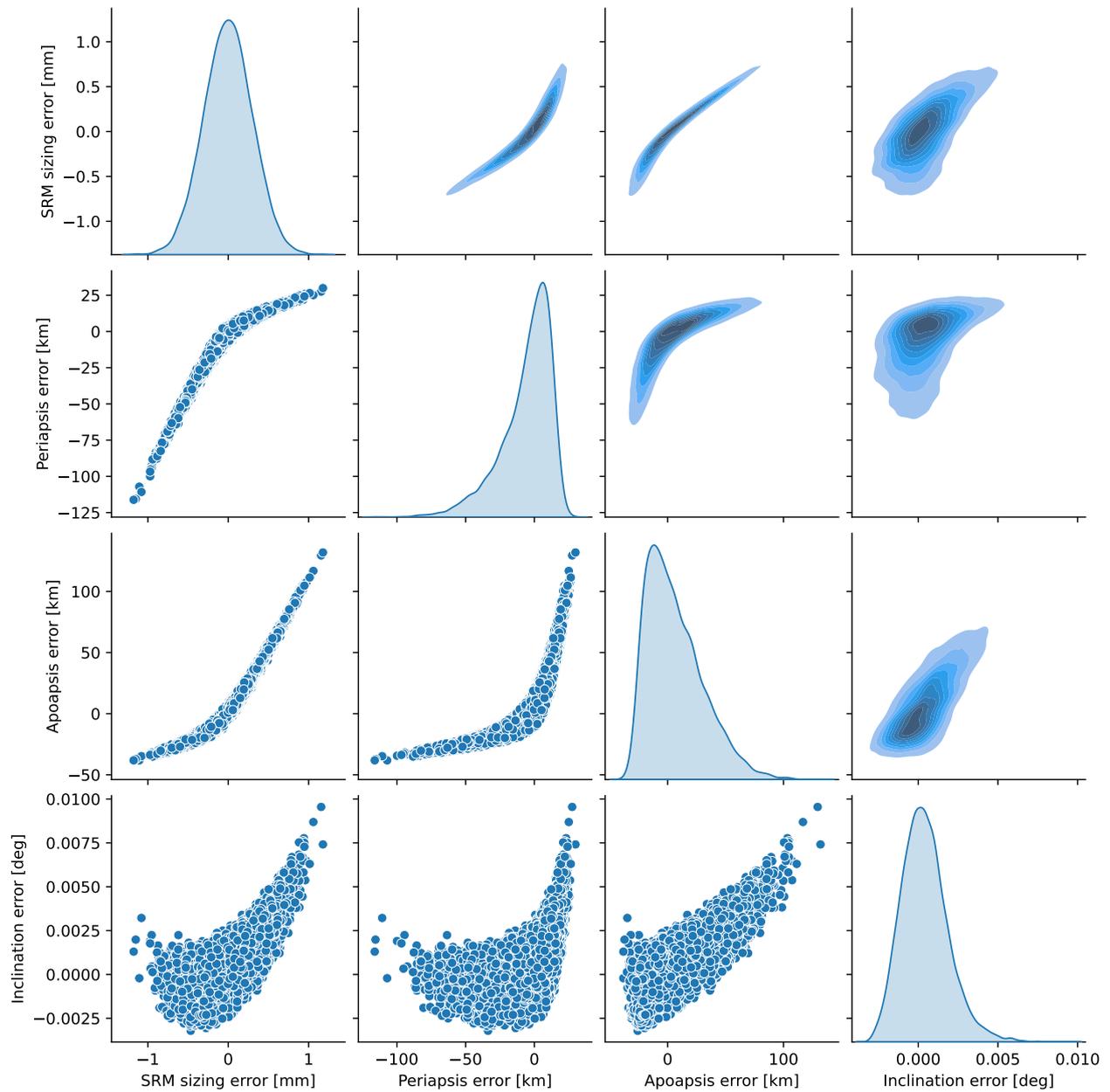
Figure A.32: SRM sizing, periapsis, apoapsis, and inclination errors, all plotted against each other.
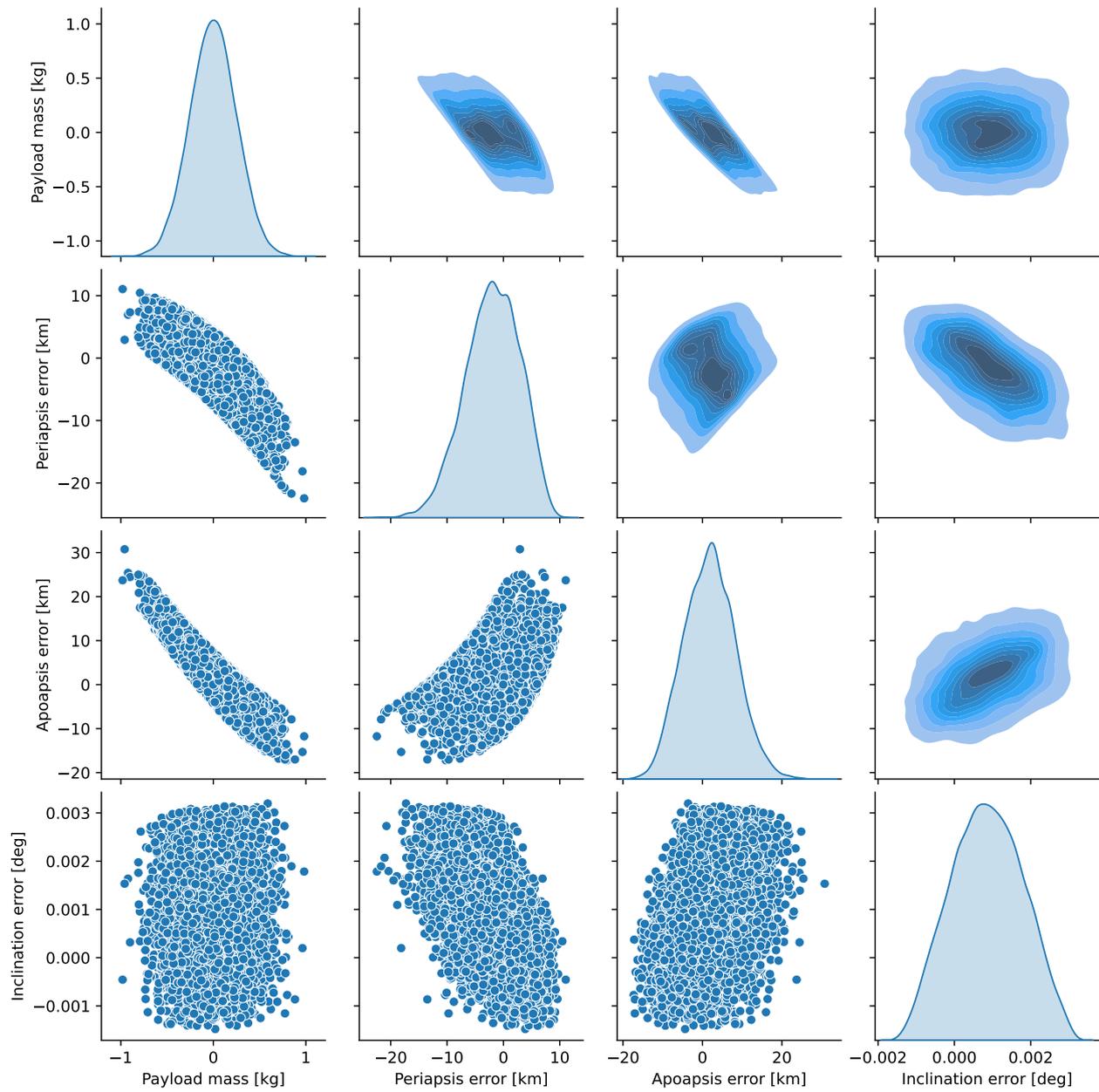
Figure A.33: Payload mass, periapsis, apoapsis, and inclination errors, all plotted against each other.