Mars Sample Return ascent trajectory optimisation

Literature Study

S.D. Petrovic

Faculty Aerospace, Section Spaceflight

Delft University of Technology

Challenge the future

Preface

Back in December 2013 Warren Gebbett gave a presentation on his work at the Jet Propulsion Laboratory (JPL) in Pasadena, California, USA and the opportunity for a new student to go and perform research at JPL. At this point I sent in my application together with eight other students. Then at the end of December I heard that I was invited for an interview in the first week of January 2014. In this interview it was concluded that I met all the requirements and that I was the perfect candidate to follow Warren up as the next student at JPL with financial backing of Dutch Space (now Airbus Defence and Space, the Netherlands). Financial backing was also going to be provided by the Stichting Prof.dr.ir. H.J. van der Maas Fonds (Aerospace Engineering Faculty, TU Delft) and the Stichting Universiteitsfonds Delft (TU Delft). Communication with JPL was thus started and in March 2015 it was clear that I would be working for the Mars Program Formulation Office under the supervision of Roby Wilson (Inner Solar System group, NASA JPL). He told me to focus on subjects that dealt with Mars missions. At that point I was doing my internship at DLR Bremen on Lunar rocket ascent and descent, which lasted till June 2015. When I came back to Delft me and my supervisors Erwin Mooij (rockets, trajectories, entry and descent, TU Delft) and Ron Noomen (mission design and orbit analysis, TU Delft) agreed that it would be best to perform a study on these Mars subjects to prepare for my visit to JPL and to formulate proposal thesis topics. The first week at JPL I presented these initial thesis topics to both people from the Inner Solar System group and the Mars program formulation office. The next few weeks were spent choosing and refining one of these topics. This document is the result of the two-month literature study on that topic to prepare for the thesis project.

S.D. Petrovic
Pasadena, California, January 2016

Contents

NOMENCLATURE

α	Angle of attack	rad or deg
α	In-plane thrust angle	rad or deg
β	Out-of-plane thrust angle	rad or deg
χ	Azimuth (or heading angle)	rad or deg
Δm_b	Required propellant mass for orbit insertion	kg
δ	Latitude	rad or deg
$\dot{\Omega}_M$	Rotational velocity of Mars	rad/s or deg/s
$\dot{m}$	Mass flow rate	kg/s
ϵ_T	Thrust elevation-gimbal angle	rad or deg
η	Numerical approximation	-
η	Step multiplication factor	-
η_a	Absolute effectivity	-
η_{cut}	Cut effectivity	-
η_r	Relative effectivity	-
γ	Flight-path angle	rad or deg
γ_a	Adiabatic index	-
$\mathbb{T}$	General transformation matrix	-
$\mathbb{T}_{\mathbf{x}}$	x-axis transformation matrix	-
$\mathbf{r}$	State vector containing coordinate values	-
$\mathbf{x}_{i+1}$	Next data point	-
$\mathbf{x}_i$	Current data point	-
μ	Standard gravitational parameter	m^3/s^2
μ_M	Standard gravitational parameter of Mars	m^3/s^2
μ_S	Standard gravitational parameter of the Sun	m^3/s^2
œ	Orbital element	m,-,deg or rad
Ω	Right ascension of the ascending node	rad or deg
ω	Argument of periapsis	rad or deg
Ω_P	Prime meridian offset at inertial reference frame definition	rad or deg
$\bar{\mathbf{X}}$	State vector	-
Φ	Increment function	-

ϕ	General rotation angle	rad or deg
ψ_T	Thrust elevation-gimbal angle	rad or deg
ρ	Air density	kg/m^3
ρ_0	Atmospheric density at zero elevation	kg/m^3
σ	Standard deviation	-
τ	Local error tolerance	-
τ	Longitude	rad or deg
θ	True anomaly	rad or deg
ξ	Non-singular orbital elements	m,-,deg or rad
a	Acceleration	m/s^2
a	Local speed of sound	m/s
a	Semi-major axis	m
A_e	Exit surface area of the nozzle	m^2
C_D	Drag coefficient	-
c_{eff}	Effective expulsion velocity	m/s
C_L	Lift coefficient	-
CR	Crossover constant	-
D	Drag	N
D	Number of variables	-
E	Eccentric anomaly	rad or deg
e	Eccentricity	-
e_{max}	Estimate of maximum truncation error	-
F	Amplification factor	-
f_N	Perturbing acceleration perpendicular to the radius vector	m/s^2
f_S	Radial perturbing acceleration	m/s^2
f_{Thrust}	Total thrust acceleration	m/s^2
f_W	Perturbing acceleration perpendicular to the orbital plane	m/s^2
G	Generation	-
G	Gravitational constant	m^3/(kg·s^2)
g	Local gravitational acceleration	m/s^2
g_0	Gravitational acceleration at Earth sea-level	m/s^2
h	(Current) step-size	s
h	Altitude	m
h	Specific relative angular momentum	m^2/s

H_s	Scale height	m
I	Retrograde factor	-
i	Inclination	rad or deg
I_{sp}	Specific impulse	s
k	Penalty function adjustable scalar	-
L	Lift	N
L	True longitude	rad or deg
M	Mach number	-
M	Mean anomaly	rad or deg
M	Order of the maximum truncation error estimate	-
m	Mass	kg
$m_{2,f}$	Final second stage mass	kg
m_2	Second stage mass after burn-out	kg
M_a	Molar mass of the Martian atmosphere	kg/mol
m_{MAV}	MAV mass	kg
M_M	Mass of Mars	kg
$Max_{n,i}$	Maximum number of not improved iterations	-
$N_{n,i}$	Number of not improved iterations	-
NP	Population size	-
p	Semi-latus rectum	m
$p_{0,s}$	Pressure of the surrounding air	Pa
p_0	Atmospheric pressure at zero elevation	Pa
p_e	Exit pressure	Pa
Q	Q-law function in Kepler elements	s^2
Q_{MEE}	Non-singular Q-law	s^2
R	Specific gas constant for Mars	J/(kg·K)
r	Radial distance	m
R_a^*	Molar gas constant	J/(mol·K)
r_a	Apocentre	m
r_{MS}	Distance between Mars and the Sun	m
R_M	(Local) radius of Mars	m
r_p	Pericentre	m
S	Surface area	m^2
S_G	Travelled ground distance	m

T	Thrust	N
t_0	Current time	s
t_1	First stage burn-time	s
t_2	Second stage burn-time	s
T_a	Absolute temperature	K
t_{c1}	First-second stage coast time	s
T_{MS}	Orbital period of Mars around the Sun	s
$T_{n,k}$	Truncation error	-
t_O	Time from inertial reference frame definition	s
t_{T2}	Second thruster control phase start time	s
$u_{i,G+1}$	Trial vector	-
V	Velocity	m/s
$v_{i,G+1}$	Mutation vector	-
$W_{œ}$	Orbital element weight	-
w_e	Expulsion velocity	m/s
W_P	Penalty function weight	-
$x_{i,G}$	Original, target, individual vector	-
x_n	nth variable	m, m/s and kg

ABBREVIATIONS

AB4	Adams-Bashforth 4th order
AB6	Adams-Bashforth 6th order
ABM4	Adams-Bashforth-Moulton 4th order
ABM12	Adams-Bashforth-Moulton 12th order
ACS	Attitude Control System
BF	Body-fixed
CNES	Centre National d'Études Spatiales
CNRS	Centre National de la Recherche Scientifique
DE	Differential Evolution
DLR	German Aerospace Center (Deutsches Zentrum für Luft- und Raumfahrt)
EA	Evolutionary Algorithm
EP	Electric propulsion
ESA	European Space Agency
EoM	Equations of Motion
GA	Genetic Algorithm
GEO	Geostationary Earth Orbit
GLOM	Gross Lift-Off Mass
GMT	Greenwich Mean Time
GRAM	Global Reference Atmospheric Model
GTO	Geostationary Transfer Orbit
ITAR	International Traffic in Arms Regulations
ISS	International Space Station
JPL	Jet Propulsion Laboratory
JVSRP	JPL Visitor Student Research Program
LEO	Low Earth Orbit
MAV	Mars Ascent Vehicle
MBH	Monotonic Basin Hopping
MCI	Mars-centred inertial
MCMF	Mars-centred Mars-fixed
MEE	Modified Equinoctial Elements
MSR	Mars Sample Return
N/A	Not Available
OS	Orbiting Sample
OTIS	Optimal Trajectories by Implicit Simulation
PaGMO	Parallel Global Multi-objective Optimizer
POST	Program to Optimize Simulated Trajectories
PSO	Particle Swarm Optimisation
RCS	Reaction Control System
RF	Frame of Reference
RHR	Right-hand-rule
RKF45	Runge-Kutta-Fehlberg 4th (5th) order
RK4	Runge-Kutta 4th order
RKN12	Runge-Kutta-Nyström 12th order
SA	Simulated Annealing
s/c	Spacecraft
SC14	Störmer-Cowell 14th order
SEP	Solar Electric Propulsion
SG	Shampine-Gordon
SNOPT	Sparse Nonlinear Optimizer
SQP	Sequential Quadratic Programming
TOSCA	Trajectory Optimisation and Simulation of Conventional and Advanced space transportation system
TRW	Thompson Ramo Wooldridge
TSI	Taylor Series integration
TVC	Thrust Vector Control

1

Introduction

What is the optimal trajectory solution for the combined trajectory problem of a high-thrust MAV and a low-thrust Mars orbiter single-revolution rendezvous in Mars orbit? This document has been written to provide back-ground information and an improved understanding of this preliminary thesis topic and the involved subjects that apply to the thesis work which will be performed at NASA JPL. All the information gathered and provided in this report is used to finalise the thesis topic and formulate a final thesis proposal. The thesis work at JPL is made possible through the JPL Visitor Student Research Program (JVSRP), which has provided the opportunity to perform research at JPL over a period of 1 year. During this period, the JPL staff and resources are available to use for the research as long as it does not involve any restricted information or software. This literature study was performed during the first few months at JPL, and the last few months will be used to carry out the thesis research. At the start of the research period it was known that the topic would involve a Mars mission. Because of restrictions and regulations it was decided that the thesis topic would include trajectory analysis and optimisation and the Mars Sample Return (MSR) mission. Eventually, the initial topic was set on combined MSR MAV ascent trajectory and Mars 2022 orbiter trajectory optimisation to rendezvous in Mars orbit. The road-map towards, and a comprehensive explanation of, the thesis topic will be explained in Chapter 2. In Chapter 3 different reference missions and research for each aspect of the thesis topic are presented to provide a perspective of what has already been done and which missions and what research could be used as reference and validation data. It is also important to gain a general understanding of different mathematical concepts that are necessary for the thesis work. Therefore Chapter 4 on optimisation and Chapter 5 on integrators were written and form the basic understanding that is required. In the optimisation chapter, different optimisation methods are described and a selection is made of the optimisation methods that will be used. The same is done for the different integrators. The thesis problem will need to be described using mathematical models to simulate the ascent and orbital conditions of the MAV and Mars 2022 orbiter respectively. Chapter 6 describes the used models, assumptions and corresponding equations for the ascent phase. In turn Chapter 7 describes the models, assumptions and equations for the orbiter. In these equations, many different reference frames and reference systems are used. This is why Chapter 8 describes the different required reference frames and systems and the transformations between them. All the required transformations, which follow from Chapters 6 and 7, are written out as well. One important aspect of the MAV ascent, which sets it apart from other sample return missions, is that Mars has an atmosphere which cannot be neglected. Accordingly Chapter 9 describes the different atmospheric models and the trade-off that was performed to decide which model to use in this thesis problem. For some of the aspects that will be treated in the thesis problem, certain software is already available. A summary of this software is provided in Chapter 10. Finally, all the information gathered is used to write a final thesis topic proposal and corresponding proposed schedule which is presented in Chapter 11. This literature study will serve as a guideline during the thesis project and provide background information for the final thesis report.

2

Overview

In Chapter 1 the initial topic was introduced: Mars Sample Return high-thrust ascent trajectory and low-thrust orbiter trajectory combined optimisation. Within this topic it is hypothesised that a rendezvous between the high-thrust Mars Ascent Vehicle (MAV) and the low-thrust Mars 2022 orbiter in less than one revolution after MAV launch would benefit both total mission mass and time. This initial topic is based on the proposed Mars Sample Return (MSR) mission by JPL[1] and is interesting because of the many different aspects of this mission. It will involve the integration and optimisation of the ascent trajectory of the MAV to low Mars orbit. Once in orbit, the MAV will have to rendezvous with an orbiting Spacecraft (s/c). The current proposal involves the Mars 2022 orbiter, which will use a low-thrust propulsion system to manoeuvre in the Martian system and will initially be used to perform scientific measurements. The orbiter must then perform several orbit changes to position itself in the desired rendezvous orbit. This low-thrust trajectory will have to be integrated and optimised as well. At the start of this literature study it is proposed to combine these two trajectory optimisations in order to find the optimal total mission mass and time. The different aspects of this proposal will be investigated in this literature study to determine the different possibilities for Mars ascent trajectories, low-thrust orbital trajectories, rendezvous in less than one revolution after launch, integration and optimisation of the trajectories. The final thesis proposal will be based on the information gathered and trade-offs perform in this study.

[1] All JPL missions: http://www.jpl.nasa.gov/missions/ [Accessed 30 July 2015]

3

Mission heritage

In early 2012 a group of scientists and engineers was set-up to devise an approach for the continuing exploration of Mars [1]. They had to combine all the requirements set by President Obama (to have humans in Mars orbit in the 2030s) and the 2011 NRC Decadal Survey for Planetary Science (science goals), while still keeping to the new proposed budget of the FY2013 U.S. Budget Submittal. Their conclusion was that a sample return mission would be the next logical step, because they deemed it to be the best compromise between human exploration, science and technology. They also suggested different mission architectures to perform an MSR mission; using three, two or one launch(es). The first concept would first launch a rover to collect samples, then an MAV and a so-called *Fetch* rover to collect the sample case from the first rover and launch them into orbit, and the final launch would then sent a sample return orbiter to collect the MAV/sample case and safely return it to Earth. The second concept would combine the last two launches into one using a smaller return orbiter propelled by a solar electric propulsion system. Finally, the third concept was to put all of these aspects into one launch, resulting in a sample collection rover with an on-board MAV system and again a small return orbiter [1]. Currently, NASA has decided to use a separate collector, orbiter and MAV system, which is what the research topic is based on. With the topic defined as in Chapter 2, it is required to determine what has already been done and what is currently being researched. This is called researching the mission heritage and is performed for every of the three phases involved: Mars atmospheric ascent (Section 3.1), low-thrust orbital transfers (Section 3.2) and rendezvous between an ascent vehicle and an orbiting vehicle above a planet (Section 3.3). Both these last two sections will focus on the Earth, the Moon and Mars and will not involve trajectories between different celestial bodies. Finally, the MAV heritage is examined in Section 3.4. To keep an overview, the reference missions and research will be discussed per phase and in the same order as mentioned.

3.1. Mars atmospheric ascent reference missions

To date there have not been any launches from the Martian surface back into space. This does not mean that there exist no reference missions or any reference research. In principle every Earth launch can be compared to a launch from the Martian surface; there is a noticeable gravitational acceleration, rotational speed and atmosphere. However, the Martian atmosphere is very thin with respect to the Earth atmosphere. In that sense, the Moon might be a better reference body. In this section research on Mars ascent will be discussed first, followed by actual flown Lunar sample return missions. Also, some Lunar ascent/sample return research will be discussed similar to the Mars studies and research.

3.1.1. Martian ascent reference research

Research has been performed on Martian ascent trajectories and is still being carried out, for instance, by the German Aerospace Center (Deutsches Zentrum für Luft- und Raumfahrt) (DLR). A list of reference research is shown in Table 3.1 followed by a short summary of each of the researched cases.

Fanning and Pierson [2] During the optimisation different ascent profiles were tested to find the optimal ascent trajectory from the Martian surface to a circular orbit around Mars. Two different models were used to observe the differences between them in the case of an optimisation for largest MAV payload

Table 3.1: Previous and current Mars ascent trajectory research.

Author	Organisation	Country	Year	Simulator
Fanning and Pierson [2]	Iowa State University	United States	1996	Unknown
Desai et al. [3]	NASA Langley and JPL	USA	1998	POST & OTIS
Whitehead [4, 5]	Lawrence Livermore National Laboratory	USA	2004	Unknown
Di Sotto et al. [6]	DEIMOS Engenharia and ESA	Portugal/Europe	2007	Unknown
Trinidad et al. [7]	Northrop Grumman and JPL	USA	2012	MCAT
Dumont [8]	DLR	Germany	2015 (Ongoing)	TOSCA

mass using a two-stage vehicle and thus minimising the propellant mass required. For the first and second stage a liquid XLR-132 and a liquid R-40B engine were assumed respectively. The target parking orbit had a radius of 3862.92 km (where they assumed a Mars radius of 3389.92 km resulting in a 473 km altitude). The two different used models were a gravity-turn model and a pitch-rate model. Using a constant Gross Lift-Off Mass (GLOM) of 1400 kg, the gravity-turn model resulted in a maximum payload mass of 367.8 kg whereas the pitch-rate model resulted in 366.4 kg (or 366.8 kg when a higher pitch-rate was used). It was mentioned that the pitch-rate model might yield a better result overall if the aerodynamic effects are also taken into account but in the end it was concluded that the gravity-turn model is a good choice to reach a preliminary design for the MAV.

Desai et al. [3] In this paper an MAV ascent to a 300 km altitude circular orbit was simulated. The aerodynamic effects were simulated in a program called the Aerodynamic Preliminary Analysis System (APAS), which provides quick estimates of the aerodynamic parameters for different configurations. APAS thus allows for rapid phase 0 conceptual studies of different vehicle designs. The aero-thermodynamic properties were simulated in the Langley Aerothermodynamics Upwind Relaxation Algorithm (LAURA), which is a computational fluid dynamics code that provides solutions to the thin-layer Navier-Stokes equations. Finally, two different trajectory optimisation programs were used: the Program to Optimize Simulated Trajectories (POST) and the Optimal Trajectories by Implicit Simulation (OTIS) program. In POST, numerical integration is used to find the optimum trajectory and in OTIS uses a collocation scheme to perform the optimisation. Mars-GRAM was used for the atmospheric model and the simulations also included mass, gravitational and propulsion models. Furthermore, a two-stage liquid propulsion system was assumed, which had to deliver a payload mass of 30 kg into space. It was concluded that aero-thermodynamic effects were minimal and the final nominal MAV lift-off mass was found to be 426 kg. Unfortunately, all the software programs mentioned in this paper fall under the International Traffic in Arms Regulations (ITAR) of the United States [1] and can thus not be used by any foreign nationals.

Whitehead [4, 5] In both papers the circular target orbit was taken at an altitude of 500 km. Also it was assumed to have a 100 kg MAV launch mass. Different trade-offs between staging, thrust, shape, and either liquid or solid propulsion were performed. In this case liquid propulsion showed a better performance with respect to the solid propulsion option. The problem was simulated as a simplified 2-D problem and coded into Fortran. It is also mentioned that miniaturisation of both liquid and solid propulsion systems would be required (down to a complete 10 kg system). A liquid single-stage to orbit is provided as a viable alternative.

Di Sotto et al. [6] This research was performed as part of the European Aurora program (human space exploration program). The possible advantages of ascending to a 300 by 2000 km altitude parking orbit are discussed with respect to an ordinary 500 km altitude circular parking orbit. The optimisation of the trajectory was split into an atmospheric part and a so-called exo-atmospheric part. The first part was

[1] Personal correspondence with JPL personnel

then again split into a vertical rise phase, constant pitch rate phase, constant pitch phase and gravity-turn phase respectively following MAV launch. The second part was also split into several phases: first an active propulsion phase, then a coast phase, followed by another active propulsion phase where the steering resulted from using an optimum guidance law derived from the Primer vector method. In this case it was assumed that both the first (four engines) and second stage (one engine) used liquid Rocketdyne RS-2101c engines. It was concluded that a 300 by 2000 km altitude orbit could be achieved using the same architecture as for the 500 km altitude circular orbit. This would then also save 1000 kg in orbiter mass because of the higher altitude rendezvous.

Trinidad et al. [7] In this paper the baseline design for the MAV is discussed. Several trade-off's were made concerning the kind of propulsion system and propellants. Also, five different launch flight-path angles (aligned on the same longitude) were simulated and compared. This trajectory analysis was performed using a three degrees of freedom tool developed by Northrop Grumman called the Mission Capabilities Analysis Tool (MCAT). It used the Mars Geodetics and the 2010 atmosphere version of Mars-GRAM. The program was used to optimise (through the change of steepness of the trajectory and relative size of each stage) for a minimum MAV GLOM. The trajectory analysis was performed at an orbit of 466 km above the Martian surface. The resulting lowest GLOM was achieved using two liquid stages resulting in an MAV mass of 227 kg (283 to 391 kg including contingencies).

Dumont [8] [2] Trajectory Optimisation and Simulation of Conventional and Advanced space transportation system (TOSCA) is a program that has been under development by DLR. It was initially created as an Earth launcher ascent simulator for early launcher design. Recently Dumont has started updating the program for both Lunar and Martian ascent, unfortunately nothing has been published yet on Martian ascent simulations. It incorporates both aerodynamic changes and propellant changes into the trajectory simulation and can also be used to optimise the trajectory based on several optimisation parameters.

3.1.2. REFERENCE FLOWN LUNAR SAMPLE RETURN MISSIONS

As mentioned before, Lunar sample return missions can be considered similar to Mars sample return missions, one of the main difference being the atmosphere. This means that Lunar flight data can be used to validate the initial ascent program by assuming a Lunar ascent (and then changing the parameters to Martian parameters and adding the atmosphere in a later phase). A comprehensive overview of all the sample return missions is presented in Table 3.2 for both manned and unmanned missions [3]. All successful soil and rock sample return missions were performed by the Soviet Union (robotic) and the United States (manned).

Table 3.2: Previous Lunar Sample Return Missions.

Launch date	Country	Mission	Returned mass [kg]
16 July 1969	USA	Apollo 11	22
14 November 1969	USA	Apollo 12	34
12 September 1970	Soviet Union	Luna 16	0.101
31 January 1971	USA	Apollo 14	43
26 July 1971	USA	Apollo 15	77
14 February 1972	Soviet Union	Luna 20	0.055
16 April 1972	USA	Apollo 16	95
7 December 1972	USA	Apollo 17	111
9 August 1976	Soviet Union	Luna 24	0.17

Both Luna and Apollo missions first landed a craft on the Moon, then collected samples and returned these samples using a Lunar ascent vehicle. Some data on the Luna missions is provided in [9]. Unfortunately, it does not provide enough information on the ascent trajectory of the Luna return vehicles and therefore the Luna missions cannot be used for validation. The Apollo missions were, however, documented in detail and do include ascent flight data [10].

[2] Information based on personal communications with author as well

[3] Overview: https://en.wikipedia.org/wiki/Sample_return_mission [Accessed 28 October 2015]

3.1.3. Lunar ascent reference research

Research on Lunar sample return can also be used as reference data. Therefore a similar table to Table 3.1 can be set-up for Lunar research. An overview of different Lunar ascent/sample return research is provided in Table 3.3. It is interesting to see that different studies use different Lunar ascent vehicle configurations and propulsion systems based on the requirements for each study case.

Table 3.3: Previous and current Lunar ascent trajectory research.

Author	**Organisation**	**Country**	**Year**	**Program name**
Sostaric and Merriam [11]	NASA Johnson Space Center	USA	2008 (Ongoing)	SORT
Dietrich et al. [12]	University of Colorado	USA	2015	Copernicus
Dumont [8]	DLR	Germany	2015 (Ongoing)	TOSCA

Sostaric and Merriam [11] In this paper manned Lunar ascent and rendezvous trajectories were simulated and optimised. This was accomplished by dividing the ascent up into three different parts. A 100 m vertical rise, followed by a single-axis rotation manoeuvre and ending in a powered explicit guidance phase to reach the target orbit. The problem was simulated in a 3-degrees of freedom simulation program called SORT. The single-axis rotation algorithm calculates a time-optimal single-axis rotation given the initial and final attitude. For the final phase of the ascent an optimiser called NPSOL was used, which is a non-linear programming solver. The initial target orbit was 15.24 by 75 km above the Lunar surface. The possibilities for emergency ascent were investigated together with a comparison between yaw steering and on-orbit plane changes. It was concluded that for large plane changes it is best to perform the plane changes on-orbit to reduce the required ΔV.

Dietrich et al. [12] In 2013 a new research study was performed called Orion/MoonRise where the MoonRise mission would be combined with a manned mission to a halo orbit around the Earth-Moon L2 point [13]. In that case the MoonRise probe would be modified slightly and would have to ascent from the surface of the Moon to the Orion s/c in its halo orbit. The simulations performed showed a maximum theoretical sample return mass of 38 kg. The ascent vehicle would use a single solid STAR 48AX motor, based on the STAR 48A and 48B motors, and would have to be specially designed by Orbital ATK. In this follow-up study two possible landing/ascent points on the Moon were investigated: Schrödinger Crater and Tsiolkovsky Crater [12]. The simulator was developed in a tool called Copernicus combined with a MATLAB interface. The trajectories were simulated to ascent from the Lunar surface to a halo orbit around L2 and were split into four parts: ascent, trajectory correction manoeuvre, halo orbit insertion and halo orbit propagation.

Dumont [8] Lunar ascent simulations were performed using TOSCA in light of the ROBEX project. In this study, a re-usable concept Lunar ascent vehicle is described using a liquid oxygen/liquid hydrogen propulsion system with a specific impulse (I_{sp}) of 440 s. The target orbit was 15 by 100 km after which the orbit would be circularised to 100 km. Ascent altitude-time profiles are provided for two different constant thrust settings: 25.9 and 34.5 kN.

3.2. Low-thrust orbital transfer reference missions

Because the planned Mars 2022 orbiter, which will likely collect the Martian sample from the MAV and return back to Earth, will have a low-thrust propulsion system it is important to identify previous missions and research. In the past few years, low-thrust propulsion systems have become increasingly popular in long-duration missions because of the high I_{sp} and the low total system mass. Since the current defined topic will focus on a low-thrust problem in the Martian system, it is useful to analyse the propulsion system and the trajectories flown by previous low-thrust missions in planetary systems (see Section 3.2.1). To date, no low-thrust missions have flown in the Martian system. However, research has been performed on low-thrust missions in both the Earth and Martian system as described in Section 3.2.2.

3.2.1. EARTH SYSTEM LOW-THRUST APPLICATIONS

Low-thrust applications in the Earth system were first implemented back in the 60s with the first (non-experimental) flight, using Electric propulsion (EP) for attitude control, performed by the Zond-2 [14]. The three main applications for EP in satellites have historically been in attitude control, (geosynchronous) station keeping and other general orbit adjustments. For the thesis problem it is interesting to investigate missions that used EP for orbit transfers and orbital phase changes. Table 3.4 provides an overview of different missions that used electric propulsion EP as a means to perform either orbital phase adjustments or orbit transfers (based on [14]). In the past few years EP has become increasingly popular for use as s/c main propulsion systems and now companies, such as Boeing and their 702SP series of communication satellites[4], are even introducing fully electric s/c [15]. These new satellites are also included in Table 3.4 as well as a number of missions which had to use their EP systems for orbital transfer manoeuvres even though the electric thrusters were not intended for it[5,6].

Table 3.4: Previous Low-thrust Earth Missions.

Launch year	**Country**	**Mission**	**EP use**
1965	USA	Vela	Phase adjustment
1967	USA	Advanced Vela	Phase adjustment
1988	USA	Gstar-3	Orbit Transfer
2000	USA	MightySat II.1	Orbit Transfer
2001	Europe	Artemis	Orbit Transfer
2010	USA	AEHF(-1)	Orbit Transfer
2015	USA	ABS-3A	Orbit Transfer
2015	USA/France	Eutelsat 115 West B	Orbit Transfer

Compared to the intended use of the low-thrust EP on the Mars 2022 orbiter, where the propulsion system shall be used to perform several orbital transfer manoeuvres including orbit raising, orbit lowering and most likely inclination changes, the orbit transfer applications for the satellites shown in Table 3.4 are simply transferring the s/c from a Low Earth Orbit (LEO) or Geostationary Transfer Orbit (GTO) to Geostationary Earth Orbit (GEO). Advanced low-thrust manoeuvres have been performed by interplanetary missions, however non-planetary missions are not included in this report.

3.2.2. LOW-THRUST (MARTIAN) PLANETARY SYSTEM REFERENCE RESEARCH

Because low-thrust as a main propulsion system has become increasingly important, many studies are currently being conducted in the area of low-thrust trajectory optimisation for the Earth system, Mars system and mainly interplanetary transfers. However, since this problem deals with the orbit transfers in a planetary system the reference research will be selected based on Earth and Martian system applications. The reference research is presented in Table 3.5 again followed by a short description.

Geffroy and Epenoy [16] The possibility of using a generalised version of averaging techniques for low-thrust optimisation was investigated. Two problems were considered: minimum-time and fuel-saving both combined with thrust direction, environmental and technological constraints. It is mentioned that rendezvous problems can be treated as well, but was not done during this research study. It was concluded that the method works well but needs to be combined with other methods to properly solve the numerical problem.

Kluever and Oleson [17] This research concerned the optimisation of low-thrust trajectories from LEO to GEO and GTO to GEO using a direct method (incorporating averaging techniques) and it was compared to the widely used SEPSPOT (which uses a shooting method to solve the two-point boundary value problem and is based on calculus of variation). The direct method used a sequential quadratic programming optimiser to optimise the problem. It was concluded that the developed direct method is a robust method that can be useful in preliminary design.

[4] Boeing company: http://www.boeing.com/resources/boeingdotcom/space/boeing_satellite_family/pdf/Bkgd_702SP.pdf [Accessed 25 November 2015]

[5] Artemis mission update: http://www.esa.int/Our_Activities/Telecommunications_Integrated_Applications/Artemis_finally_reaches_operational_orbit [Accessed 18 October 2015]

[6] AEHF-1 mission update: http://spaceflightnow.com/atlas/av019/111009.html [Accessed 18 October 2015]

Table 3.5: Previous and current Mars low-thrust trajectory research.

Author	Organisation	Country	Year	Program name
Geffroy and Epenoy [16]	CNES	France	1997	Unknown
Kluever and Oleson [17]	University of Missouri-Columbia and NYMA, Inc.	USA	1998	Unknown
Bertrand et al. [18]	CNRS & CNES	France	2001	Unknown
Whiffen [19]	NASA JPL	USA	2006	Mystic
Sims et al. [20]	NASA JPL	USA	2006	MALTO
Kos et al. [21]	NASA JPL	USA	2006	MALTO, Mystic, Copernicus, OTIS, SNAP, CHEBYTOP, VARITOP, SEPTOP, NEWSEP and Sail
Derz and Seboldt [22]	DLR	Germany	2012	InTrance

Bertrand et al. [18] Here, low-thrust propulsion was used for the heliocentric stages and the Mars insertion and escape. Chemical propulsion was however still used for the final Mars rendezvous. In this case again a low-thrust planetocentric optimisation tool was used that is based on averaging techniques in optimal control. The used tool was developed by Centre National d'Études Spatiales (CNES) and solves the two-point boundary value problem. In conclusion it is mentioned that improvements can be made to this tool: including variable power, mixed minimum-time/fuel-saving criterion, atmospheric drag and Van Allen degradation (in case of LEO insertion) and better transition between planetocentric and heliocentric mission phases. It was also concluded that a fully electric vehicle would be a better choice (thus getting rid of the chemical propulsion for rendezvous) and will then be used for rendezvous as well, but this requires more research.

Whiffen [19] This paper describes the Mystic program used for low-thrust trajectory calculations. The optimisation algorithm used is called Static/Dynamic Optimal Control and is a non-linear optimal control method that can optimise static and dynamic variables at the same time (it is based on Bellman's principle). The software can be used for planetocentric low-thrust optimisation.

Sims et al. [20] This paper describes the MALTO program used for low-thrust (and can also be used for high-thrust) trajectory optimisation. The trajectory problem is a non-linear optimisation problem which can be solved in MALTO using the non-linear programming software called SNOPT. In this paper the direct MALTO program was compared to the indirect SEPTOP program used previously for many mission design cases. MALTO showed similar final mass estimations but has much less convergence sensitivity and can incorporate many more intermediate flybys (up to 12 were tested). The software can be used for planetocentric low-thrust optimisation.

Kos et al. [21] This paper shows an overview of the different low-thrust trajectory analysis tools developed by NASA. A comparison is made between the different programs and it is specified that the users will have to decide for themselves which program fits best with their problem. SNAP is the primary tool to be used for planetocentric optimisation of low-thrust trajectories. All the programs are coded in Fortran. The paper also provides information on how to obtain the programs (also through a website, which is not online anymore).

Derz and Seboldt [22] In this paper a European MSR mission is envisioned through two different architectures (either two separate launches or one combined launch). In this research the orbiter utilised a low-thrust propulsion system as its main propulsion system. The orbiter would go into a 1000 km parking orbit around Mars waiting for the sample to be brought into orbit by the MAV. The low-thrust trajectories were optimised for minimal flight time using the InTrance program. This program optimises through the use of artificial neural networks and evolutionary algorithms. For the integration, Runge-Kutta-Fehlberg 4^{th} (5^{th}) order (RKF45)) was used and JPLs DE405 ephemerides for Earth and Mars were incorporated as well. It was concluded that a low-thrust option for MSR is a good alternative to the European Space Agency (ESA) proposed high-thrust option. But, because of the power requirements, the s/c configuration should be investigated more.

This is a selection of research and programs that have been used in the past few years. Much more research has been performed concerning the Earth raising orbits and station keeping, however this is a different application than in the proposed research problem, which is why these are not discussed in this report.

3.3. ASCENT VEHICLE AND ORBITER DIRECT RENDEZVOUS REFERENCES

In this case, the definition of direct rendezvous is to get to the same point (or at least control box) in orbit at the same time as the orbiting s/c within one revolution. Often a certain rendezvous orbit is chosen and once in this orbit, the s/c performs a so-called phasing manoeuvre to get closer to the orbiting s/c. Once inside the control box, the close rendezvous begins, which is not part of this study. In this section reference missions are provided which have performed or which came close to single-revolution rendezvous. Also, the reference research performed in this particular field will be described.

3.3.1. FLOWN MISSIONS

There have been many rendezvous missions, some of which specific to a celestial body launch and rendezvous with an orbiting vehicle. For example, all the Apollo missions were able to rendezvous with an orbiting s/c after ascending from the Moon. And in more recent years there have been many rendezvous with the International Space Station (ISS). In Table 3.6 the missions with a 4 or less revolutions before rendezvous after launch are shown. It also shows an estimate of the number of revolutions it took to successfully rendezvous with the orbiting s/c.

Table 3.6: Previous (nearly) direct rendezvous Missions.

Year	Country	Mission	Orbiting body	Revolutions
1965	USA	Gemini 6A and 7 [23]	Earth	4
1966	USA	Gemini 8 [24]	Earth	4
1967	Soviet Union	Kosmos-186 and 188 [25]	Earth	1
1969	USA	Apollo 11 [10]	Moon	2
2012	Russia	Progress-M-15M [26]	Earth	4

It is interesting to observe that so-far only the Russians have performed a single-revolution rendezvous[7]. However, it should be noted that this was achieved with two unmanned s/c. Also, for the Gemini missions, single-revolution rendezvous (called first-apogee plan) was one of the three rendezvous plans considered [24]. Though in the end it was decided that due to both extra stress on the astronauts, because all the procedures would have to take place in a shorter time, and due to the inaccuracy of the orbit insertions at the time a first-apogee rendezvous would be too risky and would need a back-up plan to deal with uncertainties. Instead it was decided to use the coelliptical plan. A more recent application is the Russian rendezvous with the ISS. As mentioned in the table, the Progress-M-15M was the first to demonstrate a shorter rendezvous strategy. Before this strategy was employed, the trip from Earth to the ISS took at least two days, however due to advancements in, among others, satellite navigation technology and high-performance computers, the transfer could now be achieved in a much shorter time [26]. This strategy is currently also applied to the manned Soyuz missions to the ISS. Even shorter transfers are being considered. In the paper it is mentioned that a 3-revolution period is possible but it could even go down to (a fraction of) 1 revolution.

3.3.2. DIRECT RENDEZVOUS REFERENCE RESEARCH

As mentioned in Section 3.3.1, research has already been performed in the past on the Gemini missions and is currently being done for missions to the ISS. However, more research has been conducted focused on single-revolution rendezvous or at least close to one revolution. An overview of this research can be found in Table 3.7 followed by a short description.

[7] Personal account: http://www.svengrahn.pp.se/trackind/K186188/K186188.html [Accessed 20 October 2015]

Table 3.7: Direct rendezvous reference research.

Author	Organisation	Country	Year	Application
de Almeida Prado [27]	Instituto Nacional de Pesquisas Espaciais	Brazil	1996	Rendezvous between two orbits
Woolley et al. [28]	NASA JPL	USA	2011	MSR

de Almeida Prado [27] In this paper, two orbits were selected and specifically two points in those orbits. Then the optimum rendezvous is computed starting in the first point in the initial (lower) orbit and arriving in the second point in the final (higher) orbit. This is achieved using less than one, one, or more revolutions. The Lambert problem is used and solved to find the required parameters for a given number of revolutions.

Woolley et al. [28] Different strategies for Martian ascent and rendezvous are presented. Provided a set of requirements, the proper launch and rendezvous trajectories were found. In this case, the MAV was positioned in an orbit in front (could be slightly below) of the orbiter, because of the line-of-sight requirement in this case. It was concluded that optical detection and orbit determination for this particular MSR architecture would be a viable option.

As presented in Table 3.7 not much research has been conducted concerning the special case of direct launch single-revolution rendezvous. Also, considering that single-revolution rendezvous has been performed in the past and that for manned missions to the ISS it is still considered, it will be interesting to compare the results of the thesis work to other rendezvous options.

3.4. MAV DESIGN

Martian sample return has never been attempted, as a matter of fact a sample return from any celestial body with an atmosphere has never been done before. When designing a Martian sample return mission, a vital part is transporting the samples off the planet, using an MAV. This vehicle can then either be sent directly to Earth or rendezvous with an s/c (either orbiting the planet or not). This section will focus on the different possible MAV designs that have been considered and/or proposed (Section 3.4.1). Most of these studies were based on a set baseline design that has changed over time due to continuing changes in mission design and proposed date. Therefore Section 3.4.2 will outline the current MAV baseline design. There is however still some flexibility in the baseline design. This design space can be used during the optimisation to change the design slightly if required. The exact design space will be described in Section 3.4.3.

3.4.1. PREVIOUS INVESTIGATIONS

Many MAV design studies have been performed. A number of these studies is shown in Table 3.8. For each research study, the main launch concept is provided and the kind of propellant(s) as well. The studies are presented in order of publication year.

From the table it is clear that all studies envision a rocket to bring the samples either into Martian orbit or directly back to Earth. Also, four clear propellant types have been investigated: the traditional solid and liquid propellant engines, the hybrid engine (which is a combination of a solid fuel and a liquid oxidizer) and the gel engine.

In [7] it is shown that the GLOM would be lowest when using a two-stage liquid rocket. Nevertheless, a two-stage gel rocket would be a reasonable alternative. However, a low GLOM is not the only requirement. A visual representation of some of the described designs is presented in Figure 3.1.

3.4.2. CURRENT MAV BASELINE DESIGN

The latest baseline design for the MAV originates from a (number of) studie(s) conducted in 2012 (also see Table 3.8). The main properties of the current baseline are presented best by [7]. In this paper, the rocket depicted in Figure 3.1,d) was analysed and put forward as the baseline design with the orbital sample container located in (and acting as) the nose of the rocket. A more detailed view of the baseline MAV is shown in Figure 3.2. Here, the samples are contained in the Orbiting Sample (OS), the thrust vector is controlled by the

Table 3.8: Previous MAV studies.

Author	Year	Launch method	Propellant(s)
Whitehead [29]	1997	two-stage rockets (comparison study)	solid and liquid (and gel recommended as well)
Guernsey [30]	1998	two-stage rocket	2x liquid
Desai et al. [3]	1998	two-stage rocket	2x liquid
Stone [31]	1999	two-stage rocket	hybrid
Stephenson [32]	2002	two-stage rocket (three different designs)	2x solid (best), solid and liquid or hybrid and 2x gel (best)
Whitehead [5]	2005	one-, two- and three-stage rockets (variational study)	solid and liquid
Stephenson and Willenberg [33]	2006	two-stage rocket	2x solid
Sengupta et al. [34]	2012	two-stage rocket	2x liquid
Trinidad et al. [7]	2012	1 to 4 stages (comparison study) two-stage rocket (best)	solid, liquid, hybrid and gel (comparison) 2x liquid (best)
Mungas et al. [35]	2012	single-stage rocket	liquid mono-propellant
Mars Program Planning Group [1]	2012	undefined rocket	solid and liquid

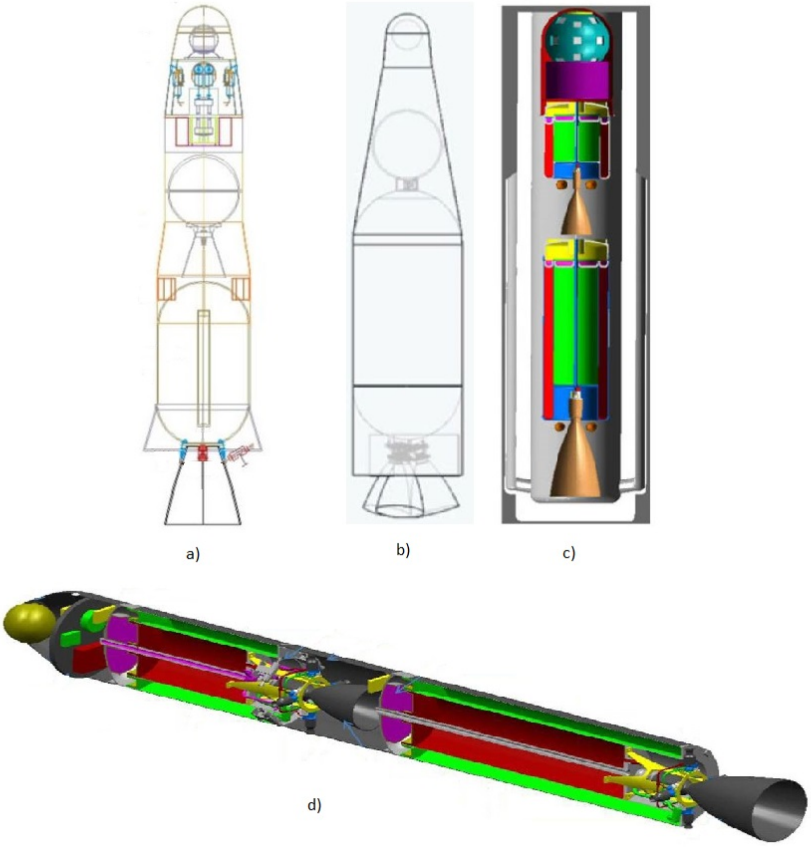


Figure 3.1: A number of rocket concepts visualized. a) Two-stage solid rocket by Lockheed Martin [32], b) Single stage mono-liquid rocket by Firestar Technologies [35], c) Two-stage gel rocket by Thompson Ramo Wooldridge (TRW) [32] and d) Two-stage bi-liquid rocket by Boeing and Northrop Grumman [7].

Thrust Vector Control (TVC) engines (or Reaction Control System (RCS)) and the final attitude adjustments are made by the Attitude Control System (ACS) thrusters.

The design resulted from the requirements for an MAV put forward back in 2010. The main characteristics

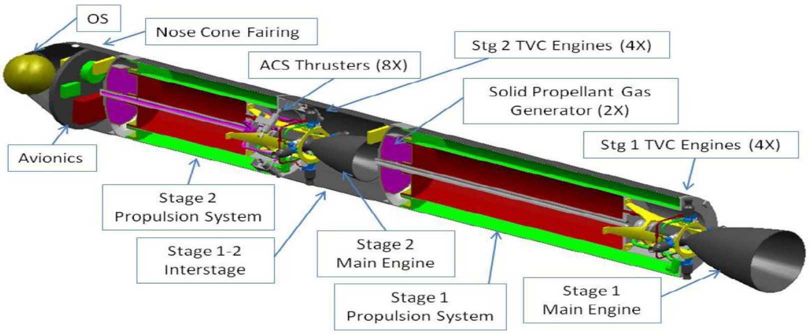


Figure 3.2: Two-stage bi-liquid baseline MAV with descriptions of different components [7].

of the baseline MAV are summarised in Table 3.9. It should be noted that since the current baseline design is a pressure-regulated system, the thrust is constant throughout the flight [36].

Table 3.9: MAV baseline design characteristics [7].

Characteristic	**Value or property**
Martian Sample Payload	5 kg
Stage 1 main engine thrust	5.3 kN (restartable)
Stage 2 main engine thrust	2.7 kN (restartable)
Stage 1 RCS engines	445 N (4x)
Stage 2 RCS engines	445 N (4x)
Stage 2 ACS engines	4.4 N (8x)
Main engine propellants	MON-25 (Oxidizer), MMH (Fuel) (2x)
RCS propellant (hot-gas)	Segmented solid propellant gas generators (also, liquid tank pressurant)
Total MAV mass	227 kg (283 kg with contingencies and 391 kg with extreme contingencies)
I_{sp}	330.5 s vacuum, 328.6 s in Martian atmosphere
Diameter	340 mm
Total length	approx. ≤ 3666 mm
Max. launch angle	± 35° off vertical axis
Target orbit	460 by 580 km

3.4.3. CURRENT DESIGN RESTRICTIONS

The baseline design described in Section 3.4.2 was based on early design restrictions set by the MAV Study Guidelines 2010 (as mentioned in [7]). These include those specified in [37]. The restrictions provide flexibility in the design space (the upper and lower bound of the values specified in Table 3.9), should any of the baseline design parameters be changed slightly to result in a more optimum ascent. In this case, some design parameters could be treated as optimisation parameters (if required).

- The total system, which is the MAV and the erector support system (tower) to launch it, have to have a total mass of less than 360 kg [7]
- Updated volume requirement: ranging from 3666 mm long with a diameter of 350 mm to 3150 mm long with a diameter of 700 mm [7]
- 45° ± 0.2° target orbit inclination [37]
- Ability to launch from ± 30° latitudes [37]
- Sample sphere diameter: 16 cm (5 kg) [37]

4

Optimisation

In this chapter, different optimisation techniques will be discussed. There are two different kinds of optimisations: local optimisation, where a minimum (or maximum) is found, which may be the minimum within a small area in the function space and which does not have to be the absolute optimum of the function, and global optimisation, where the found minimum (or maximum) is ideally the minimum value of the entire function space [38]. Such a function space can contain multiple local minima but only one of these is the global minimum. A general description of the local and global optimisation techniques is provided in Sections 4.1 and 4.2 respectively. The different optimisation techniques are compared in Section 4.3 and two methods are chosen through a trade-off in Section 4.4. Finally, the chosen methods will be described in more detail towards implementation in Sections 4.5 and 4.6. An optimisation problem can either be solved analytically (for simple problems), resulting in a direct solution, or numerically (for more intricate problems), approaching the exact solution [39]. When comparing different optimisation techniques usually cpu time is traded against the accuracy of the result or the robustness of the method. Often a more accurate or reliable result requires more time and thus increases the cpu time required. From now on only minima are discussed, but every method can also be used to find the maximum.

4.1. Local optimisation methods

Local optimisation methods aim to find the local optimum of a function within a limited search space. The methods used are called Nelder-Mead (based on a simplex around the initial guess which then progressing in the direction of the best solution), Newton-Raphson (a gradient technique that requires derivative information and works for 1-dimensional problems, the method moves in the direction of the largest derivative), steepest descent (either along the axes or in an arbitrary direction, similar to Newton-Raphson, but now for 2-dimensions) and Sequential Quadratic Programming (SQP) (an approximation of the function that has to be optimised is provided by a quadratic function, which is evaluated and updated until a local optimum is found or a certain cut-off criterion is met) [39, 40]. Local optimisation methods are often fast and accurate, producing good results in a short time. But they only focus on a small area of the function space [41]. Therefore it is important to define a proper initial parameter value for the local methods (for instance by first using a global optimiser).

If many local optimisations are used spread over the entire function search space, the chances that the global optimum is found is increased. A method based on multiple local optimisations is Monotonic Basin Hopping (MBH). This method uses the fact that local optima tend to be grouped together in many problems. The local optimum of such a group is found by hopping within this group. Once the local optimum of such a group is found, the process starts again, but this time somewhere else in the search space. In this way, the global optimum might be found.

4.2. Global optimisation methods

In [39], two main numerical global optimisation categories are mentioned: sampling methods and metaheuristics. In global optimisation the minimum value of the complete function is strived for, or at least an approximation of this value.

4.2.1. SAMPLING METHODS

The sampling methods are based on selecting a value for the variable(s) in the function and then evaluating the function. This is done for many different values until a minimum is found and can be accomplished randomly (Monte Carlo) or using a certain pattern (grid search, Latin hypercube sampling). Because Monte Carlo is based on the evaluation of the function using randomly generated values for the variables, many different points are often required, which increases the required cpu time. Also, the final resolution depends on the resolution of the random number generator. In grid search the value space is divided up into a number of different possible values, where the amount of points depends on the desired resolution. In this case the function is evaluated starting with the first values until each value has been used, and the minimum function evaluation is stored. Again, the number of points is very important and often many are required. Latin hypercube sampling is a method based on matrices [42], where an n-dimensional matrix is generated such that each random value is only used once in each row and column (for a 2-dimensional case). From this matrix, in case 2-dimensions are used, samples are chosen such that each row and column only provides one sample. The variable values can also be generated quasi-random, which often involves generating all the different values at the start of the optimisation and then evaluating the function for every of these values. An example of such a method is Sobol sequencing [43], where all the values are generated using a pre-set function developed by Sobol. Sampling methods can be used to make a rough inventory of the total search space and reduce this for a local optimiser since the resolution of sampling methods can be limited. However, should a sampling method be applied to a small function space, accurate results can still be obtained, which is why it can also be used for local optimisation.

4.2.2. METAHEURISTICS

Metaheuristics are numerical optimisation techniques that directly propagate towards the solution. Unfortunately, full convergence cannot always be achieved, and usually a local optimisation gradient method is required to refine the final solution. The different methods are: Genetic Algorithm (GA), DE (both based on Evolutionary Algorithm (EA)), Particle Swarm Optimisation (PSO), (Adaptive) Simulated Annealing (SA), Ant Colony optimisation, Dynamic Programming and Interval Analysis [1] [39, 40]. Often these methods require many iterations to reach a proper solution, however, they can be very robust and do not always require an accurate initial value.

4.3. TECHNIQUE COMPARISON

To gain a better understanding of the different methods mentioned in Sections 4.1 and 4.2, a comparison is made between all of them. Table 4.1 shows the advantages and disadvantages of each method using a similar representation as provided in [39]. For metaheuristics the general advantages are: robust, do not depend on derivatives, do not require a good initial value and propagate directly towards a solution. The general disadvantages are: need many iterations, convergence is unclear and become very slow for a large number of variables. Also, often metaheuristics require a local optimisation method to reach the precise final optimum. In the table the advantages and disadvantages that separate the different metaheuristic methods are mentioned. A global optimum is preferred.

Table 4.1: Optimisation method comparison [39] (unless mentioned otherwise).

Method	**Type**	**Advantage**	**Disadvantage**
Nelder-Mead	Local	simple	converges to local minimum
Newton-Raphson	Local	simple	convergence is slow near a flat optimum and converges to local minimum
Steepest descent	Local	simple, always finds a minimum and only requires previous gradient information (arbitrary direction results in better conversion)	can oscillate around minimum, has a slow convergence and converges to a local minimum (needs second derivatives/Hessian matrix for arbitrary directions)
SQP [44]	Local	works well on non-linearly constrained problems and neither initial point or iterate points have to be feasible	converges to local minimum

[1] Usually using a branch and bound method

MBH	Local/ Global	can locate several local minima that are close together [40]	no guarantee that best found local minimum is also global minimum and normal Basin Hopping is preferred when searching one local minimum [45]
Monte Carlo	Global/ sampling	simple to implement and does not depend on derivatives	only simple problems (otherwise very slow), it is based on luck and can have bad resolution (approximation of value)
Grid search	Global/ sampling	simple to implement and does not depend on derivatives	slow and has to trade resolution against cpu time
Latin hypercube sampling [42]	Global/ sampling	simple to implement, does not depend on derivatives and it provides more regular sampling	only simple problems (otherwise very slow)
Sobol sequencing [43]	Global/ sampling	good uniform distribution (regular sampling), faster convergence and does not depend on derivatives	all sampling values are generated at once, and performs poorly on higher-dimensional problems
GA	Global	simple, easy to understand	original formulation uses binary numbers to encode variables
DE	Global	simple, uses actual variable values	original DE does not perform well for multi-objective problems and was not intended for binary code use [46]
PSO [47]	Global	does not use mutations, very fast, simple, based on intelligence and can be used for multi-objective problems	problem has to have a specific structure
SA [48, 49]	Global	easy to program, can be applied to many different problems, can deal with many constraints	fine-tuning of the constraints and variables can be delicate, often requires much storage space and cpu time
Ant Colony Optimisation [47]	Global	can be solved using parallel computations, can handle dynamic problems	based on a sequence of dependent random decisions, and works best for experimental problems
Dynamic Programming [50]	Global	solving sub-problems to find problem solution and stores solutions to sub-problems as to only evaluate every sub-problem once	can require much storage space with increase of different sub-problems
Interval Analysis [51]	Global	includes different error estimations, can be used for infinite data sets	can be complex

4.4. CHOSEN METHODS

It is important to use an optimisation method that is best suited for the thesis problem, or a method that shows potential in becoming the best optimisation method for this particular kind of problem. The thesis problem might use two different methods and compare the performance of these methods or even combine them to produce an even more suited method. It could also be that simply one optimisation method is chosen and the focus of the thesis will be put on other aspects. This final decision of what to focus on will be made in Chapter 11. To accommodate this decision, two optimisation methods (out of the ones provided in Table 4.1) will have to be selected. To reduce the amount of method used in the trade-off and to help establish the different criteria, reference research and TU Delft (Space department) heritage will be used. Basing the decision on what knowledge already exists in the department would allow for use of resources that are already available and would make it easier to learn from experienced people. However, other used methods for similar subject problems could not be ignored, since one method might be better suited for the problem than the other. Therefore, Section 4.4.1 first describes a collection of methods that have frequently been used for similar problems and then Section 4.4.2 will discuss several methods used in the Space department. The final trade-off will be described in Section 4.4.3.

4.4.1. FREQUENTLY USED METHODS

The required method depends on the problem that needs to be solved. To help determine which methods to use for the thesis problem, reference optimisation research was consulted. An overview is presented in

Table 4.2. In this case only research focused on finding the global optimum is provided because the thesis problem will also focus on finding the global optimum. Most of the problems mentioned in Table 4.2 required metaheuristic methods. Addis et al. [52] mentions that DE is very popular because of its robustness resulting in good results, but that MBH might be a better alternative because it can produce even better results.

Table 4.2: Previous methods used for trajectory and orbit transfer optimisation problems.

Author	**Year**	**Subject**	**Method**
Gage et al. [53]	1995	Interplanetary Trajectory (Mars missions)	GA
Rauwolf and Coverstone-Carroll [54]	1996	Low-thrust orbit transfers (to Mars and Mercury)	GA
Kim and Spencer [55]	2002	Spacecraft Rendezvous	GAs
Myatt et al. [56]	2004	Mission Analysis and Design (transfer orbits)	DE (most robust)
Lee et al. [57]	2005	Low-thrust orbit transfers	Q-law with GA and Q-law with SA
Abdelkhalik and Mortari [58]	2007	Transfer orbits	GAs
Vasile et al. [59]	2008	Space trajectories	MBH
Garcia et al. [60]	2010	Optimisation of Mars entry vehicle for a mass of 40 tons	(Multi Objective) GA
Li and Peng [61]	2011	Mars entry and descent	SQP and Monte Carlo
Addis et al. [52]	2011	Space trajectories	MBH

4.4.2. TU Delft optimisation heritage

At the Space department of the TU Delft, a toolbox was developed called Tudat. This toolbox can be used to solve a variety of (aspects of) astrodynamic problems. Within this toolbox, an optimisation toolbox is used called Parallel Global Multi-objective Optimizer (PaGMO), which houses a selection a selection of optimisers mentioned in Table 4.1 and more [62]. Since the introduction of this toolbox, students have been working on these optimisation programs within this toolbox, often adjusting it slightly to fit their specific needs. A program using a PaGMO optimiser was developed by Musegaas [40]. These developed programs were also adjusted to incorporate specific problem needs such as was done by Miranda [63] where the original program used DE and PSO was added to produce a better solution. It is however unlikely that Tudat programs itself will be used during this thesis, nonetheless the experience is there and can still be used. An overview of the different research performed by previous master students and the used (and best) optimisation methods is provided in Table 4.3.

Table 4.3: Previous master theses on trajectory and orbit transfer optimisation problems.

Author	**Year**	**Subject**	**Method**
Pagano [64]	2010	Launch trajectories	PSO
Musegaas [40]	2012	Space Trajectories (including gravity assists and manoeuvres)	GA, PSO and DE (best)
Vandamme [65]	2012	Launch trajectories	PSO and DE (best)
van Kesteren [66]	2013	Launch trajectories	GA, PSO and DE (best)
Boudestijn [62]	2014	Low-thrust trajectories	DE
Gomez [67]	2015	Low-thrust trajectories	DE
Miranda [63]	2015	Hybrid rockets	DE and PSO (best)

4.4.3. Chosen candidate methods

The decision of which methods to use will be based on the heritage, reference research, problem type (here launch and space trajectories), accuracy, speed, robustness, easy implementation and novelty. Also, the more resent the research the more relevant it is deemed to be. Based on Tables 4.2 and 4.3 it was decided to include

GA, PSO, DE and MBH in the trade-off. The trade-off is visualised using Table 4.4. Every criterion is assigned a weight depending on how important the criterion is. Then the different methods are either green (1), yellow (0.5) or red (0) for every criterion. This results in a final score.

Table 4.4: Optimiser trade-off table.

	Heritage	**Ref. research**	**Trajectories**	**Accuracy**	**Speed**	**Robust**	**Impl.**	**Novelty**	**Score**
Weight	*4*	*4*	*5*	*4*	*3*	*3*	*2*	*5*	
GA	🟥	🟩	🟨	🟨	🟨	🟨	🟩	🟥	13.5
PSO	🟨	🟥	🟨	🟨	🟩	🟨	🟩	🟨	15.5
DE	🟩	🟩	🟩	🟨	🟨	🟩	🟩	🟨	24
MBH	🟥	🟩	🟩	🟩	🟨	🟩	🟩	🟩	24.5

Currently, the method that is used most is DE, which also results in a high score in the trade-off table. However based on the reference research it is clear that even though DE is very popular even outside of the TU Delft, MBH could be an attractive alternative. Indeed, MBH scored similarly to DE in the trade-off (and even slightly better). This is why it was decided to investigate both DE and MBH. This is also interesting from an academic point of view, since there is currently a discussion on which method is better for trajectory problems. The next two sections will discuss each of the chosen methods in more detail and explain how they can be implemented.

4.5. Implementation of DE

Differential Evolution is a direct global method based on EAs and was first introduced by Storn and Price back in 1995 [68]. The method became increasingly popular after performing well in two world-wide optimisation competitions [69] and finally caught the interest of the scientific and engineering community in 1997 when two more papers were published of which the final paper [70] provided a clear and easy-to-understand description of the method. This description will also be used in Section 4.5.1 to explain the basics of DE. In Section 4.5.2 variations on the original DE method are explained.

4.5.1. Workings of DE

When using DE for a minimisation problem the objective is to find the minimum global function evaluation. For instance, if the aim is to find the minimum propellant mass to reach a certain point in a certain orbit then the function will be written to compute the required propellant mass given a combination of different variables (such as launch azimuth and lift-off mass). These variables values are initially randomly chosen by the DE algorithm and their combination is optimised to reach a minimum propellant mass solution. In such a case, the minimum propellant mass function is called the cost function. Because DE is based on EAs it works with "individuals", "population size", and "generations". An individual is a vector, with all the different variables in it, of size D (the number of variables). The values of these variables are initially randomly chosen for each individual before the DE optimisation, within the different variable constraints (parameter space). The population size (NP) determines the number of individuals used during the optimisation. Before the start of the optimisation the first generation is generated randomly within the mentioned constraints. Each time an optimisation run is performed, a new generation (G) is created with new (improved) individuals, or including the best individuals of the previous generation depending on how well they performed. This means that the optimisation will run until a maximum number of generations is achieved, which means that the maximum number of generations is a set value by the user. The individual vectors per generation can be written as shown in Equation (4.1) [70].

$$x_{i,G} \quad \text{with} \quad i = 1, 2, \ldots, NP \tag{4.1}$$

To generate new individual vectors during the optimisation (new generation), DE adds the weighted difference between two individual vectors to a third vector, a process called mutation, and then mixes this new mutation vector $v_{i,G+1}$ with the original, target, individual vector $x_{i,G}$ resulting in a trial vector $u_{i,G+1}$, a process called crossover. Here, the three individual vectors used to compile the mutation vector are all distinctly different from the original individual vector. This then calls for the requirement that NP is at least equal to 4. In equation form this means that first the mutant vector is created through the use of Equation (4.2) (visualized for a two-dimensional cost function in Figure 4.1).

$$v_{i,G+1} = x_{r_1,G} + F \cdot \left(x_{r_2,G} - x_{r_3,G}\right) \tag{4.2}$$

where in Equation (4.2) $r_1, r_2, r_3 \in \{1, 2, \ldots, NP\}$ are all different and different from i. Also $F \in [0, 2]$, where F is called the amplification factor (or weight) which determines the impact of the difference vector to the first chosen $x_{r_1,G}$ vector and is selected by the user. It should be noted that in case of a strictly constraint problem, such as the proposed thesis topic problem (constraints on MAV design), each of the new variables created through mutation should be checked for plausibility. This means that the value for the different variables should always be within the respective parameter space.

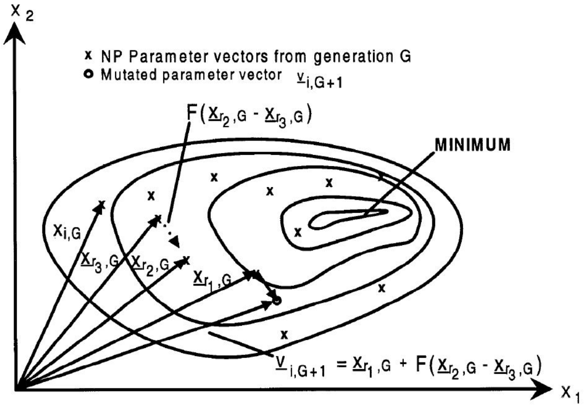

X 2
× NP Parameter vectors from generation G
○ Mutated parameter vector v i,G+1
F(x r2,G - x r3,G)
MINIMUM
x i,G
x r3,G
x r2,G
x r1,G
v i,G+1 = x r1,G + F(x r2,G - x r3,G)
X 1

Figure 4.1: Construction of the mutation vector in a two-dimensional case [70].

The crossover/trial vector is composed of the variables from both $x_{i,G}$ and $v_{i,G+1}$. Each different variable is denoted by its separate position number j (thus $j = 1, 2, \ldots, D$). Using this notation the trial vector can be described as $u_{i,G+1} = \left(u_{1i,G+1}, u_{2i,G+1}, \ldots, u_{Di,G+1}\right)$ where the determination of which variable value originates from which vector ($x_{i,G}$ or $v_{i,G+1}$) is determined by Equation (4.3) and visualized by Figure 4.2 for a 7-dimensional case.

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1}, & \textbf{if } \left(randb\left(j\right) \leq CR\right) \textbf{ or } j = rnbr\left(i\right) \\ x_{ji,G}, & \textbf{if } \left(randb\left(j\right) > CR\right) \textbf{ and } j \neq rnbr\left(i\right) \end{cases} \tag{4.3}$$

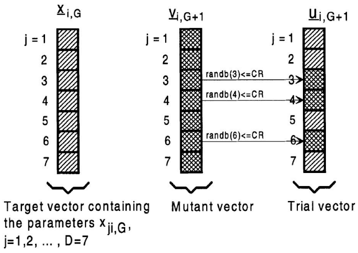


Figure 4.2: Construction of the trial vector in a 7-dimensional case [70].

where in Equation (4.3) $randb\left(j\right) \in [0, 1]$ and is the jth evaluation obtained from a uniform random generator [70]. Also, CR is called the crossover constant which is chosen by the user $\in [0, 1]$ and $rnbr\left(i\right)$ is again a random chosen number. This ensures that at least one variable from the mutation vector will become part of the trial vector. The last step is to determine whether the trail vector $u_{i,G+1}$ generates a better value, or lower propellant mass in case of the example, than the original, target, individual vector $x_{i,G}$. If the trial vector performs better, then the ith individual vector in the next generation $x_{i,G+1}$ becomes $u_{i,G+1}$. However, if it performs worse than $x_{i,G}$, then $x_{i,G+1}$ becomes $x_{i,G}$. This is done for every individual vector in the population. Once the entire population in a generation has been examined, and has either been replaced by a better combination or not, the next generation is examined until the final generation is reached. A pseudo-code for this process written in C is provided in the paper of Storn and Price [70] and a specific example of this DE application is presented in [71] for both C and Matlab.

4.5.2. Variations of DE

In their 1997 publication, Storn and Price also mention that the DE method described in Section 4.5.1 is not the only way to implement DE. In fact they make a selection of three distinct changes that can be made to vary the method. The first change would be that, instead of taking a random individual vector $x_{r_1,G}$ as the vector that will be mutated shown in Equation (4.2), one could also mutate the best individual vector (so the individual vector with the lowest value for the cost function that is not equal to the current (i) vector being examined). In that case Equation (4.2) would change to Equation (4.4).

$$v_{i,G+1} = x_{best,G} + F \cdot \left(x_{r_1,G} - x_{r_2,G}\right) \tag{4.4}$$

The second change would be to use more difference vectors. In the case of Equations (4.2) and (4.4) only one difference vector is used. Should, for instance, two difference vectors be used then Equation (4.4) would change into Equation (4.5) [70]. This would however also affect the constraint for NP, which would in this case increase to a minimum of 6 (every added difference vector raises the minimum NP by 2).

$$v_{i,G+1} = x_{best,G} + F \cdot \left(x_{r_1,G} + x_{r_2,G} - x_{r_3,G} - x_{r_4,G}\right) \tag{4.5}$$

The final change would be in how the crossover is determined. Currently this is done using an independent binary (bin) experiment. The notation used to differentiate between the different options would be DE/rand/1/bin for the method described in Section 4.5.1, DE/best/1/bin if Equation (4.4) were to be used and if Equation (4.5) would be used, the notation would change to DE/best/2/bin, just to give a few examples.

Concerning the different control variables (NP, F and CR), a different optimiser could also be incorporated to optimise these parameters each time to gain the fastest result depending on the kind of problem. Storn and Price [70] do, however, provide a fair range and approximation of values for the control variables. As by Storn and Price: "F = 0.5 is usually a good initial choice. If the population converges prematurely, then F and/or NP should be increased. Values of F smaller than 0.4, like those greater than 1, are only occasionally effective. A good first choice for CR is 0.1, but since a large CR often speeds convergence, to first try CR = 0.9 or CR = 1.0 is appropriate to see if a quick solution is possible. For fastest convergence, it is best to pick the initial parameter range such that it covers the region of the suspected global optimum, although this choice doesn't seem to be mandatory." They also mention that a reasonable choice for NP is between 5 times D and 10 times D, however NP should always suffice to the minimum requirement as per number of difference vectors.

A more recent publication by A. Qing in 2009 [72] shows a clear overview of the advances in the field of DE since 1997. In this book four different DE categories are described: classic, dynamic, modified and hybrid. Classic is the classic method described in Section 4.5.1 and slight modifications to it. As already mentioned it is important to deal with the fact that variables might need to stay within a certain parameter space. Qing describes two methods to deal with mutants that are not feasible: random reinitialization (where a new random variable is generated if the current variable is invalid) and bounce-back (where the value of the variable is brought back into the parameter space on either the lower bound or the upper bound, depending on where it left the parameter space). It is also mentioned that should the optimum be known, the program can be terminated whenever the solution is found within a certain accuracy. The program could also be terminated if the population diversity is very small (smaller than a set limit).This means that almost all the individuals are the same and a solution might not be found. In [72] it is mentioned that Storn and Price recommend that NP = 10 times D, F = 0.8 and CR = 0.9. However, F = 0.5 and CR = 0.1 are also claimed to be a good first choice, which is similar to the recommendations found in [70]. [72] also states that it is claimed that convergence is more likely to occur but generally takes longer with larger populations and weaker mutation (smaller F). CR = 1 results in faster convergence if convergence occurs but CR = 0 is required to make DE robust enough for particular problems. Dynamic DE is similar to classic but is not based on generations. Dynamic DE has one population where the better children continuously replace the parents until a certain number of function evaluations has been reached. Also, the best individual is immediately updated every iteration round. Because it is all part of the new iteration, computations cannot be done in parallel any more. In modified DE one or several of the following core aspects of classic DE is modified: population initialization, differential mutation, crossover, objective and constraint function evaluation, and selection. An example of modified DE is provided in [71]. And finally hybrid DE involves combining DE with a deterministic and/or a stochastic optimiser [72]. Qing also shows different ways to determine the control parameters NP, F and CR, plus different ways to perform a multi-objective DE, where one would like to optimise different objectives at the same time.

It is clear that DE is flexible and can be adjusted to different kinds of problems, and what specifically to change depends on the user and the problem itself. It is the choice of the user to specify in what kind of configuration DE is to be used.

4.6. IMPLEMENTATION OF MBH

Monotonic Basin Hopping is an optimisation method initially developed by Leary in 2000 [73] based on Basin Hopping first introduced by Wales and Doye in 1997 [74]. It depends on many local optimisations to eventually find a global optimum [40, 59, 75], which is why it can also be classified as a combination of a local and global optimiser (depending on the chosen definition). In 2008 Addis et al. [76] states that MBH outperforms DE for given optimisation problems, which makes comparing these two methods for this particular thesis topic very interesting. In Section 4.6.1 the workings of MBH will be explained and any available variations on this method will be discussed in Section 4.6.2.

4.6.1. WORKINGS OF MBH

The basic form of MBH is probably best explained through the description provided in [75]. The method begins by setting a so-called improvement counter to zero: $N_{n,i} = 0$. This counter is used to determine how many times a local optimisation does not result in a better solution than the existing best minimum. Then a set of randomly chosen variable values is generated to use as an initial point/guess. This initial guess vector is called x and is used as the first input for a (by the user) chosen local optimiser. This local optimiser then tries to determine the local minimum point x^* (also called a "basin"), and if it finds a feasible solution, this local minimum is set as the current point and called $x_{current}$. If the local optimiser does not find a feasible solution, the program starts over again and chooses a new random point to serve as a new initial guess. This first step is visualized in Figure 4.3 for a one-dimensional case.

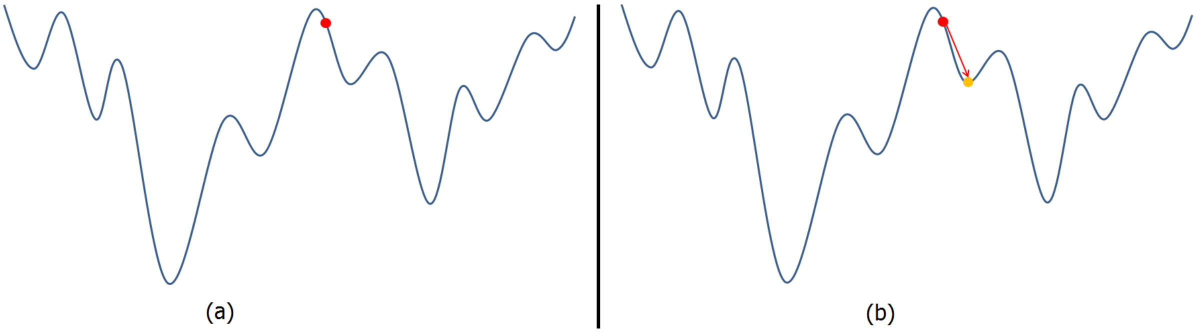


Figure 4.3: The first step of MBH where (a) first a random initial point x (red dot) is generated after which (b) the local minimum x^* (dark yellow point) is found through the use of a local optimiser. The local optimisation is visualized by a solid arrow. x^* becomes $x_{current}$ if the local optimum is feasible.

The second step is to perturb (or mutate as it is called in Evolutionary algorithms) $x_{current}$ such that a new vector with variable values, called x', is created. This perturbation is achieved by randomly choosing a deviation of the variable value through the use of a uniform probability distribution in $[-\sigma, \sigma]$, where σ is the standard deviation. This process is visualized in Figure 4.4.

Using this new initial point, the local optimiser is run again. If this new x^* is feasible *and* better than the current minimum $x_{current}$, then $x_{current}$ becomes x^* and the second and third steps are repeated. This can be seen in Figure 4.5. Also, at this point the $N_{n,i}$ is set to zero again, because the minimum was improved.

However, it could be possible that at a certain point the minimum does not improve, such as shown in Figure 4.6. In this case $x_{current}$ is not updated and remains the same, and the loop starts again with the second step, thus perturbing the same minimum point as before. Also, for each consecutive moment that this occurs, $N_{n,i}$ is increased by one. It could be that the minimum value does not improve for a large number

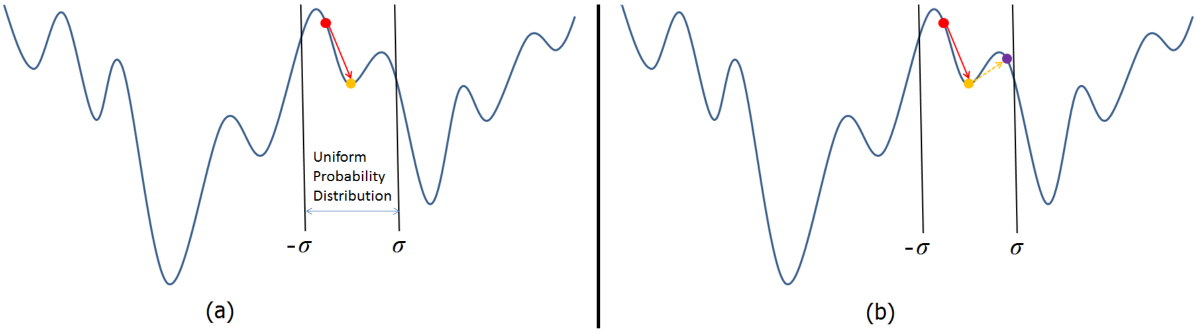


Figure 4.4: The second step of MBH where (a) $x_{current}$ is perturbed within a uniform probability distribution creating (b) a new initial point x' (purple dot). The perturbation is visualized by a dashed arrow.

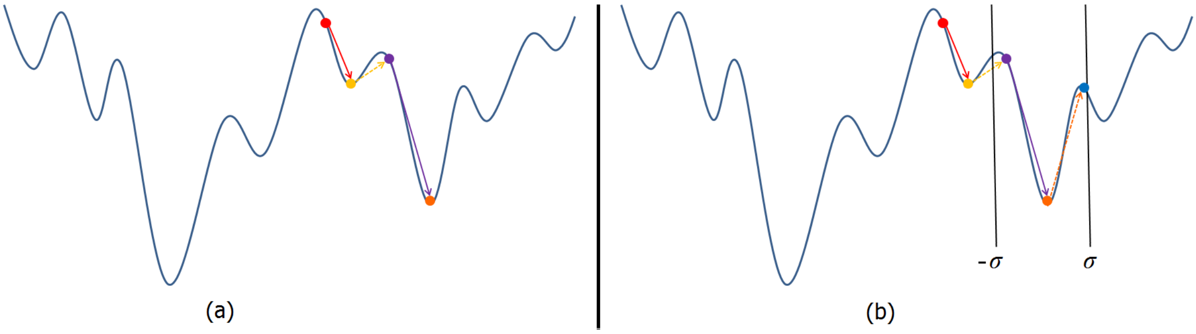


Figure 4.5: The third step of MBH where (a) the new local minimum x^* (orange dot) is compared to the current local minimum $x_{current}$ and is set as $x_{current}$ if it is better. Then (b) the second step repeats from the new $x_{current}$ creating a new perturbed point x' (dark blue dot).

of iterations, which means that it is stuck at the best local minimum for that region. MBH works with the assumption that basins tend to be close together and thus finds a certain minimum within that set of basins (called a "funnel", which could be the right part of the function curve illustrated in Figure 4.6).

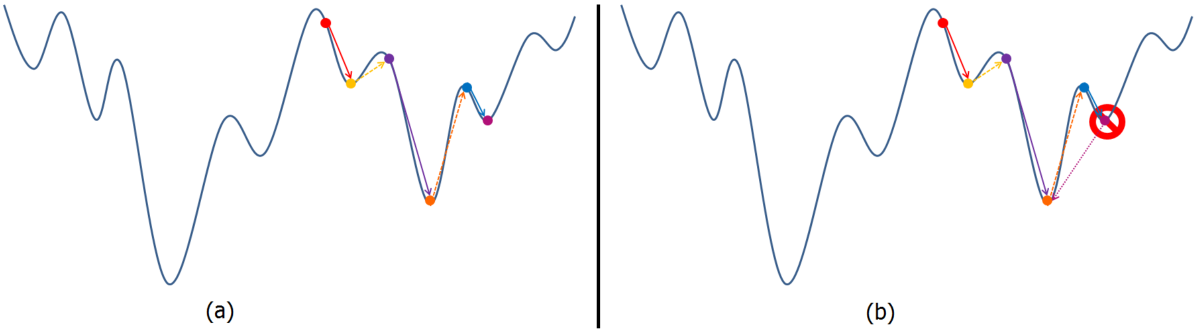


Figure 4.6: (a) A new local optimum is found (magenta dot), but (b) it is not better than the current optimum (orange dot), which is why it is disregarded visualized by the dotted arrow.

However, if the minimum point does not improve, it means that it is stuck in such a funnel. The search space however likely contains a number of these funnels, which all have their own best minimum and one of these funnels thus contains the global minimum, which is the minimum that the algorithm attempts to find. So to escape a certain funnel and try to find the next one a maximum on the number of not-improved iterations is set called $Max_{n,i}$. Once this maximum is reached, MBH is set to start at the first step again while saving the best found local minimum for that particular funnel.

This process is repeated (see Figure 4.7), until either the maximum number of total iterations (steps 1 to 3), or a maximum CPU time is reached. These two criteria are called the cut-off criteria.

At the end of the MBH run, the best found local minimum is presented and should be the global minimum (unless the global minimum could not be reached before one of the cut-off criteria was met). One of the

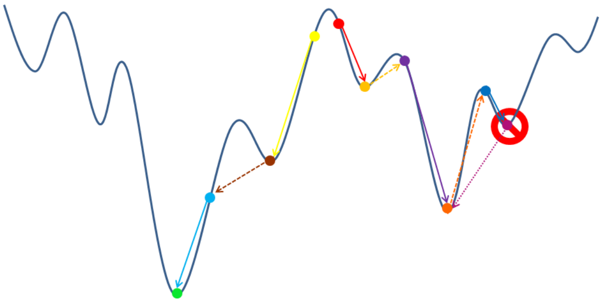


Figure 4.7: The process is repeated for another initial guess (yellow dot). In this case, the current funnel contains the global minimum, however the process is still repeated until one of the cut-off criteria has been met.

advantages of this method is that, because MBH uses a local optimiser to determine the final values of the variables, the obtained global minimum will be very accurate and convergence to *a* minimum is guaranteed (whether it is the global minimum or not). An example pseudo-code for the described process is provided in [75]. It is thus important to choose a proper local optimiser for MBH. The choice of the local optimiser is further discussed in the next section, together with other methods to create variations on the MBH method.

4.6.2. VARIATIONS OF MBH

There are several ways in which the MBH method can be changed. First the manner in which the values for point x are created can be bound to a certain parameter space for each of the variables. This could result in more feasible points and thus less iterations. Secondly the local optimiser can be chosen by the user themselves depending on the problem that needs to be solved, so different MBH methods with different local optimisers can exist. Many reference research that has been focussed on space trajectory problems use the Sparse Nonlinear Optimizer (SNOPT), such as [77–80] [2]. SNOPT was introduced in 2002 by Gill et al. [81] as an SQP (see Sections 4.1 and 4.3) method. SNOPT requires the first function derivatives and is very effective with highly constrained problems such as trajectory optimisation. More information on the detailed workings of SNOPT can be found in [81, 82]. Musegaas [40] also mentioned that SNOPT should be very useful in future research given the fact that it is a powerful local optimiser. Based on all the reference research and provided that similar problems to the thesis topic have all used or recommended SNOPT, this local optimisation tool is chosen to be used in the MBH method. Finally, a different distribution from the original uniform probability distribution [80], used to perturb $x_{current}$, can be used to improve the efficiency and robustness of MBH. In [75] two different distributions called Cauchy [3] and bi-polar Pareto [3] were used to find a better performance of the MBH optimiser. In conclusion the paper recommends a bi-polar Pareto distribution, because it improved the performance most compared to the original distribution.

[2] The last reference is not a space problem application

[3] Also used in [78]

5
Integrators

Propagation refers to the process of modelling/predicting the manner in which a function will progress (in time) usually given an initial condition. In orbital computations, numerical integration is often used to achieve this, because of the irregularities in the dynamic environment and the absence of analytical solutions. The solution then provides an estimate of the trajectory and the position of the s/c[83]. A short summary of available integration methods will be provided in Section 5.1. Determining a suitable integration method requires a trade-off between accuracy of the result and cpu time [84]. In Section 5.2 a comparison will be made between the different integration methods, which will provide an indication of the accuracy and required cpu time. In Section 5.3 a selection is made of the most appropriate integrators based on the TU Delft heritage and external experience with problems similar to the subject provided in Chapter 2. Unfortunately, because the methods are numerical, and thus never result in an perfectly accurate answer, there is always a certain inaccuracy. This inaccuracy is represented by the local (and total) truncation error. These truncation errors are usually the largest errors, however there can also be an additional error caused by the fact that computers round-off values up to a certain number of decimal figures. The more computations are performed in sequence, the bigger this round-off error will become. There are also specific errors that are associated with the integration of a space-related problem as described in [85]. If the simulated system is chaotic or if the step-size is too large, instability errors can occur. There can also be errors in the physical model used to simulate the system caused by mistakes in the assumptions made to create the physical model such as forces and disturbances that are not (properly) taken into account, but this is not an integration error (integrator independent). Many methods exist to handle, or at least provide an approximation of, these errors. These methods are then sometimes combined with existing integration methods to create new integration methods. This chapter will thus focus on the different integration methods and not the different methods of determining the different errors.

5.1. Different integrators

Many different numerical integration methods are available. In this section the methods have been split into single-step (Section 5.1.1) and multi-step methods (Section 5.1.2) based on [84]. The methods can further be categorised by either using a fixed or a variable step-size and by being explicit or implicit. An explicit method only uses the information of the current $\mathbf{x}_i$ (and sometimes past) point(s) to determine the next point value $\mathbf{x}_{i+1}$. An implicit method also uses the next point value to determine this same next point value, which requires iteration. Numerical integration to the next point can be defined by the current point plus the step-size h times the increment function Φ as shown by Equation (5.1). The increment function changes depending on the used method. Here, η represents the numerical approximation.

$$\mathbf{x}(t_0 + h) \approx \mathbf{x}_0 + h\boldsymbol{\Phi} = \eta(t_0 + h) \tag{5.1}$$

There are however also promising analytical integration methods, and one of these will be mentioned in Section 5.1.3.

5.1.1. SINGLE-STEP

In single-step methods, only the information at the current (starting) point is taken into account and the information of previous points is neglected and not saved [84]. Some simple explicit, fixed step-size, single-step methods are Euler, Mid-point and RK4 [83]. Euler uses the properties of the initial point to directly calculate the value at the next point. Mid-point already takes an extra point at half a step-size into account, and RK4 takes the weighted average of four points (the starting point, two mid-points and a final point) into account. Many derivative methods exist based on these functions. A method based on the Mid-point method for instance is the high-order extrapolation (a.k.a. DIFEX2) (explicit) [86]. Examples based on the original RK4 integrator are Runge-Kutta-Nyström (RKN, with variations such as DOPRIN a.k.a. RKN7(6)9) (implicit) [87, 88], Runge-Kutta-Nyström 12^{th} order (RKN12) (implicit) [87] and Runge-Kutta-Fehlberg 4^{th} (5^{th}) order (RKF45) [89, 90]. This last integrator is slightly different since it is still explicit, but uses a variable step-size.

5.1.2. MULTI-STEP

A multi-step method uses the information from the current point and the information of previous points, usually reaching as far back as the previous three points such as the Adams-Bashforth 4^{th} order (AB4) method [84]. This explicit method is similar to RK4 where it uses a weighted average of four points, but in this case uses three previous points. A derivative of this method is the Adams-Bashforth 6^{th} order (AB6) (explicit) method. An implicit, fixed step-size, multi-step method is the Adams-Moulton method which uses a polynomial to interpolate the function values [84]. Combining both these explicit and implicit methods creates what is called a Predictor-Corrector where the initial guess for the next point value follows from the explicit function and the implicit function is then used to correct or improve the estimate. Such methods are Adams-Bashforth-Moulton 4^{th} order (ABM4) and Adams-Bashforth-Moulton 12^{th} order (ABM12) [84, 87]. These all use a fixed step-size, however there are also variable step-size methods based on these methods such as Shampine-Gordon (SG) (sometimes referred to as DE, not to be confused with Differential Evolution), which is an explicit method [91, 92] and Störmer-Cowell 14^{th} order (SC14), which is again both implicit and explicit (based on Predictor-Corrector) [91, 93].

5.1.3. TAYLOR SERIES INTEGRATION

Another example of an implicit multi-step method is the Taylor Series integration (TSI) method, which uses a variable step-size and is based on Taylor Series expansion to compute the next set of variables [94]. However, compared to the methods mentioned in Sections 5.1.1 and 5.1.2 this method is not numerical. It is still important to include this analytical method, because unlike other analytical integration methods, TSI has shown great promise in the field of trajectory integration.

5.2. TECHNIQUE COMPARISON

A representation (based on [84]) of the mentioned methods is provided in Table 5.1 and can be used to compare the different methods. The sources for these advantages and disadvantages are the same as mentioned in Sections 5.1.1 and 5.1.2, and should an extra source be used, it will be mentioned separately. Also, please note that the comparison is sometimes based on astrodynamic problems specifically and are not necessarily true for arbitrary physical problems. The information in Table 5.1 was provided in [84] unless mentioned otherwise.

Table 5.1: Integration method comparison.

Method	Type	Advantage	Disadvantage
Euler	Single-step, fixed step-size, explicit	simple, easy to implement	poor accuracy, better solution requires very small step-sizes, so an increase in cpu time
Mid-point	Single-step, fixed step-size, explicit	simple, easy to implement	not very accurate
RK4	Single-step, fixed step-size, explicit	simple, stable, and has small round-off error accumulation	not very accurate for large problems and step-size has to be determined through trial and error which requires more cpu time
DIFEX2	Single-step, fixed step-size, explicit	useful for numerically stable second order differential equations	not very accurate and not very fast

DOPRIN	Single-step, fixed step-size, implicit	includes local truncation error estimate [88], can be used for a wide range of accuracies [87]	not very fast
RKN12	Single-step, fixed step-size, implicit	high accuracy, efficient (fast)	high order (can be complex) and not the best method if the system is velocity dependent [87]
RKF45	Single-step, variable step-size, explicit	includes local truncation error estimate	poor accuracy and slow
AB4	Multi-step, fixed step-size, explicit	simple, stable, efficient if previous results are stored, faster than RK4 for identical step-size and has small round-off error accumulation	step-size has to be determined through trial and error which required more cpu time, and needs a different technique to determine the previous point values at the beginning
AB6	Multi-step, fixed step-size, explicit	very fast	very poor accuracy and instability at large step-sizes [87]
TSI	Multi-step, variable step-size, explicit	very fast and very accurate [94]	can be complex
Adams-Moulton	Multi-step, fixed step-size, implicit	accurate, stable, and has small round-off error accumulation	step-size has to be determined through trial and error which requires more cpu time, and needs a different technique to determine the previous point values at the beginning
ABM4	Multi-step, fixed step-size, both implicit and explicit	very fast and includes local truncation error estimate	poor accuracy and not efficient if high accuracies are required
ABM12	Multi-step, fixed step-size, both implicit and explicit	fast, high accuracy and includes local error estimate	not efficient if low accuracies are required
SG	Multi-step, variable step-size, explicit	very accurate, high efficiency, includes local truncation error estimate and stable [87]	not very fast
SC14	Multi-step, variable step-size, both implicit and explicit	very accurate, fast and very stable [87]	can be complex

A graphic performance comparison, performed by [87], was made between (several of) these methods for single-step and multi-step techniques (Figure 5.1). It should be mentioned though that the graphs show more methods than mentioned in this and the previous section, because more variations exist, but it was chosen to discuss a selection of different methods only.

5.3. CHOSEN METHODS

For the integration method selection, it is beneficial to know which methods are used in industry and the scientific community (other universities included) and which methods have already been implemented within the Space department at the TU Delft. Some of the references used in Sections 5.1 and 5.2 were already focussed on space missions but discussed a number of different methods. Section 5.3.1 elaborates on the implemented methods for space problems. The past TU Delft experience is discussed in Section 5.3.2 and finally the methods are chosen in Section 5.3.3.

5.3.1. FREQUENTLY USED METHODS IN RELATED SPACE PROBLEMS

Table 5.2 provides an overview of reference research and the used integrators. Should a reference discuss different methods, then the method that the author deems most suited for a space related mission will be cited. However, in some cases several methods are mentioned, since the detailed scope of the subjects was not set yet. In those cases, the proper application of the methods will be provided as well.

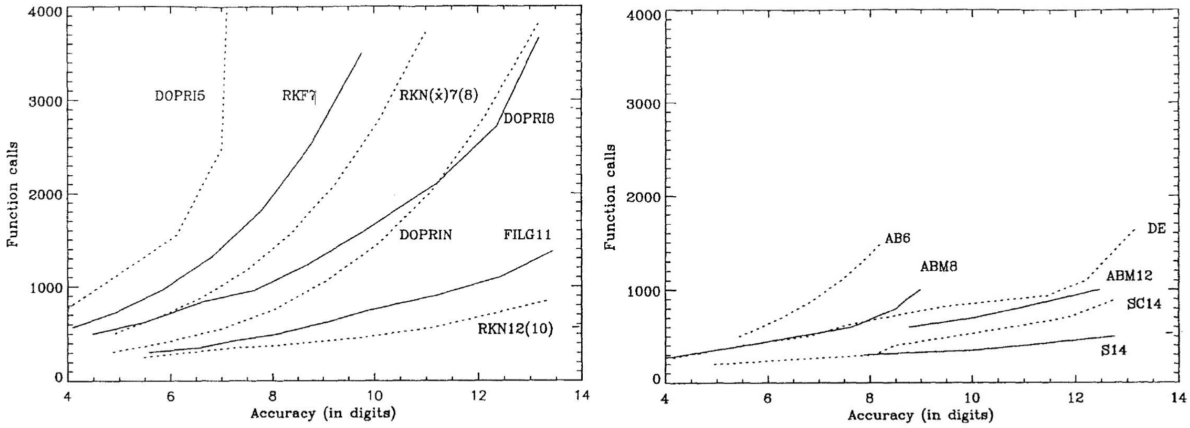


Figure 5.1: Comparison of single-step (left) and multi-step (right) methods for an eccentricity of 0.1 [87].

Table 5.2: Previous integration methods used for trajectory and orbit (transfer) problems.

Author	Year	Subject	Method
Quinn et al. [95]	1991	orbit propagation	Störmer (so without Cowell implicit correction)
Montenbruck [87]	1992	orbital motion (very low and very high elliptic orbits)	RKN12 (if EoM $\neq f(V)$) else SG
Van der Houwen et al. [96]	1998	high-precision orbit computations	SC
González et al. [97]	2000	long-term satellite orbit prediction	high-order RKN
Berry [91]	2004	space surveillance	SC
Scott and Martini [94]	2008	orbital motion, planetocentric and heliocentric trajectories	RKF89 and TSI

A completely different category of integrators is the symplectic integration methods. These are used in orbit mechanics when dealing with a pure Hamiltonian system [83] and in the case of very long-term orbit integrations. However, since they deal with such a specific problem they are not considered useful for this thesis problem.

5.3.2. TU DELFT INTEGRATION HERITAGE

For the selection of the integration method, it is important to understand what has been done by previous master students and researchers at the Space department of the TU Delft. Because, information that already exists can be accessed more easily and it is also important to determine where improvements still have to be made. An overview of the different integrators used for research similar to the proposed thesis research is presented in Table 5.3.

It is interesting to see that even though RK4 is not very accurate for large problems, it is still the preferred method for many master theses.

5.3.3. CHOSEN METHODS

Based on the reference research and the TU Delft heritage it has been decided to include RK4, RKF (any order) and TSI in the trade-off. To provide flexibility in deciding on the final thesis topic, two different integration methods will be chosen. Then in Chapter 11 it will determined whether to focus on two different integration methods or simply choose one. The decision of which methods to use will be based on the heritage, reference research, problem type (here launch and space trajectories), accuracy, speed, easy implementation and novelty. Also, the more resent the research the more relevant it is deemed to be. The trade-off is visualised using Table 5.4. Every criterion is assigned a weight depending on how important the criterion is. Then the different methods are either green (1), yellow (0.5) or red (0) for every criterion. This results in a final score.

Table 5.3: Previous integration methods used within the Space department.

Author	**Year**	**Subject**	**Method**
Pagano [64]	2010	launch trajectories	RKF45
Römgens et al. [98]	2011	satellite trajectories and collision avoidance	TSI
Gondelach [99]	2012	low-thrust trajectories	RK4
Vandamme [65]	2012	launch trajectories	RK4
Van Kesteren [66]	2013	launch trajectories	RK4
Hofsteenge [83]	2013	long-term propagation of space debris	DOPRI8 (similar to DOPRIN)
Gebbett [100]	2014	low-thrust trajectories	Adams-Moulton
Boudestijn [62]	2014	low-thrust trajectories	RK4, RKF78 and Haun's method (extension of Euler)(best)
Gomez [67]	2015	low-thrust trajectories	Haun's method, ABM4 and RKF45 (best)
Miranda [63]	2015	hybrid rockets	RK4

Table 5.4: Optimiser trade-off table.

	Heritage	**Ref. research**	**Trajectories**	**Accuracy**	**Speed**	**Impl.**	**Novelty**	**Score**
Weight	*4*	*4*	*5*	*4*	*3*	*2*	*5*	
RK4	🟩	🟥	🟩	🟥	🟨	🟩	🟥	12.5
RKF	🟩	🟨	🟩	🟨	🟨	🟩	🟥	16.5
TSI	🟨	🟨	🟨	🟩	🟩	🟥	🟩	18.5

From the trade-off table it is clear that TSI shows potential for trajectories. However, as determined from the reference research and the TU Delft heritage, TSI has not yet been used for launch trajectory integrations specifically. In [94] TSI was compared to RKF89 because the used program (SNAP) had this last integration method already incorporated. For an s/c spiralling out of the Earth's gravity field, the cpu time required to integrate the problem with TSI was found to be at least 19 times less than the time required to perform the same integration using RKF89. TSI also proved to be more accurate for most cases.

RKF89 is a higher-order Runge-Kutta-Fehlberg method based on the original RKF45. Both Pagano and Gomez used this original formulation of the RKF45 method. Boudestijn used a 7^{th} order version of this method. Table 5.4 shwos that in this case it is also interesting to compare TSI and RKF. Based on the fact that previous students have used RKF45 for both launch trajectories and (low-thrust) space trajectories ([64, 67]) this method will also be used for the integration of this thesis problem as well as TSI. These two methods can then be compared to each other.

To gain a better understanding of TSI, the implementation of this method will be discussed in Section 5.4. RKF45 was originally based on RK4 and was described by Fehlberg in 1969 [89]. Because it is based on RK4, this method will be described in Section 5.5. Initially, RK4 can also be used to test the program. Later, both RKF45 and TSI can be implemented and compared. The decision on whether to use different integration methods or just one of these will be made in the final topic trade-off in Chapter 11.

5.4. IMPLEMENTATION OF TSI

TSI has been used to solve ordinary differential equations since the early 1960s [94]. However, the first modern implementation of TSI in a space trajectory problem was provided by Montenbruck in 1992 [101, 102]. In 2008 Scott and Martini were able to implement this TSI method into the SNAP trajectory propagator [94] which is the implementation that will be used during this explanation. The method used is described in Section 5.4.1 and possible improvements on the method are mentioned in Section 5.4.2.

5.4.1. WORKINGS OF TSI

In [94] an example situation is used to describe the TSI method: the thrust-less motion around a central body. For consistency the same example and formulation will be used. The state vector is represented by $\overline{\mathbf{X}}$ with the corresponding vector for the initial conditions $\overline{\mathbf{X}}_0$. Each of the variables can at any point be represented by $x_n(t)$ and thus the Taylor series expansion for an order $K \in \mathbb{R}$ (chosen by the user) can be written as shown by Equation (5.2) with $n = 1, \ldots, 7$ in this case (7 variables: three position, three velocity and one mass) and $T_{n,K}$ is the truncation error for the n^{th} variable using K terms.

$$x_n(t) = \sum_{k=0}^{K} \frac{x_n^{(k)}(t_0)}{k!} (t - t_0)^k \quad + \quad T_{n,K} \tag{5.2}$$

This particular TSI method uses recurrence relations to determine the k^{th} order derivatives and only requires the first derivatives shown in Equation (5.3) for the 7 variables and two newly introduced variables to ease the use of the recurrence relations. Here GM represents the standard gravitational parameter (also known as μ).

$$\begin{aligned}
x_1' &= x_4 \\
x_2' &= x_5 \\
x_3' &= x_6 \\
x_4' &= -GM\frac{x_1}{\left(x_1^2 + x_2^2 + x_3^2\right)^{3/2}} = -GM\frac{x_1}{x_9} \quad \text{with} \quad x_9 = x_8^{3/2} \quad \text{and} \quad x_8 = x_1^2 + x_2^2 + x_3^2 \\
x_5' &= -GM\frac{x_2}{\left(x_1^2 + x_2^2 + x_3^2\right)^{3/2}} = -GM\frac{x_1}{x_9} \\
x_6' &= -GM\frac{x_3}{\left(x_1^2 + x_2^2 + x_3^2\right)^{3/2}} = -GM\frac{x_1}{x_9} \\
x_7' &= 0 \\
x_8' &= 2x_1x_4 + 2x_2x_5 + 2x_3x_6 \\
x_9' &= \frac{3}{2}\frac{x_9x_8'}{x_8}
\end{aligned} \tag{5.3}$$

The general recurrence relations for products ($w(t) = f(t)g(t)$) and quotients ($w(t) = \frac{f(t)}{g(t)}$) that are required are provided in Equation (5.4) respectively.

$$\begin{aligned}
&\text{For products} \quad W(k) = \sum_{j=0}^{k} F(j)G(k-j) \\
&\text{For quotients} \quad W(k) = \frac{1}{g(t_0)}\left[F(k) - \sum_{j=1}^{k} G(j)W(k-j)\right] \\
&Both\ with \quad W(k) = \frac{w^{(k)}(t_0)}{k!}, \quad F(j) = \frac{f^{(j)}(t_0)}{j!} \quad \text{and} \quad G(k-j) = \frac{g^{(k-j)}(t_0)}{(k-j)!}
\end{aligned} \tag{5.4}$$

Now let $u_n^{(k-1)} = x_n^k$ for $k = 1, \ldots, K$, then $\frac{u_n^{(k-1)}}{(k-1)!} = \frac{x_n^k}{(k-1)!}$, and also $\frac{x_n^k}{k!} = X_n(k)$. Combining this results in Equation (5.5).

$$U_n(k-1) = kX_n(k) \Rightarrow X_n(k) = \frac{U_n(k-1)}{k} \tag{5.5}$$

Then also introducing $w_4 = \frac{x_1}{x_9}$, $w_5 = \frac{x_2}{x_9}$, $w_6 = \frac{x_3}{x_9}$, $w_{8,1} = x_1x_4$, $w_{8,2} = x_2x_5$, $w_{8,3} = x_3x_6$ and $w_9 = \frac{x_9u_8}{x_8}$, the equations presented in Equation (5.3) can be rewritten to arrive at the recurrence relations using the provided definitions. These relations are presented in Equation (5.6).

$$\begin{aligned} U_1(k) &= X_4(k) = \frac{U_4(k-1)}{k} & U_4(k) &= -GMW_4(k) & U_7(k) &= 0 \\ U_2(k) &= X_5(k) = \frac{U_5(k-1)}{k} & U_5(k) &= -GMW_5(k) & U_8(k) &= 2W_{8,1}(k) + 2W_{8,2}(k) + 2W_{8,3}(k) \\ U_3(k) &= X_6(k) = \frac{U_5(k-1)}{k} & U_6(k) &= -GMW_6(k) & U_9(k) &= \frac{3}{2}W_9(k) \end{aligned} \tag{5.6}$$

Where the expressions for $W_4(k)$, $W_5(k)$, $W_6(k)$, $W_{8,1}(k)$, $W_{8,2}(k)$, $W_{8,3}(k)$ and $W_9(k)$ are shown in Equation (5.7).

$$\begin{aligned} W_4(k) &= \frac{1}{x_9}\left[X_1(k) - \sum_{j=1}^{k} X_9(j)\, W_4(k-j)\right] = \frac{1}{x_9}\left[\frac{U_1(k-1)}{k} - \sum_{j=1}^{k}\frac{U_9(j-1)}{k} W_4(k-j)\right] \\ W_5(k) &= \frac{1}{x_9}\left[X_2(k) - \sum_{j=1}^{k} X_9(j)\, W_5(k-j)\right] = \frac{1}{x_9}\left[\frac{U_2(k-1)}{k} - \sum_{j=1}^{k}\frac{U_9(j-1)}{k} W_5(k-j)\right] \\ W_6(k) &= \frac{1}{x_9}\left[X_3(k) - \sum_{j=1}^{k} X_9(j)\, W_6(k-j)\right] = \frac{1}{x_9}\left[\frac{U_3(k-1)}{k} - \sum_{j=1}^{k}\frac{U_9(j-1)}{k} W_6(k-j)\right] \\ W_{8,1}(k) &= \sum_{j=0}^{k} X_1(j)\, X_4(k-j) = x_1\frac{U_4(k-1)}{k} + x_4\frac{U_1(k-1)}{k} + \sum_{j=1}^{k-1}\frac{U_1(j-1)}{j}\frac{U_4(k-j-1)}{k-j} \\ W_{8,2}(k) &= \sum_{j=0}^{k} X_2(j)\, X_5(k-j) = x_2\frac{U_5(k-1)}{k} + x_5\frac{U_2(k-1)}{k} + \sum_{j=1}^{k-1}\frac{U_2(j-1)}{j}\frac{U_5(k-j-1)}{k-j} \\ W_{8,3}(k) &= \sum_{j=0}^{k} X_3(j)\, X_6(k-j) = x_3\frac{U_6(k-1)}{k} + x_6\frac{U_3(k-1)}{k} + \sum_{j=1}^{k-1}\frac{U_3(j-1)}{j}\frac{U_6(k-j-1)}{k-j} \\ W_9(k) &= \frac{1}{x_8}\left[\sum_{j=0}^{k} X_9(j)\, U_8(k-j) - \sum_{j=1}^{k} X_8(j)\, W_9(k-j)\right] \\ &= \frac{1}{x_8}\left[\sum_{j=0}^{k}\frac{U_9(j-1)}{j} U_8(k-j) - \sum_{j=1}^{k}\frac{U_8(j-1)}{j} W_9(k-j)\right] \end{aligned} \tag{5.7}$$

With the Taylor series coefficients now defined as provided in Equation (5.5), Equation (5.2) can be used to determine the parameter values at time t (or t_1) for the known previous parameter values at time t_0. The same can then be done to determine the values at t_2 using the parameter values at t_1, etc. In its simplest form, a constant step-size h (defined as $t - t_0$) is taken which determines the next t.

5.4.2. Variations of TSI

Many of the different aspects of TSI can be varied upon: the manner in which the truncation error is estimated, the order K until which the series is evaluated, the use of a variational step-size and the method of determining the next step size.

As mentioned before K should be chosen by the user, however in [94] it is mentioned that the maximum number of series terms cannot exceed 30 using this particular method.

When considering the variable step-size, [94] uses two different methods to determine the next step size. The first method is using Equation (5.8), where η is the chosen step multiplication factor and has to be smaller than 1, h is the current step-size, τ is the chosen local error tolerance (or preferred accuracy), e_{max} is the estimate of maximum truncation error and M is the order of the maximum truncation error estimate which follows from the maximum number of series terms.

$$h_{next} = \eta h \left(\frac{\tau}{e_{max}}\right)^{\frac{1}{M}} \tag{5.8}$$

In this case e_{max} is determined through Equation (5.9). The maximum estimated truncation error over all variables n is chosen.

$$e_{max} = Max_n \left[\left| X_n (K-1) \right| h^{K-1} + \left| X_n (K) \right| h^K \right] \tag{5.9}$$

Another method mentioned in the same paper is to directly use the local error tolerance τ to determine the next step size. This method assures that the step-size is small enough that all variables satisfy the error condition directly. The step-size is determined using a so-called fixed-point iteration performed using Equation (5.10). The initial step-size h_1 can be chosen to be the current step-size h as an easy estimate. Then the iteration is performed over l until a certain required convergence is reached.

$$h_{l+1} = exp\left(\frac{1}{K-1} ln \left[\frac{\tau}{\left| X_n (K-1) \right| + h_l \left| X_n (K) \right|} \right] \right) \tag{5.10}$$

Again this is done for all variables and the smallest required step-size is chosen. This is then used to determine the next step-size through $h_{next} = \eta h_{chosen}$. Scott and Martini preferred this second method over the first mentioned method of determining the next step-size because the first method requires previous step information (e_{max}), nonetheless the performance of both methods was very similar [94].

Finally, there are several techniques to estimate the (local) truncation error that results from every series evaluation. Simply put e_{max} is the estimation of the maximum value that $T_{n,K}$ can take. One example of how to determine this estimate was already shown by Equation (5.9). Another approach would be to calculate the estimate from the K+1 term in the series, however, this is discouraged in [94] where it is said that it would require an extra computation and is not a reliable error estimate.

5.5. IMPLEMENTATION OF RK4

RK4 is known as the work-horse of engineering problems when it comes to the integration of functions. It is a slightly more complicated form of the Euler and Mid-point methods. In this case four points/state vectors are used to determine the next parameter value(s). The exact workings of RK4 will be explained in Section 5.5.1 and other variations are discussed in Section 5.5.2.

5.5.1. WORKINGS OF RK4

RK4 is based on the formulation provided by Equation (5.1) where in this case the increment function for RK4 is presented in Equation (5.11). The principle behind RK4 is well described by [84]. This method is a single-step, fixed step-size, explicit method and thus does not use any previous step information to determine the next step. Only the current parameter values are used to predict the next step as is visualised in Figure 5.2. In this figure it can also be seen that this method uses four derivative evaluations of the function to determine the next point indicated as $x(t_0 + h)$.

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

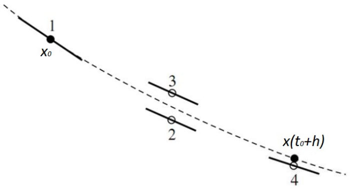


Figure 5.2: Principle of RK4 for a single parameter [84].

First the time-derivative of the current point/state vector is taken and called k_1 as shown by Equation (5.12) (including the whole sequence). This is used to evaluate the local state vector halfway through the interval at $h/2$ from the current time and for a parameter value of $x_0 + h\frac{k_1}{2}$ and is called k_2. This k_2 is then used instead of k_1 to perform the same evaluation as before, resulting in k_3. Finally, k_3 is used to evaluate the local state vector at the end of the interval resulting in k_4. These four derivative values are then added in a weighted fashion to produce the increment function as shown by Equation (5.11), which is in turn used to determine the next point/state vector (see Equation (5.1)).

$$\begin{aligned}
k_1 &= f'(t_0, x_0) \\
k_2 &= f'(t_0 + h/2, x_0 + hk_1/2) \\
k_3 &= f'(t_0 + h/2, x_0 + hk_2/2) \\
k_4 &= f'(t_0 + h, x_0 + hk_3)
\end{aligned} \tag{5.12}$$

In Equation (5.12) f' depicts the evaluation of the derivative. It is also mentioned in [84] that performing this integration will result in a local truncation error that is $O(h^5)$. Therefore, Equation (5.1) can be adapted to include the truncation error as depicted by Equation (5.13).

$$\mathbf{x}(t_0 + h) = \mathbf{x}_0 + h\mathbf{\Phi} + \mathbf{T} \tag{5.13}$$

5.5.2. METHODS BASED ON RK4

The RK4 method is often seen as the classic method and has thus been expanded and varied upon substantially (also see Section 5.1.1). There are three direct means in which to modify the method: the order (so number of intermediate points used to evaluate the final point), including the next estimation to improve the next parameter values (making it implicit) and the manner in which the step-size is chosen, which can even be done such that the method changes to a variable step-size method. Examples of implicit methods based on RK4 are DOPRIN and RKN12, and an example of a method which is based on RK4 but uses a variable step-size is RKF45. The RKF45 uses an estimate of the truncation error to determine the next step-size [89], similar to a variation of TSI. This is accomplished by computing a fourth and fifth order Runge-Kutta and then using the difference between those two as an estimate for the truncation error. Then the step-size can be adjusted if required.

6

Atmospheric (Martian) launch

As mentioned in Section 3.1 no Martian launches have occurred to date, however, there have been numerous Earth launches. Also, similar research has already been performed using different simulation programs as was discussed in Section 3.1. Section 6.1 will elaborate on this research. For the launch in the Martian atmosphere, a model similar to the one used for Earth launches can be implemented provided it is adapted for the Martian atmosphere. The model and the assumptions including the resulting EoM are provided in Section 6.2. The launch trajectory itself can consist of different phases with their own characteristics. The preferred launch trajectory for the desired final conditions is described in Section 6.3 and finally Section 6.4 discusses the initial conditions of the launch site on the Martian surface based on the landing sites currently under consideration for the Mars 2020 rover.

6.1. Previous research

In Section 3.1 the different successful Lunar sample return missions were provided. The flight data of these missions (readily available for all Apollo missions in the Apollo mission reports) can be used for initial validation of the dynamic model and the EoM. The reference Lunar research can be used for initial verification. Other verification will have to be performed using the reference Mars ascent research mentioned in Section 3.1.1. The most important characteristics that could be used for the validation are mentioned in Table 6.1 for each of the different papers. Unfortunately, Dumont [8] has not yet performed any Marian ascent simulations which is why it cannot be used for Mars validation at this time. Should information not be available at the moment, the cell will read Not Available (N/A). For Fanning and Pierson the thrust values originate from a second source[1] , [2].

Table 6.1: Reference characteristics from Mars Ascent research.

Characteristic	**Fanning and Pierson [2]**	**Desai et al. [3]**	**Whitehead [4, 5]**	**Di Sotto et al. [6]**	**Trinidad et al. [7]**
GLOM [kg]	1400	426	100	919.2	227
Payload mass [kg]	367.8	30	Max. 20	4	5
Target orbit altitude [km]	473 (circular)	300 (circular)	500 (circular)	300 by 2000	460 by 580
Propulsion type	Liquid	Liquid	Liquid	Liquid	Liquid
No. of stages	2	2	1 or 2	2	2
Thrust per engine [N] first/second stage	16700 [1]/ 4000 [2]	3559 (max. vacuum)(2x)/222 (max. vacuum)(4x)	1000	1503 (vacuum)(4x)/1687 (vacuum)	5300 (plus 4x 445 TVC)/2700 (plus 4x 445 TVC plus 8x 4.4 ACS)

[1] First stage engine thrust: http://www.astronautix.com/engines/xlr132.htm [Accessed 10 November 2015]

[2] Second stage engine thrust: http://www.astronautix.com/engines/r40b.htm [Accessed 10 November 2015]

I_{sp} per engine [s]	338/309	323 (vacuum)/308 (vacuum)	310	306/306	328.6 (330.5 vacuum)/328.6 (330.5 vacuum)
Point-mass?	Yes	No	No	N/A assumed yes	N/A assumed no
Initial vertical rise? (Time [s])	Yes (3±2)	N/A	Yes (N/A)	Yes (2.3)	Yes (in 90° case 2.15 incl. 1.5 coast)
Launch latitude [°]	0	between 30 N and 15 S	N/A assumed 0	N/A	N/A assumed 0
Launch azimuth [°]	90	N/A	N/A assumed 90	N/A	90 (?)
Target inclination [°]	N/A assumed 0	30	N/A assumed 0	45	N/A assumed 0
Drag coefficient	0.88	0.7 (at Mach 0.8), 1.38 (> Mach 3.0)	0.2 (max. 0.6 and 0.3 > Mach 4)	N/A	N/A
Cross-sectional area [m^2]	N/A	2.27	0.2	N/A	0.091

6.2. Dynamic model and Equations of Motion

There are many different approaches to the ascent problem of the MAV. Therefore it is important to choose the proper dynamic model to simulate the ascent and formulate the corresponding EoM. The model and initial assumptions are discussed in Section 6.2.1, and the EoM are described in Section 6.2.2.

6.2.1. Dynamic model and initial assumptions

When performing trajectory calculations it is important to understand in which RF the parameters are defined and computed and which set of state variables (or coordinate system) is used. The different reference frames and coordinate systems will be discussed in more detail in Chapter 8. During the ascent phase of a rocket there are two coordinate systems that can be used, either the Cartesian coordinate system [103] or the spherical coordinate system [2, 66, 104]. The Cartesian coordinate system is often used during simulations in an inertial system because there is a fast and straightforward propagation for the position: x, y, z and velocity: V$_x$ ($\dot{x}$), V$_y$ ($\dot{y}$), V$_z$ ($\dot{z}$). Also, the use of Cartesian coordinates provides for a stable simulation because there are no singularities. However, this coordinate system is less suited for the interpretation of the trajectory of the vehicle, because it does not provide a proper insight into the flight characteristics during ascent. This is where spherical coordinates are more useful, presented in Figure 6.1 (left). Here the position is represented by the distance r, the longitude τ and the latitude δ. The velocity is defined by the ground speed V, the flight-path angle γ and the azimuth (or heading angle) χ. In this definition, if $\gamma = 0°$ it means that the vehicle is pointing straight up, which can also be easily visualized on, for instance, a piece of paper. In the Cartesian coordinate system, the flight direction is only defined by a change in position/velocity and is thus more difficult to interpret. In a computer environment, Cartesian coordinates can however also be used to visualize (and compare) the trajectories, but using spherical coordinates provides more information on the actual ascent behaviour. The reason that spherical coordinates should not be used for simulation purposes (in this thesis problem) is because during the integration and optimisation singularities can occur whenever γ and/or $\delta = \pm 90°$ and/or V=0 m/s. Also, the associated equations are slower and less straightforward to compute, compared to the ones associated with Cartesian coordinates.

It was decided to use the Cartesian coordinate system for the simulation. The results of each iteration (position and velocity) can easily be transformed to spherical coordinates (see Chapter 8) to gain more insight into the trajectory. When formulating the EoM the definition of the different forces acting on the MAV are required (the EoM will then be formulated using the Cartesian coordinates). A two-dimensional representation of the forces acting on the MAV is provided in Figure 6.1 (Right). In this case it will be assumed that there is no wind acting on the MAV. This can be done, because wind is random and can not be predicted easily.

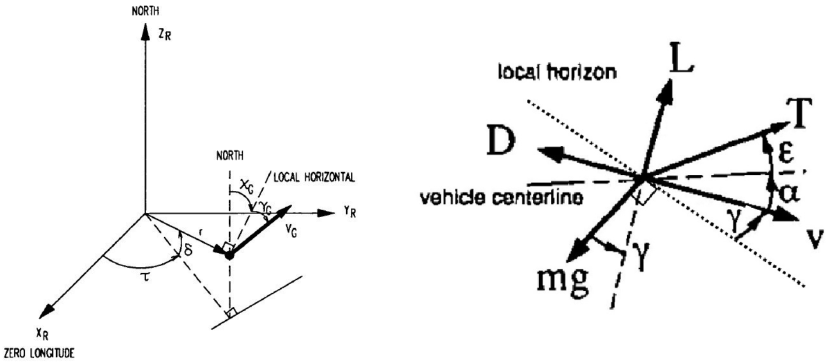


Figure 6.1: (Left) Graphical definition of the 3D spherical coordinate system. The angles are all defined positive in the direction of the arrow[105] (Right) 2D graphical definition of the force model [2].

In the model presented in Figure 6.1 ϵ (from now on ϵ_T) is the thrust elevation-gimbal angle and the lift L, drag D and thrust T are defined by Equation (6.1) [106].

To simulate out-of-plane motion and thus making it a three-dimensional model, a second thrust-gimbal angle is introduced as seen in Figure 6.2. This angle is defined in the x-y plane of the body frame, or B-frame, where it should be noted that the positive x-axis is defined through the vehicle centreline in the direction of flight, the z-axis is defined pointing down in Figure 6.1 (left) perpendicular to the vehicle centreline and the y-axis is then pointing out of the x-z plane by the convention of the right-hand-rule RHR. This second gimbal angle is called the thrust azimuth gimbal angle ψ_T as by [105].

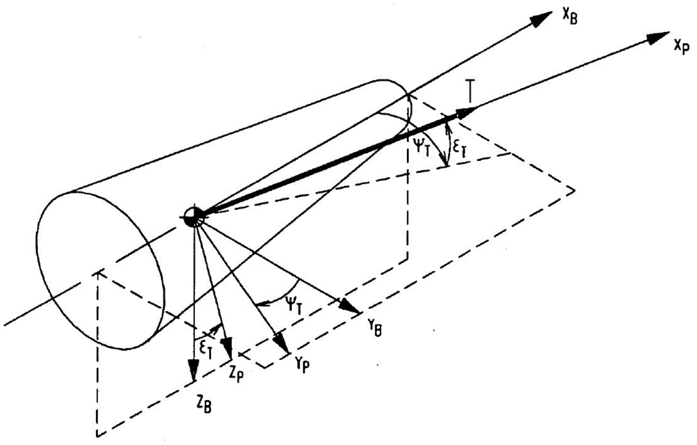


Figure 6.2: Definition of the thrust angles [105].

In the equations for L, D and T, ρ is the air density, S is the reference surface area, C_L the lift coefficient, C_D the drag coefficient, $\dot{m}$ the mass flow rate $\left(= -\dfrac{dm}{dt}\right)$, g_0 the gravitational acceleration at Earth sea-level (not Mars), c_{eff} the effective expulsion velocity, w_e the expulsion velocity, A_e the exit surface area of the nozzle, p_e the exit pressure and $p_{0,s}$ the pressure of the surrounding air.

$$\begin{aligned} L &= \frac{1}{2}\rho V^2 S C_L \\ D &= \frac{1}{2}\rho V^2 S C_D \\ T &= \dot{m} g_0 I_{sp} = \dot{m} c_{eff} = \dot{m} w_e + A_e \left(p_e - p_{0,s}\right) \end{aligned} \tag{6.1}$$

All these equations depend on the altitude h: $\rho = f(h)$ and $p_{0,s} = f(h)$. Also, both coefficients are a function of the Mach number M and the angle of attack α. In this thesis problem, it is assumed that $\alpha = 0°$ (also see Section 6.2.2). This means that the drag coefficient is only a function of the Mach number. If a simple "pencil" shape is assumed, which is a reasonable assumption when examining the baseline model in Figure 3.2, then [4] provides an estimation of the drag coefficient for different Mach numbers as can be seen in Figure 6.3.

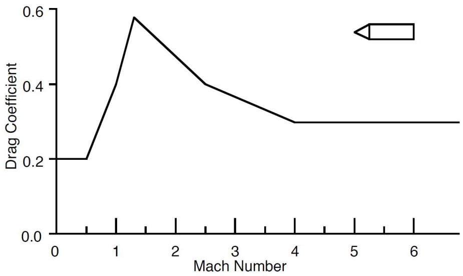


Figure 6.3: Drag coefficient as a function of Mach number [4].

This drag coefficient plot can be used to determine the drag at different velocities (since $M = f(V, a)$ where a is the speed of sound in the Martian atmosphere at that particular point which can be obtained through the atmospheric model that will be chosen). Thus given a certain atmospheric model, the air density and pressure at different altitudes and the Mach number can be computed. The atmospheric model will be selected and discussed in Chapter 9.

The gravity force mg depends on the local gravitational acceleration g. This g can be determined assuming a homogeneous gravity field and a spherical Mars in the rotating Mars RF using Equation (6.2) where g_{M0} is the standard gravitational acceleration on the surface of Mars (3.71 m/s^2)[3] , R_M is the (local) radius of Mars (3389.5 km), G the gravitational constant (6.67408·$10^{-11}\text{m}^3/(\text{kg}\cdot\text{s}^2)$), M_M the mass of Mars (6.4174·10^{23} kg), μ_M the standard gravitational parameter of Mars (4.283·10^{13} m^3/s^2) and r the position radius of the vehicle. The J_2 for Mars is equal to 1960.45· 10^{-6} [107], which would result in an average acceleration in the radial direction (same as gravity) in the order of 10^{-2} m/s^2 (assuming an ascent time of 5 minutes and based on the equations provided in [108]). This in turn translates to a ΔV in the order of 10^0 m/s and a Δr in the order of 10^3 m. This difference is so small that the perturbations due to irregularities in Mars' gravity field (J2, J3, etcetera) can be neglected.

$$g = g_{M0}\left(\frac{R_M}{r}\right)^2 = -\frac{GM_M}{r^2} = -\frac{\mu_M}{r^2} \tag{6.2}$$

This equation can then be used to determine the gravitational acceleration at any distance from Mars. It is assumed that the mass of the MAV, the orbiter(s) and the two Martian moons can be neglected with respect to the mass of Mars (the mass of Phobos is an order of 7 smaller than the mass of Mars) [107] so third-body perturbations are not considered. Also, during the ascent the perturbations due to irregularities in Mars' gravity field can be neglected, because the time spent in ascent is relatively short.

6.2.2. Corresponding EoM and initial assumptions

The general equations of motion can be derived using the described model, and are based on [105]. Because the motion of the Mars 2022 orbiter will be described in an inertial Mars frame and because the final orbit will be described using Kepler elements it is useful to write the EoM for the MAV ascent in the same frame. The current conditions can be described as shown by Equation (6.3), where m_{MAV} is the mass of the MAV and the subscript I refers to the inertial frame.

$$\mathbf{r} = \begin{pmatrix} x_I \\ y_I \\ z_I \end{pmatrix} \qquad \mathbf{V} = \begin{pmatrix} V_{x_I} \\ V_{y_I} \\ V_{z_I} \end{pmatrix} \qquad m_{MAV} \tag{6.3}$$

The EoM can now be described by Equation (6.4). Here, a represents the acceleration.

[3] All constants from NASA Mars fact sheet [107] unless otherwise mentioned

$$\begin{aligned} \dot{x}_I &= V_{x_I} & \ddot{x}_I &= \dot{V}_{x_I} = a_{x_I} & & \\ \dot{y}_I &= V_{y_I} & \ddot{y}_I &= \dot{V}_{y_I} = a_{y_I} & \dot{m}_{MAV} &= -\frac{T}{g_0 I_{sp}} \\ \dot{z}_I &= V_{z_I} & \ddot{z}_I &= \dot{V}_{z_I} = a_{z_I} & & \end{aligned} \tag{6.4}$$

The accelerations follow from Newton's $F = m \cdot a$ and can thus be described as a function of the mass and the different forces acting on the MAV. These are all described in different reference frames unfortunately, which means that the forces acting on the MAV will have to be translated into the inertial Mars frame. This is done through reference frame transformations (which will be discussed in detail in Chapter 8). A transformation is always performed by rotating around the different axes in a particular order which requires the different rotational angles. Each rotation around a certain axis has a corresponding transformation matrix, for instance $\mathbb{T}_{\mathbf{x}}(\phi)$ where $\mathbb{T}$ is the symbol for a transformation matrix, the subscript x shows that it is a rotation around the x-axis and the ϕ is in this case the angle of rotation. This particular rotation is a positive rotation around the x-axis (following the RHR). Should the rotation be negative then the transformation matrix would have been presented as $\mathbb{T}_{\mathbf{x}}(-\phi)$. First it is important to define the reference frames in which the different forces are acting. In this case the thrust is acting in a propulsion frame or P-frame (with the corresponding gimbal angles as rotation angles), the drag is acting in the aerodynamic frame or A-frame and the gravity is acting in the Mars rotational frame or R-frame. Before describing the relation between the different frames, a number of assumptions are made.

- No wind; it is assumed that there is no wind and thus no corresponding effects or changes in the MAV attitude to simplify the simulation. This also means that during the verification of the program, wind will not be taken into account.
- No angle of attack; the angle of attack is assumed to be zero, the MAV is assumed to always move in the direction of flight.
- No lift; it can be assumed that there is no lift, based on the assumption that there is no angle of attack and that the MAV has a symmetric shape.
- No bank angle; because the system is a point mass system and since there is no lift, the bank angle can be assumed equal to zero [66].
- No side-slip; because there is no wind and no banking, there will be no side-slip. Therefore the side-slip angle can be assumed to be zero [105].

From these assumptions it is clear that the A-frame is in this case identical to the body frame or B-frame. Therefore the body is always pointing into the same direction as the velocity vector. This provides another set of angles to transform to the inertial frame as can be seen in Figure 6.1. These angles are the flight-path angle and the heading angle. The subscript G shows that this frame is set-up for the flight-path angle, azimuth and velocity based on the ground-speed as opposed to airspeed, but since there is no wind this is irrelevant in this case. Rotating over these angles would result in transforming to the vertical frame or V-frame. In the vertical frame the x-axis points towards the northern hemisphere, the z-axis points towards the centre of mass of the orbiting body and the y-axis points towards the east [105]. Figure 6.1 then also shows the rotation angles required to transform from the V-frame to the R-frame. These angles are the latitude δ and the longitude τ in a rotating frame. The final transformation involves the rotation from the R-frame to the I-frame. This rotation depends on the location of the prime meridian (the meridian of zero longitude) at the time that the inertial reference frame was set Ω_P, the rotational velocity of Mars $\dot{\Omega}_M$ and the time between when the inertial frame was set and the current time t_O. The rotational angle between the R-frame and the I-frame can then be defined to be $\dot{\Omega}_M t_O - \Omega_P$ (where the motion and angles are defined towards the east). Now including all the forces, accelerations and rotations and using the notation for transformation matrices as defined earlier, the expressions for the accelerations in the I-frame can be described by Equation (6.5). In this notation, each vertical line depicts when a new reference frame is reached and the letter shows which one. Also, because the transformations require matrix multiplications, the transformation should be executed from right to left.

$$\begin{pmatrix} a_{x_I} \\ a_{y_I} \\ a_{z_I} \end{pmatrix} = \Big|_{\mathbf{I}} \mathbb{T}_{\mathbf{z}}\left(-\dot{\Omega}_M t_O + \Omega_P\right)\Big|_{\mathbf{R}} \left[\begin{pmatrix} -\mu_M \dfrac{x_R}{r^3} \\ -\mu_M \dfrac{y_R}{r^3} \\ -\mu_M \dfrac{z_R}{r^3} \end{pmatrix} + \Big|_{\mathbf{R}} \mathbb{T}_{\mathbf{z}}(-\tau)\, \mathbb{T}_{\mathbf{y}}\left(\frac{\pi}{2} + \delta\right)\Big|_{\mathbf{V}} \mathbb{T}_{\mathbf{z}}(-\chi)\, \mathbb{T}_{\mathbf{y}}(-\gamma)\Big|_{\mathbf{B}} \left\{ \begin{pmatrix} -\dfrac{D}{m_{MAV}} \\ 0 \\ 0 \end{pmatrix} + \ldots \right.\right.$$

$$\left.\left. \cdots + \Big|_{\mathbf{B}} \mathbb{T}_{\mathbf{z}}\left(-\psi_T\right) \mathbb{T}_{\mathbf{y}}(-\epsilon_T)\Big|_{\mathbf{P}} \begin{pmatrix} \dfrac{T}{m_{MAV}} \\ 0 \\ 0 \end{pmatrix} \right\} \right] \tag{6.5}$$

The corresponding transformation matrices will be written out in Chapter 8. Also, the position and velocity will change each time step, which means that the angles τ, δ, χ and γ will also change. These angles can all be described in the R-frame. Thus to determine the updated values, the position and velocity will have to be transformed from the I-frame to the R-frame. Then using the transformation between Cartesian and spherical coordinates, the angles can be obtained. This transformation is also described in Chapter 8. Following from the haversine formula based on great circles, the change in the ground distance travelled S_G (in the R-frame) can be computed as a function of the longitude and latitude as shown by Equation (6.6).

$$\dot{S}_G = 2R_M \arcsin\left(\sqrt{\sin^2\left(\frac{\dot{\delta}}{2}\right) + \frac{1}{2}\left(\cos\left(\dot{\delta}\right) + \cos\left(2\delta + \dot{\delta}\right)\right)\sin^2\left(\frac{\dot{\tau}}{2}\right)}\right) \tag{6.6}$$

6.3. REQUIRED LAUNCH TRAJECTORY

When examining the different possible launch trajectories, one mainly looks at different ways to control the ascent vehicle in such a manner that reduces losses. Four different possible launch trajectories (usually followed by each other) are a vertical rise phase, a constant pitch-rate phase, a constant pitch phase and a gravity-turn phase. Also, sometimes a control program for the pitch/pitch-rate is followed instead of a constant pitch/pitch-rate. The pitch angle is defined as $\alpha + \gamma$, however since there is no angle of attack in this case, the pitch angle is equal to the flight-path angle γ (in the R-frame and V-frame). The ascent is then followed by a coasting phase and a final burn phase to place the s/c into a desired orbit [2, 6, 103, 106]. In this case however, since it is assumed that the engines can be gimballed (ψ_T and ϵ_T) the phase between the vertical rise and the gravity-turn can be optimised and does not need a constant pitch-rate and constant pitch phase. This transition phase shall be known as the first thruster control phase. When all the propellants of the first stage are used, the stage-separation will take place (which often happens during the gravity-turn (see Section 6.3.3)). Then the second stage ignites and will continue the gravity-turn. At some point in the second stage burn, it could be more beneficial to change the flight program to a second thruster control phase. This is to avoid an early zero flight-path angle and to position the MAV such that a certain apocentre can be reached. After the second stage is burned out, the coasting phase will start. At the end of this coasting phase (at apocentre of the ascent) a final burn is provided to propel the MAV into the desired final orbit. It should, however, be noted that a vertical rise might not be optimal, which is why the set of equations mentioned in Equation (6.5) allow for an initial heading (χ) and launch angle (γ) to be defined at the beginning of the simulation. The different ascent phases will be discussed in this section.

6.3.1. VERTICAL RISE

The vertical rise phase can be used to penetrate the thickest layers of the atmosphere before performing the gravity-turn [2, 106]. This will reduce the effect of drag significantly but will increase the gravity losses. During the vertical ascent the flight-path angle is equal to 90° (or $\frac{\pi}{2}$). The heading angle during this phase is undefined, which means that the orientation of the MAV is not defined at this stage. Also, during the vertical ascent both gimbal angles are set equal to zero. Therefore the flight-path angle and heading angle do not change during the vertical ascent and since the value of the heading angle is not important, it can be set equal to zero. As soon as the first thruster control phase starts, the heading angle will be defined automatically as a result of the gimbal angles, and the flight-path angle will change as well. With these trajectory characteristics the expressions in Equation (6.5) can be rewritten for the vertical phase (see Equation (6.7)). It is worth mentioning that the first two rotations are now both around the y-axis, which means that they can be performed at once (resulting in a positive rotation δ) however, for clarity it is written such that the V-frame is included.

$$\begin{pmatrix} a_{x_I} \\ a_{y_I} \\ a_{z_I} \end{pmatrix} = \left|\mathbb{T}_{\mathbf{z}}\left(-\dot{\Omega}_M t_O + \Omega_P\right)\right|_{\mathbf{I}}\Bigg|_{\mathbf{R}} \left[\begin{pmatrix} -\mu_M \dfrac{x_R}{r^3} \\ -\mu_M \dfrac{y_R}{r^3} \\ -\mu_M \dfrac{z_R}{r^3} \end{pmatrix} + \Bigg|_{\mathbf{R}} \mathbb{T}_{\mathbf{z}}(-\tau)\,\mathbb{T}_{\mathbf{y}}\left(\frac{\pi}{2}+\delta\right)\Bigg|_{\mathbf{V}} \mathbb{T}_{\mathbf{y}}\left(-\frac{\pi}{2}\right)\Bigg|_{\mathbf{B}} \begin{pmatrix} \dfrac{T-D}{m_{MAV}} \\ 0 \\ 0 \end{pmatrix}\right] \tag{6.7}$$

For the optimisation of this phase the only control variable can be the vertical ascent time t_v (if not preset).

6.3.2. First thruster control phase

To transition from the vertical phase to the gravity-turn, a change in the heading and flight-path angle is required. These changes can be set in motion by gimballing the thruster, or in other words by providing a thrust elevation (ϵ_{T1}, where the subscript 1 refers to the first thruster control phase) and/or thrust azimuth angle (ψ_{T1}). These two angles are both optimisation parameters. In this case, it is necessary to use the complete set of equations for the accelerations (Equation (6.5)). A third optimisation parameter for the first thruster control phase is the time spent in this phase t_{T1}. During this phase a reduced thrust could be required. This would then introduce another optimisation parameter: the thrust level during the first thruster control phase T_{T1}. At the end of this phase, the thrust level is again set to full thrust (if it was changed) and both ϵ_{T1} and ψ_{T1} are again set to zero which will be the start of the gravity-turn phase.
The current baseline design does not assume an engine that can be gimballed (see Section 3.4), but uses control thrusters instead. However, for this thesis it can be assumed that in the future either gimballing will be introduced on the MAV or the trajectory adjustments will indeed have to be made by control thrusters instead.

6.3.3. Gravity-turn and second thruster control phase

After the vertical rise and the first thruster control phase, a so-called gravity-turn is initiated. The gravity-turn results in a rotation of the MAV into a (near) horizontal flight orientation which reduces the gravity losses and is usually initiated after passing through the thickest layers of the atmosphere to reduce drag. Because at the beginning of this phase the flight-path angle is $0 \leq \gamma \leq \frac{\pi}{2}$ (in the R-frame), the gravity force will cause a change in this flight-path angle. This will cause the MAV to turn towards a lower flight-path angle automatically without the use of the gimbal feature. Therefore, during this manoeuvre, the gimbal angles are both zero, which means that the thrust is again acting directly in the B-frame simplifying the acceleration equation. However, in this case there is a flight-path angle and there is also a heading angle. The resulting acceleration equation can be represented by Equation (6.8).

$$\begin{pmatrix} a_{x_I} \\ a_{y_I} \\ a_{z_I} \end{pmatrix} = \left|\mathbb{T}_{\mathbf{z}}\left(-\dot{\Omega}_M t_O + \Omega_P\right)\right|_{\mathbf{I}}\Bigg|_{\mathbf{R}} \left[\begin{pmatrix} -\mu_M \dfrac{x_R}{r^3} \\ -\mu_M \dfrac{y_R}{r^3} \\ -\mu_M \dfrac{z_R}{r^3} \end{pmatrix} + \Bigg|_{\mathbf{R}} \mathbb{T}_{\mathbf{z}}(-\tau)\,\mathbb{T}_{\mathbf{y}}\left(\frac{\pi}{2}+\delta\right)\Bigg|_{\mathbf{V}} \mathbb{T}_{\mathbf{z}}(-\chi)\,\mathbb{T}_{\mathbf{y}}(-\gamma)\Bigg|_{\mathbf{B}} \begin{pmatrix} \dfrac{T-D}{m_{MAV}} \\ 0 \\ 0 \end{pmatrix}\right] \tag{6.8}$$

At the start of the gravity-turn, there is also a certain heading angle. This heading angle (in the I-frame) will determine the final inclination of the orbit together with the MAV latitude at the start of the gravity-turn. The relation for the inclination is shown by Equation (6.9). During the gravity-turn (and the rest of the ascent phase) both the heading angle and latitude will change, however the eventual inclination which the MAV will reach will stay the same (assuming $\psi_{T2} = 0$ during the second thruster control phase). If at the beginning of the gravity-turn, the desired inclination has not been reached, it means that the first two phases were not ideal and will have to be changed. Also, in this case the inclination is an optimisation parameter since different orbital inclinations will have to be investigated combined with the Mars 2022 orbiter trajectory.

$$\chi_0 = \arcsin\left(\frac{\cos(i)}{\cos(\delta_0)}\right) \tag{6.9}$$

At some point the first stage will burn out (which often happens during the gravity-turn), be ejected (simulated by an instant reduction in mass; the empty mass of stage 1) and the second stage will be ignited and burn until its burn-out. Should this separation happen during the gravity-turn, the second stage could either continue following this trajectory, follow a second thruster control program or both. Because there is no need to change the inclination (which was already set by the first thruster control phase) this second thruster control phase will only use the elevation-gimbal angle ϵ_{T2} to change the MAV attitude, resulting in the acceleration equation given by Equation (6.10).

$$\begin{pmatrix} a_{x_I} \\ a_{y_I} \\ a_{z_I} \end{pmatrix} = \Big|_{\mathbf{I}} \mathbb{T}_{\mathbf{z}}\left(-\dot{\Omega}_M t_O + \Omega_P\right) \Big|_{\mathbf{R}} \left[\begin{pmatrix} -\mu_M \frac{x_R}{r^3} \\ -\mu_M \frac{y_R}{r^3} \\ -\mu_M \frac{z_R}{r^3} \end{pmatrix} + \Big|_{\mathbf{R}} \mathbb{T}_{\mathbf{z}}(-\tau)\, \mathbb{T}_{\mathbf{y}}\left(\frac{\pi}{2} + \delta\right) \Big|_{\mathbf{V}} \mathbb{T}_{\mathbf{z}}(-\chi)\, \mathbb{T}_{\mathbf{y}}(-\gamma) \Big|_{\mathbf{B}} \left\{ \begin{pmatrix} -\frac{D}{m_{MAV}} \\ 0 \\ 0 \end{pmatrix} + \dots \right.\right.$$

$$\left.\left. \dots + \Big|_{\mathbf{B}} \mathbb{T}_{\mathbf{y}}(-\epsilon_T) \Big|_{\mathbf{P}} \begin{pmatrix} \frac{T}{m_{MAV}} \\ 0 \\ 0 \end{pmatrix} \right\} \right] \tag{6.10}$$

This manoeuvre could also be chosen to begin some time after the ignition of the second stage. Another aspect that could be used for the optimisation is the coasting time between the separation of the first stage and the second stage. All these different options result in a number of optimisation parameters which can be chosen to be active or not (either optimised or set as a constant parameter). These parameters are: burn-time of the first stage t_1, the first coasting time between the first and second stage t_{c1}, the elevation-gimbal angle ϵ_{T2} (with possible thrust control T_{T2}), the burn-time of the second stage t_2 and (if chosen to delay the second thruster manoeuvre) the time at which the second thruster control step starts t_{T2}.

6.3.4. COASTING AND ORBIT BURN

The final coasting phase starts at the second stage burn-out. There will still be some remaining propellant mass in the second stage at that point which will be used to perform the final orbit injection burn. The required propellant mass is a direct result of the necessary change in velocity required to propel the MAV into its final orbit from the current coasting trajectory. To compute this propellant mass, first the apocentre and pericentre of the current (elliptic) trajectory have to be determined [2] (see Equation (6.11)).

$$r_{ap} = \frac{-\mu_M r \pm \sqrt{(\mu_M r)^2 + r\left(rV^2 - 2\mu_M\right)\left(rV\cos(\gamma)\right)^2}}{rV^2 - 2\mu_M} \tag{6.11}$$

Here r_{ap} represents the pericentre r_p for the negative and apocentre r_a for the positive sign. At this point, a check is performed to assure that the reached r_a is equal to the radius of the desired orbit (in case of a circular target orbit) or equal to the pericentre of the desired orbit (in case of an elliptic target orbit). The general equations that can be used for this check are provided in Equation (6.12) [109] where a and e are the semi-major axis and eccentricity of the desired target orbit respectively, which can be optimisation parameters as well, and the underscore c stands for check.

$$\begin{aligned} r_{a,c} &= a(1+e) \\ r_{p,c} &= a(1-e) \end{aligned} \tag{6.12}$$

The velocity at the apocentre can then be computed using Equation (6.13).

$$V_a = \sqrt{V^2 - 2\mu_M\left(\frac{1}{r} - \frac{1}{r_a}\right)} \tag{6.13}$$

The required ΔV to reach the desired orbit is then given by Equation (6.14).

$$\Delta V = V_a - \sqrt{\mu_M\left(\frac{2}{r_a} - \frac{1}{a_{target}}\right)} \tag{6.14}$$

Now that the required velocity change is known, the corresponding required propellant mass Δm_b can be computed through the use of Tsiolkovsy's equation as the difference between the mass of the second stage at burn-out m_2 and the final mass of the second stage $m_{2,f}$ after the orbit burn. The expression for Δm_b is then provided by Equation (6.15).

$$\Delta m_b = m_2 \left(1 - exp\left(\frac{-\Delta V}{g_0 I_{sp}}\right)\right) \tag{6.15}$$

During the coasting and the final burn, the only optimisation parameters are the eccentricity and the semi-major axis of the desired orbit. Thus the maximum number of optimisation parameters for the ascent phase is 14 and are: t_v, t_{T1}, T_{T1}, ϵ_{T1}, ψ_{T1}, t_1, t_{c1}, t_2, t_{T2}, ϵ_{T2}, T_{T2}, a, e and i. Most of these parameters are free, but it can be chosen to fix some of these as well. The number of optimisation parameters can also be increased using the design space for the MAV provided in Section 3.4.3 which would include the dimensions of the MAV. Also, to add more flexibility, more intermediate thrust angles could be set as optimisation parameters. This would simply add (or even mix up) the different kinds of trajectories.

6.4. MARS 2020 LANDING SITE CONDITIONS

In the beginning of August 2015 the second Workshop on the Mars 2020 Landing Site was held in Monrovia, California [4]. Directly following this workshop, a meeting was held to select eight potential landing sites for the Mars 2020 rover. These sites are (alphabetically) presented in Table 6.2 including the position data. Please note that the latitude and longitudes are within an accuracy of approximately 5 km.

Table 6.2: Current candidate landing sites (source:JPL)

Landing site	**Altitude w.r.t. MOLA geoid [km]**	**Latitude [°N]**	**Longitude [°E]**
Columbia Hills/Gusev	-1.9	-14.4	175.6
Eberswalde	-1.4	-23.0	327.0
Holden	-2.1	-26.4	325.1
Jezero	-2.5	18.5	77.4
Mawrth	-2.3	24.0	341.1
NE Syrtis	-2.2	17.8	77.1
Nili Fossae	-0.6	21.0	74.5
SW Melas	-1.9	-12.2	290.0

There are also several landing site constraints that are important to the thesis problem:

- **Elevation** The elevation of the landing site must be below +0.5 km MOLA elevation, which is with respect to the MOLA geoid (defined to be at a radius of 3396 km [110–112], not to be confused with the volumetric mean radius mentioned in [107]).
- **Latitude** The landing site latitude has to be between ±30° of the equator.

These constraints result in a band where the landing sites can be chosen. The location of the eight chosen potential landing sites are indeed located in this band as shown by Figure 6.4 [4].

The position data is all provided in the R-frame. With the different latitudes, the respective local rotational velocity $V_{y_{R,0}}$(due East or in the y-direction in the R-frame), as viewed from the I-frame can be determined. These velocities provide the values for the initial velocity vector. The relation for $V_{y_{R,0}}$ is given in Equation (6.16) and the initial velocity vector is then provided through Equation (6.17). Here, $\dot{\Omega}_M$ is the rotational velocity of Mars, the L-frame is attached to the landing site and R_M is the (local) radius of Mars.

$$V_{y_{R,0}} = \dot{\Omega}_M R_M \cos(\delta_0) \tag{6.16}$$

$$\begin{pmatrix} V_{x_{I,0}} \\ V_{y_{I,0}} \\ V_{z_{I,0}} \end{pmatrix} = \Bigg|_{\mathbf{I}} \mathbb{T}_{\mathbf{z}}\left(-\dot{\Omega}_M t_O + \Omega_P\right) \Bigg|_{\mathbf{R}} \mathbb{T}_{\mathbf{z}}\left(-\tau_0\right) \Bigg|_{\mathbf{L}} \begin{pmatrix} 0 \\ V_{y_{R,0}} \\ 0 \end{pmatrix} = \begin{pmatrix} -\sin\left(\tau_0 + \dot{\Omega}_M t_O - \Omega_P\right) V_{y_{R,0}} \\ \cos\left(\tau_0 + \dot{\Omega}_M t_O - \Omega_P\right) V_{y_{R,0}} \\ 0 \end{pmatrix} \tag{6.17}$$

For the initial analysis and simulations, only one (the current primary candidate) landing site will be used. Other landing sites can be used as MAV if extra time is available at the end of the thesis work. At this point NASA [4] suggests that only the Nili Fossae site is guaranteed not to require terrain relative navigation, which will make the landing easier. Therefore, at this time it is assumed that this is the primary landing site candidate. This will however have to be confirmed.

[4] Landing site workshop: http://marsnext.jpl.nasa.gov/ [Accessed 19 November 2015]

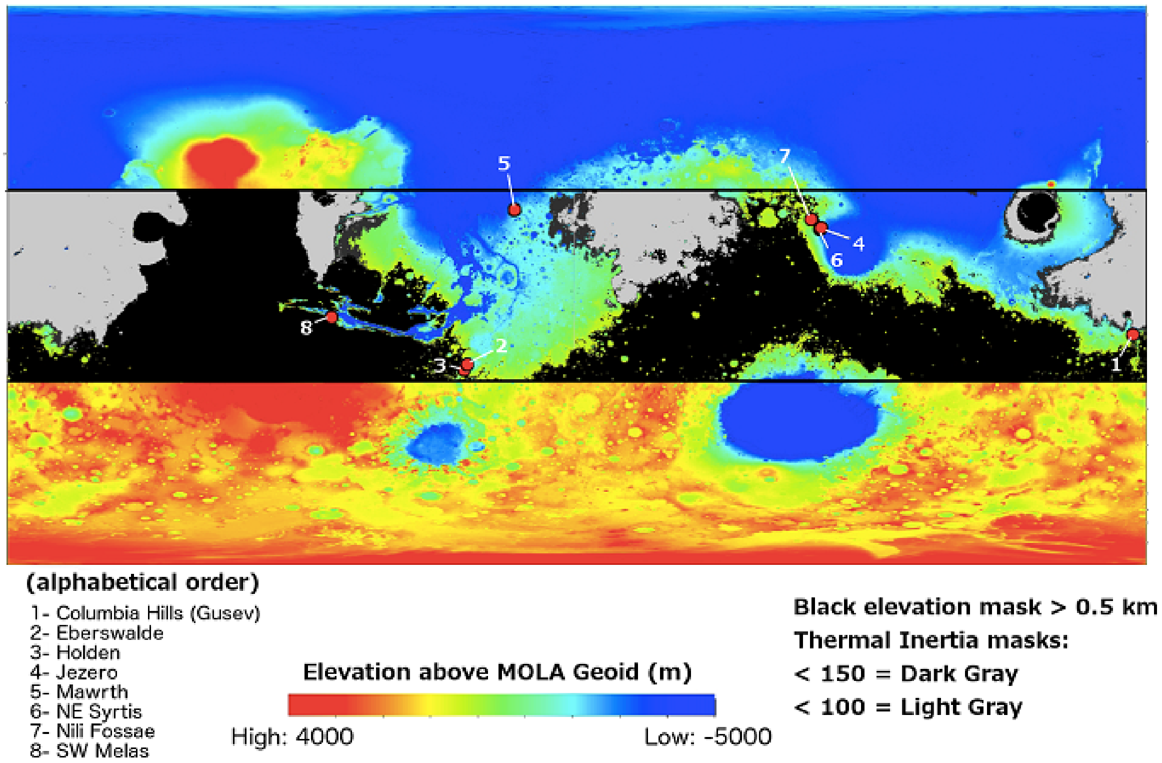


Figure 6.4: Locations of the eight potential landing sites within the constrained area.

7

Transfer orbits

The second part of the thesis problem involves (transfer) orbits around Mars. After the ascent phase of the MAV, the vehicle will be inserted into an orbit around Mars. Since it will use its high-thrust second-stage engine to perform the final burn to do this it is required to understand the different possible orbits that can be reached using such high-thrust. Therefore Section 7.1 will discuss the different high-thrust orbits. At that time, the orbiter should already be in position to rendezvous with the MAV. The orbiter will therefore have to travel from its original (operational science) orbit to the rendezvous orbit around Mars. However, compared to the MAV, the orbiter will be using a low-thrust propulsion system. These propulsion systems can be turned on for longer amounts of time to provide a continuous thrust. The low-thrust trajectories are discussed in Section 7.2. Because low-thrust transfers are relatively new, all available planetocentric reference missions have been flown in the Earth system (Section 3.2). The reference mission for low-thrust missions will be discussed and the most important information will be summarized in Section 7.2 as well. Section 7.3 describes the models and the general EoM required to represent the motion of the orbiter. The thrust history for the orbiter can be constructed using the Q-law. This control law is described in Section 7.4. Finally, the Mars 2022 orbiter initial conditions and constraints will be provided in Section 7.5.

7.1. High thrust

The MAV will be using a high-thrust propulsion system in the last phase of the ascent, which is the burn into the target orbit. Basically, the MAV will transfer from an elliptical (ascent trajectory) 'orbit' to either a circular or a different elliptical orbit around Mars. This orbit injection burn, as it is often called, was described by the equations presented in Section 6.3.4. This burn does not however include the plane change that might need to be performed. It should be mentioned that any required inclination change between the MAV orbit and the orbit of the Mars 2022 orbiter should be performed by the orbiter because a low Mars orbit inclination change with a high-thrust propulsion system requires significantly more energy and thus propellant than an inclination change performed by the low-thrust propulsion system of the Mars 2022 orbiter at a (much) higher orbit [113].

7.2. Low thrust

The Mars 2022 orbiter will be using a low-thrust electric propulsion system and will be orbiting the planet to gather scientific data and perform observations. It will also be required to collect the Martian sample from the MAV and bring it back to Earth. On a Martian planetocentric level this means that the orbiter will have to travel from its operational science orbit to the lower MAV orbit and then back to a higher Martian orbit to escape Mars' gravity well and to reach a suitable Earth return orbit. The low-thrust transfer orbit is described in Section 7.2.1 and Section 7.2.2 describes the reference missions for the Mars 2022 orbiter transfer orbits.

7.2.1. Low-thrust electrical propulsion transfer

Low-thrust electric propulsion is characterised by a high specific impulse (I_{sp}) but very low thrust levels. It is very useful for long-duration missions and these engines can thrust continuously if necessary. This continuous thrusting would result in a spiral orbit as depicted in Figure 7.1, however the thrust can also be applied in increments where there are thrusting periods and coasting periods in the transfer orbit. This was for instance done by the Dawn spacecraft in combination with gravity assists (see Figure 7.2).

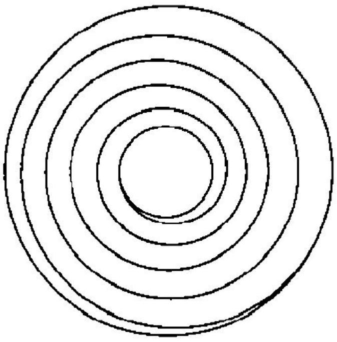


Figure 7.1: Example of continuous low-thrust trajectory [114].

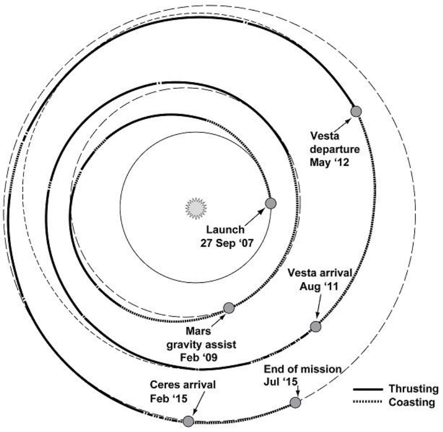


Figure 7.2: Example of a segmented low-thrust interplanetary trajectory: the Dawn transfer [113].

7.2.2. REFERENCE MISSIONS

The Mars 2022 orbiter will be propelled by an EP system, thus a spiralling transfer motion can be expected. The reference missions and research mentioned in Section 3.2 also used EP. Deep Space 1 is considered to be the first modern s/c to use EP as its main propulsion system used to perform orbital transfer manoeuvres. The engine was a Xenon ion engine. Ever since this mission Xenon engines have become increasingly popular and almost all modern low-thrust missions now use such engines [113]. These engines are either Hall thrusters or full ion engines. In Table 7.1 all flown reference missions that used either of these engines are mentioned and their engine characteristics are specified. This table also includes interplanetary missions.

There are already many different programs available that can simulate low-thrust trajectories. In three of the mentioned papers in Section 3.2.2 the different used methods, dynamic equations and EoM are described. [16] and [17] used the classical non-singular equinoctial orbital elements, instead of the Kepler elements, and their corresponding EoM to determine the orbit trajectories, whereas [19] describes a method of using Cartesian coordinates and the corresponding EoM. These reference methods can be used in the next section to set-up the dynamic model and corresponding EoM for the thesis problem.

[1] Artemis mission update: http://m.esa.int/Our_Activities/Telecommunications_Integrated_Applications/Artemis_finally_reaches_operational_orbit [Accessed 18 October 2015]

[2] AEHF-1 mission update: http://spaceflightnow.com/atlas/av019/111009.html [Accessed 18 October 2015]

[3] Online catalogue Aerojet Rocketdyne: https://www.rocket.com/propulsion-systems/electric-propulsion [Accessed 25 November 2015]

[4] Assumed 1 provided that the maximum available power is 9 kW, or 2 if no other system is active, which is less likely

[5] Boeing company: http://www.boeing.com/space/boeing-satellite-family/index.page [Accessed 18 October 2015]

Table 7.1: Flown low-thrust missions using Xenon engines and their characteristics (provided per thruster).

Year	Mission/s/c	Engine	Thrusters	Thrusters active at once	Engine power [kW]	I_{sp} [s]	Thrust [mN]	Ref.
1998	Deep Space 1	Ion	1	1	2.5 (max)	3120	92.4	[14, 113, 115]
2001	Artemis	Ion	4	1	0.57	3370	15 ±0.9%	[116] [1]
2003	Hayabusa	Ion	4	3	0.38	3160	8	[113]
2003	SMART-1	Hall	1	1	1.19	1631	68	[113]
2007	Dawn	Ion	3	1	2.6	3058	92	[113]
2010	AEHF(-1)	Hall	4	2	4.5	2020	267	Updates [2], [3]
2015	Boeing 702SP	Ion	4	1[4]	4.5	3500	165	[15, 117] [5]

7.3. DYNAMIC MODEL AND EQUATIONS OF MOTION

For the simulation of the orbiter motion it is important to have an accurate representation of reality. However, it should also still be possible to find a solution to the problem, which means that the model should also not be too complex. Most of the low-thrust model will be based on the work of Gebbett [100] and thus the same assumptions will be made and equations will be used. The dynamic model is described in Section 7.3.1 and the corresponding EoM are provided in Section 7.3.2.

7.3.1. DYNAMIC MODEL AND INITIAL ASSUMPTIONS

The motion of an s/c around a body can be described by Kepler elements (Figure 8.10) where a is the semi-major axis, e is the eccentricity, i the orbit inclination, ω the argument of periapsis, Ω the right ascension of the ascending node and θ the true anomaly. In a normal stable orbit all these elements except for θ are constant. The elements are in this case described in an inertial planet-centred reference frame.

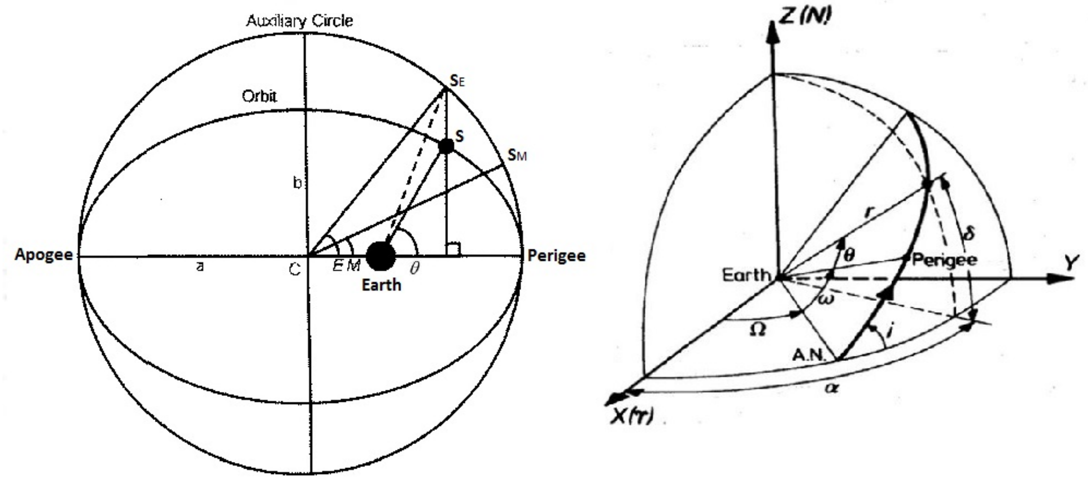


Figure 7.3: Definition of the Kepler elements in 2D (left) and 3D (right) [118, 119].

A stable orbit can however be disturbed which will cause the orbital elements to change and thus result in a different orbit. These disturbances are called orbital perturbations. During the orbital change of the Mars 2022 orbiter, the low-thrust propulsion system will cause a perturbation to change from one orbit to the other, however there are also other effects that can cause the orbit to change. The most important perturbations that were taken into account by [100] and that are meaningful to take into account in an orbit around Mars are the J_2 gravitational (flattening) effect, and the third-body perturbation caused by the Sun. Both effects will be discussed in the next section. Another element that is taken into account in the low-thrust orbital model is the shadowing effect of the planet. The orbiter will be using a propulsion technology called Solar Electric Propulsion (SEP) which requires such a great amount of power that it can only be used when the s/c is in direct sunlight. Therefore the propulsion system cannot be used during eclipse, which has to be taken into account when designing the propulsion scheme for the low-thrust orbit trajectory. More information on the application of the shadowing effect in the simulation tool can be found in [100].

7.3.2. Corresponding EoM and initial assumptions

To model the changes in the orbital elements as a result of the thrust and the different perturbations, Gauss' form of the Lagrange's planetary equations as shown by Equation (7.1) are recommended [100, 113, 120]. Here f_S, f_N and f_W are defined as in Figure 7.4 and p is the semi-latus rectum defined as $a\left(1-e^2\right)$.

$$
\begin{aligned}
\frac{da}{dt} &= 2\frac{a^2}{\sqrt{\mu_M p}}\left[e\sin(\theta)\,f_S+\frac{p}{r}f_N\right]\\
\frac{de}{dt} &= \frac{1}{\sqrt{\mu_M p}}\left[p\sin(\theta)\,f_S+\left(\left(p+r\right)\cos(\theta)+re\right)f_N\right]\\
\frac{di}{dt} &= \frac{r}{\sqrt{\mu_M p}}\cos(\theta+\omega)\,f_W\\
\frac{d\omega}{dt} &= \frac{1}{e\sqrt{\mu_M p}}\left[-p\cos(\theta)\,f_S+\left(p+r\right)\sin(\theta)\,f_N\right]-\frac{r\sin(\theta+\omega)\cos(i)}{\sqrt{\mu_M p}\sin(i)}f_W\\
\frac{d\Omega}{dt} &= \frac{r}{\sqrt{\mu_M p}\sin(i)}\sin(\theta+\omega)\,f_W\\
\frac{d\theta}{dt} &= \frac{\sqrt{\mu_M p}}{r^s}+\frac{1}{e\sqrt{\mu_M p}}\left[p\cos(\theta)\,f_S-\left(p+r\right)\sin(\theta)\,f_N\right]
\end{aligned}
\tag{7.1}
$$

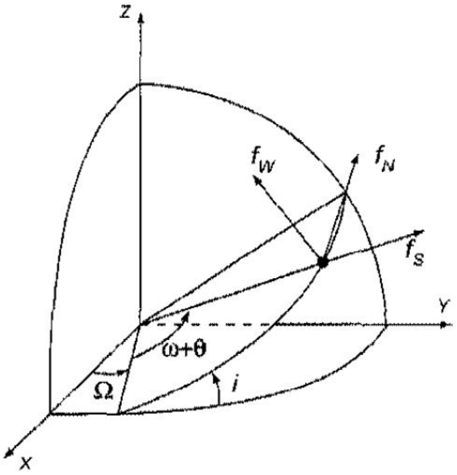


Figure 7.4: Definition of the perturbing accelerations [113].

However, it is clear that for $e=0$ and/or $i=0°$ or $180°$ this set of equations results in singularities. Therefore a slightly modified set of elements is used to eliminate these singularities. In [100] the Modified Equinoctial Elements (MEE) were deemed most suited based on the survey performed in [121] because this set is non-singular. The definition of the six elements is provided in Equation (7.2), where the semi-latus rectum is still the same as defined before but now directly used. More information on the MEE will be provided in Chapter 8.

$$
\begin{aligned}
p &= a\left(1-e^2\right) & h &= \tan^I\left(\frac{i}{2}\right)\cos(\Omega)\\
f &= e\cos(\omega+I\Omega) & k &= \tan^I\left(\frac{i}{2}\right)\sin(\Omega)\\
g &= e\sin(\omega+I\Omega) & L &= \omega+\theta+I\Omega
\end{aligned}
\tag{7.2}
$$

In Equation (7.2) I is the retrograde factor and has a value of +1 for posigrade orbits and -1 for retrograde orbits. This is done to avoid singularities when $i = 180°$ and thus making the set non-singular.

The corresponding differential equations are also provided in [121] and corrected by [100]. These corrected equations are shown in Equation (7.3) where $s^2 = 1 + h^2 + k^2$ and $w = p/r = 1 + f\cos(L) + g\sin(L)$. These equations are used to propagate the orbit.

$$
\begin{aligned}
\frac{dp}{dt} &= \frac{2p}{w}\sqrt{\frac{p}{\mu_M}}f_N \\
\frac{df}{dt} &= \sqrt{\frac{p}{\mu_M}}\left[\sin(L)\, f_S + \big((w+1)\cos(L) + f\big)\frac{f_N}{w} - (h\sin(L) - k\cos(L))\, g\frac{f_W}{w}\right] \\
\frac{dg}{dt} &= \sqrt{\frac{p}{\mu_M}}\left[-\cos(L)\, f_S + \big((w+1)\cos(L) + g\big)\frac{f_N}{w} + (h\sin(L) - k\cos(L))\, f\frac{f_W}{w}\right] \\
\frac{dh}{dt} &= \sqrt{\frac{p}{\mu_M}}\frac{s^2 f_W}{2w}\cos(L) \\
\frac{dk}{dt} &= \sqrt{\frac{p}{\mu_M}}\frac{s^2 f_W}{2w}\sin(L) \\
\frac{dL}{dt} &= \sqrt{\mu_M p}\left(\frac{w}{p}\right)^2 + \frac{1}{w}\sqrt{\frac{p}{\mu_M}}(h\sin(L) = k\cos(L))\, f_W
\end{aligned}
\tag{7.3}
$$

When it comes to the other perturbations, these can simply be added as an extra effect in any of the three perturbation acceleration directions. J_2 on Mars has a value of 1960.45$\cdot 10^{-6}$ [107]. The general equations for gravitational perturbations in normal Kepler elements are discussed in [108], however in this case, the same equations are needed for the MEE. Fortunately, these are provided by [122] and are given in Equation (7.4). Here s is again defined as for Equation (7.3).

$$
\begin{aligned}
f_{S,J_2} &= -\frac{3\mu_M J_2 R_M^2}{2r^4}\left(1 - 12\frac{(h\sin(L) - k\cos(L))^2}{s^4}\right) \\
f_{N,J_2} &= -\frac{12\mu_M J_2 R_M^2}{r^4}\left(\frac{(h\sin(L) - k\cos(L))\,(h\cos(L) + k\sin(L))}{s^4}\right) \\
f_{W,J_2} &= -\frac{6\mu_M J_2 R_M^2}{r^4}\left(\frac{(h\sin(L) - k\cos(L))\left(1 - h^2 - k^2\right)}{s^4}\right)
\end{aligned}
\tag{7.4}
$$

The perturbing acceleration caused by a third body are important to take into account if the third body has a high mass compared to the primary (orbiting) body. Mars has two natural satellites: Phobos and Deimos. However, the mass of these two satellites is relatively small compared to that of Mars [107]. The gravitational acceleration of Phobos acting on an orbiting s/c is on the order of 10^{-8} (compared to the accelerations caused by the J2-effect, which is in the order of 10^{-3}). Thus it is assumed that the Martian moons do not have to be taken into account in the third-body perturbations, however should this decision change, they can always be added in the future. Also, the orbit of Mars 2022 will be different from that of the two moons, which will reduce the effects even more. The Sun however is a body that will be taken into account, because the gravitational accelerations are in the order of 10^{-7} and are almost always present. In [100] a method by Battin [123] is used to determine the accelerations caused by the third body, which in this case is the Sun. Battin's function is described in Equation (7.5) where $\mathbf{r}$ is the position vector of the orbiter with respect to Mars and $\mathbf{r_{MS}}$ is the position vector of the Sun with respect to Mars.

$$
f(q) = q\left[\frac{3 + 3q + q^2}{1 + (1+q)^{\frac{3}{2}}}\right] \qquad \text{where} \qquad q = \frac{\mathbf{r}\cdot(\mathbf{r} - 2\mathbf{r_{MS}})}{\mathbf{r_{MS}}\cdot\mathbf{r_{MS}}}
\tag{7.5}
$$

The third-body acceleration can then be described by Equation (7.6). Please note that these accelerations are described in the inertial Mars-centred frame in Cartesian coordinates and will thus have to be transformed to the s/c centred orbital frame (Figure 7.4) to be added to the other accelerations that are acting on the orbiter. This transformation will be described in Chapter 8.

$$\mathbf{a}_S = -\frac{\mu_S}{\left(\mathbf{r}-\mathbf{r_{MS}}\right)^3}\left(\mathbf{r}+f\left(q\right)\mathbf{r_{MS}}\right) \tag{7.6}$$

Here the standard gravitational parameter for the perturbing body, in this case the Sun, is used. For a given position of the celestial bodies and the orbiting s/c the most important perturbations can now be computed.

7.4. Q-LAW

During the transfer of the orbiter, the low-thrust engine will be used to perform the orbital changes. However, unlike high-thrust engines, low-thrust engines can be active for a long period of time. They can also be turned off and on again. Determining the time that the engines have to be active and in which direction the thrust has to be pointed can be very difficult and time consuming. This is why, at JPL back in 2003, Petropoulos developed a feedback control law called Q-law [124], which can very accurately provide a good representation of the thrusting behaviour required to get from a starting orbit to a target orbit. The specific point in that orbit θ can not be pre-set. In the years following, the Q-law was refined ([120, 125]) and the latest version of the fundamental function Q is presented in Equation (7.7).

$$Q = \left(1+W_P P\right)\sum_{\text{œ}} W_{\text{œ}} S_{\text{œ}}\left[\frac{d\left(\text{œ},\text{œ}_T\right)}{\tilde{\dot{\text{œ}}}_{xx}}\right]^2, \qquad \text{for} \qquad \text{œ} = a, e, i, \omega, \Omega \tag{7.7}$$

Here W_P and $W_{\text{œ}}$ are weights which have a scalar value equal to or greater than 0. P is a penalty function depending on the problem, but in [120] it was used to define a minimum periapsis radius which the orbit had to satisfy. This function is defined in Equation (7.8). $S_{\text{œ}}$ is a scalar function as defined by Equation (7.9) and $d\left(\text{œ},\text{œ}_T\right)$ is a distance function (see Equation (7.10)) where the subscript T stands for target. Finally, $\tilde{\dot{\text{œ}}}_{xx}$ represents the maximum rate of change of the orbital elements over the thrust direction and true anomaly as defined by Equation (7.11).

$$P = exp\left[k\left(1-\frac{r_p}{r_{p,min}}\right)\right] \tag{7.8}$$

Here k is a scalar that can be adjusted if required. For instance, if at the end of the propagation it turns out that during the propagation the orbit went 2 km below the allowed minimum perigee radius and it is preferred to only go 1 km below the allowed minimum perigee radius, then the value for k should be increased. This is typically done by trial and error.

$$S_{\text{œ}} = \begin{cases} \left[1+\left(\dfrac{a-a_T}{m a_T}\right)^n\right]^{\dfrac{1}{r}}, & \text{for œ} = a \\ 1, & \text{for œ} = e, i, \omega, \Omega \end{cases} \tag{7.9}$$

In Equation (7.9) the nominal values for m, n and r are respectively 3, 4 and 2.

$$d\left(\text{œ},\text{œ}_T\right) = \begin{cases} \text{œ}-\text{œ}_T, & \text{for œ} = a, e, i \\ \arccos\left[\cos\left(\text{œ}-\text{œ}_T\right)\right], & \text{for œ} = \omega, \Omega \quad \text{where} \quad \cos\left(\text{œ}-\text{œ}_T\right) \in [0,\pi] \end{cases} \tag{7.10}$$

$$\tilde{\dot{\text{œ}}}_{xx} = \begin{cases} \dot{\text{œ}}_{xx} = \max\limits_{\alpha,\beta,\theta}, & \text{for œ} = a, e, i, \Omega \\ \left(\dot{\omega}_{xxi} + b\dot{\omega}_{xxo}\right)/\left(1+b\right), & \text{for œ} = \Omega \end{cases} \tag{7.11}$$

Here $\dot{\text{œ}}$ are provided by Equation (7.1), $\alpha \in [-\pi, \pi]$ is the in-plane thrust angle and $\beta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ is the out-of-plane thrust angle [100]. The subscript x means that it is a maximum. In this case two x's are used because first the maximum is taken using a combination of α and β resulting in $\dot{\text{œ}}_x$ and then the maximum is taken using θ thus resulting in the notation $\dot{\text{œ}}_{xx}$. Also, b is a non-negative constant with a nominal value of 0.01 and both $\dot{\omega}_{xxi}$ and $\dot{\omega}_{xxo}$ are defined in Equation (7.12).

$$\begin{aligned} \dot{\omega}_{xxi} &= \max_{\alpha,\theta}\left(\dot{\text{œ}}|_{\beta=0}\right) \\ \dot{\omega}_{xxo} &= \max_{\theta}\left(\dot{\text{œ}}|_{\beta=\pi/2}\right) \end{aligned} \tag{7.12}$$

Now using Equation (7.13), the accelerations in Equation (7.3) can be written such that Q is not a function of α, β and θ as described in [120].

$$\begin{aligned} f_S &= f_{Thrust}\cos(\beta)\sin(\alpha) \\ f_N &= f_{Thrust}\cos(\beta)\cos(\alpha) \\ f_W &= f_{Thrust}\sin(\beta) \end{aligned} \tag{7.13}$$

Q is zero at the target orbit and positive elsewhere. The greater the value of Q the further away the s/c is from the target orbit. The idea is to reduce Q to zero as soon as possible. Therefore at every time step, the largest reduction of Q is required. This can be computed through Equation (7.14). Here the subscript n denotes that it is a minimum.

$$\begin{gathered} \frac{dQ}{dt} = \dot{Q} = \sum_{\text{œ}} \frac{\delta Q}{\delta \text{œ}} \dot{\text{œ}} \\ \dot{Q}_n = \min_{\alpha,\beta} \dot{Q} \end{gathered} \tag{7.14}$$

From this it can be determined in which direction the thrust has to be directed. The next step is to determine whether it is useful to provide this thrust at this instance or not. This is done through the so-called effectivity. The absolute and relative effectivity are defined as in Equation (7.15).

$$\begin{aligned} \eta_a &= \frac{\dot{Q}_n}{\dot{Q}_{nn}} &\qquad where \quad \dot{Q}_{nn} &= \min_{\theta} \dot{Q}_n \\ \eta_r &= \frac{\dot{Q}_n - \dot{Q}_{nx}}{\dot{Q}_{nn} - \dot{Q}_{nx}} &\qquad where \quad \dot{Q}_{nx} &= \max_{\theta} \dot{Q}_n \end{aligned} \tag{7.15}$$

It is up to the user to determine which of the two effectivities to use, because it is problem dependent. But both values will have to be higher than a certain cut-off value η_{cut} for the control law to accept it as a point where thrust has to be applied. If the effectivity is lower than this η_{cut} it means that no thrust will be applied and the s/c is thus in a coasting phase. According to Petropoulos it is better to use the relative (rather than absolute) effectivity for near-circular orbits, which is based on his experience.
The equation for the Q-law itself is used to determine the direction in which thrust has to be applied and the Q-law effectivity determines whether this thrust should be applied or not. This results in a thrust which is incorporated in the orbit propagation. However, in its current form, singularities can occur, which is why [100] wrote the Q-law in MEE using the Gauss' equations provided in Equation (7.3). This form (described by [100]) will also be used in the thesis problem to avoid any singularities and to be able to optimise for every kind of orbit.

The simplified form of the Q-law using MEE is provided by Equation (7.16)[100]. This formulation is similar to the original Q-law formulation that was introduced in 2003 [124].

$$Q_{MEE} = \sum_{\xi} W_{\xi} \left[\frac{\xi - \xi_t}{\dot{\xi}_x} \right]^2, \qquad for \qquad \xi = p, f, g, h, k \tag{7.16}$$

Again W_ξ are weights with a scalar value equal to or greater than 0 and ξ depicts the MEE. Also, since only one maximum is taken (only the maximum with respect to L) the notation for $\dot{\xi}_x$ has one x in the subscript. In this case Gebbett found that it is more computationally efficient to use the expressions for the accelerations in the different directions directly instead of using the two thrust angles. Therefore, the expressions for $\dot{\xi}_x$ are found through the EoM given by Equation (7.3) and applying Equation (7.17).

$$\dot{\xi}_x = \max_L \left(\dot{\xi}\right) \qquad for \qquad \xi = p, f, g, h, k \tag{7.17}$$

For $\dot{p}_x$, $\dot{h}_x$ and $\dot{k}_x$ the analytical expressions for the critical points can be found for which L is maximum. The value for $\dot{p}_x$, $\dot{h}_x$ and $\dot{k}_x$ is then the maximum of both critical values resulting from the equations mentioned in Equation (7.18) [100].

$$\begin{aligned}
\dot{p}_x &= \max\left(\dot{p}_{cr}\right), & \dot{p}_{cr} &= \frac{2p}{1 \pm \dfrac{f}{\sqrt{\dfrac{g^2}{f^2}+1}} \pm \dfrac{g}{f\sqrt{\dfrac{g^2}{f^2}+1}}}\sqrt{\frac{p}{\mu_M}} f_{Thrust} \\
\dot{h}_x &= \max\left(\dot{h}_{cr}\right), & \dot{h}_{cr} &= \frac{\pm\sqrt{1-g^2}}{1 \pm f\sqrt{1-g^2}-g^2}\sqrt{\frac{p}{\mu_M}\frac{s^2}{2}} f_{Thrust} \\
\dot{k}_x &= \max\left(\dot{k}_{cr}\right), & \dot{k}_{cr} &= \frac{\pm\sqrt{1-f^2}}{1 \pm g\sqrt{1-f^2}-f^2}\sqrt{\frac{p}{\mu_M}\frac{s^2}{2}} f_{Thrust}
\end{aligned} \tag{7.18}$$

Because $\dot{f}$ and $\dot{g}$ are a function of all the three acceleration directions, a direct analytical solution was deemed impossible to find by [100]. Therefore the functions have to be evaluated numerically for L to determine the critical values for the three different accelerations. The equations for $\dot{f}$ and $\dot{g}$ (Equation (7.3)) can be slightly rewritten to single out the different accelerations as shown in Equation (7.19). Here $H(L)$, $J(L)$ and $K(L)$ are simply the rewritten parts of the original expression and are shown in Equation (7.20) for both $\dot{f}$ and $\dot{g}$ respectively.

$$\begin{aligned}
\dot{f} &= H_f(L) f_S + J_f(L) f_N + K_f(L) f_W \\
\dot{g} &= H_g(L) f_S + J_g(L) f_N + K_g(L) f_W
\end{aligned} \tag{7.19}$$

$$\begin{aligned}
H_f &= \sqrt{\frac{p}{\mu_M}}\sin(L) & H_g &= -\sqrt{\frac{p}{\mu_M}}\cos(L) \\
J_f &= \sqrt{\frac{p}{\mu_M}}\frac{(w+1)\cos(L)+f}{w} & J_g &= \sqrt{\frac{p}{\mu_M}}\frac{(w+1)\cos(L)+g}{w} \\
K_f &= -\sqrt{\frac{p}{\mu_M}}\frac{h\sin(L)-k\cos(L)}{w}g & K_g &= \sqrt{\frac{p}{\mu_M}}\frac{h\sin(L)-k\cos(L)}{w}f
\end{aligned} \tag{7.20}$$

It is mentioned that this method is very slow, but it does result in a robust system that can handle both elliptic and circular orbits. To have a slightly faster evaluation [100] decided to evaluate $\dot{f}_x$ and $\dot{g}_x$ only once every revolution because the orbital elements do not show a significant difference in that one revolution. This is only an approximation but in Equation (7.16) the weights can be used to make the contribution of both $\dot{f}_x$ and $\dot{g}_x$ less important and have a lower influence on the solution. Q_{MEE} is again zero at the target orbit and positive elsewhere. In this case Equation (7.21) (based on Equation (7.15)) is used to determine the largest rate of change of Q_{MEE}. For further information see [100].

$$\begin{aligned}
\frac{dQ_{MEE}}{dt} &= \dot{Q}_{MEE} = \sum_{\xi}\frac{\delta Q_{MEE}}{\delta \xi}\dot{\xi} \\
\dot{Q}_{MEE,n} &= \min_{f_S, f_N, f_W} \dot{Q}_{MEE}
\end{aligned} \tag{7.21}$$

With the minimum value of $\dot{Q}_{MEE}$ now known, the expressions for the absolute and relative effectivity (see Equation (7.15) used for the normal Q-law) can be applied here again to determine whether the thrust has to be applied or not. However, as was mentioned in Section 7.3.1 the orbiter is unable to thrust when it is in the shadow of the planet. Therefore when using the minima and maxima (with respect to L instead of θ) there is also a limit on L to avoid positions in the orbit that would be in the shadow of Mars [100]. Another important consideration is the fact that the program might decide to show thrust-on-off behaviour called on-off-chatter, which is caused by an η value close to the cut-off value [120]. A second type of chatter is the thrust direction chatter, where first the thrust is directed forward and then backward and then forward again (for instance) etcetera. Both types of chatter can be dealt with as described by Petropoulos [120] and Gebbett [100] but can never be completely eliminated.

The non-singular form of the Q-law Q_{MEE} does not include a penalty function as is the case with the normal Q-law. The described penalty function in Equation (7.8) provides a minimum condition for the pericenter radius. As will be discussed in the next section, the initial orbit of the Mars 2022 orbiter is at a relatively low altitude. This is why, in consultation with Petropoulos, it was decided that this penalty function might have to be added to Q_{MEE} depending on the behaviour during the simulation. This would change the partial derivatives for $\frac{\delta Q_{MEE}}{\delta \xi}$ and therefore the program written by Gebbett [100] would have to be updated.

7.5. MARS 2022 ORBITER INITIAL CONDITIONS

According to JPL the Mars 2022 orbiter shall be used to bring a Martian sample back to Earth. In the current architecture the orbiter will be launched from Earth two years before the MAV will. This means that the orbiter will also perform several scientific missions in the Martian system. Therefore, at the start of the thesis simulations the orbiter will be in a science operational orbit. In this case that means that the orbiter will be in a sun-synchronous orbit at an altitude of 320 km [6]. The precise orientation of the orbit with respect to an inertial Mars reference frame (ω & Ω) will depend on the simulated exact date at the beginning of the simulation. It is assumed that this initial orbit will be a circular orbit ($e = 0$). The precise point in this orbit (θ) will have to be chosen before the start of the simulation. Given that the starting orbit is a sun-synchronous orbit at an altitude of 320 km, the orbit characteristics a and i can be computed. In [107] it is mentioned that the volumetric mean radius of Mars R_M is 3389.5 km thus resulting in a semi-major axis of 3709.5 km. With this information, Equation (7.22) [108] can be solved for a pure J_2 (for Mars 1960.45$\cdot$ 10^{-6} [107]) sun-synchronous orbit around Mars with the orbital period of Mars around the Sun $T_{MS} = 5.9355\cdot 10^7$ s and $\mu_M = 4.283\cdot 10^{13}$ m^3/s^2.

$$\cos(i) = -\frac{4\pi}{3T_{MS}J_2}\left(\frac{a\left(1-e^2\right)}{R_M}\right)^2\sqrt{\frac{a^3}{\mu_M}} \tag{7.22}$$

From this it can be computed that the inclination will be 92.6979°. Thus the initial parameters will be a = 3,709.5 km, e = 0 and i = 92.6979°, and ω, Ω and θ are not specifically defined yet.

Besides the characteristics of the initial orbit, the engine parameters and the total mass of the orbiter at the start of the simulation are required. Currently, two different Xenon engines are under consideration. Table 7.2 shows the different characteristics depending on the type of engine that will be chosen. The max. engine power is the maximum power available for the worst case (which is at Mars aphelion).

Table 7.2: The two different engines currently under consideration for use on the Mars 2022 orbiter[6].

Engine	**Start Mass [kg]**	**Max. engine power [kW]**	**I_{sp} [s]**	**Thrust [mN]**
Hall	2500	10.0877	2600	250
Ion	2000	7.2055	3300	170

The ion engine is based on the same engine used by Boeing's 702SP satellite.

[6] Personal correspondence with JPL personnel

8

Reference systems and transformations

When considering the position, velocity and acceleration of a space vehicle, different reference frames are usually used. Why there are different frames, which frames are required, what they are used for and how to transfer from one frame to the other will be discussed in Sections 8.1 and 8.2. In case of, for instance, a car travelling on the road, a certain coordinate system has to be used to measure the velocity in a certain direction. For the car it can be said that a Cartesian system is used with x in the direction of motion, z pointing towards the ground and y then pointing to the right if looking in the direction of motion (using the so-called RHR to complete the frame). The speed is then measured and expressed in the x-direction. This works well when a vehicle is travelling in a straight line or on a flat plane, but in orbital mechanics the motion usually has to be described just above or in an orbit around a (spherical) body. Therefore it is important to know which coordinate system is used and how to change between these coordinate systems should the other system become more convenient to use at a certain point. In Section 8.3 the difference between these coordinate systems and how to transfer from one to the other is explained.

8.1. Required reference systems

In the example of the car, the Frame of Reference (RF) is fixed to the Earth. However, when considering two cars driving on the same road, one might like to investigate the difference in velocity between both vehicles. In that case an RF is chosen that is fixed to one of the two cars. This makes it easier to determine the relative velocity of one car with respect to the other. The same can be done for s/c (think of formation flying). There are therefore a number of different RFs that can be used. In this section the required RFs and their application will be presented.

8.1.1. Mars-centred inertial RF (I-frame)

The Mars-centred inertial (MCI) reference frame is an RF that is, as the name implies, fixed to the centre of mass of Mars [126]. It is a frame in which Mars still rotates with respect to that frame, or in other words it is an inertial frame with respect to Mars. The x-axis is defined through a certain point which can be set by the user. This means that the user can determine his or her own inertial frame. A frame that is often used (and which is based on J2000[1]) is the MARSIAU frame [127, 128]. This is a Mars inertial frame that is translated from the Earth J2000 frame. For the translation of J2000 a point through the mean equator of Mars is specified where the equator ascends through the mean equator of Earth. The x-axis is defined through this point. The z-axis is then defined through the rotational axis of Mars and the y-axis completes the frame through the RHR. The translation is described in [127, 128] using the J2000 numbers from [129]. The generic inertial frame is shown in Figure 8.1. The centre of the frame is represented by the letter ’A’. The MARSIAU frame is required if certain data from outside sources will have to be incorporated into the simulation, however for the actual simulation and optimisation it will be easier and much more straightforward to define a custom Mars inertial reference frame.

[1] J2000 is an Earth referenced inertial frame that was set through the rotational axis of Earth (z-axis) and through the vernal equinox (x-axis). The reference was set on the Julian date of 2451545.0 or 01-01-2000 at 12:00 Greenwich Mean Time (GMT).

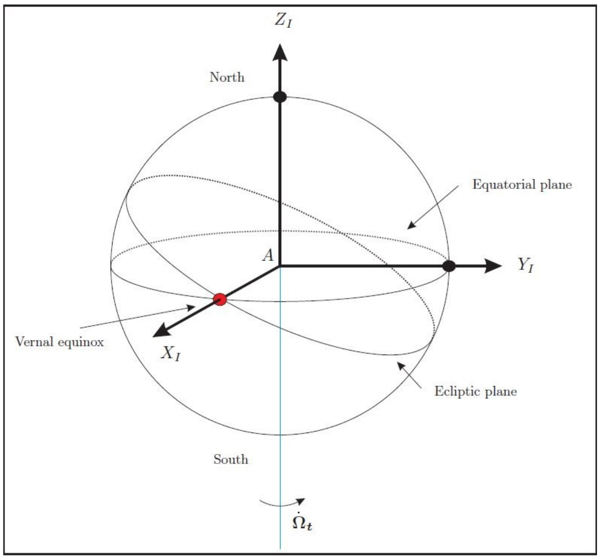


Figure 8.1: Graphical definition of the Mars-centred inertial reference frame [126].

This reference system is very useful if the motion of the s/c with respect to the ground-track is not important.

8.1.2. MARS-CENTRED MARS-FIXED RF (R-FRAME)

The Mars-centred Mars-fixed (MCMF) frame is the same as the MCI frame except for one distinct difference. The x-axis of the MCMF frame (x_R) is defined through the 0.5 km wide Airy-0 crater [130] which has a latitude of 5.07829°S [131]. The associated meridian is called the prime meridian. On 1 January 2000 at 12:00 the prime meridian of Mars was 176.630° east of the x-axis in the MARSIAU frame (which is important when transforming from the MARSIAU frame to the current MCMF frame)[128, 129]. Because the Airy-0 crater is fixed to the surface of Mars and rotates around the Martian rotational axis (in both MCI and MCMF the z-axis), the MCMF is also a rotational frame. This frame rotates with Mars and can therefore be very useful for ground observation purposes. In the case of the MAV it is important to take the rotational velocity of Mars into account, since Mars will be rotating underneath the MAV causing rotational effects. A graphical representation of the MCMF frame is provided in Figure 8.2. An angle Ω_P is defined as the relative angle between the prime meridian and the x-axis of the MCI frame at the time that the inertial frame was defined.

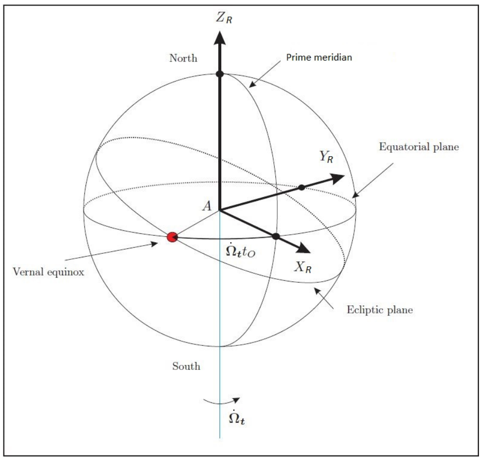


Figure 8.2: Graphical definition of the Mars-centred Mars-fixed reference frame [126]. Source: NASA

8.1.3. VERTICAL RF (V-FRAME)

This next RF can be used to describe the motion of an s/c in orbit around a planet. The centre of the vertical frame is defined in the centre of gravity of the s/c [126] but is not rotationally fixed to it. This means that the frame can rotate with respect to the s/c itself. This is because the x-axis is set to point to the north pole of Mars. The z-axis is perpendicular to Mars' surface (or points to the centre of Mars assuming Mars is a sphere) and the y-axis then points due east. The x-y plane is tangent to Mars' geoid. Figure 8.3 shows the vertical frame denoted by the letter 'V' at an arbitrary position around Mars compared to the MCI frame. The centre of the frame is defined by the letter 'G'. The vertical frame relates to the Mars rotational frame through the latitude and longitude (also see Figure 6.1).

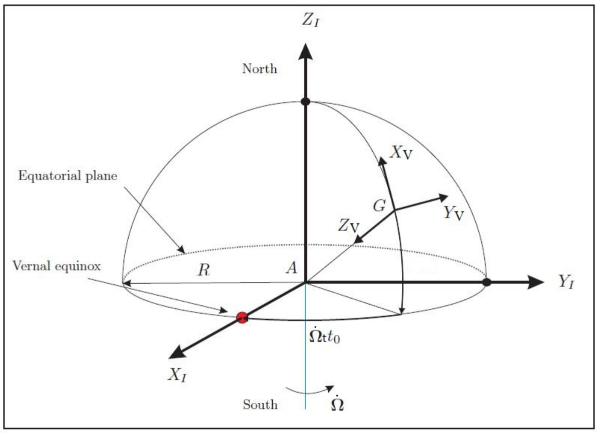


Figure 8.3: Graphical definition of the Vertical reference frame compared to the Mars-centred inertial frame [126].

8.1.4. BODY-FIXED RF (B-FRAME)

The body-fixed BF frame is fixed to the s/c body and therefore rotates with the s/c as well. The origin of the BF RF can be defined in the centre of symmetry (or the middle of the s/c). The axis orientation can be chosen to be in any direction. In this case the x-axis is defined through the vehicle-centre line and goes through the "nose" of the s/c, the z-axis is defined towards the orbiting body (when assuming an orbiting vehicle) and the y-axis completes the frame through the RHR. An example of a body frame that can be applied to the MAV is provided in Figure 8.4.

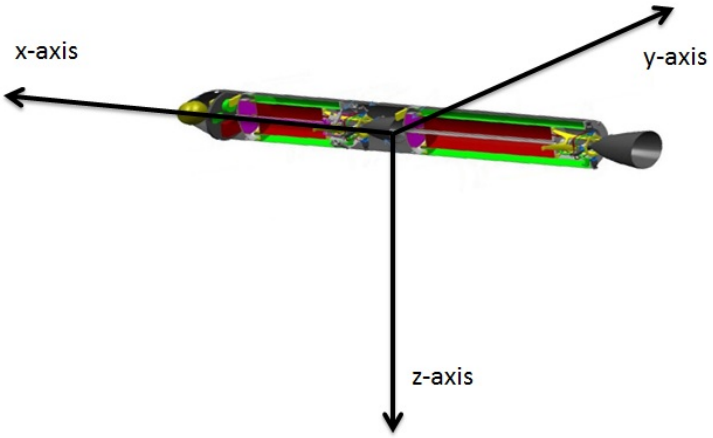


Figure 8.4: Graphical example of a Body-fixed reference frame for the MAV [7].

The axes of the body frame are denoted by a 'B' in this report. In the thesis problem the body frame relates directly to the vertical frame through the flight-path angle γ and the heading (or azimuth) angle χ, because the x-axis of the body frame coincides with the velocity vector of the s/c (also see Figure 6.1).

8.1.5. PROPULSION RF (P-FRAME)

The propulsion or thrust frame is a body frame where the x-axis is defined by the thrust vector. The y-axis is defined in the same plane as the x and y-axes of the B-frame. Then the z-axis completes the frame through the RHR as depicted in Figure 8.5.

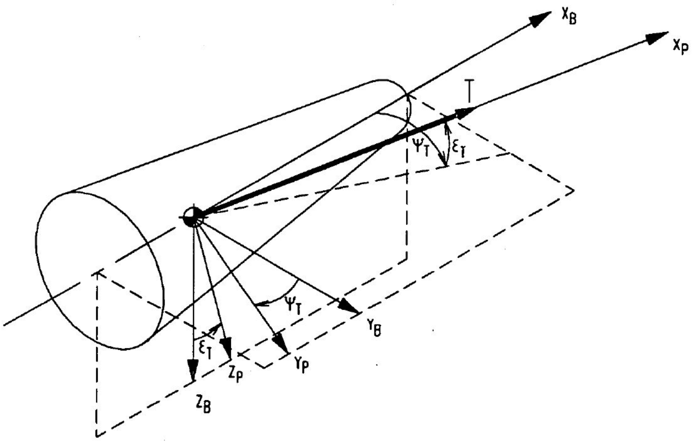


Figure 8.5: Propulsion frame relative to the body frame [105].

The B-frame and P-frame are related through the thrust elevation gimbal angle ϵ_T and the thrust azimuth gimbal angle ψ_T.

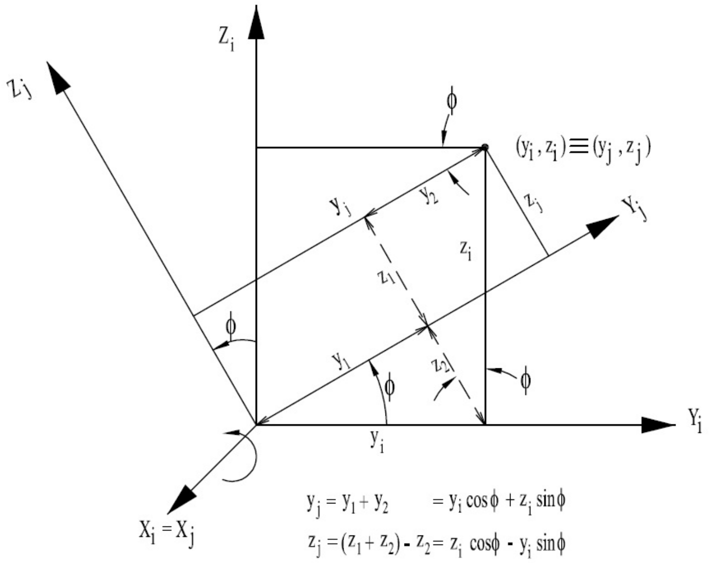
8.2. TRANSFORMATION BETWEEN REFERENCE FRAMES

Because different problems and situations are easier to understand and to formulate in different RFs, it is important to understand how to transform from one RF to the other. In this case only transformations for a set point in time, called static transformations, are required for the described thesis problem. The necessary transformations from Chapter 6 are described in this section.

For a transformation between two RFs, a correlation has to exist. This correlation (orientation) can be described using a set of angles, called the Euler angles, between the different RFs [132]. In a 3-dimensional space, there are three axes. So to get from one RF to the other, a maximum of 3 rotations (translations/-transformations) have to take place; one over each axis. However, often fewer rotations are necessary because some of the axes are already properly aligned. Sometimes two reference frames are not directly related through known angles, which means that other (intermediate) RFs will have to be used. In this case more than 3 rotations are possible, but again a maximum of three rotations are needed to go from one RF to an intermediate RF given the known angles. Since a 3-dimensional space is described, 3-dimensional coordinates have to be translated to the next reference frame. For this, Cartesian coordinates are used. A rotation around the x-axis can be described by the change in the y- and z-coordinates over an angle ϕ in the yz-plane as visualized in Figure 8.6. The rotation around the x-axis follows the RHR of rotation, which states that if the thumb of your right hand is pointed in the axis direction, then the rest of your fingers show the direction of positive orientation. Therefore the described rotation is positive. Should the rotation have been in the other direction (clockwise around the x-axis as seen in Figure 8.6), the angle would have been negative. This is very important in transformations, and its usefulness becomes more obvious when more transformations are performed after each other. But first a system of equations is required to transform from the 'i' frame to the 'j' frame.

From Figure 8.6 a relation can be described between the coordinates in the 'i' frame and the 'j' frame in matrix form. This relation (convention as by [132]) is shown by Equation (8.1). Here, $\mathbf{r}$ is the state vector containing the coordinate values and $\mathbb{T}_{\mathbf{x}}(\phi)$ represents the x-axis transformation matrix and is the standard matrix to be used if a transformation around an x-axis is performed. The only parameter that changes is the angle.

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_j = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}_i \Rightarrow \mathbf{r}_j = \mathbb{T}_{\mathbf{x}}(\phi)\mathbf{r}_i \tag{8.1}$$



Z_i
Z_j
ϕ
(y_i, z_i)≡(y_j, z_j)
y_j
y_2
z_j
Y_j
z_i
z_1
ϕ
y_1
ϕ
z_2
ϕ
Y_i
y_i
X_i = X_j
y_j = y_1 + y_2 = y_i cosϕ + z_i sinϕ
z_j = (z_1 + z_2) - z_2 = z_i cosϕ - y_i sinϕ

Figure 8.6: General rotation around the x-axis [132].

This rotation described by Figure 8.6 and Equation (8.1) can also be described for rotations around the y-axis and the z-axis. The complete set of axis transformation matrices is then presented in Equation (8.2).

$$\mathbb{T}_{\mathbf{x}}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}, \mathbb{T}_{\mathbf{y}}(\phi) = \begin{bmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{bmatrix}, \mathbb{T}_{\mathbf{z}}(\phi) = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8.2}$$

From now on, the cosine notation will be shortened to 'c' and the sine notation will be shortened to 's' to make the transformation matrices more comprehensive. Also, because the matrices are orthogonal, the inverse of the matrix is simply the transpose of the matrix. The inverse of a rotation matrix is the rotation in the other direction.

The order, or sequence, in which these transformations are performed is very important. This is because the sequence results in a matrix multiplication equation, and from linear algebra it is known that $\mathbf{A}\cdot\mathbf{B} \neq \mathbf{B}\cdot\mathbf{A}$. Therefore, if the sequence of transformations is not correct, the resulting RF will not be the desired RF. Also, as mentioned before, it is not always possible or easy to transform from one RF directly into the other. So usually one of the afore mentioned RF is used as an intermediate RF. It is therefore possible to transform from the P-frame all the way to the I-frame using all the other mentioned RFs with a maximum of three rotations between each frame. For convenience it has been chosen to adopt the convention mentioned in [126] and [132] to describe the transformations. Therefore the I-frame, R-frame, V-frame, B-frame and P-frame will be known as F_I, F_R, F_V, F_B and F_P respectively when used to indicate the RFs in equations. And because these equations involve matrix multiplications, they have to be executed from right to left. Therefore it can be said that the transformation matrix to go from $F_B \rightarrow F_R$ is $\mathbb{T}_{\mathbf{RB}} = \mathbb{T}_{\mathbf{RV}}\mathbb{T}_{\mathbf{VB}}$. This is one of the transformations required for the MAV trajectory. The other two transformations are more straightforward: $F_P \rightarrow F_B$ is $\mathbb{T}_{\mathbf{BP}}$ and $F_R \rightarrow F_I$ is $\mathbb{T}_{\mathbf{IR}}$. For the $\mathbb{T}_{\mathbf{IR}}$ matrix , a transition is performed from $F_R \rightarrow F_I$. This is done over the angle $\dot{\Omega}_t t_O$ as defined in Figure 8.2. $\dot{\Omega}_t$ is the rotational velocity of Mars and t_O is the time from the set inertial frame. This is also where Ω_{P0} is required (not depicted) since this angle describes the position of the prime meridian at the time that the inertial frame was set. Because the angle is defined east of the x_I-axis, it has to be subtracted from $\dot{\Omega}_t t_O$ since Ω_{P0} is defined positive in the same direction as $\dot{\Omega}_t t_O$ which is defined to be positive going from x_I to x_R. The rotation around the z-axis is positive in the same direction according to the RHR for rotation. In this case the rotation is performed in the negative direction around the z-axis with an angle $-\dot{\Omega}_M t_O + \Omega_P$. The complete RF transformation with the written out matrix for this rotation is given by Equation (8.3). Please note that the signs of the angles changed in the matrix due to the sine and cosine functions.

$$\mathbb{T}_{\mathbf{IR}} = \Big|_{\mathbf{I}} \; \mathbb{T}_{\mathbf{z}}\left(-\dot{\Omega}_M t_O + \Omega_P\right) \Big|_{\mathbf{R}} = \begin{bmatrix} c\left(\dot{\Omega}_M t_O - \Omega_P\right) & -s\left(\dot{\Omega}_M t_O - \Omega_P\right) & 0 \\ s\left(\dot{\Omega}_M t_O - \Omega_P\right) & c\left(\dot{\Omega}_M t_O - \Omega_P\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8.3}$$

Here the vertical lines with the letter designations depict when a certain reference frame is reached. The same notation was already shown in Chapter 6. The transformations for $\mathbb{T}_{\mathbf{RB}}$ and $\mathbb{T}_{\mathbf{BP}}$ can now be written in a similar manner and are described in Equations (8.4) and (8.5) respectively.

$$\begin{aligned} \mathbb{T}_{\mathbf{RB}} &= \Big|_{\mathbf{R}} \; \mathbb{T}_{\mathbf{z}}(-\tau)\,\mathbb{T}_{\mathbf{y}}\left(\frac{\pi}{2}+\delta\right) \Big|_{\mathbf{V}} \; \mathbb{T}_{\mathbf{z}}(-\chi)\,\mathbb{T}_{\mathbf{y}}(-\gamma) \Big|_{\mathbf{B}} \\ &= \begin{bmatrix} c\tau\left(-s\delta c\chi c\gamma + c\delta s\gamma\right) - s\tau s\chi c\gamma & c\tau s\delta s\chi - s\tau c\chi & c\tau\left(-s\delta c\chi s\gamma - c\delta c\gamma\right) - s\tau s\chi s\gamma \\ s\tau\left(-s\delta c\chi c\gamma + c\delta s\gamma\right) + c\tau s\chi c\gamma & s\tau s\delta s\chi + c\tau c\chi & s\tau\left(-s\delta c\chi s\gamma - c\delta c\gamma\right) + c\tau s\chi s\gamma \\ c\delta c\chi c\gamma - c\delta s\gamma & -c\delta s\chi & c\delta c\chi s\gamma - s\delta c\gamma \end{bmatrix} \end{aligned} \tag{8.4}$$

$$\mathbb{T}_{\mathbf{BP}} = \Big|_{\mathbf{B}} \; \mathbb{T}_{\mathbf{z}}\left(-\psi_T\right)\mathbb{T}_{\mathbf{y}}\left(-\epsilon_T\right) \Big|_{\mathbf{P}} = \begin{bmatrix} c\psi_T c\epsilon_T & -s\psi_T & c\psi_T s\epsilon_T \\ s\psi_T c\epsilon_T & c\psi_T & s\psi_T s\epsilon_T \\ -s\epsilon_T & 0 & c\epsilon_T \end{bmatrix} \tag{8.5}$$

Occasionally transformation matrices can become rather complex and it is easy to make mistakes. Fortunately the motion of the orbiter is already known in Kepler (or Modified Equinoctial) elements. This means that for the gravitational accelerations caused by the Sun described in Equation (7.6), which are in the MCI RF, they can be directly transformed to the body-fixed f_S, f_N and f_W RF. From now on this RF is referred to as the Gaussian frame or F_G, which can be similar to the F_V frame but does not have to be. The transformation from $F_I \rightarrow F_G$ follows from Figure 8.7 and is provided by the transformation convention and matrices described by Equation (8.6). In this case an intermediate reference frame is required which will be called $F_{I''}$, because two z-axis rotations are needed, but they are not in the same plane. It can also be seen that the position of the Prime meridian on the 1st of January 2000 at 12:00 is not important in this direct transformation and will only have to be taken into account in the transformation from $F_R \leftrightarrow F_I$ for the MAV.

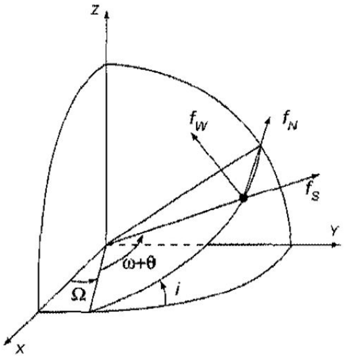


Figure 8.7: Relation between the MCI and the Gaussian body frame [113].

$$\begin{aligned} \mathbb{T}_{\mathbf{GI}} = \mathbb{T}_{\mathbf{GI''}}\mathbb{T}_{\mathbf{I''I}} &= \Big|_{\mathbf{G}} \; \mathbb{T}_{\mathbf{z}}(\omega+\theta) \Big|_{\mathbf{I''}} \; \mathbb{T}_{\mathbf{x}}(i)\,\mathbb{T}_{\mathbf{z}}(\Omega) \Big|_{\mathbf{I}} \\ &= \begin{bmatrix} c(\omega+\theta)c\Omega - s(\omega+\theta)cis\Omega & c(\omega+\theta)s\Omega + s(\omega+\theta)cic\Omega & s(\omega+\theta)si \\ -s(\omega+\theta)c\Omega - c(\omega+\theta)cis\Omega & c(\omega+\theta)cic\Omega - s(\omega+\theta)s\Omega & c(\omega+\theta)si \\ sis\Omega & -sic\Omega & ci \end{bmatrix} \end{aligned} \tag{8.6}$$

8.3. Transformation between different coordinate systems

In the previous sections the different RF and transformations were expressed using x, y and z-axes. These x, y and z-coordinates belong to the Cartesian coordinate system. However, during an integration, simulation or analysis it might be more useful and meaningful to express position (and velocity) in a different coordinate system. Two other systems are the spherical and Keplerian coordinate systems. How to transform back and forth from the Cartesian system to these other two systems will be explained in Sections 8.3.1 and 8.3.2 respectively and are based on [118]. The transformation from the spherical coordinate system to Kepler elements is described in Section 8.3.3. Kepler elements are very useful for orbit computations, unfortunately singularities can occur if this coordinate system is used for certain situations. Therefore Section 8.3.4 will discuss a non-singular form of the Kepler elements.

8.3.1. Spherical and Cartesian

Spherical coordinates can be useful to determine the location of a satellite above Mars assuming that Mars is a perfect sphere. In this thesis problem, however, spherical coordinates are used during the MAV ascent simulation. The coordinate relation between spherical and Cartesian coordinates is shown in Figure 8.8. Based on this diagram, the relations to go from the spherical system to the Cartesian system are derived and are provided in Equation (8.7). Please note that $\lambda \neq \tau$ except for a transformation to (or from) the R-frame.

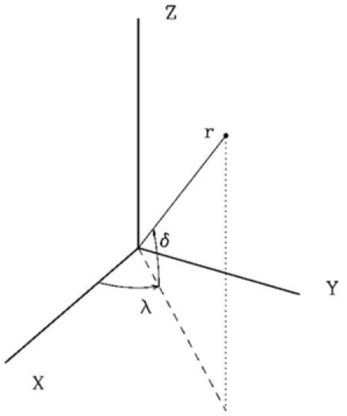


Figure 8.8: Relation between spherical position coordinates and Cartesian position coordinates [118].

$$\begin{aligned} x &= r\cos\delta\cos(\lambda \\ y &= r\cos\delta\sin(\lambda \\ z &= r\sin\delta \end{aligned} \tag{8.7}$$

The velocities can be obtained by taking the time derivatives of these functions, resulting in the velocity expressions in x-, y- and z-direction given by Equation (8.8). Differentiating those expressions again will result in the accelerations.

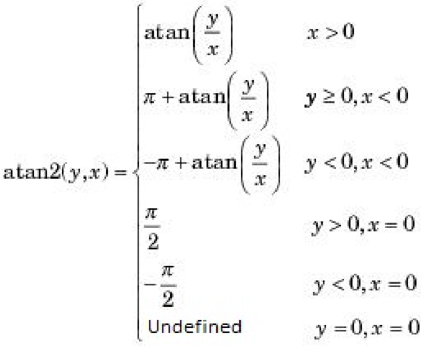
$$\begin{aligned} \dot{x} &= \dot{r}\cos\delta\cos(\lambda - r\dot{\delta}\sin\delta\cos(\lambda - r\dot{\lambda}\cos\delta\sin(\lambda \\ \dot{y} &= \dot{r}\cos\delta\sin(\lambda - r\dot{\delta}\sin\delta\sin(\lambda + r\dot{\lambda}\cos\delta\cos(\lambda \\ \dot{z} &= \dot{r}\sin\delta + r\dot{\delta}\cos\delta \end{aligned} \tag{8.8}$$

The transformation from Cartesian coordinates to spherical coordinates can again be derived from Figure 8.8, resulting in the expressions presented in Equation (8.9).

$$\begin{aligned} r &= \sqrt{x^2 + y^2 + z^2} \\ \delta &= \arcsin(\frac{z}{r}) \\ \lambda &= \arctan(\frac{y}{x}) \end{aligned} \tag{8.9}$$

However, the expression for λ only results in a value between $-\pi/2$ and $\pi/2$. Also, if x = 0, a singularity occurs. To solve this issue, the so-called 'atan2' function can be used. This function incorporates both sin and cosine values to provide a value for λ between 0 and 2π [118]. λ can then be expressed by Equation (8.10) where atan2 is defined by Figure 8.9[2].

$$\lambda = \text{atan2}(y, x) \tag{8.10}$$



$$\text{atan2}(y,x) = \begin{cases} \text{atan}\left(\frac{y}{x}\right) & x > 0 \\ \pi + \text{atan}\left(\frac{y}{x}\right) & y \geq 0, x < 0 \\ -\pi + \text{atan}\left(\frac{y}{x}\right) & y < 0, x < 0 \\ \frac{\pi}{2} & y > 0, x = 0 \\ -\frac{\pi}{2} & y < 0, x = 0 \\ \text{Undefined} & y = 0, x = 0 \end{cases}$$

Figure 8.9: atan2 function evaluation conditions.

The time derivatives of these functions can be taken to find the velocity expressions. The resulting velocity expressions are shown in Equation (8.11). The left expressions in this equation define the velocity characteristics as often expressed in the spherical coordinate system required for transformations.

$$\begin{aligned} \dot{r} &= \frac{x\dot{x} + y\dot{y} + z\dot{z}}{\sqrt{x^2 + y^2 + z^2}} & V &= \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} \\ \dot{\delta} &= \frac{x\dot{y} - y\dot{x}}{x^2 + y^2} & \gamma &= \arcsin\left[\frac{\dot{r}}{V}\right] \\ \dot{\lambda} &= \frac{r\dot{z} - z\dot{r}}{r^2\sqrt{1 - \left(\frac{z}{r}\right)^2}} & \chi &= \arccos\left[\frac{r\dot{\delta}}{V\cos\gamma}\right] \end{aligned} \tag{8.11}$$

8.3.2. Keplerian and Cartesian

The relation between the Keplerian and Cartesian systems is slightly more complex. Figure 8.10 shows the Kepler elements: a (semi-major axis), e (eccentricity, not shown but is a ratio that determines the elliptic properties of an orbit), i (inclination), ω (argument of perigee), Ω (right ascension of the ascending node) and θ (true anomaly). Often, however, the mean anomaly M is used instead of the true anomaly. The mean anomaly is defined to be the angle between the semi-major axis and the line between the centre of the orbit and an imaginary point (S_M) on the auxiliary circle that can be drawn around the orbit. This point is determined by assuming a constant rotational velocity in this circular orbit with the same orbital time as the actual (elliptical) orbit.

The transformation from Kepler elements to Cartesian coordinates consists of a number of steps starting with the mean anomaly and eccentricity. Both these values have to be used to determine the eccentric anomaly E and in turn the true anomaly θ. The eccentric anomaly is computed by rewriting the first expression of Equation (8.12) into the second expression [118].

$$\begin{aligned} M &= E - e \cdot \sin E \\ E_{i+1} &= E_i + \frac{M - E_i - e \cdot \sin E_i}{1 - e \cdot \cos(E_i)} \end{aligned} \tag{8.12}$$

The determination of E is an iterative process. First a reasonable estimate of E has to be provided (it is best to use the value for M for this) and then a new E can be computed. This has to be done until the desired level of precision is reached. Then the true anomaly can be computed using this E as shown by Equation (8.13).

[2] Online blog: http://guihaire.com/code/?p=1168 [Accessed 13 November 2015]

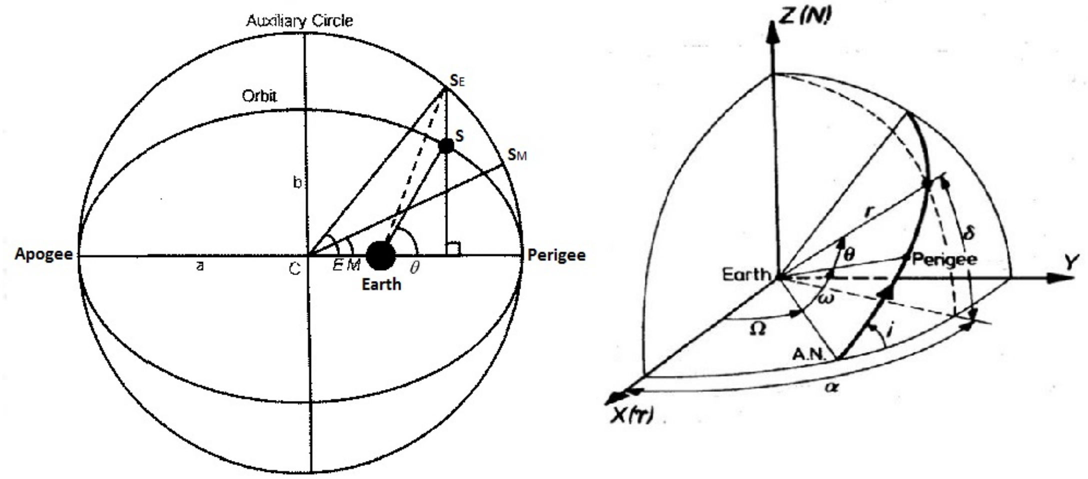


Figure 8.10: Definition of the Kepler elements in 2D (left) and 3D (right) [118, 119].

$$\theta = 2 \cdot \arctan\left[\tan\left(\frac{E}{2}\right)\sqrt{\frac{1+e}{1-e}}\right] \tag{8.13}$$

With the true anomaly known, the radial distance from the orbiting body to the centre of Mars r can be determined using Equation (8.14)

$$r = \frac{a \cdot (1-e^2)}{1 + e\cos\theta} \tag{8.14}$$

The Cartesian coordinates can now be computed using the expression in Equation (8.15) with the matrix entries provided by Equation (8.16)[118].

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{bmatrix} l_1 & l_2 \\ m_1 & m_2 \\ n_1 & n_2 \end{bmatrix} \begin{pmatrix} r\cos\theta \\ r\sin\theta \end{pmatrix} \tag{8.15}$$

$$\begin{aligned} l_1 &= \cos\Omega\cos\omega - \sin\Omega\sin\omega\cos i \\ l_2 &= -\cos\Omega\sin\omega - \sin\Omega\cos\omega\cos i \\ m_1 &= \sin\Omega\cos\omega + \cos\Omega\sin\omega\cos i \\ m_2 &= -\sin\Omega\sin\omega + \cos\Omega\cos\omega\cos i \\ n_1 &= \sin\omega\sin i \\ n_2 &= \cos\Omega\sin i \end{aligned} \tag{8.16}$$

Equation (8.17) then provides the expressions for x, y and z.

$$\begin{aligned} x &= r \cdot [\cos\Omega\cos(\omega+\theta) - \sin\Omega\sin(\omega+\theta)\cos i] \\ y &= r \cdot [\sin\Omega\cos(\omega+\theta) - \cos\Omega\sin(\omega+\theta)\cos i] \\ z &= r\sin i\sin(\omega+\theta) \end{aligned} \tag{8.17}$$

For the velocity values, an extra parameter is required. This parameter is called the specific relative angular momentum h. Assuming that the mass of the orbiting body can be neglected with respect to the planet (or other celestial body) it is orbiting, h can be expressed as a function of the standard gravitational parameter of that planet μ, a and e as shown by Equation (8.18).

$$h = \sqrt{\mu a \cdot (1-e^2)} \tag{8.18}$$

Then with h and the expression from Equation (8.16), the velocities can be expressed by Equation (8.19).

$$\begin{aligned}
\dot{x} &= \frac{\mu}{h}[-l_1 \sin\theta + l_2 \cdot (e+\cos\theta)] \\
\dot{y} &= \frac{\mu}{h}[-m_1 \sin\theta + m_2 \cdot (e+\cos\theta)] \\
\dot{z} &= \frac{\mu}{h}[-n_1 \sin\theta + n_2 \cdot (e+\cos\theta)]
\end{aligned} \tag{8.19}$$

It is also possible to convert from the Cartesian system back to the Kepler system, however, this requires more steps and intermediate expressions. First, it is convenient to express the position and velocity both in one vector each as shown by Equation (8.20).

$$\begin{aligned}
\mathbf{r} &= \begin{bmatrix} x & y & z \end{bmatrix} \\
\mathbf{V} &= \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}
\end{aligned} \tag{8.20}$$

The cross product of these two ($\mathbf{r} \times \mathbf{V}$) will result in the specific relative angular momentum vector $\mathbf{h}$ and the scalar values, or lengths, can then be represented by $r = \|\mathbf{r}\|$, $V = \|\mathbf{V}\|$ and $h = \|\mathbf{h}\|$. Also, a vector $\mathbf{N}$ can be defined as shown by Equation (8.21)[118].

$$\mathbf{N} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \mathbf{h} \tag{8.21}$$

With these parameters set, the first four Kepler elements (a, e, i and Ω) can be computed as shown by Equation (8.22). Here, the value for the eccentricity can be obtained by taking the length of the vector: $e = \|\mathbf{e}\|$. Also, $\mathbf{h}(3)$ is the third value of the specific relative angular momentum vector, so the value in the z-direction. The same principle holds for the values from $\mathbf{N}$.

$$\begin{aligned}
a &= \left(\frac{2}{r} - \frac{V^2}{\mu}\right)^{-1} \\
\mathbf{e} &= \frac{\mathbf{V} \times \mathbf{h}}{\mu} - \frac{\mathbf{r}}{r} \\
i &= \arccos\left(\frac{\mathbf{h}(3)}{h}\right) \\
\Omega &= \operatorname{atan2}\left(\mathbf{N}(2), \mathbf{N}(1)\right)
\end{aligned} \tag{8.22}$$

For the computation of the values for ω and θ, three unit vectors are needed: $\hat{\mathbf{N}} = \frac{\mathbf{N}}{\|\mathbf{N}\|}$, $\hat{\mathbf{e}} = \frac{\mathbf{e}}{\|\mathbf{e}\|}$ and $\hat{\mathbf{r}} = \frac{\mathbf{r}}{\|\mathbf{r}\|}$. This leads to the conditional expressions seen in Equation (8.23).

$$\begin{aligned}
\omega &= \begin{cases} \arccos\left(\hat{\mathbf{e}} \cdot \hat{\mathbf{N}}\right), & \text{if } \left(\hat{\mathbf{N}} \times \mathbf{e}\right) \cdot \mathbf{h} > 0 \\ -\arccos\left(\hat{\mathbf{e}} \cdot \hat{\mathbf{N}}\right), & \text{otherwise} \end{cases} \\
\theta &= \begin{cases} \arccos\left(\hat{\mathbf{r}} \cdot \hat{\mathbf{e}}\right), & \text{if } \left(\mathbf{e} \times \mathbf{r}\right) \cdot \mathbf{h} > 0 \\ -\arccos\left(\hat{\mathbf{r}} \cdot \hat{\mathbf{e}}\right), & \text{otherwise} \end{cases}
\end{aligned} \tag{8.23}$$

The eccentric anomaly can now be computed using θ and e as shown by Equation (8.24), which is the rewritten form of Equation (8.13).

$$E = 2 \cdot \arctan\left[\tan\left(\frac{\theta}{2}\right)\sqrt{\frac{1-e}{1+e}}\right] \tag{8.24}$$

And finally, the mean anomaly can be found using Equation (8.12)[118].

8.3.3. SPHERICAL AND KEPLERIAN

Often it is required to transform the spherical coordinates into Kepler elements. Wakker [109] provides this direct transformation through the relations provided in Equation (8.25). In this case the expressions are written explicitly for Mars by using μ_M.

$$
\begin{aligned}
a &= \frac{r}{2 - \frac{rV^2}{\mu_M}} \\
e &= \sqrt{1 - \frac{rV^2}{\mu_M}\left(2 - \frac{rV^2}{\mu_M}\right)\cos^2\gamma} \\
i &= \arccos\left(\cos\delta \sin\chi\right) \\
\theta &= 2\arctan\left[\sqrt{\frac{1+e}{1-e}}\tan\left(\frac{E}{2}\right)\right], \qquad \text{where} \qquad E = \arctan\left[\sqrt{\frac{a}{\mu_M}}\frac{rV\sin\gamma}{a-r}\right] \\
\omega &= \frac{\sin\delta}{\sin(i)} - \theta \\
\Omega &= \lambda - \arccos\left[\frac{\cos\chi}{\sin(i)}\right]
\end{aligned}
\tag{8.25}
$$

Then the spherical coordinates can be computed by rewriting the expressions provided in Equation (8.25) to the spherical coordinates. These rewritten expressions are shown in Equation (8.26).

$$
\begin{aligned}
r &= a\left(1 - e\cos(E)\right), \qquad \text{where} \qquad E = 2\arctan\left[\sqrt{\frac{1-e}{1+e}}\tan\left(\frac{\theta}{2}\right)\right] \\
V &= \sqrt{\mu_M\left(\frac{2}{r} - \frac{1}{a}\right)} \\
\gamma &= \arcsin\left[e\sin(E)\frac{\sqrt{\mu_M a}}{rV}\right] \\
\delta &= \arcsin\left[\sin(\omega+\theta)\sin(i)\right] \\
\lambda &= \arcsin\left[\frac{\tan\delta}{\tan(i)}\right] - \Omega \\
\chi &= \arccos\left[\cos(\omega+\theta) - \frac{\sin(i)}{\cos\delta}\right]
\end{aligned}
\tag{8.26}
$$

8.3.4. NON-SINGULAR KEPLER ELEMENTS

As mentioned in Chapter 7, during the calculation of the change in orbital elements some of the Kepler elements cannot assume every possible value. Singularities occur when $e = 0$, and/or $i = 0°$ or $180°$. In Section 7.3 it was mentioned that MEEs are chosen to avoid these singularities. The transformation from the ordinary Kepler elements to the MEEs was already provided in Equation (7.2) and is repeated here once more as shown in Equation (8.27). This equation also includes the expressions to revert back to normal Kepler elements (see left equations). Here I is the retrograde factor and has a value of +1 for posigrade orbits and -1 for retrograde orbits. This is done to avoid singularities when $i = 180°$ and thus making the set non-singular.

$$
\begin{aligned}
p &= a\left(1 - e^2\right) & a &= \frac{p}{1 - \left(f^2 + g^2\right)} \\
f &= e\cos(\omega + I\Omega) & e &= \sqrt{f^2 + g^2} \\
g &= e\sin(\omega + I\Omega) & i &= 2\tan^{-I}\left[\sqrt{h^2 + k^2}\right] \\
h &= \tan^{I}\left(\frac{i}{2}\right)\cos(\Omega) & \omega &= L - \theta - I\Omega \\
k &= \tan^{I}\left(\frac{i}{2}\right)\sin(\Omega) & \Omega &= \operatorname{atan2}(k, h) \\
L &= \omega + \theta + I\Omega & \theta &= L - \arccos\left[\frac{f}{e}\right]
\end{aligned}
\tag{8.27}
$$

9 Mars atmospheric model

During the ascent the MAV will pass through the Martian atmosphere, which will result in a drag force opposite to the velocity direction. This drag force $D = f(\rho, V, a)$ where ρ is the air density, V the vehicle velocity and a is the speed of sound ($D = f(\rho, V, C_D)$, $C_D = f(M, \alpha)$ and $M = f(a, V)$). The speed of sound can be written as Equation (9.1) for an ideal gas, where γ_a is the adiabatic index (or isentropic expansion factor) of the Martian atmosphere, R_a is the molar gas constant, T_a is the absolute temperature and M_a is the molar mass of the Martian atmosphere.

$$a = \sqrt{\gamma_a R_a T_a} \qquad \text{where} \qquad R_a = \frac{R_a^*}{M_a} \tag{9.1}$$

The temperature depends on the altitude and should be provided by the atmospheric model. Both γ_a and M_a depend on the composition of the atmosphere. According to [107] the atmosphere has a volumetric composition of 95.32% Carbon dioxide, 2.7% Nitrogen, 1.6% Argon, 0.13% Oxygen and 0.08% Carbon monoxide. This results in a mean molecular mass of 0.04334 kg/mol for the Martian atmosphere. Also, in [133] the adiabatic index γ_a is assumed to have a constant value of ~1.35 for Mars.

Now the only two variables that are required from the atmospheric model are the density ρ and the absolute temperature T_a at a certain point in the atmosphere. Another method would be to write the speed of sound as a function of the air density and the pressure p_a (instead of absolute temperature). In this case the expression for a is provided by Equation (9.2).

$$a = \sqrt{\frac{\gamma_a p_a}{\rho}} \tag{9.2}$$

It would be an asset if the speed of sound can be provided directly, because this alleviates the uncertainty caused by the adiabatic index γ_a. In this chapter the atmospheric models that were used in reference research will be described in Section 9.1 and the preferred atmospheric model will be determined in Section 9.2. Finally, Section 9.3 will describe different interpolation methods.

9.1. Reference atmospheric models

During the simulation of the MAV launch, the reference research mentioned in Chapter 3 used different models and assumptions. It is useful to understand which atmospheric model was used and why. It is also important to determine why a certain model would be appropriate to use in this thesis problem. The reference research and their models are provided in Table 9.1. It is interesting to see that either an exponential density model was used, or a model called Mars-GRAM. Another model (not mentioned before) is the EMCD model, which is a European Mars atmospheric model. However, because this model is not available at JPL this model will not be discussed.

9.1.1. Exponential atmosphere

The exponential density model is based on the assumption that the density decreases exponentially with increasing altitude. Also, in an exponential atmosphere model, the temperature is constant (which means

Table 9.1: Mars atmospheric models used in reference research.

Author	Year	Atmospheric model used
Fanning and Pierson [2]	1996	Exponential density model (scale height 11.17 km) and constant C_D (≈0.88)
Desai et al. [3]	1998	Mars-GRAM
Whitehead [4, 5]	2004	Exponential fit to Viking density data (scale height 8.3 km) and constant a (250 m/s)
Trinidad et al. [7]	2012	2010 Mars-GRAM

that the speed of sound is constant, see Equation (9.1)). For a given altitude h the local density can be directly computed using Equation (9.3) [2]. Here, ρ_0 is the atmospheric density at zero elevation and H_s is the scale height.

$$\rho = \rho_0 exp\left(-\frac{h}{H_s}\right) \tag{9.3}$$

Similarly, the pressure can be assumed to decrease exponentially with increased altitude as well, as shown by Equation (9.4). Here p_0 is the atmospheric pressure at zero elevation.

$$p = p_0 exp\left(-\frac{h}{H_s}\right) \tag{9.4}$$

In Equations (9.1) to (9.4) the same H_s is used, which means that a constant temperature is assumed and also results in a constant speed of sound ([107] mentions a scale height of 11.1 km). However, a range of temperatures can be taken into account by using different scale heights for both density and pressure. The scale height can also be changed to simulate different seasons. For each season a different scale height can be selected to simulate the change in atmospheric conditions. Using this method, both pressure and density can be evaluated directly for every point in the trajectory, however some accuracy is lost due to all the assumptions.

9.1.2. MARS-GRAM

Mars-GRAM is a more sophisticated model that can be used for high-fidelity simulations [134][1]. The model is based on NASA Ames Mars General Circulation Model (for altitudes between 0-80 km) and Mars Thermospheric General Circulation model (for altitude above 80 km). It can provide density, temperature and pressure data (among other data) [135]. It can be set for different atmospheric conditions and also different seasons. In [135] the 2005 version of the model was validated against Thermal Emission Spectrometer data. The results for both density and temperature are provided and show that Mars-GRAM is within ± 10% difference from the spectrometer data for temperature data between an altitude of 0-60 km and for density between an altitude of 0-40 km. After 40 km the density differences increase to a maximum difference of 35% at an altitude of 60 km. This is still very accurate. However, since it is such a detailed model, each evaluation requires considerable cpu time and also computes unnecessary data, or overhead. One option to reduce the cpu time would be to only evaluate Mars-GRAM once at the beginning of a new optimisation, from now on referred to as the limited Mars-GRAM option. In that case a detailed table will be generated at the beginning of the simulation, and during the simulation the different required values will be acquired through interpolation (more information on interpolation methods will be provided in Section 9.3). Another option would be to use a hybrid form of both the exponential atmosphere model and Mars-GRAM. Mars-GRAM could for instance be used to determine the scale height and then the exponential model could determine the density and pressure at each path point.

9.2. CHOSEN ATMOSPHERIC MODEL

It was already mentioned that the two main output requirements are either a and ρ, or T_a and ρ, or p_a and ρ. Two other requirements are the speed at which the model can be evaluated for the current path point and the

[1] NASA website: https://see.msfc.nasa.gov/model-Marsgram [Accessed 14 December 2015]

accuracy at which this is done. Finally, the model should be available to use in the thesis problem. With these requirements, a trade-off table was created to compare both models and determine the most appropriate one, see Table 9.2. Every requirement is assigned a weight depending on how important the requirement is. Then the different models are either green (1), yellow (0.5) or red (0) for every requirement. This results in a final score. It can be seen that the limited Mars-GRAM model fits best with the set requirements and is therefore the preferred model to be used in the thesis problem.

Table 9.2: Atmospheric model trade-off table.

	ρ	p_a	T_a	a	**Evaluation speed**	**Accuracy**	**Availability**	**Score**
Weight	*5*	*2*	*4*	*1*	*3*	*4*	*5*	
Exponential	🟩	🟩	🟥	🟥	🟩	🟥	🟩	15
Mars-GRAM	🟩	🟩	🟩	🟥	🟥	🟩	🟩	20
Limited Mars-GRAM	🟩	🟩	🟩	🟥	🟩	🟩	🟩	23
Hybrid	🟩	🟩	🟥	🟥	🟩	🟨	🟩	17

9.3. INTERPOLATION METHODS

In Section 9.2 it was determined that a limited Mars-GRAM option will best suit this thesis problem. This model involves the generation of an atmospheric table with altitude, temperature, pressure and density (and can also include latitude and longitude data) at the start of the simulation. It could however be that during the simulation the information corresponding to a certain altitude is not directly available from the table. In that case the value will have to be interpolated, and because the pressure, temperature and density all correlate to a certain altitude value this interpolation is 1-dimensional. This does assume that the atmosphere is constant on every altitude regardless of the latitude and longitude. Including the longitude and latitude as well will change the problem to a 3-dimensional interpolation problem, which requires multivariate interpolation [136]. There are many different interpolation methods, however in this case the generated table can be very precise, which again means that the interpolation method does not require to be very accurate. One of the simplest methods is the linear interpolation. This method has been used before in the determination of atmospheric parameter values [137] and can even be used in a 3-dimensional interpolation problem [136]. Using a simple integration method during the simulation will result in a lower cpu time compared to more precise or more complex methods. Another reason to use linear interpolation is that often different layers of the atmosphere display linear behaviour in the change of temperature, pressure and density values. Therefore, at this point (multivariate) linear interpolation is chosen.

Linear interpolation is based on the idea that there exists a linear relation between two (or more) data points. The equation for a 1-dimensional linear interpolation can be expressed as shown by Equation (9.5). In this example the altitude h_t is the current altitude and is known. This altitude lies between altitude h_1 and h_2 with their corresponding temperature values T_1 and T_2. T_t is the desired temperature value at the current time and altitude.

$$T_t = T_1 + \left(\frac{h_t - h_1}{h_2 - h_1}\right)(T_2 - T_1) \tag{9.5}$$

This process can be repeated three times in case of the 3-dimensional problem.

10

Proposed software

Time can be saved by using already written and validated software. Because of the available software from both TU Delft and JPL and the evaluation speed it was decide to use C++ as the main programming language. Many of the software packages that are used at JPL fall under the ITAR. This means that the software cannot be used directly by foreign nationals, however some of the tools that are used in those software packages are commercially available. The JPL software can nonetheless still be used as a verification tool for the software that will be written during the course of this thesis project. In that case, a JPL employee will simulate the trajectory with the same conditions as the written software and the results can be compared to each other. For the optimisation, there is software available that can be integrated into the newly written software. In Section 10.1 the different existing software packages that will be used are described and any changes that will have to be made to the software. Section 10.2 describes the different aspects of the simulation program that will have to be developed during the thesis work.

10.1. Existing software packages

As was discussed in Section 4.6.2 there is already software available for the local optimisation of the MBH method. As a matter of fact, MBH itself is already an option within the PaGMO package [62, 138]. However, should PaGMO not be available or should MBH have to be modified, it is still usefull to have a seperate version of the local optimisation software available. During the ascent, Mars-GRAM 2005 (originally developed in the '90s) will be used to simulate the Martian atmospheric conditions. An overview of the software packages is provided in Table 10.1.

Table 10.1: Required existing software

Subject	Software	Developer	Year	C++ source code/-compatible	Provided by
Optimisation	PaGMO	ESA Advance Concept Team [138]	2012	Yes	TU Delft [1]
Optimisation	SNOPT	Gill et al. [81]	2002	Yes	TU Delft and JPL [2]
Ascent	Mars-GRAM 2005	Justus [139]	2005	Yes	TU Delft and JPL

Both PaGMO and SNOPT are available through Tudat which can be provided by the TU Delft. However, at this point it is assumed that the simulation program software will mostly be based on JPL software.

In Chapter 4 it was already mentioned that SNOPT will be used as the local optimisation tool for MBH. This means that if the pre-set PaGMO version of MBH is used, SNOPT will have to be incorporated into it. However, should PaGMO not be available after all, then SNOPT will simply be integrated into the self developed MBH software. It is not expected that the SNOPT software itself will require any adjustments. Also,

[1] For more information on PaGMO please see the ESA website: http://esa.github.io/pagmo/ [Accessed 4 February 2016]

[2] SNOPT is also available throughPaGMO [138]

Mars-GRAM 2005 will solely be used for its outputs, which means that its can be run separately from the optimiser and also will not need to be changed. It could be that more changes are required when actually using the software, but these are the (possible) required changes envisioned at this time.

10.2. DEVELOPMENT OF NEW SOFTWARE

The final thesis topic is similar to the Earth ascent problem studied by Pagano [103]. Unfortunately, it is not similar enough, which means that new software will have to be developed for this thesis problem. All the tools that will have to be developed will be written in C++ and will have to be combined in the end, together with the existing software, to form the final simulation software. There are several aspects of the simulation program that will have to be written. These include: ascent dynamic equations, linear interpolation, the different integration methods, and might also include MBH. Preferably these will all be written as separate tools and such that they are flexible enough to also be applied to other problems. The top-level architecture of the program as it is currently envisioned can now be set up, and is visualised in Figure 10.1.

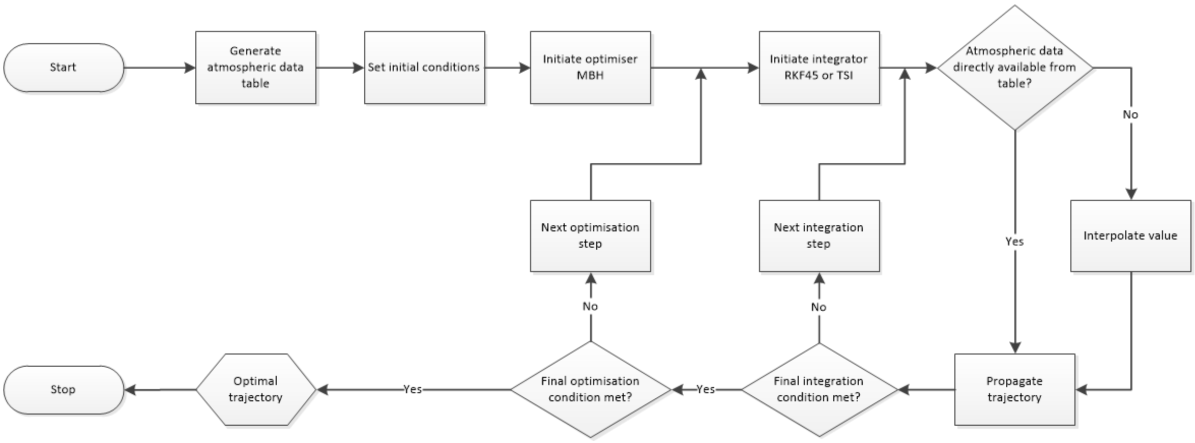

Start
Generate atmospheric data table
Set initial conditions
Initiate optimiser MBH
Initiate integrator RKF45 or TSI
Atmospheric data directly available from table?
No
Interpolate value
Yes
Next optimisation step
Next integration step
No
No
Stop
Optimal trajectory
Yes
Final optimisation condition met?
Yes
Final integration condition met?
Propagate trajectory

Figure 10.1: Proposed general simulation software architecture.

10.3. VERIFICATION AND VALIDATION

Each of the written software tools will have to be verified and validated. The verification of a piece of software assures that the software meets the set requirements for the software. If a piece of software is validated, it does what it is suppose to do and the results are correct. For example, it is required of the interpolation tool that it provides a value for T, for example, provided a certain h that lies in between the known values. So if the tool is run and it indeed provides a value for T in between the known values, then the tool is verified. However, this does not mean that the value for T is (or comes close to) the correct value (missing minus sign for instance). Therefore, the results have to be compared to a source from which it is known that the values are correct. If the same values are obtained, the tool is validated. Also, each time software is added to the program, the whole program has to be verified and validated again. In this case the interpolation, trajectory propagation, integrators and optimiser all have to be verified and validated. This can consist of different steps. Each of these steps is provided for each of the software tools:

- **Interpolation** *Verification:* when the interpolation tool provides a previously unknown value in between two known values after it is run, then it is verified (inspection). *Validation:* two different tables can be generated using Mars-GRAM 2005, each with slightly different data points. The interpolation tool will then use the data points of the first table to determine the parameter values of the unknown values corresponding to the h, for instance, of the data points of the second table. If the results correspond to the parameter values provided in the second table, the interpolation tool is validated (testing). This can be done for different data point resolutions to determine the minimum resolution required for the table.

- **Integrator** *Verification:* to test the integrator by itself, a simple function (with a known outcome) is required. If the integration tool produces results it is verified (inspection). *Validation:* if the results match the known function values then the integrator is also validated. This can be done for increasingly more difficult functions and problems. Examples of validation functions are provided in [84] (testing).

- **Trajectory propagation** *Verification:* once the integrators have been validated, the trajectory propagation can be incorporated, and it can be checked if it produces proper results (instpection). *Validation:* initially, Apollo ascent flight data [10] can be used to validate the program for non-atmospheric and Lunar conditions (testing). Then the program shall be validated for Mars ascent using data provided by Woolley [28] for the corresponding conditions (testing). A second Mars validation can be performed by setting specific conditions and simulate the trajectory using other JPL software. These results can then be used to validate the thesis software (testing).

- **Optimiser** *Verification:* for an optimiser it is important that it produces a result within a reasonable time. Therefore if the optimiser converges it is verified (inspection). This will initially be done for a benchmark function as provided by [39]. Once integrated into the simulation program it itself will have to be verified. Again, the desired outcome is a convergence within a reasonable time. *Validation:* for the initial validation the results will have to match the results of the known function as provided by [39] (testing). Then the final validation of the complete program will be achieved by creating the same results as presented in [28], but this time the optimal condition will not be set. Instead the simulation program will have to determine its own optimum conditions, which will then have to correspond to the optimum trajectory results provided in [28]. A further validation of the entire system will then take place using JPL optimisation software. In this case an initial condition is set in both the thesis software and the JPL software and the optimum result of both programs will be compared to determine the validity of the results obtained (testing).

For each piece of software, a walk-through will be performed to verify that each part of the tool is working properly before attempting the final verification. Also, additional validation of the software can be performed using the reference research discussed in Chapter 3.

11

Final thesis topic proposal

The final subject definition is based on the work performed in this literature study and is an adaptation of the the original general subject introduced in Chapter 2. Not all the work presented in this literature study can be performed in the provided 7 months, which is why a trade-off was necessary. This topic trade-off is presented in Section 11.1. The corresponding research objective is described in more detail in Section 11.2 followed by the research perspective in Section 11.3. The research questions associated with the chosen thesis topic are presented in Section 11.4 and finally Section 11.5 will describe the research strategy.

11.1. Final refined thesis topic

In this section the final proposal is chosen. There were several aspects of the original proposal which will have to be traded. First of all, either a high-thrust ascent trajectory optimisation or a low-thrust trajectory optimisation has to be selected. This has to be done because the work associated with combining these two aspects cannot be completed within the required 7 months. The second trade-off is whether to focus on optimisation or integration comparison.

One of the most important criterion here is academic value: how (and how much) will this research contribute to the scientific community? This criterion is closely related to the second criterion, novelty: has this ever been done before? Novelty also includes the amount of previous master theses that have been written on the subject. The third criterion is challenge: is the topic challenging or is it too easy? It is also critical that the work can actually be performed within the 7 months, which is why the fourth criterion is called allocated time. And finally it is important that the work is related to the current work done at JPL and is useful to them. Therefore this last criterion is called JPL value. Every criterion is assigned a weight depending on how important the criterion is. Then the different topics are either green (1), yellow (0.5) or red (0) for every criterion. This results in a final score. The trade-off is provided in Table 11.1.

Table 11.1: Optimiser trade-off table.

	Academic value	**Novelty**	**Challenge**	**Allocated time**	**JPL value**	**Score**
Weight	*5*	*4*	*2*	*4*	*3*	
Ascent and optimisation focus	🟨	🟩	🟨	🟩	🟨	13
Ascent and integration focus	🟩	🟩	🟩	🟩	🟨	18
Orbiter and optimisation focus	🟩	🟥	🟨	🟨	🟩	11
Orbiter and integration focus	🟩	🟨	🟩	🟥	🟩	12

The result of the trade-off shows that a Martian ascent optimisation with the focus on integration method comparison is the most logical choice. This means that the optimisation method from Chapter 4 with the

most potential will be used, which is MBH. Furthermore, the two integration methods that will be compared are RKF45 and TSI.
With the topic set, the final working title can now be described as: ***"Mars Sample Return MAV ascent trajectory optimisation using Taylor Series integration"***.

11.2. Research objective

The objective of this research is to find the optimum solution to the high-thrust MAV launch trajectory problem by using MBH for the optimisation for lowest GLOM and integrating the trajectory using both RKF45 and TSI and comparing these two integration methods.

11.3. Research perspective

It will be both theory-testing research (MBH and TSI compared with RKF45) and problem-analysis research (optimum problem). The problem will be solved through the writing and verification of an optimisation program based on Newtonian mechanics and from a(n) (initial) mission design standpoint. It is therefore an engineering problem, and was finalised through discussions between researcher, supervisors, stakeholders (JPL) and using the work performed in this literature study.

11.4. Research questions

The main research question is: how does TSI perform compared to RKF45 in a Mars ascent optimisation problem? The corresponding sub-questions are split up into primary and secondary sub-questions. The primary sub-questions have to be answered during the thesis work, the secondary sub-questions can be answered provided there is enough time. The primary sub-questions are listed here:

- Would TSI be a good alternative for use in future ascent optimisation programs?
  - How much more (or less) accurate is TSI compared to the RKF45?
  - How much faster (or slower) is TSI compared to RKF45?
  - What are the advantages and disadvantages (besides the first two sub-questions) when using TSI in this kind of trajectory optimisation problem?
- What is the optimal MAV configuration and combination of launch parameters to reach the desired target orbit, based on the design choices, that results in the lowest MAV GLOM?
- What is the performance of MBH in an ascent problem?
  - How well do the solutions of MBH converge in an ascent problem?
  - What is the cpu time required by MBH in an ascent problem?

The secondary sub-questions are listed here:

- Which of the current proposed Mars 2020 landing sites would be best considering the MSR mission.
  - Which proposed landing sites provide the best optimum solution for the thesis problem?
  - Which proposed landing sites provide the worst optimum solution for the thesis problem?

11.5. Research strategy

The first step will be to design the detailed simulation and optimisation program architecture. Then the Mars ascent simulation program will be written and validated. First for the Moon using Apollo reference data reaching a specific orbit, and then the same is done for Mars using the same conditions in JPL software. At that time, both integrators are included and validated using reference research data and JPL software. After this the optimiser is included as well and is then validated separately from the simulator as well. More information on the verification and validation of the software will be provided in Chapter 10. During the entire process, everything will be documented. Once the simulation program has been completely validated, the initial conditions are set and the optimisation process is started. For the candidate landing site, and each of the integrators, the lowest GLOM condition can be found. The integrators will then be compared based on their individual results.

12

Proposed time schedule

The proposed time schedule is based on the research strategy described in Section 11.5 and the verification and validation process described in **??**. The total time that has to be spent on the thesis project is 1176 hours, or approximately 30 weeks. This schedule does not include any specific dates yet but describes the work that will have to be done per week up to the 30th week of the thesis project. It is assumed that a day has an effective work time of 8 hours. The schedule is presented in Table 12.1. The workload is split up into different work-packages or WP.

Table 12.1: Proposed time schedule thesis project

Task	**Task description**	**Days required**	**Week number**
WP 1	**Goal:** Acquire all the required software and get familiar with them **Input:** List of required software packages **Tasks:** Acquire the required software, install the software onto the computer, familiarise myself with the software **Output:** Knowledge on the different software packages	2,2,2	1-2
WP 2	**Goal:** Learn C++ **Input:** Current programming experience **Tasks:** Familiarise myself with the programming language, the different possibilities, the different functions and write some simple scripts **Output:** Knowledge on C++ and C++ programming skills	3,2,2,1	1-3
WP 3	**Goal:** Design the detailed simulation and optimisation program architectures **Input:** Knowledge on the different software packages, theoretical knowledge of the to be developed tools, and knowledge on C++ **Tasks:** Design the detailed architecture of each of the software tools that have to be developed **Output:** Detailed architectures of the to be developed software tools	8	3-5
WP 4	**Goal:** Write the interpolation tool **Input:** Knowledge on C++, C++ programming skills, theoretical knowledge on the interpolation method, knowledge on Mars-GRAM and the detailed architecture of the interpolation tool **Tasks:** Write the interpolation tool, verify it and validate using Mars-GRAM data, document everything **Output:** Interpolation tool and documentation on the tool	1,0.5,0.5,1	5

WP 5	**Goal:** Write the RK4 integration tool **Input:** Knowledge on C++, C++ programming skills, theoretical knowledge on the RK4 method and the detailed architecture of the RK4 integration tool **Tasks:** Write the RK4 integration tool, verify it and validate, document everything **Output:** RK4 integration tool and documentation on the tool	1,0.5,0.5,1	6
WP 6	**Goal:** Write the RKF45 integration tool **Input:** Knowledge on C++, C++ programming skills, theoretical knowledge on the RKF45 method and the detailed architecture of the RKF45 integration tool **Tasks:** Write the RKF45 integration tool, verify it and validate, document everything **Output:** RKF45 integration tool and documentation on the tool	3,1,1,2	6-7
WP 7	**Goal:** Write the TSI integration tool **Input:** Knowledge on C++, C++ programming skills, theoretical knowledge on the TSI method and the detailed architecture of the TSI integration tool **Tasks:** Write the TSI integration tool, verify it and validate, document everything **Output:** TSI integration tool and documentation on the tool	3,1,1,2	8-9
WP 8	**Goal:** Include trajectory propagation **Input:** Knowledge on C++, C++ programming skills, theoretical knowledge on the ascent equations, the RK4 integration tool, the interpolation tool, Mars-GRAM and the detailed architecture of the trajectory propagation tool **Tasks:** Write the trajectory propagation tool, verify it and document everything **Output:** Trajectory tool, verified ascent simulation program, and documentation on the trajectory and ascent simulation program	3,2,2	9-10
WP 9	**Goal:** Validation of the ascent simulation program**Input:** Knowledge on C++, C++ programming skills, Apollo flight data, Mars ascent simulation reference data, verified ascent simulation program **Tasks:** Validate for the Moon, Validate for Mars and document everything **Output:** Validated ascent simulation program and documentation on the ascent program	5,7,3	10-13
WP 10	**Goal:** Write the optimisation tool **Input:** Knowledge on C++, C++ programming skills, theoretical knowledge on the optimisation method, and the detailed architecture of the optimisation tool **Tasks:** Write the optimisation tool, verify it and validate, document everything **Output:** Optimisation tool and documentation on the tool	2,1,2,2	13-15

WP 11	**Goal:** Finish the complete ascent simulation and optimisation program **Input:** Knowledge on C++, C++ programming skills, Mars ascent simulation reference data, validated ascent simulation program and the detailed architecture of the optimisation and simulation program **Tasks:** Integrate optimisation tool into the ascent program, verify it, validate and document everything **Output:** Validated ascent simulation and optimisation program and documentation on the ascent program	2,2,10,3	15-18
WP 12	**Goal:** Obtain optimised RKF45 ascent trajectory **Input:** Knowledge on C++, C++ programming skills, MAV baseline data, initial conditions, target orbit and validated ascent simulation and optimisation program **Tasks:** Optimise the ascent trajectory using RKF45 and document results **Output:** Optimised RKF45 ascent trajectory and documented results	3,2	18-19
WP 13	**Goal:** Obtain optimised TSI ascent trajectory **Input:** Knowledge on C++, C++ programming skills, MAV baseline data, initial conditions, target orbit and validated ascent simulation and optimisation program **Tasks:** Optimise the ascent trajectory using TSI and document results **Output:** Optimised TSI ascent trajectory and documented results	3,2	19-20
WP 14	**Goal:** Analysis and comparison of results **Input:** Optimised RKF45 ascent trajectory, optimised TSI ascent trajectory **Tasks:** Analyse the different trajectories, compare the performance of both integrators and documentation on the results **Output:** Analysis and comparison of results and documentation on the results	2,2,2	20-21
WP 15	**Goal:** Finish complete thesis report **Input:** All documentation of the previous work-packages and the results of the analysis **Tasks:** Write the thesis report, draw conclusions and write recommendations **Output:** Finished draft thesis report	15,3,2	21-25
Delay buffer	Have a few days as a buffer in case any delays occur	22	25-30
Draft thesis hand-in	Hand-in of the draft thesis report	1	30
WP 16	**Goal:** Finish final version of thesis report **Input:** Feedback and draft thesis report **Tasks:** Implement all feedback into the draft thesis report **Output:** Finished final report	10	32-33
Final thesis hand-in	Hand-in of the final thesis report	1	34
Thesis defence	**Goal:** Graduate **Input:** Final thesis report and experience **Tasks:** Defend my thesis **Output:** MSc title	1	34

Bibliography

[1] Mars Program Planning Group, *Summary of the final report,* (2012), [online database], URL: http://www.nasa.gov//sites/default/files/files/ [cited 14 October 2015].

[2] J. Fanning and B. Pierson, *A model comparison for optimal mars ascent trajectories,* in *Engineering Optimization+ A35*, Vol. 26 (1996) pp. 271–285.

[3] P. Desai, R. Braun, W. Engelund, F. Cheatwood, and J. Kangas, *Mars ascent vehicle flight analysis,* in *7th AIAA/ASME Joint Thermophysics and Heat Transfer Conference* (1998).

[4] J. C. Whitehead, *Mars ascent propulsion trades with trajectory analysis,* in *40th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit* (American Institute of Aeronautics and Astronautics, 2004).

[5] J. C. Whitehead, *Trajectory analysis and staging trades for smaller mars ascent vehicles,* in *Journal of spacecraft and rockets*, Vol. 42 (2005) pp. 1039–1046.

[6] E. Di Sotto, J. C. Bastante, and R. Drai, *System and gnc concept for rendezvous into elliptical orbit for mars sample return mission,* in *AIAA Guidance, Navigation, and Control Conference and Exhibit, Colorado* (2007).

[7] M. Trinidad, E. Zabrensky, and A. Sengupta, *Mars ascent vehicle system studies and baseline conceptual design,* in *Aerospace Conference, 2012 IEEE* (IEEE, 2012) pp. 1–13.

[8] E. Dumont, *Design of a modular transportation system for future lunar robotic missions,* in *30th ISTS/34th IEPC/6th NSAT Joint Conference, Kobe, Japan* (2015).

[9] B. Harvey, *Soviet and Russian Lunar Exploration*, 1st ed., edited by J. Mason (Springer and Praxis Publishing, Chichester, UK, 2007).

[10] Mission Evaluation Team, *Apollo 11 mission report,* (1971), [online database], URL: http://history.nasa.gov/alsj/a11/ [cited 15 October 2015].

[11] R. R. Sostaric and R. S. Merriam, *Lunar ascent and rendezvous trajectory design,* in *31st Annual AAS Guidance and Control Conference, Breckenridge, Colorado* (2008).

[12] A. Dietrich, K. Davis, and J. Parker, *Ascent trajectories from the lunar far-side to earth-moon l2 halo orbits,* in *Advances in Space Research*, Vol. 56, University of Colorado (Elsevier, 2015) pp. 2595–2611.

[13] L. Alkalai, B. Solish, J. Elliott, T. McElrath, J. Mueller, and J. Parker, *Orion/moonrise: A proposed human and robotic sample return mission from the lunar south pole-aitken basin,* in *Aerospace Conference, 2013 IEEE* (IEEE, 2013) pp. 1–10.

[14] M. Martinez-Sanchez and J. E. Pollard, *Spacecraft electric propulsion-an overview,* in *Journal of Propulsion and Power*, Vol. 14 (1998) pp. 688–699.

[15] S. Schaeff and S. Erb, *Low thrust trajectory optimization for orbit raising applications,* Presentation, Second U.K. Workshop on Optimisation in Space Engineering (2014), [online database], URL: http://www.turing-gateway.cam.ac.uk/documents/ [cited 18 October 2015].

[16] S. Geffroy and R. Epenoy, *Optimal low-thrust transfers with constraints—generalization of averaging techniques,* in *Acta Astronautica*, Vol. 41 (1997) pp. 133–149.

[17] C. A. Kluever and S. R. Oleson, *Direct approach for computing near-optimal low-thrust earth-orbit transfers,* in *Journal of Spacecraft and Rockets*, Vol. 35 (1998) pp. 509–515.

[18] R. Bertrand, J. Bernussou, S. Geffroy, and R. Epenoy, *Electric transfer optimization for mars sample return mission,* in *Acta Astronautica*, Vol. 48 (2001) pp. 651–660.

[19] G. J. Whiffen, *Mystic: Implementation of the static dynamic optimal control algorithm for high-fidelity, low-thrust tra jectory design,* in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Keystone, Colorado* (NASA Jet Propulsion Laboratory, 2006).

[20] J. A. Sims, P. A. Finlayson, E. A. Rinderle, M. A. Vavrina, and T. D. Kowalkowski, *Implementation of a low-thrust trajectory optimization algorithm for preliminary design,* in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Keystone, Colorado* (NASA Jet Propulsion Laboratory, 2006).

[21] L. D. Kos, T. Polsgrove, R. Hopkins, D. Thomas, and J. A. Sims, *Overview of the development for a suite of low-thrust trajectory analysis tools,* in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit* (NASA Jet Propulsion Laboratory, 2006).

[22] U. Derz and W. Seboldt, *Mars sample return mission architectures utilizing low thrust propulsion,* in *Acta Astronautica*, Vol. 77 (2012) pp. 83–96.

[23] E. Grayzeck, *Gemini 6a,* Website (2014), [online database], URL: http://nssdc.gsfc.nasa.gov/nmc/spacecraftDisplay.do?id=1965-104A [cited 20 October 2015].

[24] J. Mayer and R. Parten, *Development of the gemini operational rendezvous plan,* in *Journal of spacecraft and rockets*, Vol. 5 (NASA Manned Spacecraft Center, 1968) pp. 1023–1028.

[25] E. Ezell and L. Ezell, *The partnership: A history of the apollo-soyuz test project,* website (1978), [online database], URL: http://www.hq.nasa.gov/pao/History/SP-4209/ch3-8.htm [cited 20 October 2015].

[26] R. Murtazin and N. Petrov, *Usage of pre-flight data in short rendezvous mission of soyuz-tma spacecrafts,* in *Acta Astronautica*, Vol. 93 (2014) pp. 71–76.

[27] A. F. de Almeida Prado, *Optimal rendezvous maneuvers for space vehicles,* in *Revista Brasileira de Ciencias Mecanicas/Journal of the Brazilian Society of Mechanical Sciences*, Vol. 18 (1996) pp. 297–301.

[28] R. C. Woolley, R. L. Mattingly, J. E. Riedel, and E. J. Sturm, *Mars sample return - launch and detection strategies for orbital rendezvous,* in *AAS/AIAA Astrodynamics Specialist Conference*, Vol. 142 (Univelt, Inc., 2011).

[29] J. Whitehead, *Mars ascent propulsion options for small sample return vehicles,* in *33rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit* (AIAA Joint Propulsion Conference and Exhibit, 1997).

[30] C. S. Guernsey, *Mars ascent propulsion system (maps) technology program: Plans and progress,* in *34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit* (1998).

[31] C. Stone, *Rice mars,* presentation/website (1999), [online database], URL: http://www.slideshare.net/cliffordstone/rice-mars-nov99 [cited 25 October 2015].

[32] D. Stephenson, *Mars ascent vehicle - concept development,* in *38th Joint Propulsion Conference and Exhibit, Indianapolos, Indiana*, Vol. 4318 (AIAA, 2002).

[33] D. D. Stephenson and H. J. Willenberg, *Mars ascent vehicle key elements of a mars sample return mission,* in *Aerospace Conference, 2006 IEEE* (IEEE, 2006) p. 11.

[34] A. Sengupta, A. Kennett, M. Pauken, M. Trinidad, and E. Zabrensky, *Systems engineering and technology considerations of a mars ascent vehicle,* in *2011 IEEE Aerospace Conference* (2012).

[35] G. Mungas, D. Fisher, J. Vozoff, and M. Villa, *Nofbx single stage to orbit mars ascent vehicle,* in *Aerospace Conference, 2012 IEEE* (IEEE, 2012) pp. 1–11.

[36] A. J. Colozza, *Comparison of Mars Aircraft Propulsion Systems*, Tech. Rep. (NASA Glenn Research Center, 2003) : NASA/CR-2003-212350.

[37] D. J. Anderson, E. Pencil, D. Vento, J. Dankanich, M. Munk, and D. Hahne, *Propulsion technology development for sample return missions under nasas ispt program,* in *47th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit* (2011).

[38] H. Visser, *Aircraft performance optimization course ae4447,* Reader, Delft University of Technology (2014), [Internal publication].

[39] R. Noomen, *ae4-878.optimistion.v4-7,* Lecture slides, Delft University of Technology (2013), [Internal publication] Course: Mission Geometry and Orbit Design.

[40] P. Musegaas, *Optimization of Space Trajectories Including Multiple Gravity Assists and Deep Space Maneuvers,* Master's thesis, Delft University of Technology (2012).

[41] S. Boyd and L. Vandenberghe, *Convex Optimization*, 1st ed. (Cambridge University Press, New York, 2004).

[42] J. Helton and F. Davis, *Latin hypercube sampling and the progagation of uncertainty in analysis of complex systems,* in *Reliability Engineering and System Safety 81* (Elsevier, 2003) pp. 23–69.

[43] I. Sobol, *On quasi-monte carlo integrations,* in *Mathematics and Computers in Simulation*, Vol. 47 (Elsevier, 1998) pp. 103–112.

[44] P. Boggs and J. Tolle, *Sequential quadratic programming,* in *Acta Numerica*, Vol. 4 (Cambridge University Press, 1995) pp. 1–51.

[45] M. Iwamatsu and Y. Okabe, *Basin hopping with occasional jumping,* in *Chemical Physics Letters*, Vol. 339 (Elsevier, 2004) pp. 396–400.

[46] C. C. Coello, G. Lamont, and D. V. Veldhuizen, *Evolutionary Algorithms for solving multi-objective problems*, 2nd ed., edited by D. Goldberg and J. Koza (Springer Science+Business Media, New York, 2007).

[47] V. Selvi and R. Umarani, *Comparative analysis of ant colony and particle swarm optimization techniques,* in *International Journal of Computer Applications*, Vol. 5 (2010).

[48] F. Busetti, *Simulated annealing overview,* (2003), [online database], URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.66.5018 [cited 13 December 2015].

[49] L. Ingber, *Simulated Annealing: Practice versus Theory,* in *Mathematical and Computer Modelling*, Vol. 18 (Pergamon Press Ltd., 1993) pp. 20–57.

[50] B. Bhowmik, *Dynamic programming–its principles, applications, strengths, and limitations,* in *International Journal of Engineering Science and Technology*, Vol. 2 (2010) pp. 4822–4826.

[51] R. Horst and P. Pardalos, *Handbook of Global Optimization*, Vol. 1 (Kluwer Academic Publishers, Boston, 1995) available at JPL library.

[52] B. Addis, A. Cassioli, M. Locatelli, and F. Schoen, *A global optimization method for the design of space trajectories,* in *Computational Optimization and Applications*, Vol. 48 (Springer, 2011) pp. 635–652.

[53] P. Gage, R. Braun, and I. Kroo, *Interplanetary trajectory optimization using a genetic algorithm,* in *Journal of Astronautical Sciences*, Vol. 43 (1995) pp. 59–76.

[54] G. A. Rauwolf and V. L. Coverstone-Carroll, *Near-optimal low-thrust orbit transfers generated by a genetic algorithm,* in *Journal of Spacecraft and Rockets*, Vol. 33 (1996) pp. 859–862.

[55] Y. H. Kim and D. B. Spencer, *Optimal spacecraft rendezvous using genetic algorithms,* in *Journal of Spacecraft and Rockets*, Vol. 39 (2002) pp. 859–865.

[56] D. Myatt, V. M. Becerra, S. J. Nasuto, and J. Bishop, *Advanced global optimisation for mission analysis and design*, Tech. Rep. (University of Reading, 2004) ariadna id 03/4101.

[57] S. Lee, P. von Ailmen, W. Fink, A. E. Petropoulos, and R. J. Terr, *Design and optimization of low-thrust orbit transfers,* in *Aerospace Conference, 2005 IEEE* (IEEE, 2005) pp. 855–869.

[58] O. Abdelkhalik and D. Mortari, *N-impulse orbit transfer using genetic algorithms,* in *Journal of Spacecraft and Rockets*, Vol. 44 (2007) pp. 456–460.

[59] M. Vasile, E. Minisci, and M. Locatelli, *On testing global optimization algorithms for space trajectory design*, in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Honolulu, Hawaii* (2008).

[60] J. A. Garcia, J. L. Brown, D. J. Kinney, J. V. Bowles, L. C. Huynh, X. J. Jiang, E. Lau, and I. C. Dupzyk, *Co-optimization of mid lift to drag vehicle concepts for mars atmospheric entry*, in *10th AIAA/ASME Joint Thermophysics and Heat Transfer Conference, Chicago, Illinois* (2010).

[61] S. Li and Y. Peng, *Mars entry trajectory optimization using doc and dcnlp*, in *Advances in Space Research*, Vol. 47 (Elsevier, 2011) pp. 440–452.

[62] E. Boudestijn, *Development of a low-thrust earth-centrered transfer optimizer for the prelimiary mission design phase*, Master's thesis, Delft University of Technology (2014).

[63] F. Miranda, *Design Optimization of Ground and Air-Launched Hybrid Rockets*, Master's thesis, Delft University of Technology (2015).

[64] A. Pagano and E. Mooij, *Global launcher trajectory optimization for lunar base settlement*, in *AIAA/AAS Astrodynamics Specialist Conference, Toronto, Canada, 2-5 August 2010; AIAA 2010-8387* (American Institute of Aeronautics and Astronautics (AIAA), 2010).

[65] J. Vandamme, *Assisted-Launch Performance Analysis*, Master's thesis, Delft University of Technology (2012).

[66] M. van Kesteren, *Air Launch versus Ground launch: a multidisciplinary design optimization study of expendable launch vehicle on cost and performance*, Master's thesis, Delft University of Technology (2013).

[67] M. Gomez, *Optimization of Low-Thrust Trajectories in Earth-Centered Orbit*, Master's thesis, Delft University of Technology (2015).

[68] R. Storn and K. Price, *Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces*, Tech. Rep. (ICSI, USA, 1995) tR-95-012.

[69] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential evolution: a practical approach to global optimization*, edited by G. Rozenberg (Springer, Berlin, 2005).

[70] R. Storn and K. Price, *Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces*, in *Journal of global optimization*, Vol. 11 (Springer, 1997) pp. 341–359.

[71] V. Feoktistov, *Differential Evolution, In Search of Solutions*, edited by D. Du, Vol. 5 (Springer, New York, 2006).

[72] A. Qing, *Differential evolution: fundamentals and applications in electrical engineering* (John Wiley & Sons, Singapore, 2009).

[73] R. H. Leary, *Global optimization on funneling landscapes*, in *Journal of Global Optimization*, Vol. 18 (2000) pp. 367–383.

[74] D. J. Wales and J. P. Doye, *Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms*, in *The Journal of Physical Chemistry A*, Vol. 101 (1997) pp. 5111–5116.

[75] J. A. Englander and A. C. Englander, *Tuning monotonic basin hopping: Improving the efficiency of stochastic search as applied to low-thrust trajectory optimization*, in *24th International Symposium on Space Flight Dynamics, Laurel, MD* (2014).

[76] B. Addis, A. Cassioli, M. Locatelli, and F. Schoen, *Global optimization for the design of space trajectories*, DB FILE 2008/11/2150 (2008), [online database], URL: http://www.optimization-online.org/ [cited 23 October 2015].

[77] C. Yam, D. Lorenzo, and D. Izzo, *Low-thrust trajectory design as a constrained global optimization problem*, in *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 225 (2011) pp. 1243–1251.

[78] J. A. Englander, B. A. Conway, and T. Williams, *Automated interplanetary trajectory planning,* in *AAS/AIAA Astrodynamics Specialist Conference, Minneapolis, Minnesota* (2012) pp. 4517–4537.

[79] J. A. Englander, M. A. Vavrina, B. Naasz, R. G. Merill, and M. Qu, *Mars, phobos, and deimos sample return enabled by arrm alternative trade study spacecraft,* in *AIAA/AAS Astrodynamics Specialist Conference, San Diego, CA* (2014).

[80] A. Jalal Uddin Jamali, A. Grosso, M. Locatelli, and F. Schoen, *Packing identical circles in a minimized circular container by monotonic basin hopping heuristic approach,* in *ICCIT'09. 12th International Conference on Computers and Information Technology, Dhaka* (2009) pp. 1–6.

[81] P. E. Gill, W. Murray, and M. A. Saunders, *Snopt: An sqp algorithm for large-scale constrained optimization,* in *SIAM journal on optimization*, Vol. 12 (2002) pp. 979–1006.

[82] P. E. Gill, W. Murray, and M. A. Saunders, *Users guide for snopt version 7 - software for large-scale nonlinear programming,* (2008), [online database], URL: http://web.stanford.edu/group/SOL/guides/ [cited 23 October 2015].

[83] R. Hofsteenge, *Computational Methods for the Long-Term Propagation of Space Debris Orbits*, Master's thesis, Delft University of Technology (2013).

[84] R. Noomen, *ae4-878.integrators.v4-3,* Lecture slides, Delft University of Technology (2013), [Internal publication] Course: Mission Geometry and Orbit Design.

[85] A. Milani and A. M. Nobili, *Integration error over very long time spans,* in *Celestial mechanics*, Vol. 43 (Springer, 1987) pp. 1–34.

[86] P. Deuflhard, U. Nowak, and U. Poehle, *Program descriptions of elib,* (1994), [online database], URL: http://elib.zib.de/pub/elib/codelib/difex2/readme [cited 30 October 2015].

[87] O. Montenbruck, *Numerical integration methods for orbital motion,* in *Celestial Mechanics and Dynamical Astronomy*, Vol. 53 (1992) pp. 59–69.

[88] J. Dormand, M. El-Mikkawy, and P. Prince, *Families of runge-kutta-nystrom formulae,* in *IMA Journal of Numerical Analysis*, Vol. 7 (1987) pp. 235–250.

[89] E. Fehlberg, *Low order classical Runge-Kutta formulas with stepwise control*, Tech. Rep. (Marschall Space Flight Center, NASA, 1969) : NASA TR R-316.

[90] E. Fehlberg, *Classical fifth-, sixth-, seventh-, and eighth-order Runge-Kutta formulas with stepsize control*, Tech. Rep. (Marschall Space Flight Center, NASA, 1968) : NASA TR R-287.

[91] M. M. Berry, *A variable-step double-integration multi-step integrator*, Ph.D. thesis, Virginia Polytechnic Institute and State University (2004).

[92] J. Meijaard, *A comparison of numerical integration methods with a view to fast simulation of mechanical dynamical systems,* in *Real-Time Integration Methods for Mechanical System Simulation*, Vol. 69 (Springer, 1991) pp. 329–343.

[93] H. Ramos and J. Vigo-Aguiar, *Variable stepsize störmer-cowell methods,* in *Mathematical and Computer Modelling*, Vol. 42 (2005) pp. 837–846.

[94] J. R. Scott and M. C. Martini, *High speed solution of spacecraft trajectory problems using taylor series integration,* in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Honolulu, Hawaii* (2008).

[95] T. R. Quinn, S. Tremaine, and M. Duncan, *A three million year integration of the earth's orbit,* in *The Astronomical Journal*, Vol. 101 (1991) pp. 2287–2305.

[96] P. J. van der Houwen, E. Messina, and J. J. de Swart, *Parallel Störmer-Cowell methods for high-precision orbit computations*, Tech. Rep. 12 (Centrum voor Wiskunde en Informatica, 1998) : MAS-R9812.

[97] A. B. Gonzalez, P. Martin, and D. J. Lopex, *On the numerical integration of orbital problems with high order runge–kutta–nystrom methods,* in *Applied numerical mathematics*, Vol. 35 (Elsevier, 2000) pp. 1–10.

[98] B. A. Romgens, E. Mooij, and M. C. Naeije, *Verified interval orbit propagation in satellite collision avoidance,* in *AIAA Guidance, Navigation, and Control, Portland, OR* (2011).

[99] D. Gondelach, *A hodographic-shaping method for low-thrust trajectory design*, Master's thesis, Delft University of Technology (2012).

[100] W. Gebbett, *Multi-revolution, low-thrust trajectory design using Lyapunov feedback control*, Master's thesis, Delft University of Technology (2014).

[101] O. Montenbruck, *Numerical integration of orbital motion using taylor series,* in *Spaceflight mechanics* (1992) pp. 1217–1231, available at JPL library.

[102] J. R. Scott and M. C. Martini, *High-speed solution of spacecraft trajectory problems using taylor series integration,* in *Journal of Spacecraft and Rockets*, Vol. 47 (2010) pp. 199–202.

[103] A. Pagano, *Global Launcher Trajectory Optimization for Lunar Base Settlement*, Master's thesis, Delft University of Technology (2010).

[104] M. Balesdent, *Multidisciplinary design optimization of launch vehicles*, Ph.D. thesis, Ecole Centrale de Nantes (2011).

[105] E. Mooij, *The motion of a vehicle in a planetary atmosphere* (Delft University Press, Delft, 1994).

[106] H. Wittenberg, B. Ambrosius, M. Naeije, and E. Mooij, *Ae4-870a: Rocket motion,* Lecture Notes (2014), course: Rocket Motion.

[107] D. Williams, *Mars fact sheet,* website (2015), [online database], URL: http://nssdc.gsfc.nasa.gov/planetary/factsheet/marsfact.html [cited 1 November 2015].

[108] R. Noomen, *Ae2-104, lecture hours 15+16: Kepler orbit, specialized orbits,* Lecture slides, Delft University of Technology (2011), [Internal publication] Course: Flight and Orbital Mechanics.

[109] K. Wakker, *Astrodynamics-i course ae4874, part 1,* Reader, Delft University of Technology (2010), [Internal publication] Course: Astrodynamics I.

[110] D. Smith and M. Zuber, *The relationship between mola northern hemisphere topography and the 6.1-mbar atmospheric pressure surface of mars,* in *Geophysical research letters*, Vol. 25 (1998) pp. 4397–4400.

[111] D. E. Smith, M. T. Zuber, H. V. Frey, J. B. Garvin, J. W. Head, D. O. Muhleman, G. H. Pettengill, R. J. Phillips, S. C. Solomon, H. J. Zwally, *et al.*, *Mars orbiter laser altimeter: Experiment summary after the first year of global mapping of mars,* in *Journal of Geophysical Research*, Vol. 106 (2001) pp. 23689–23722.

[112] N. Barlow, *Mars: An introduction to its interior, surface and atmosphere*, Vol. 8 (Cambridge University Press, Cambridge, 2008).

[113] K. Wakker, *Astrodynamics-ii course ae4874, part 2,* Reader, Delft University of Technology (2010), [Internal publication] Course: Astrodynamics-II.

[114] J. Wertz, *Orbit & Constellation Design & Management*, 2nd ed. (Microcosm Press and Springer, USA, 2009).

[115] J. Polk, R. Kakuda, J. Anderson, J. Brophy, V. Rawlin, M. Patterson, J. Sovey, and J. Hamley, *Validation of the nstar ion propulsion system on the deep space one mission: overview and initial results,* in *35th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, Vol. 99 (1999) p. 2274.

[116] R. Killinger, R. Kukies, M. Surauer, A. Tomasetto, and L. van Holtz, *Artemis orbit raising inflight experience with ion propulsion,* in *Acta Astronautica*, Vol. 53 (Elsevier, 2003) pp. 607–621.

[117] ESA, *All-electric propulsion and multi-satellite launch system for communication satellites,* Website (2015), [online directory], URL: https://directory.eoportal.org/web/eoportal/satellite-missions/a/all-electric [cited 25 November 2015].

[118] R. Noomen, *ae4-878.basics.v4-14,* Lecture slides, Delft University of Technology (2013), [Internal publication] Course: Mission Geometry and Orbit Design.

[119] O. E. Akcasu and I. Akcay, *Solar time using gps technology,* Lonestar Inventions, L.P., Austin, TX, U.S. Patent (2013), patent application, Docket No. WO 2013070518 A1, filed 2 Nov. 2012.

[120] A. E. Petropoulos, *Refinements to the q-law for low-thrust orbit transfers,* in *Advances in the Astronautical Sciences*, Vol. 120 (2005) pp. 963–983, available at JPL library.

[121] G. R. Hintz, *Survey of orbit element sets,* in *Journal of guidance, control, and dynamics*, Vol. 31 (2008) pp. 785–790.

[122] J. A. Kechichian, *Minimum-time constant acceleration orbit transfer with first-order oblateness effect,* in *Journal of Guidance, Control, and Dynamics*, Vol. 23 (2000) pp. 595–603.

[123] R. Battin, *An introduction to the mathematics and methods of Astrodynamics* (AIAA, 1999) revised version, available at JPL library.

[124] A. E. Petropoulos, *Simple control laws for low-thrust orbit transfers,* in *AAS/AIAA Astrodynamics Specialists Conference, Big Sky, MT* (2003) available at JPL library.

[125] A. E. Petropoulos, *Low-thrust orbit transfers using candidate lyapunov functions with a mechanism for coasting,* in *AIAA/AAS Astrodynamics Specialis Conference and Exhibit, Providence, Rhode Island* (2004).

[126] J. Mulder, W. van Staveren, J. van der Vaart, E. de Weerdt, A. in 't Veld, and E. Mooij, *Flight dynamics, lecture notes,* Reader, Delft University of Technology (2013), [Internal publication] Course: Flight Dynamics.

[127] N. Bachman, *Naif spice toolkit hypertext documentation, required reading documents, reference frames,* website (2014), [online database], URL: http://naif.jpl.nasa.gov/pub/naif/ [cited 18 December 2015].

[128] J. D. del Rio, *Generic frame definition kernel file for esa planetary missions,* (2008), [online database], URL: http://naif.jpl.nasa.gov/pub/naif/VEX/kernels/fk/RSSD0002.TF [cited 8 December 2015].

[129] B. A. Archinal, M. Ahearn, E. Bowell, A. Conrad, G. J. Consolmagno, R. Courtin, T. Fukushima, D. Hestroffer, J. L. Hilton, G. A. Krasinsky, *et al.*, *Report of the iau working group on cartographic coordinates and rotational elements: 2009,* in *Celestial Mechanics and Dynamical Astronomy*, Vol. 109 (2011) pp. 101–135.

[130] O. Morton, *Mapping Mars: Science, Imagination, and the Birth of a World*, 1st ed. (Picador, New York, 2003) avaiable at JPL library.

[131] T. Duxbury, P. Christensen, D. Smith, G. Neumann, R. Kirk, M. Caplinger, A. Albee, N. Seregina, G. Neukum, and B. Archinal, *The location of airy-0, the mars prime meridian reference, from stereo photogrammetric processing of themis ir imaging and digital elevation data,* in *Journal of Geophysical Research: Planets*, Vol. 119 (John Wiley & Sons, 2014) pp. 2471–2486.

[132] E. Mooij, *Ae3202-week1-lecture1b-frames,* Lecture slides, Delft University of Technology (2013), [Internal publication] Course: Flight Dynamics.

[133] C. Ho, N. Golshan, and A. Kliore, *Radio Wave Propagation Handbook for Communication on and Around Mars*, 1st ed. (NASA Jet Propulsion Laboratory, Pasadena, CA, 2002) handbook JPL Publication 02-5.

[134] H. L. Justh and C. G. Justus, *Utilizing mars global reference atmospheric model (mars-gram 2005) to evaluate entry probe mission sites,* Presentation (2008), [online database], URL: https://smartech.gatech.edu/bitstream/handle/1853/26375/34-186-1-PB.pdf?sequence=1 [cited 10 December 2015].

[135] C. Justus, A. Duvall, and V. W. Keller, *Mars Aerocapture and Validation of Mars-GRAM with TES Data,* in *53rd JANNAF Propulsion Meeting/2nd Liquid Propulsion Subcommittee/Spacecraft Propulsion Joint Meeting* (2005).

[136] M. Sagliano, T. Oehlschlagel, S. Theil, and E. Mooij, *Real Time Adaptive Feedforward Guidance for Entry Vehicles,* in *3rd CEAS EuroGNC, Specialist Conference on Guidance, Navigation and Control, Toulouse* (2015).

[137] E. Mooij, *Passivity Analysis for Non-Linear, Non-Stationary Entry Capsule: Rotational Motion,* in *AIAA Guidance, Navigation, and Control Conference, Porltand, Oregon* (2011).

[138] D. Izzo, *PyGMO and PyKEP: Open source tools for massively parallel optimization in astrodynamics (the case of interplanetary trajectory optimization),* in *5th International Conference on Astrodynamics Tools and Techniques (ICATT)* (2012).

[139] C. Justus, *A mars global reference atmospheric model (mars-gram) for mission planning and analysis,* in *28th Aerospace Sciences Meeting, Reno, Nevada* (1990).