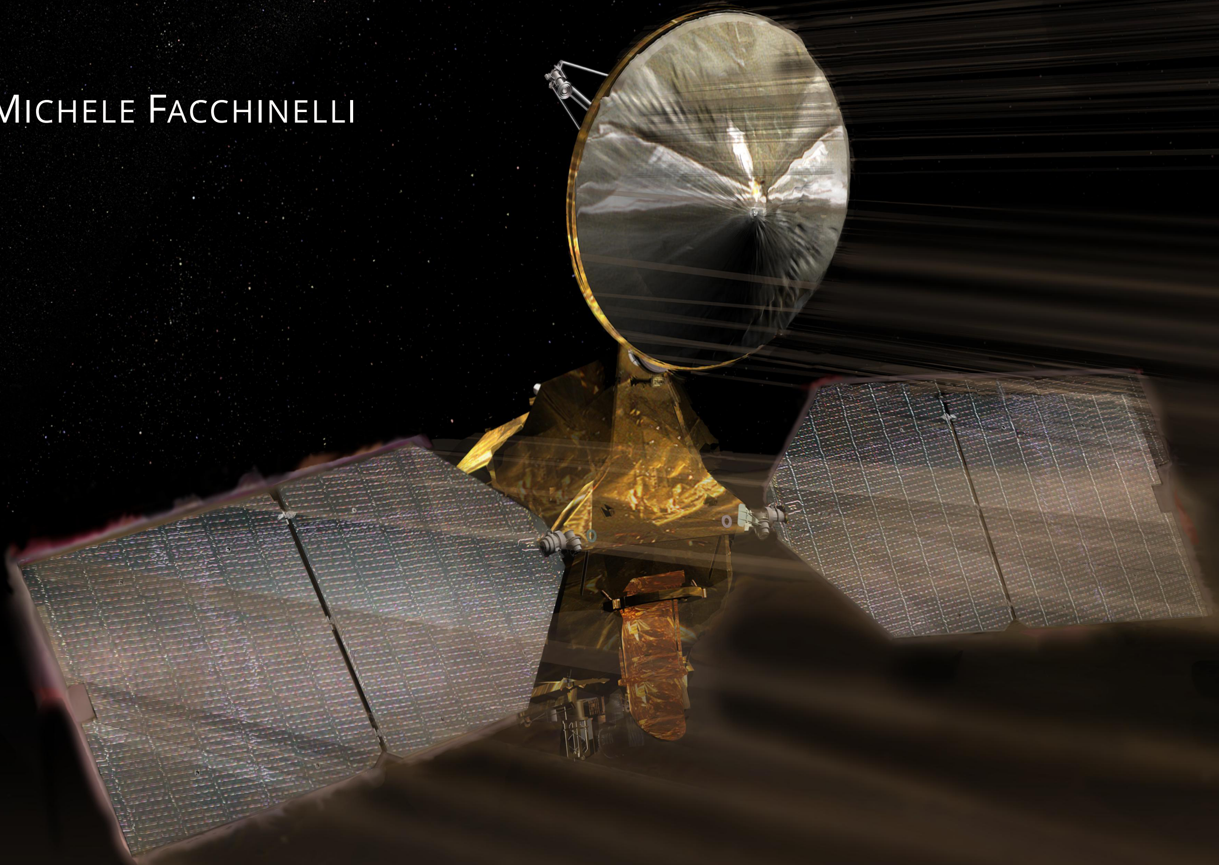
Aerobraking Navigation, Guidance and Control

A Comparison of State Variables Performance in Propagation

Delft University of Technology

MICHELE FACCHINELLI



TUDelft

Page Intentionally Left Blank

Aerobraking Navigation, Guidance and Control

A Comparison of State Variables Performance in Propagation

by

MICHELE FACCHINELLI

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday, November 30, 2018 at 9:30am.

Student number:	4303040	
Project duration:	March 19, 2018 – November 30, 2018	
Thesis committee:	Prof. Dr. Ir. P. Visser,	TU Delft, chair
	Dr. Ir. E. Mooij,	TU Delft, supervisor
	Dr. Ir. E. van Kampen,	TU Delft

Cover picture by Jet Propulsion Laboratory, depicting Mars Reconnaissance Orbiter during aerobraking.
Available at https://goo.gl/tKorKM.
Part picture by Jet Propulsion Laboratory, depicting the Victoria Crater at Meridiani Planum on Mars.
Available at https://goo.gl/yCSk7M.

PAGE INTENTIONALLY LEFT BLANK

Preface

I am proud to present to you what will likely be my biggest single-handed achievement. After what has been a year of emotional roller coasters, I have finally reached this target. I am not entirely sure about how satisfied I am with what I have achieved, but I must admit I am a very hard person to please, and that one can only do so much in one year of work. Overall, it has nonetheless been a worthwhile growing experience.

For starters, I would truly like to thank my friends Zyanya, Kapeel and Ali, for their unconditional support throughout this period. Not only do I have to say that this would have not been possible without them, but I do not think I would have been able to go through the darkest times, without their presence and advice. I have both loved and hated working on this thesis, but as always the negative feelings tend to prevail. My gratitude is also directed to Jorg and Justine, for their great spirit and for the shared moments over WMT. In addition, a big shout-out to the control and simulation students of room 2.56, with whom I have aggregated over many lunches, and spent many of my complaining sessions, otherwise known as coffee breaks.

Great appreciation also goes to my supervisor, Erwin, for keeping me motivated, believing in me and understanding the many meanings behind my "*I'm okay*"s. Our meetings have been something I have looked forward to every week, and I am certainly going to miss them. Furthermore, I would like to thank Dominic Dirkx, Assistant Professor at the faculty, for his help and guidance in the development of my thesis software. Also, thanks for offering the teaching assistant job: it has been a crucial element in helping me keeping my motivation and focus.

Finally, special thanks go to Dr. Jah of the University of Texas at Austin, for his availability and assistance in trying to understand the navigation system he so passionately developed. It is always great to see people so motivated and excited by their work, even more than ten years after the original publication.

Now it is time for my next adventure. Moving has never been an easy experience, but after having developed some of the deepest connections I have ever felt, this time it will be particularly tough. I am nevertheless confident that my life in Cambridge will bring new and exciting moments and that these friendships will transcend the spatial distance. I especially look forward to the rush of adrenaline in getting almost run over every time I try to cross the street.

Michele Facchinelli
Delft, November 16, 2018

Page Intentionally Left Blank

Abstract

The amount of propellant required to perform deep-space maneuvers usually consists of a large percentage of the total spacecraft mass. In the early '90s, a new technique was experimented at the end of the Magellan mission, which employed the atmosphere of Mars itself, to reduce the eccentricity of the orbit over a long period of time. Said action was named 'aerobraking' and the design of the required onboard software to perform this maneuver, is the main topic of this thesis. In fact, the research question that this work tries to answer is as follows:

Can the mass of an interplanetary spacecraft be decreased by using a robust guidance, navigation and control system for aerobraking, without jeopardizing the safety of the spacecraft and maintaining a low navigation error?

where the use of colors will become clear later on. To solve the problem this question poses, the guidance, navigation and control (GNC) systems detailed below have been developed:

- **Navigation System**
  The purpose of this part of the software is to determine the current spacecraft state (position, velocity and attitude). Two navigation systems have been implemented in this research and they differ in the fact that one is purely based on data available onboard (such as accelerometer measurements and a simplified environment model), whereas the other uses observations from an unspecified ranging system, to complement the onboard models. These systems are also supplemented by an unaided navigation technique, which consists of direct integration of the accelerations estimated by the simplified environment.
- **Guidance System**
  The tasks of the guidance system can be summarized as: estimating the minimum and maximum values of the pericenter altitude and targeting the midpoint of said range. The first function is achieved by considering the thermodynamic limitations of the spacecraft (for the lower bound) and the restrictions on aerobraking duration (for the upper one), and by splitting the aerobraking maneuver into a few different phases, depending on the eccentricity of the orbit. To attain the second purpose, in contrast, use is made of a maneuver estimator. As the name suggests, this element estimates the magnitude and direction of the required maneuvers.
- **Control System**
  Both an attitude and an orbit controller are designed for the mission. A proportional, integral and derivative controller is used to regulate the rotational motion of the spacecraft, whereas, orbit control is realized by assuming the impulsive shot approximation holds, and is modeled by simply adding or subtracting the velocity change estimated by the maneuver estimator to and from the spacecraft velocity.

Based on the analyses performed, it can be concluded that the research question above has only been partly answered. In particular, if one focuses on the four highlighted words, the following can be said about the completion of this thesis:

Mass The GNC system is able to reduce the mass of an interplanetary spacecraft, by requiring the employment of 82 instead of more than 1000 m s^{-1} of velocity increments. Nonetheless, the guidance algorithm does, at times, result in an inefficient usage of propellant. A more optimal design would thus be able to decrease the mass even further.

Error A low navigation error is achieved when the second navigation system described above, is selected. In fact, the root mean square (RMS) error in position is kept below 50 m. The same, however, cannot be said for the first navigation system, nor for the unaided method. In fact, for the latter, RMS errors of up to 1.5 km are observed.

Robust In the computation of the lower and upper bounds for pericenter height, the combination of uncertainties in the navigation system estimations lead to inaccurate estimations of the expected aerothermodynamic conditions and of the required maneuver values. Therefore, the robustness of the guidance system would need to be improved, by performing an extensive sensitivity analysis to the uncertainty in navigation variables.

Safety The excessive heating experienced during the atmospheric phases of the mission, poses a large threat to the thermal and overall integrity of the spacecraft, possibly jeopardizing the whole mission.

Finally, due to the large duration of an aerobraking mission, the usage of an efficient set of propagation coordinates is of high interest. For this reason, the performance of the unified state model (USM) variables was also analyzed. As it turned out, due to the small time step needed to simulate the onboard processing of the various GNC routines listed above, no advantage of using the USM over more conventional propagators, such as Cowell with Cartesian coordinates, was observed. This is also related to the fact that aerobraking is a highly dynamic and very long lasting maneuver. In fact, the USM coordinates showed the most attractive results in terms of RMS error in position (and velocity), in case of short propagation durations and a limited number of perturbing accelerations.

Contents

List of Symbols

Latin Symbols:

a	Semi-major axis	[m]
a	Speed of sound	[m s^{-1}]
$\mathbf{a}$	Acceleration	[m s^{-2}]
A	Cross-sectional area	[m^{2}]
A	Wave amplitude	[−]
$\mathbf{A}$	Design matrix	variable
$\mathbf{b}$	Bias	variable
c	Speed of light	[m s^{-1}]
C	Coefficient	[−]
C_h	Heat transfer coefficient	[W m^{-2} K^{-1}]
C_p	Specific heat at constant pressure	[J kg^{-1} K^{-1}]
C	Velocity component perpendicular to radius vector for the unified state model	[m s^{-1}]
$\mathbf{C}$	Direction cosine matrix	[−]
D	Diameter	[m]
e	Eccentricity	[−]
$\hat{\mathbf{e}}$	Euler eigenaxis	[−]
$\mathbf{e}$	Exponential map	[−]
E	Eccentric anomaly	[rad]
f	Fraction	[−]
$\mathbf{f}$	System function	variable
$\mathcal{F}$	Reference frame	[−]
$\mathbf{F}$	Force	[N]
g	Gravitational acceleration	[m s^{-2}]
G	Universal gravitational constant	[kg m^{3} s^{-2}]
h	Altitude	[m]
h	Step size	[s]
$\mathbf{h}$	Measurement function	variable
H	Scale height	[m]
$\mathbf{H}$	Specific angular momentum	[m^{2} s^{-1}]
i	Inclination	[rad]
I	Impulse	[s]
I	Interpolating function	variable
$\mathbf{I}$	Identity matrix	[−]
J	Gravitational moment	[−]
$\mathbf{J}$	Inertia matrix	[kg m^{2}]
$\mathbf{J}$	Jacobian	variable
k	Boltzmann constant	[J K^{-1}]
$\mathbf{k}$	Runge-Kutta estimates	variable
Kn	Knudsen number	[−]
$\mathbf{K}$	(Kalman) Gain	variable
L	Length, characteristic vehicle dimension	[m]
L	Lifetime	[s]
L	Luminosity	[W]
$\mathbf{L}$	Moment arm	[m]
m	Mass	[kg]
M	Mach number	[−]
M	Mean anomaly	[rad]
$\mathscr{M}$	Molar mass	[kg mol^{-1}]
$\mathbf{M}$	Misalignment matrix	[−]

$\mathbf{M}$	Moment	[N m]
n	Mean motion	[rad s^{-1}]
n	Number density	[m^{-3}]
$\hat{\mathbf{n}}$	Line of nodes	[−]
$\hat{\mathbf{n}}$	Surface normal	[−]
N_A	Avogadro's number	[mol^{-1}]
$\mathcal{N}$	Orbit number	[−]
$\mathcal{O}$	Order	[−]
p	Semi-latus rectum	[m]
p	Auxiliary variable for the unified state model	[−]
$\mathbf{p}$	Pressure	[Pa]
P	Orbital period	[s]
P	Polynomial function	variable
$\mathcal{P}_n$	n-th degree Legendre polynomial	[−]
$\mathcal{P}_n^m$	n-th degree and m-th order associated Legendre function	[−]
$\mathbf{P}$	Covariance matrix	variable
q	Heat flux	[W m^{-2}]
$\bar{q}$	Dynamic pressure	[N m^{-2}]
$\mathbf{q}$	Quaternion	[−]
Q	Heat load	[J m^{-2}]
Q	Quadrant	[−]
r	Recovery factor	[−]
$\mathbf{r}$	Orbital distance	[m]
R	Gas constant	[J kg^{-1} K^{-1}]
R	Radius	[m]
Re	Reynolds number	[−]
$\mathscr{R}$	Universal gas constant	[J mol^{-1} K^{-1}]
$\mathbf{R}$	Rotation matrix	[−]
$\mathbf{R}$	Velocity along the semi-latus rectum direction for the unified state model	[m s^{-1}]
s	Molecular speed ratio	[−]
$\mathbf{s}$	Infinitesimal element	[−]
$\mathbf{s}$	Scale factor	[−]
S	Dust storm parameter	[−]
St	Stanton number	[−]
$\mathbf{S}$	Surface area	[m^{2}]
t	Time	[s]
T	Temperature	[K]
u	Argument of latitude	[rad]
$\mathbf{u}$	Control vector	variable
U	Gravitational potential	[−]
$\mathbf{v}$	Measurement noise	variable
$\mathbf{V}$	Velocity	[m s^{-1}]
w	Weight	[−]
$\mathbf{w}$	Uncorrelated white noise, system noise	variable
$\mathbf{W}$	Weighing matrix	[−]
x, y, z	Cartesian coordinates	[m]
x	State variable	variable
$\mathbf{x}$	State vector	variable
$\mathcal{X}$	State sigma points	variable
$\mathbf{z}$	Measurement vector	variable
Z	Relaxation number	[−]
$\mathcal{Z}$	Measurement sigma points	variable
$\mathbf{0}$	Zero matrix	[−]

Greek Symbols:

α	Angle of attack	[rad]
α	Angular scattering parameter	[−]
β	Angle of sideslip	[rad]
β	Inverse scale height	[m^{-1}]
γ	Flight-path angle	[rad]
γ	Specific heat ratio	[−]
γ	Auxiliary variable for the unified state model	[−]

δ	Latitude	[rad]
ε	Tolerance/accuracy	[–]
$\boldsymbol{\varepsilon}$	Imaginary part of quaternion	[–]
ζ	Atmosphere multiplication factor	[–]
η	Real part of quaternion	[–]
ϑ	Incidence angle	[rad]
ϑ	True anomaly	[rad]
κ	Constant	variable
λ	Levenberg–Marquardt damping coefficient	[–]
λ	Mean free path	[m]
λ	Right ascension of latitude	[rad]
μ	Standard gravitational parameter	[m^{3} s^{-2}]
μ	Viscosity	[kg m^{-1} s^{-1}]
ξ	Rotation angle	[rad]
$\boldsymbol{\Xi}$	Intermediate variable	[–]
ρ	Density	[kg m^{-3}]
ρ	Levenberg–Marquardt gain factor	[–]
σ	Accommodation coefficient	[–]
σ	Bank angle	[rad]
σ	Molecular collision diameter	[m]
σ	Standard deviation	variable
$\boldsymbol{\sigma}$	Modified Rodrigues parameters	[–]
τ	Longitude	[rad]
$\boldsymbol{\tau}$	Shear stress	[N m^{-2}]
Φ	Flux	[W m^{-2}]
χ	Heading angle	[rad]
ω	Argument of periapsis	[rad]
ω	Temperature-dependence of viscosity	[–]
$\boldsymbol{\omega}$	Angular velocity	[rad s^{-1}]
Ω	Right ascension of ascending node	[rad]

Sub- and Superscripts:

a	Apoapsis
a	Axial
aero	Aerodynamic
A	Airspeed
c	Control
c	Covariance
d	Derivative
D	Drag
e	Edge of boundary layer
e	Error
f	Shear, friction
grav	Gravitational
guess	Guessed value
h	Heat
i	Integral
l	Roll
L	Lift
m	Order
m	Pitch
max	Maximum
min	Minimum
n	Degree
n	Normal
n	Yaw
N	Normal
p	Periapsis
p	Pressure
p	Proportional
peak	Peak, local maximum
pert	Perturbing

r	Rotational
rad	Radiation pressure
s	Shadow
s	Stagnation point
sp	Specific
S	Side force
t	Translational
t	Transverse
trans	Transition
third	Third body
T	Tangential
v	Vibrational
w	Wall
0	Initial or total
∞	Free-stream conditions

Reference Frames:

$\mathcal{F}_{\mathcal{A}}$	Aerodynamic reference frame
$\mathcal{F}_{\mathcal{B}}$	Body-fixed reference frame
$\mathcal{F}_{\mathcal{I}}$	Inertial planetocentric reference frame
$\mathcal{F}_{\mathcal{J}}$	International celestial reference frame
$\mathcal{F}_{\mathcal{L}}$	Local orbital reference frame
$\mathcal{F}_{\mathcal{O}}$	Orbit reference frame
$\mathcal{F}_{\mathcal{R}}$	Rotating planetocentric reference frame
$\mathcal{F}_{\mathcal{T}}$	Trajectory reference frame
$\mathcal{F}_{\mathcal{V}}$	Vertical reference frame

Special Characters:

$\bar{\square}$	Average
$\square^{\mathrm{a}}$	Augmented value
$\hat{\square}$	Estimated value
$\square'$	Intermediate value
$\tilde{\square}$	Median
$\overline{\square}$	Normalized
$\square^{\dagger}$	Post-maneuver value
$\check{\square}$	Target or allowed value

$\hat{\blacksquare}$	Unit vector
$\blacksquare^{\mathrm{S}}$	Skew-symmetric matrix in terms of vector elements
$\blacksquare^{\mathrm{T}}$	Matrix transpose
$\\|\blacksquare\\|$	Magnitude of vector

$\odot$	Sun
Υ	Vernal equinox

$\triangleq$	Defined as

Chemical Formulas:

Ar	Argon
$\mathrm{CH_4}$	Methane
CO	Carbon monoxide
$\mathrm{CO_2}$	Carbon dioxide
H	Monoatomic hydrogen
$\mathrm{H_2}$	Diatomic hydrogen
$\mathrm{H_2O}$	Dihydrogen monoxide
He	Helium
$\mathrm{N_2}$	Nitrogen
O	Monoatomic oxygen
$\mathrm{O_2}$	Diatomic oxygen
$\mathrm{O_3}$	Ozone

List of Abbreviations

AAB	Autonomous Aerobraking
AB	Aerobraking
ABM	Aerobrake Trim Maneuver
AE	Atmosphere Estimator
AOA	Angle of Attack
ASF	Atmospheric Safety Factor
C	Control
CG	Center of Gravity
CM	Center of Mass
COTM	Corridor Orbit Trim Maneuver
CP	Center of Pressure
CSH	Constant Scale Height
DAIA	Dynamic Atmospheric Interface Altitude
DCM	Direction Cosine Matrix
DSMC	Direct Simulation Monte Carlo
DSN	Deep Space Network
EE	Ephemeris Estimator
EKF	Extended Kalman Filter
EM	Exponential Mapping
ESA	European Space Agency
G	Guidance
GAN	GRS-assisted Navigation
GNC	Guidance, Navigation and Control
GNSS	Global Navigation Satellite System
GRAM	Global Reference Atmospheric Model
GRS	Generic Ranging System
GS	Ground Station
IERS	International Earth Rotation and Reference Systems Service
IMAN	Inertial Measurements for Aeroassisted Navigation
IMU	Inertial Measurement Unit
JPL	Jet Propulsion Laboratory
LEO	Low Earth Orbit
MAG	Magellan
ME	Maneuver Estimator
MGS	Mars Global Surveyor
MOY	Mars Odyssey
MRP	Modified Rodrigues Parameters
MRO	Mars Reconnaissance Orbiter
N	Navigation
NA	Not Applicable or Not Available
NASA	National Aeronautics and Space Administration
OBC	Onboard Computer

ODE	Ordinary Differential Equation
PID	Proportional, Integral and Derivative
PTE	Periapse Time Estimator
REQ	Requirement
RK4	Runge-Kutta Fourth-order Integration
RKF5(6)	Runge-Kutta-Fehlberg Fifth- and Sixth-order Integration
RKF7(8)	Runge-Kutta-Fehlberg Seventh- and Eighth-order Integration
RMS	Root Mean Square
RW	Reaction Wheel
SA	Solar Array
SC	Spacecraft
SEM	Shadow Exponential Mapping
SMRP	Shadow Modified Rodrigues Parameters
SPARTA	Stochastic Parallel Rarefied-gas Time-accurate Analyzer
SPICE	Spacecraft Planet Instrument Camera-matrix Events
SOI	Sphere of Influence
STD	Standard Deviation
TBD	To Be Defined or To Be Determined
TGO	Trace Gas Orbiter
Tudat	TU Delft Astrodynamics Toolbox
TU Delft	Delft University of Technology
UKF	Unscented Kalman Filter
USM	Unified State Model
USMEM	Unified State Model with Exponential Mapping
USM6	Unified State Model with Modified Rodrigues Parameters
USM7	Unified State Model with Quaternions
VEX	Venus Express
VSS	Variable Soft Sphere

1 Introduction

Interplanetary flights are very expensive. Other than the costs needed to develop the mission and the spacecraft design, one also needs to take into account the propellant needed to get to the target body. It is not the propellant itself that is expensive, but taking it along the flight is what results in the high costs. All of the maneuvers required (transition from parking orbit to transit orbit, and break to a specific orbit around the target), result in an extensive amount of propellant. The total fuel weight can be as much as 1500 kg for a Mars-bound satellite, which was the case for the MAVEN spacecraft[1]. Each kilo of propellant saved, can potentially be used to increase the payload mass (which for MAVEN was a mere 2.5 % of the total mass).

A very ingenious way of saving propellant was introduced in the 1990s with the Magellan spacecraft. Instead of using a conventional periapsis maneuver to reduce the ellipticity of the orbit, the spacecraft used the atmosphere of Venus. In this manner, the mission managed to save about 500 kg of fuel (Spencer and Tolson, 2007). The maneuver performed was named 'aerobraking', which since the 1990s has been used a total of six times, and whose procedure is still being refined for future missions.

The main reason why aerobraking sparks so much interest is its propellant-saving ability. However, aerobraking also comes with some disadvantages, these being: it can only be used for missions bound to an atmosphere-provided celestial body and it is a slow process. In general, nevertheless, aerobraking (AB) consists of using the atmosphere of a planet to dissipate orbital energy and thus, to reduce the eccentricity of a highly elliptical orbit, such that a more science-friendly quasi-circular orbit can be achieved. In this situation, the perigee of the orbit is gradually lowered in the highest layers of the atmosphere, where atmospheric density has a small instantaneous effect, but very large long-term consequences. In fact, aerobraking is a slow procedure and requires hundreds if not thousands of orbits to complete. The main reason why the perigee of the orbit is not lowered any further down the atmosphere has to do with heating. Aerodynamic drag, even at very high altitudes, has a major effect on the temperature of the spacecraft, which usually is very thermally sensitive (especially the solar arrays).

Based on the brief discussion above, primary and secondary research questions and subquestions were formulated. These are shown below:

PRIMARY:

○ Can the mass of an interplanetary spacecraft be decreased by using a robust guidance, navigation and control system for aerobraking, without jeopardizing the safety of the spacecraft and maintaining a low navigation error?
  - What are the constraints on the braking effect?
  - How long can the navigation errors be kept below the requirement threshold, i.e., before ground station intervention is necessary?
  - What is the sensitivity to deviations from the initial conditions?
  - What is the sensitivity to aerodynamic coefficients uncertainty?
  - How do uncertainties in the atmospheric modeling affect the system?

○ Which state variables provide the best compromise between computation time and accuracy in propagation?

SECONDARY:

○ What is the optimal sensor configuration?

From the research question, a research objective can be extracted. Hence, the scope of this Master thesis is:

[1] University of Colorado Boulder: Laboratory for Atmospheric and Space Physics: MAVEN - https://goo.gl/DCXWYn [last accessed on February 26, 2018]

to design a robust guidance, navigation and control system for transitioning from a highly elliptical trajectory, to a quasi-circular orbit, by using the body's atmosphere to dissipate energy, with the aim of reducing the launch mass for interplanetary missions and increasing their autonomy .

The research question above, requires some clarification. In particular, the focus is brought to the words highlighted.

ELLIPTICAL AND QUASI-CIRCULAR Understanding of astrodynamics is fundamental for the success of any mission. Astrodynamics, however, is built upon the even more fundamental flight dynamics, which explains how frames and state variables are defined and how they are used. Astrodynamics, on the other hand, describes the motion of an object in space, w.r.t. said frames and in terms of said variables. One can find the former topic in Chapter 3, and the latter in Chapter 4. As it turns out, some of these state variables can provide better performance than others, depending on the problem dynamics. This is what the subtitle of this report is based upon: a comparison will be carried out among the various state variables introduced in Chapter 4.

ATMOSPHERE By lowering the periapsis of the orbit in the high layers of the atmosphere, the orbital energy can be gradually reduced. This means that the spacecraft uses drag, instead of an engine to provide enough ΔV to circularize the orbit. One important fact to consider, however, is the unpredictability and variability of the atmospheric environment, which can have a strong influence on the design of an aerobraking maneuver. Hence, it is important to properly model the atmosphere, such that a realistic scenario is considered. Description of how the atmosphere and other perturbations are modeled is given in Chapter 4. The orbital energy that is dissipated, however, does not disappear into thin air. Instead, it is transformed into heat, vibrations and noise. Of these, only the consideration of spacecraft heating is fundamental. This is due to the high thermal sensitivity of the some elements, such as solar arrays. Aerothermodynamics hence, is a key concern and will be treated by Chapter 5.

GUIDANCE, NAVIGATION AND CONTROL SYSTEM A navigation systems has the purpose of providing the onboard computer with the knowledge of the spacecraft state. The reader will become familiar with the navigation system for the aerobraking maneuver designed in this thesis by reading Chapter 6, where the implemented sensors and algorithms will be described. Once the state of the spacecraft is known, the guidance system can use it to target the desired conditions. In case of aerobraking, these are updated every orbit, based on the heating conditions mentioned under 'atmosphere'. Finally, the control system combines the commanded and navigation values, to correct for any error in the current state. Chapter 7 will see the description of both the guidance and the control system introduced here. Clearly, there are a lot of interactions between these onboard systems. For this reason, Chapter 8 was included in this report. Here, one will find more details on the interfaces between the various GNC components. Finally, in Chapter 9 one will find the verification and validation of the developed software.

ROBUST AND AUTONOMY A robust system is one that is able of coping with relatively large deviations from the optimal or nominal conditions autonomously. Furthermore, it is capable of independently computing corrective maneuvers, and updating its position, based on onboard sensors and actuators, without the aid of continuous coverage from an Earth-based ground station.

LAUNCH MASS A lower launch mass is achieved thanks to the lower amount of propellant carried onboard. Since the atmosphere is used to slow down the spacecraft, a much smaller amount of ΔV needs to be generated by the propulsion system (the only ΔV used is to make small changes in the periapsis altitude). Thus, a lower velocity budget requires lower propellant mass, which in turn, could lead to substantial reductions in interplanetary mission costs. Alternatively, one could keep the launch mass constant, and increase the payload mass.

When starting a new project, any researcher needs to know what is the state-of-the-art in the field that is being explored. As already mentioned, a few NASA and ESA missions have relied on aerobraking in the past, thus, one has to look at how their techniques evolved and what were important lessons learned. This, in fact, is the first topic covered in this literature research and can be found in Chapter 2.

The overall structure of the report is divided into three main parts. Part I is meant to provide some background information on both the aerobraking problem and the environment that will be encountered during this maneuver. Part II, on the other hand, takes this information and builds upon it the whole onboard system, comprising of the navigation, guidance and control elements. Finally, in Part III, the results of implementing the software developed in Part II in the environment of Part I will be analyzed. In particular, the reader's attention is brought to Fig. 1.1, where the flow of information from the first two parts to the third part is highlighted.

From Fig. 1.1, one can deduce the reasoning behind the structure of Part III: the order follows directly from the sequence of chapters of the previous two parts. Therefore, one can see that the results displayed Part III represent 1) the analysis of the performance of state variables in propagation, which can be found in Chapter 10, 2) the aero-

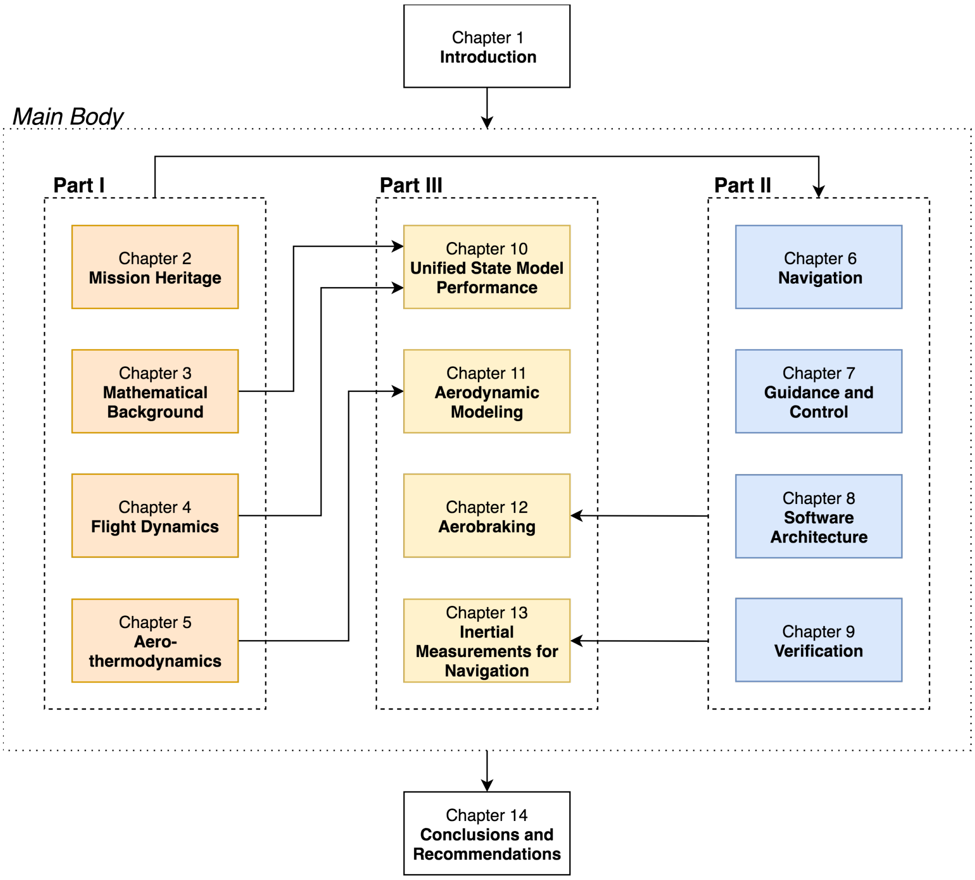


Figure 1.1: *Structure of the present Master thesis report, detailing the flow of information from Parts I and II to Part III.*

dynamic coefficients of the selected spacecraft model, in Chapter 11, 3) the application of the developed aerobraking GNC algorithm, which are split over Chapters 12 and 13. The main body of the report is then brought to an end by the conclusions and recommendations chapter.

The very last elements of the report are appendices. They are reported in order in which they are mentioned in the report, and provide auxiliary information to the main body of this research.

PAGE INTENTIONALLY LEFT BLANK

I

Background

“*I like to imagine that the world is one big machine. You know, machines never have any extra parts. They have the exact number and type of parts they need. So I figure if the entire world is a big machine, I have to be here for some reason. And that means you have to be here for some reason, too.*”

Brian Selznick, The Invention of Hugo Cabret, 2007

2
Mission Heritage

Analyzing previous missions is an important part of research. One wants to learn from the mistakes and achievements of previous projects, and wants to build up on what is already available. Doing otherwise would increase the required development time, to exaggerated extents. Therefore, missions and research from NASA and ESA (the only space agencies that have used aerobraking) will be analyzed in this chapter.

This chapter is divided into three sections: past, present and future. First, missions whose aerobraking phases have already elapsed will be analyzed. For these missions, which are described in Section 2.1, more documents are available, detailing the results and lessons learned. Then, research for future developments will be introduced by Section 2.2. For these missions, no actual flight results will be available, but a more in-depth review of the techniques considered will be carried out. At the end of the chapter, i.e., in Section 2.3, a reference mission will be chosen. Based on this mission and other parameters from deeper analysis in the rest of the document, some requirements will be set.

After the description of the missions, the reader will find Table 2.1, which summarizes the specifications of the aerobraking phase and spacecraft (SC) details. Of particular interest are the details of MRO, which as explained in Section 2.3, will be used as reference mission. References for Table 2.1 are:

(a) MAG: (Young, 1990)
(b) MAG: (Giorgini et al., 1995)
(c) MGS: [1]
(d) MOY: (Smith and Bell, 2005)
(e) MRO: (Long et al., 2008)
(f) MRO: (Lyons, 2002)
(g) MRO: [2]
(h) VEX: (Svedhem, 2015)
(i) VEX: (Sivac and Schirmann, 2007)
(j) TGO: (Long et al., 2012)

2.1 | Past Missions

Six missions have used aerobraking as means to control the shape and orbital period of their orbits. The first spacecraft to ever perform an aerobraking maneuver was the Japanese Hiten, that flew-by Earth once in 1991[3]. The Magellan (MAG) spacecraft, on the other hand, was the first to perform an aerobraking maneuver around another planet (Spencer and Tolson, 2007). Magellan was then followed, in order of launch, by Mars Global Surveyor (MGS), Mars Odyssey (MOY), Mars Reconnaissance Orbiter (MRO), and Venus Express (VEX). For each spacecraft, the aerobraking configuration will be shown. In these figures, the black arrow shows the nominal direction of the airflow, whereas the dashed line points toward the planet surface.

Note that the main references for the missions are mentioned in the title name. References for specific facts can be found in the text.

Magellan (Lyons et al., 1995)

Magellan was a three-axis stabilized spacecraft, controlled by three reaction wheels (RWs) for attitude control, and pairs of thrusters used for desaturation of the reaction wheels and for corridor orbit trim maneuvers (COTMs) (Giorgini et al., 1995). This set of thrusters, was the main driver of the aerobraking maneuver. Magellan was inserted in a highly-elliptical orbit around Venus in 1990, but to increase the mapping coverage, the orbit needed to be circularized (Doody, 1995). Not being designed for major thrust usage, the only way to lower the apoapsis altitude by almost 8000 km (from 8500 to 541 km), was aerodynamic drag. The aerobraking maneuver was divided into three phases: walk-in, main and endgame.

[1] Jet Propulsion Laboratory: Magellan: Spacecraft Overview – https://goo.gl/hj61uc [last accessed on October 25, 2017]

[2] Jet Propulsion Laboratory: Mars Reconnaissance Orbiter: Spacecraft Parts – https://goo.gl/VCfgEh [last accessed on October 24, 2017]

[3] National Aeronautics and Space Administration: Hiten-Hagoromo: In Depth – https://goo.gl/QP11tv [last accessed on October 23, 2017]

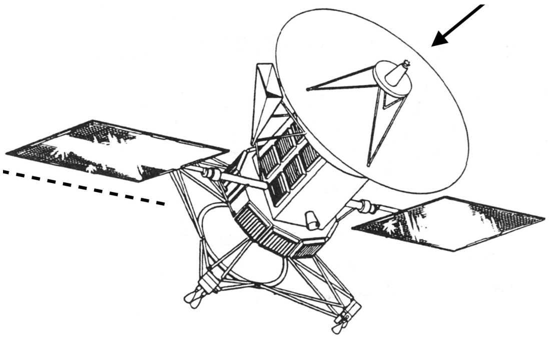


Figure 2.1: *Magellan spacecraft (adapted from Spencer and Tolson, 2007). Note that during aerobraking, the solar array was oriented perpendicular to the airflow.*

The walk-in phase was added to slowly lower the periapsis from the initial 172 to 150 km, due to the lack of accurate density models. For the main phase, a periapsis dynamic pressure corridor was established. At first it was bounded by 0.27 and 0.32 N m^{-2}, but was then shifted to 0.29 and 0.35 N m^{-2}, due to the lower than expected density fluctuations and solar array heating. The main drivers for the upper and lower bounds were in fact, the maximum temperatures for solar array joints (179 and 160 °C for temperature sensors), antenna delamination (180 °C), and limits on the attitude control system, for the upper bound. The lower bound was limited by the constraint on 100 days for the total aerobraking time. The main goal of the eight-day *endgame* was to propulsively raise the periapsis, to almost nullify atmospheric influence.

Figure 2.1 shows the Magellan spacecraft. For aerobraking, the configuration was such that the maximum solar panel area was exposed to the flow. Another important feature of this configuration was its longitudinal stability, thanks to the position of the center of pressure (CP) (aft of the center of mass, CM).

As stated by Giorgini et al. (1995), six possible COTMs were stored for orbit corrections at apoapsis on the spacecraft computer. The nominal $1n$ maneuver would provide a 0.34 m s^{-1} of ΔV, and the $1/2\ n$ and $2n$ maneuvers would provide half or two times as much ΔV. Moreover, its value could be positive or negative, resulting in a higher or lower periapsis altitude, respectively.

In general, the main challenges faced by the Magellan team were the uncertainties in atmospheric and gravitational conditions at Venus and lack of automation in the orbit determination and guidance system.

Mars Global Surveyor (Lyons et al., 1999)

Mars Global Surveyor was the first satellite for which aerobraking was specifically used to reach the desired orbit. Figure 2.2 shows the planned configuration of MGS during aerobraking. To achieve aerodynamic stability, the solar panels were swept by 30 deg, and to increase the effect of the atmosphere, drag flaps were attached at the solar array extremities.

Due to a problem in the deployment of the solar arrays after launch, the solar array was tilted 20 deg w.r.t. the nominal position. During the walk-in phase, no major deflection of the solar panels was noticed, and the MGS team deemed it safe to continue. However, after an atmospheric sweep with a much higher than expected density, the solar panels underwent substantial damage. It was therefore decided to stop aerobraking until the damage was fully assessed and a new strategy was chosen.

After almost one month, aerobraking was resumed, since the spacecraft was relying on aerobraking to achieve its

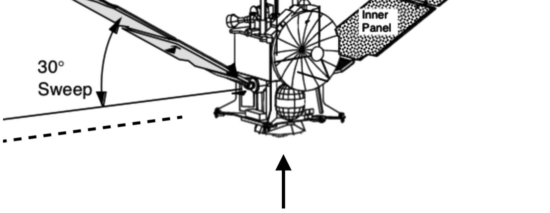


Figure 2.2: *Mars Global Surveyor (adapted from Spencer and Tolson, 2007).*

final orbit. To make sure that no further damage would occur, the maximum allowed dynamic pressure was reduced to one-third of the initial value, resulting in a one year delay.

Mars Odyssey (Smith and Bell, 2005)

Mars Odyssey used aerobraking to provide an equivalent ΔV of 1.08 km s^{-1}, to reduce the orbital period from 18.6 to less than 2 h. Figure 2.3 shows the aerobraking configuration of the spacecraft, where the solar array was placed perpendicular to the flow, to increase the reference area to 11 m^2. Aerobraking was divided into four phases: walk-in, main, endgame and walk-out.

The walk-in phase, was used to lower the periapsis altitude from 292 to 111 km. This was done gradually, to calibrate models and evaluate spacecraft performance.

The limitations on the amount of braking in the main phase, were dictated by heating of the solar array. The unpredictability of atmospheric conditions lead to reference thermal constraints much lower than the actual solar panel limits. Peak heat rate was used as limit in the main phase, whereas cumulative heating was used in the *endgame* phase.

As soon as the predicted time to reach 300 km apoapsis altitude (defined as lifetime) was estimated to be less than one day, the walk-out phase started. Below 300 km apoapsis altitude, spiraling inwards toward the planet becomes almost inevitable. During the walk-out phase, the periapsis altitude is gradually increased outside the atmosphere.

The main constraints on aerobraking used by MOY were:

- **Thermal**: periapsis hear rate corridor, where upper limit was based on maximum temperature of solar arrays (175 °C) with safety margin of 80-100 %, and lower limit based on total aerobraking time
- **Maneuvers**: aerobrake trim maneuvers (ABMs) were used to keep the periapsis in the hear rate corridor; maneuver magnitudes were chosen from a predefined list (updated weekly)
- **Lifetime**: the time needed to reach 300 km apoapsis altitude had to be greater than one day

Figure 2.4 shows the maximum heat rate experienced by MOY during each pass through the atmosphere. One can notice two interesting features:

1. The upper corridor limit was exceeded several times during aerobraking, however another two limits were posed. The 'flight allowable thermal limit' is the actual maximum limit for the solar panels (which is twice as much as the upper corridor limit, due to the 100 % margin, corresponding to a safety factor of 2), and the 'immediate action limit' corresponds to the threshold where the ground center would have commanded, an altitude raise for the next pass.
2. Heat rates during the endgame and walk-out phases became more strict because of the different strategies applied, i.e., cumulative heating and orbit lifetime limitation.

Mars Odyssey was also equipped with a periapse time estimator (PTE) (Johnson and Willcockson, 2003). The role of PTE was to be a technology demonstrator, to see if aerobraking could be partially automized. In fact, the PTE was used to correct the successive atmospheric sweep, based on the periapsis time error computed during the previous pass. The timing error was computed by analyzing the acceleration data collected by the onboard inertial measurement unit (IMU). The correction for the successive sweep was simply to shift the predefined maneuvers in

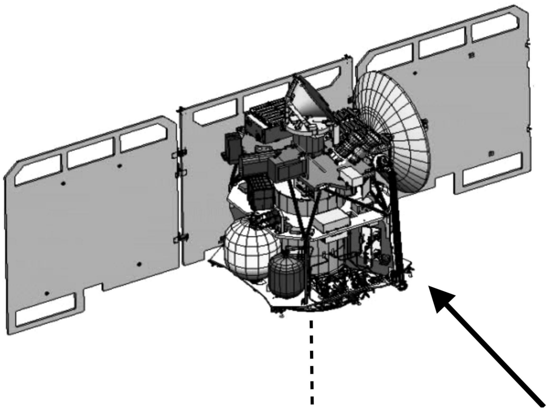


Figure 2.3: *Mars Odyssey (adapted from Spencer and Tolson, 2007).*

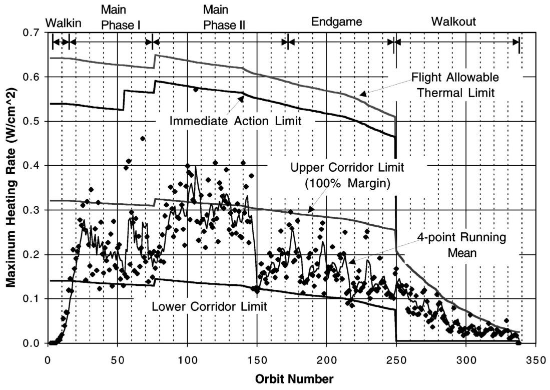


Figure 2.4: *Maximum heat rate during each atmosphere sweep, for the aerobraking phase of 2001 Mars Odyssey (Smith and Bell, 2005).*

time, by the error computed by the PTE. Despite its simplicity, this algorithm proved to be quite useful, especially in reducing the workload for the ground station personnel.

Mars Reconnaissance Orbiter (Long et al., 2008)

Mars Reconnaissance Orbiter used aerobraking in 2006 to reduce its apoapsis altitude. Aerobraking maneuvers (ABMs) were used for three main reasons in particular: increase or decrease atmospheric drag (i.e., increasing or decreasing periapsis altitude), collision avoidance with other Mars-orbiting spacecraft and for lifetime maintenance. Aerobraking was divided into four phases: walk-in, main, walk-out and transition. In total, a ΔV of 44 m s^{-1} was used for aerobraking, spread over corridor control and and collision avoidance.

The walk-in phase was used to gradually lower the periapsis altitude into the Martial atmosphere, and the main phase to control the periapsis altitude, within the corridor control.

During walk-out, ABMs were mainly carried out to avoid collisions and maintain lifetime above the two day limit. Finally, the *transition* phase, initiated at an apoapsis altitude of 485 km, was used to position the spacecraft on the final scientific orbit.

Figure 2.5 shows the configuration of MRO during aerobraking, with a total surface area of 37.5 m^2. The configuration was kept the same during the whole aerobraking phase, to avoid entering the atmosphere with the wrong configuration.

Orbit determination was carried out with the use of deep space network (DSN) tracking and Double-Precision Orbit Determination Program. Daily navigation operations included orbit determination and maneuver analysis. The aerobraking trim maneuvers were chosen from a pre-defined list (20 possible ΔV magnitudes ranging from 0.03 to 4.2 m s^{-1}), to allow for quick selection and execution of the maneuvers. One of the main requirements for navigation was for the periapsis time error to be within 225 s. A more advanced PTE algorithm used IMU data to compute periapsis timing errors and the actual drag ΔV (used in turn to compute actual orbital period change), onboard. These two values were then used to update the timing sequence of the subsequent pass.

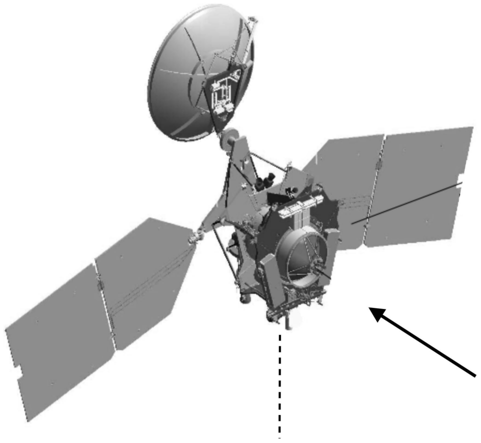
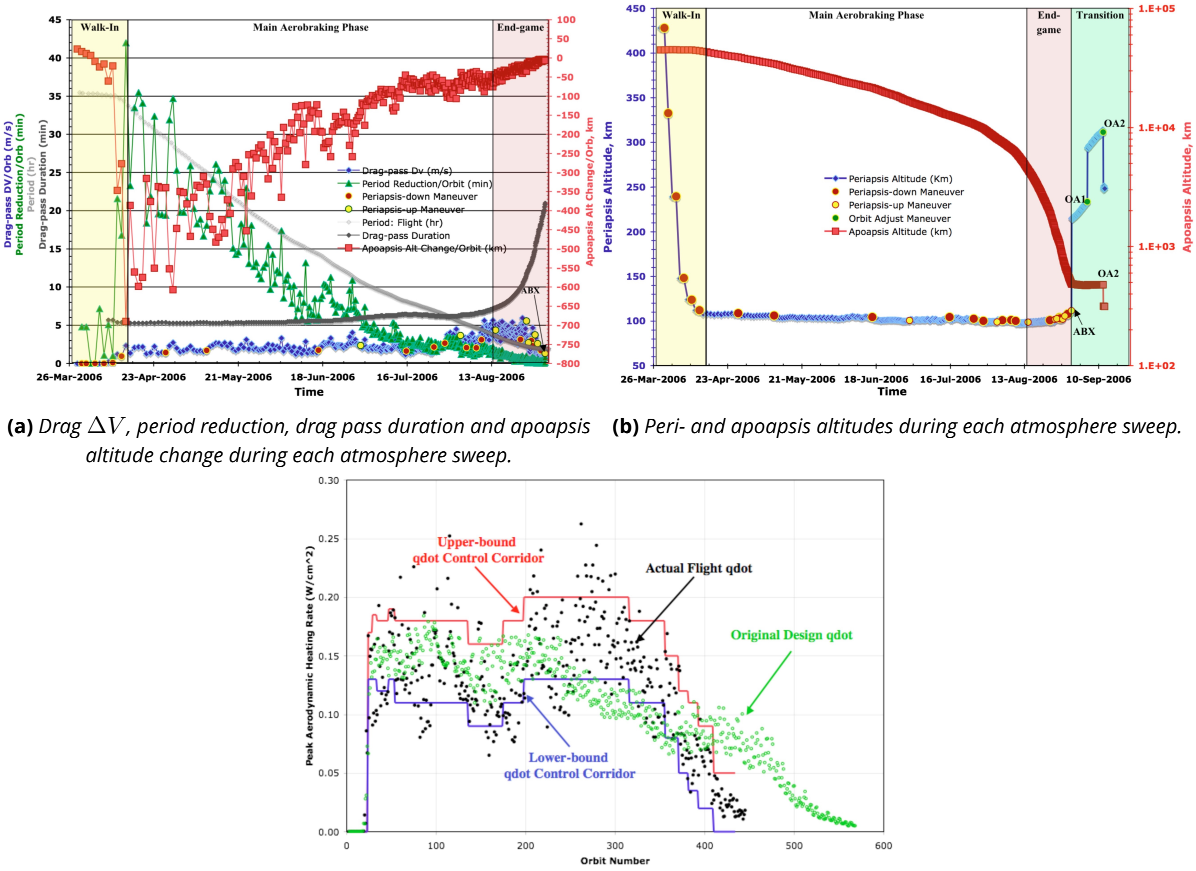


Figure 2.5: *Mars Reconnaissance Orbiter (adapted from Spencer and Tolson, 2007).*

The trim maneuvers, or ABMs were divided into four categories:

- **Nominal corridor control**: used to maintain the periapsis inside the defined corridor; this is the only category that was actually used
- **Immediate action**: used in case either the heat rate overcame the immediate action threshold, or in case of a dust storm prediction; in case of this maneuver, the periapsis altitude would have been risen outside the atmosphere
- **Manual pop-up**: commanded from the ground, in case of anomalies
- **Autonomous pop-up**: commanded by the onboard computer (OBC) in case of safe mode



(a) *Drag ΔV, period reduction, drag pass duration and apoapsis altitude change during each atmosphere sweep.*

(b) *Peri- and apoapsis altitudes during each atmosphere sweep.*

(c) *Maximum heat rate during each atmosphere sweep.*

Figure 2.6: *Details on aerobraking phase of Mars Reconnaissance Orbiter (Long et al., 2008).*

Figure 2.6a shows the behavior of many parameters as a function of time. At the beginning of main phase, the changes in apoapsis altitude (red line) and orbital period (green line) were very large. However, one can see that the variability of these values is also extreme. This is because, during this phase, aerobraking is very sensitive to atmospheric fluctuations. Another interesting parameter is the actual drag-pass ΔV (blue line). At first, the average drag effect is of $2\,\mathrm{m\,s^{-1}}$, and has a very large effect on the orbit, whereas its value during the endgame (or walk-out) reaches up to $6\,\mathrm{m\,s^{-1}}$, but its effect is considerably smaller. From Fig. 2.6b, one can also see that the number of apoapsis ABMs is limited. This is because the only fluctuations that affect the periapsis are gravitational in nature (spherical harmonics and third body). In fact, one can see from the same figure, that the periapsis value remained almost constant at 100 km during the whole duration.

Finally, Fig. 2.6c, shows the maximum heat rate experienced during each atmosphere pass. The most interesting characteristic is the high variability of the heat rate corridor constraints. In fact, these constraints were adjusted weekly, based on the observed atmosphere uncertainty and to account for lifetime constraints.

Venus Express (Val Serra et al., 2011)

The planned aerobraking configuration for VEX is shown in Fig. 2.7. Here, one can see that, as usual, the solar array are positioned perpendicular to the flow. Note that in all cases studied so far, the actual solar cells are facing the

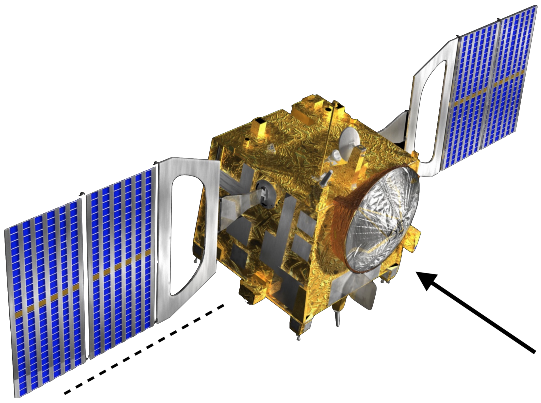


Figure 2.7: *Venus Express (adapted from Svedhem, 2015). Note that during aerobraking, the main engine (bottom side) was facing the flow.*

opposite direction (to prevent damage). Aerodynamic stability is achieved, due to the position of the CP, w.r.t. the CM. Aerodynamic stability is very important, since one does not want to continuously compensate for the drag torques generated during aerobraking. In particular, the pointing accuracy was relaxed by ± 15 deg during the atmospheric sweeps, to make sure that the attitude control subsystem would not try to balance out the torques. Thus, during aerobraking, the satellite was left to oscillate about the stable zero angle-of-attack (AOA) position.

The main constraint during aerobraking is the heating of the insulating material, covering the SC and the solar array. However, since the solar flux at Venus is much larger than at Mars, the solar panels experience high temperatures even when not performing aerobraking. The orbit configuration was therefore, also an important consideration.

To control the spacecraft, two options were investigated: rotating and fixed. A rotating guidance would have the spacecraft always aligned with the current velocity vector, whereas a fixed guidance, would have the spacecraft always aligned with the velocity vector at apoapsis. Despite the relatively large error in attitude when entering the atmosphere, the second guidance was preferred.

Aerobraking was divided into five phases (Damiani et al., 2015): initiation, walk-in, main, walk-out and termination. During initiation, the altitude was slowly lowered, and during walk-in, the atmosphere variabilities and related heating effects were examined, to see if they were as expected. Furthermore, the aerodynamic torques would be used as to calibrate the CP-CM alignment. During walk-out, ESA also decided to use a minimum lifetime as driving constraint. Finally, with termination, the pericenter altitude was increased above the atmosphere.

Orbit determination was performed with the ESA deep space antennae, using two-way Doppler (Damiani et al., 2015). The atmospheric model used was an exponential model. After each atmospheric passage, the model was adjusted for the observed discrepancies, and a new scale height was used for the successive sweep. In particular, high variability of the scale height was noticed at high altitudes, due to the small effects of density. This analysis was also used to check that the dynamic pressure did not exceed the set threshold.

Trace Gas Orbiter (Lyons, 2012)

Trace Gas Orbiter, shown in Fig. 2.8, is a spacecraft part of the ESA/Roscosmos ExoMars mission[4]. After the detachment of the Schiaparelli lander, TGO performed a Mars orbit injection (MOI) and soon after initiated an aerobraking maneuver. Aerobraking was suspended in July and August due to a solar conjunction (Mars, as seen from Earth, is hidden by the Sun). Aerobraking resumed in September and continued until March 2018.

Key constraints for the mission are: aerodynamic heating of 2800 W m^{-2} and integrated heating of 500 kJ m^{-2} (although the actual corridor constraints are set to half their values, for safety), 3 h eclipse time and two day lifetime. The limit on integrated heating became more important during the end of the aerobraking main phase, since more time was spent in the atmosphere, whereas the lifetime limit prevailed in the walk-out phase, when the spacecraft is close to spiraling toward the surface.

[4] European Space Agency: Aerobraking: Back to the Future – https://goo.gl/jAqCZg [last accessed on October 25, 2017]

[5] European Space Agecy: ExoMars 2016: Trace Gas Orbiter and Schiaparelli – https://goo.gl/Mk7Mst [last accessed on October 24, 2017]

Table 2.1: *Specifications of the past and present aerobraking missions. Numbers in parenthesis represent backups. References are listed at the beginning of the chapter.*

	MAG[a]	**MGS**[c]	**MOY**[d]	**MRO**[e]	**VEX**[h]	**TGO**[j]
			General			
Dry Mass [kg]	1035	677	380	1031[f]	675	3755 (wet)
Surface [m^2]	23.3	12.8	11.0	37.5[f]	7.6	24.0
Drag Coeff. [–]	—	—	—	2.2[f]	2.2	—
			Attitude Control			
Sensors	1 star scanner, 4 gyroscopes	star sensor, horizon sensor, sun sensors, IMUs	1 star camera, 1 sun sensor, 1 IMU	8(+8) sun sensors, 1(+1) star trackers, 1(+1) IMU[g]	2 star trackers, 2 IMUs, 2 sun sensors[i]	NA
Actuators	3 orthogonal reaction wheels, thruster system	reaction wheels, thruster system	3(+1) reaction wheels, thruster system	3(+1) reaction wheels, 8 reaction control thrusters[g]	4 reaction wheels, 2 solar array drive mechanisms, 4(+4) thrusters[i]	reactions wheels, 20 thrusters
			Orbit Control			
Sensors	Doppler (GS)	Doppler (GS)	Doppler (GS), PTE	Doppler (GS), PTE	Doppler (GS)[i]	Doppler (GS)
Actuators	main engine, thruster system	main engine, thruster system	main engine, thruster system	6 high thrust, 6 medium thrust, 8 low thrust engines[g]	main engine, 4(+4) thrusters[i]	main engine, 20 thrusters
			Aerobraking			
Year	1993	1997-98	2001	2006	2015	2017-18
Planet	Venus	Mars	Mars	Mars	Venus	Mars
Duration [d]	70[b]	167	76	154	24	approx. 300
Periapsis [km]	135 to 145[b]	115 to 135	90 to 110	90 to 110	130 to 140	105 to 125
Apoapsis [km]	8450 to 541[b]	56 000 to 400	26 200 to 540	45 000 to 320	67 000 to 37 000	34 000 to 400
ΔV *saved* [km s^{-1}]	1.22[b]	1.22	1.09	1.19	—	—
Main Corridor	dynamic pressure[b] [N m^{-2}]	dynamic pressure [N m^{-2}]	heat rate [W m^{-2}]	heat rate [W m^{-2}]	dynamic pressure [N m^{-2}]	NA
Constraints	maneuvers, star pairs, DSN scheduling, atmospheric uncertainties[b]	solar array damage, final orbit	maneuvers, lifetime, daily and final descending node times	atmospheric uncertainties, final orbit, lifetime, collision avoidance	atmospheric uncertainties, power, Sun third-body	eclipse time, lifetime

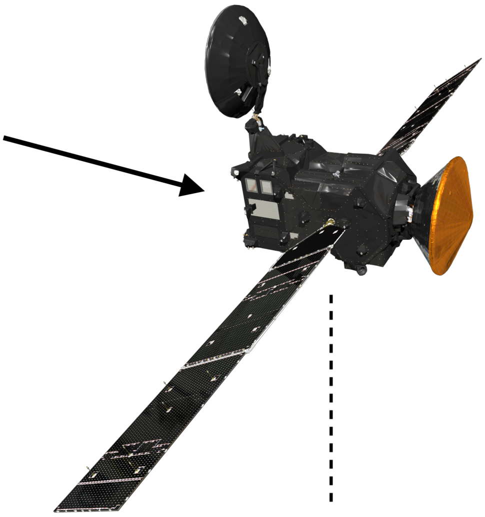


Figure 2.8: *Trace Gas Orbiter (adapted from* [5]*). Note that during aerobraking, the antenna is folded. Also, the Schiaparelli lander (in orange) was ejected before aerobraking started.*

2.2 | Future Missions

No future missions are at present, planned to use aerobraking (AB). However, intense research has been carried out with the main focus being to achieve autonomous aerobraking (AAB).

As pointed out by Spencer and Tolson (2007), one of the main disadvantages of aerobraking is the cost associated with its planning and execution. Before launch, aerobraking requires extensive investments for the planning and development of the aerobraking phases and its software. Furthermore, during the braking part of the orbit, continuous coverage by deep space antennae is needed to accurately compute the position and effect of drag on the spacecraft. Not to mention the operational costs of having a team of experts working 24/7 on the tracking of the SC, analysis of aerobraking results and estimation of atmospheric properties. The purpose of AAB would then be to reduce the workload at the ground station (GS), by giving more tasks to the OBC.

This subsection is divided into two parts. First, research performed by the NASA Langley Research Center is shown, then a few papers published as part of the ESA study on 'Robust Autonomous Aerobraking Strategies' will be summarized.

2.2.1 | NASA Strategies

Recent NASA studies are focusing on the development of techniques for *ephemeris* and *atmospheric* density estimation, *maneuver* calculations and thermal modeling.

It soon became clear during the literature study, that development and/or implementation of a thermal model would require extensive analysis and insight into the aerothermodynamics of the spacecraft and material properties. Thus, this part of the research was excluded from this preliminary analysis. One will find below, a summary of the functions of the three units typeset in italic above (Cianciolo et al., 2013):

- **Ephemeris**
  An ephemeris estimator (EE) can be used to accurately model the spacecraft position and velocity over time. Instead of using continuous GS coverage, an EE would use accurate dynamics, filtering algorithms and IMU data, to propagate an initial accurate knowledge (from GS), in real time, for up to one week, when new accurate GS data is received.

- **Atmosphere**
  An atmosphere estimator (AE) has the role of using data from previous atmospheric sweeps, to estimate what the conditions of the next passage will be, based on an ever-changing exponential model. This estimator uses arithmetic mean and standard deviation of the previous seven orbits, to compute the uncertainty. The mean is used, instead of e.g. median, because it is affected by large offsets in density. In fact, the sole purpose of the AE is to know the variability (standard deviation) of the atmospheric model.

- **Maneuver**
  A maneuver estimator (ME) is used to compute the magnitude and direction of the maneuver at apocenter,

such that the *temperature/heat rate/dynamic pressure/density/altitude* corridor is respected. In particular, the ME also considers the result of the AE, by computing the heat rate (or other surrogate parameter) corresponding to a multiple of the standard deviation of the density model. This multiplication factor ζ is specified by the user and can either be used or ignored. If triggered, the $\zeta\sigma$ heat rate is compared to the 'immediate action limit', and in case this limit is overcome, the $\zeta\sigma$ density is used for computation of the maneuver.

The reason why in the *maneuver* unit the corridor is defined with multiple parameters, is because the decision generally depends on the application. As was shown in Table 2.1, in past missions dynamic pressure:

$$\bar{q} = \frac{1}{2}\rho V^2 \tag{2.1}$$

and an aerodynamic heat rate indicator:

$$q_{\text{aero}} = \frac{1}{2}\rho V^3 \tag{2.2}$$

have been used to describe the corridor constraints, in the main phase. In the endgame, walk-out and termination phases, integrated heating:

$$\Delta Q = \int q \, \mathrm{d}t \tag{2.3}$$

was generally used. The upper boundary was derived from the maximum temperature of the SC, particularly of the solar array. The lower boundary, on the other hand, was derived from limitations on the total aerobraking duration and final orbit configuration. However, new parameters could be used for future missions, such as the temperature of critical areas of the SC. Since most processing would be done onboard, thermistor sensors could be read in real time, and used to compute the next pericenter altitude, considering the atmosphere and thermal models onboard.

2.2.2 | ESA Strategies

The work done by the ESA contractors, can be divided into four separate categories: navigation, corridor control, atmosphere estimation and guidance logic. These are explained below.

- **Navigation**
  Similarly to the approach used by NASA, the contact with GS is reduced by using onboard ephemeris estimation. The method described by Chabot et al. (2013) explains how only propagation of semi-major axis a and eccentricity e is done by the spacecraft. This is because a and e are highly affected by each drag pass, and thus the measured accelerations need to be used for proper modeling of their behavior. On the other hand, elements i, ω and Ω have a higher dependence on gravitational perturbations and can thus be predicted on the ground and periodically (every seven days at the beginning of aerobraking and every three days at the end) sent to the SC. This approach would reduce the computational load for the SC.

- **Corridor Control** (Cichocki et al., 2011, 2012)
  Two methods for corridor control were investigated, both using surrogate variables: 1D corridor and 2D corridor. The 1D corridor is solely based on heat flux: the upper limit of the corridor is constrained by the maximum allowable heat rate. On the other hand, the 2D corridor control is dependent on both heat rate and integrated heat. Despite its higher complexity, the 2D approach has two advantages: it consists of a fixed corridor and it results in a slightly more efficient aerobraking. Thus, this second method is deemed preferable by the authors.

- **Atmosphere Estimation** (Cichocki et al., 2012)
  First of all, the SC computes an atmospheric safety factor (ASF) based on the measured (via heat flux sensor) and predicted heat flux. A three-day moving average of the computed ASFs is used as multiplying factor for the predicted atmospheric density. The onboard atmosphere model would then be a given by a simple exponential function.

- **Guidance Logic** (Cichocki et al., 2012)
  The guidance logic is such that aerobraking is divided into control intervals, which last a few days, depending on atmosphere variability. During each control interval, the control variable is estimated for all pericenter passages. In case one or more corridor violations are detected, an apocenter maneuver is scheduled such that all passes in the current interval are within the limits. Furthermore, before each pericenter pass, another prediction is made, such that sudden changes in atmospheric conditions can be counteracted, as well.

2.3 | Reference Mission and Requirements

The reference mission is the one that will be used to compare the results obtained. Since the aim of this thesis is the achievement of a robust and autonomous aeroassisted maneuver, a good reference mission should be one that also aimed for autonomy. Given that no mission has ever performed an autonomous aeroassisted maneuver yet, the mission chosen is the one for which 1) most data is available and 2) had the highest degree of autonomy. Given that TGO is (at the time of writing) still performing its aerobraking phase, and that not enough information is available, the most reasonable choice is MRO. Hence, MRO is taken as reference, and the research carried out will focus mainly on the Martian environment.

Based on the shape and physical characteristics of MRO, Fig. 2.9 was developed. This image shows the discretized spacecraft model used for the aerodynamic and physical analysis carried out in this thesis.

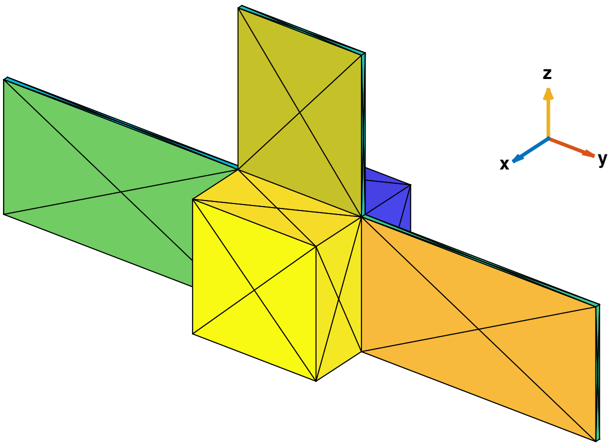


Figure 2.9: *Discretization of the design of Mars Reconnaissance Orbiter, used as reference for the various spacecraft analyses carried out in this work.*

Table 2.2 lists the requirements on the aerobraking GNC system to be designed during this thesis work. These requirements were derived from the references cited in the pages above. In particular, superscripts are used to represent references, and they correspond to the ones used for Table 2.1, to which the following are added:

(k) (Lyons, 2012) (l) (Murri, 2013) (m) (Groves, 2013) (n) (Lee et al., 2003)

The reader will notice that some of numerical values in the the requirements of Table 2.2 have been replaced with TBD, i.e., to be defined or to be determined. This is mainly for two reasons:

- The results of this thesis will be used to **define** such requirements.
- They will be **determined** by experimenting with different settings.

Take REQ-N-7, for instance. This requirement fits in the first category. Its value will be defined based on the results obtained in Chapter 12. REQ-N-3, on the other hand, will be determined during the design of the navigation system, as one looks for the best refresh rate of state estimation to find the compromise between low navigation errors and computation time.

Table 2.2: *Requirements on the aerobraking phase and on the guidance, navigation and control systems, for a spacecraft orbiting Mars.*

ID	Requirement	Rationale
	SPACECRAFT DESIGN	
REQ-SC-1	The spacecraft shall have a dry mass of 1000 kg.	Based on MRO mass properties[f].
REQ-SC-2	The spacecraft shall have a mass moment of inertia of $\mathrm{diag}\,(5750, 1215, 5210)$ kg m^2 around its body axes.	Derived in Appendix B, based on MRO shape and mass properties[f].
REQ-SC-3	The spacecraft shall carry 50 kg of propellant for aerobraking.	Based on amount of propellant used by MRO during aerobraking[e].
REQ-SC-4	The spacecraft shall be provided with eight vectorial thrusters for attitude control about the three body axes.	Based on MRO attitude determination and control subsystem design[g].
REQ-SC-5	The spacecraft shall be provided with one inertial measurement unit.	See above.
REQ-SC-6	The spacecraft shall be provided with one reaction wheel per body axis.	See above.
REQ-SC-7	The spacecraft shall be provided with two orthogonal star trackers.	See above.
REQ-SC-8	The aerodynamic coefficients shall be available for the rarefied and transition regimes, for an angle of attack range of −30 to +30 deg.	Based on expected angle of attack range.
	AEROBRAKING	
REQ-AB-1	Aerobraking shall last no longer than 180 days.	Based on duration of MRO aerobraking phase[e].
REQ-AB-2	Aerobraking shall reduce the apoapsis altitude to 320 km $\pm$ TBD m.	Based on MRO final orbit configuration[e].
REQ-AB-3	The periapsis post-aerobraking shall be 255 km $\pm$ TBD m.	Based on MRO final orbit configuration[e].
REQ-AB-4	The spacecraft shall resist heat flux of up to 2800 W m^{-2}.	Based on TGO heat flux constraint[k].
REQ-AB-5	The spacecraft shall resist integrated heating of up to 500 kJ m^{-2}.	Based on TGO heat load constraint[k].
REQ-AB-6	The minimum dynamic pressure experienced during an atmospheric pass shall be 0.19 N m^{-2}.	Based on MRO dynamic pressure constraint[f].
REQ-AB-7	The spacecraft lifetime during aerobraking shall be larger than 2 days.	Based on MRO lifetime constraint[e].
REQ-AB-8	The atmospheric interface altitude shall be 200 km.	Altitude above which the average density is lower than 10^{-13} kg m^{-3}.

Table 2.2: *(Continuation) Requirements on the aerobraking phase and on the guidance, navigation and control systems, for a spacecraft orbiting Mars.*

ID	Requirement	Rationale
	GUIDANCE	
REQ-G-1	The guidance system shall take into account a 90 % atmospheric variability.	Based on MRO aerobraking requirements[(e)].
REQ-G-2	The guidance system shall be able to take into account possible dust storm activity.	See above.
REQ-G-3	Maneuvers to raise and lower periapsis altitude shall be performed at apoapsis.	See above.
REQ-G-4	The guidance system shall be capable of autonomously performing a contingency periapsis raise maneuver.	See above.
	NAVIGATION	
REQ-N-1	The navigation system shall be able to operate autonomously for at least 1 day.	Based on autonomous aerobraking requirements[(l)].
REQ-N-2	The navigation system shall be able to estimate periapsis time with an uncertainty of at most 225 s.	Based on MRO aerobraking requirements[(e)].
REQ-N-3	The state estimation shall run at a rate of TBD Hz.	—
REQ-N-4	The inertial measurement unit shall have a sampling rate of TBD Hz.	Based on average space-grade IMU characteristics[(m)].
REQ-N-5	The inertial measurement unit shall have an accuracy in translational acceleration of $2 \times 10^{-4}\,\mathrm{m\,s^{-2}\,Hz^{-0.5}}$ (3σ).	See above.
REQ-N-6	The inertial measurement unit shall have an accuracy in rotational acceleration of $3 \times 10^{-7}\,\mathrm{rad\,s^{-1}\,Hz^{-0.5}}$ (3σ).	See above.
REQ-N-7	The position error during autonomous navigation shall be less than TBD m.	—
REQ-N-8	The translational velocity error during autonomous navigation shall be less than TBD $\mathrm{m\,s^{-1}}$.	—
REQ-N-9	The attitude error during autonomous navigation shall be less than TBD rad.	—

Table 2.2: *(Continuation) Requirements on the aerobraking phase and on the guidance, navigation and control systems, for a spacecraft orbiting Mars.*

ID	Requirement	Rationale
	CONTROL	
REQ-C-1	The spacecraft shall have attitude control on all three axes.	Based on MRO aerobraking requirements[e].
REQ-C-2	The spacecraft longitudinal axis shall always be aligned with the periapsis airspeed vector.	To maximize drag, minimize lift and keep a steady configuration.
REQ-C-3	Each attitude thruster shall provide a maximum thrust of 22 N.	Based on MRO attitude determination and control subsystem design[g].
REQ-C-4	The star trackers shall have a sampling rate of TBD Hz.	—
REQ-C-5	The star trackers shall have an accuracy in attitude determination of 20 arcsec (3σ).	Based on MRO attitude knowledge requirements[n].
REQ-C-6	Each reaction wheel shall provide a maximum angular momentum of at least TBD N m s.	—

Page Intentionally Left Blank

3
Mathematical Background

The mathematical background to flight dynamics consists of the description of reference and coordinate systems. Their main roles are to define how motion is described (for the latter) and w.r.t. what reference (for the former). Choice of coordinate system may seem trivial, but different state variables have different advantages. For instance, it will be shown later, that there is a set of state variables that has the potential of reducing significantly the computation time needed for propagation of the state of a satellite in certain conditions. Other advantages or disadvantages, might arise due to singularities in the corresponding equations of motion or complexity of the related expressions. Therefore the need of this chapter: to figure out which frames and coordinates are needed or best suited for an aerobraking mission.

This chapter is structured as follows. In Section 3.1, one will find a short introduction to Keplerian orbits, the simplest form of orbital motion. This section is meant to give some insight in the terminology of classifications of orbits used throughout this chapter and the report. Then, Section 3.2 will introduce the reference frames needed in this report for the description of an aerobraking mission. This will be followed in Section 3.3 by the introduction of state variables, which will be used to describe the position, velocity and orientation of the spacecraft. Conversion between different reference and coordinate frames are deemed of less interest for the main body of the report, and will be detailed in Appendix A.

3.1 | Keplerian Orbits

It was Johannes Kepler to first observe that planets move along paths that could be represented as elliptical trajectories, with the Sun at one of the foci. This is now known as Kepler's first law, and its expression is (Curtis, 2013):

$$r = \frac{a\left(1 - e^2\right)}{1 + e\cos\vartheta} \tag{3.1}$$

Equation (3.1) describes the distance r between two objects as one orbits the other, by using three main parameters. These are a, the semi-major axis, e, the eccentricity and ϑ, the true anomaly, and are enough to describe the orbit in two dimensions. To fully constrain the orbit, however, the three-dimensional aspect of the motion has to be described, as well. To do this, another three parameters are introduced. These are the inclination i, the argument of periapsis ω and the right ascension of the ascending node Ω. The set of these six parameters is called Keplerian elements, and they fully describe the motion of a body in three-dimensional space, w.r.t. a fixed reference. The reader is referred to Fig. 3.1 for an illustration of these elements.

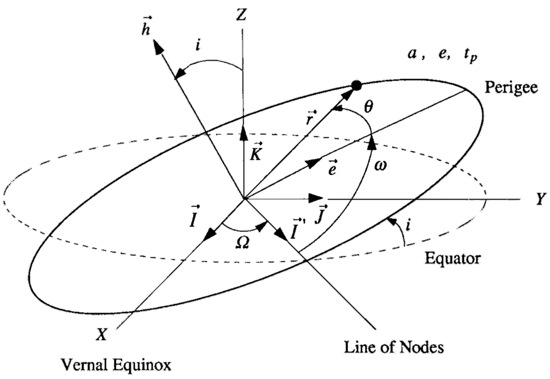


Figure 3.1: *Geometrical representation of an elliptical orbit and definition of Keplerian elements (Wie, 2008).*

As it turns out, the shape of the orbit can differ from the elliptical shape Kepler noticed for the planets. In particular, the possible shapes are four: circular ($e = 0$), elliptical ($0 < e < 1$), parabolic ($e = 1$) and hyperbolic ($e > 1$). For the purpose of this analysis, only elliptical orbits will be described, since one can consider circular (and parabolic) orbits special cases of elliptical orbits, which can only be achieved for an instant, due to their strict definitions.

3.2 | Reference Frames

To express the position and velocity of an object, a fixed and clearly defined reference needs to be established. These references are called reference frames, and they exist in many different forms, based on the application and context. The nine frames used in this research are the described in this section.

Note that each frame is orthogonal and right-handed, and that (if not mentioned) the $\hat{\mathbf{y}}$ axis, always completes the right-handed system. Reference for this section is Mooij (1994), unless otherwise stated.

- **International Celestial Reference Frame** or $\mathcal{F}_{\mathcal{J}}$ (Ma and Feissel, 1997)
  This frame was defined by the International Earth Rotation and Reference Systems Service (IERS), based on the adopted locations of 295 extragalactic radio sources. It is an inertial frame centered at the solar system barycenter, whose alignment coincides almost perfectly with the J2000 frame. In most applications, the two are in fact considered as equal. The $\hat{\mathbf{x}}_{\mathcal{J}}$ axis points toward the vernal equinox Υ and $\hat{\mathbf{z}}_{\mathcal{J}}$ is perpendicular to the equatorial plane (i.e., Earth's equator). Figure 3.1 shows this frame with axes named X, Y and Z.
- **Inertial Planetocentric Reference Frame** or $\mathcal{F}_{\mathcal{I}}$
  This is a frame centered at the center of mass (CM) of the planet that is being orbited. Its $\hat{\mathbf{x}}_{\mathcal{I}}$ axis points toward the zero-latitude, zero-longitude point at a reference time (for Earth this is the J2000 epoch[1]) and the $\hat{\mathbf{z}}_{\mathcal{I}}$ axis is parallel to the rotation axis. The need of this reference frame arises to describe the position of the spacecraft w.r.t. the planet.
- **Orbit Reference Frame** or $\mathcal{F}_{\mathcal{O}}$ (Vittaldev, 2010)
  This frame results from rotating the $\mathcal{I}$-frame about the line of nodes of the spacecraft orbit (line of intersection between the orbital plane and the $\hat{\mathbf{x}}_{\mathcal{I}} - \hat{\mathbf{y}}_{\mathcal{I}}$ plane) by the inclination of the orbit, such that the resulting $\hat{\mathbf{x}}_{\mathcal{O}}$ axis is still pointing toward Υ and $\hat{\mathbf{z}}_{\mathcal{O}}$ is perpendicular to the orbital plane. This frame is used for the definition of the unified state model, discussed in Subsection 3.3.3.
- **Local Orbital Reference Frame** or $\mathcal{F}_{\mathcal{L}}$ (Vittaldev, 2010)
  This frame is also centered at the CM of the planet, but it is a rotating frame, whose $\hat{\mathbf{x}}_{\mathcal{L}}$ axis is always pointing toward the spacecraft. The $\hat{\mathbf{z}}_{\mathcal{L}}$ axis is still perpendicular to the orbital plane. This frame is also used for the definition of the unified state model, discussed in Subsection 3.3.3.
- **Rotating Planetocentric Reference Frame** or $\mathcal{F}_{\mathcal{R}}$
  Similar to the inertial planetocentric frame, with the exception that the $\hat{\mathbf{x}}_{\mathcal{R}}$ axis is always pointing toward the zero-latitude, zero-longitude point. This frame is used to define the position w.r.t. the surface of the planet.
- **Vertical Reference Frame** or $\mathcal{F}_{\mathcal{V}}$
  The vertical reference frame is centered at the CM of the spacecraft, with its $\hat{\mathbf{z}}_{\mathcal{V}}$ axis pointing along the radial vector, but in opposite direction, and the $\hat{\mathbf{x}}_{\mathcal{V}}$ axis towards the north direction of the planet. This frame is used to define the local horizon and local vertical (LHLV).
- **Trajectory Reference Frame** or $\mathcal{F}_{\mathcal{T}}$
  This frame is also centered at the CM of the spacecraft, but its $\hat{\mathbf{x}}_{\mathcal{T}}$ axis points along the velocity vector (relative to the atmosphere) and its $\hat{\mathbf{z}}_{\mathcal{T}}$ is in the orbital plane, pointing toward the surface of the planet. This frame is used to define velocity in spherical coordinates.
- **Aerodynamic Reference Frame** or $\mathcal{F}_{\mathcal{A}}$
  Also centered at the CM of the spacecraft, with the $\hat{\mathbf{x}}_{\mathcal{A}}$ axis aligned with the velocity vector (relative to the atmosphere), and the $\hat{\mathbf{z}}_{\mathcal{A}}$ axis parallel but opposite in direction to the lift vector. Note that the relation $\hat{\mathbf{x}}_{\mathcal{A}} \equiv \hat{\mathbf{x}}_{\mathcal{T}}$ is always true, but the other axes coincide only when specific conditions are met. This frame is used to define the aerodynamics angles.
- **Body-fixed Reference Frame** or $\mathcal{F}_{\mathcal{B}}$
  The body-fixed reference frame is centered at the CM of the spacecraft. The orientation of the axes is usually defined such that the $\hat{\mathbf{x}}_{\mathcal{B}}$ and $\hat{\mathbf{z}}_{\mathcal{B}}$ axes lie in the plane of symmetry. Definition of this frame for the spacecraft used in this thesis will be given in Appendix B.

To convert between the various reference frames listed above, one can use the expressions shown in Appendix A.1.

[1] The epoch J2000 corresponds to January 1, 2000 at 11:58:55.816 UTC.

3.3 | State Variables

As the name suggests, state variables are used to define the state of a system. In particular, they are used to define the position, velocity, attitude, rotational rates, etcetera, of dynamical systems. In this section, the state variables used in the thesis work will be introduced, together with a brief description. Also, one will find Appendix A.2 of particular interest, in case conversion between the various state variables introduced here, is needed.

3.3.1 | Attitude

Attitude describes the orientation of a reference frame w.r.t. another frame. For this report, four sets of parameters are considered, and they are described below.

Direction Cosine Matrix (Schaub and Junkins, 2002) The orientation of a reference frame w.r.t. another is, in general, described in terms of a direction cosine matrix (DCM). A DCM is denoted by $\mathbf{C}$, and it expresses the set of basis vectors of the second reference frame, in terms of components of the set of basis vectors of the first reference frame.

An interesting property of DCMs is their orthonormality. An orthonormal matrix is one whose inverse is its transpose, i.e., $\mathbf{C}^{-1} = \mathbf{C}^{\mathrm{T}}$ and whose determinant is ± 1 (note that for right-handed reference frames, $\det(\mathbf{C}) = 1$). Thus, the inverse of any DCM can be expressed in two ways: by the transpose of the matrix itself, or by the transformation about the same axis with negative angle: $\mathbf{C}_{\hat{\mathbf{x}}}^{-1}(\xi) = \mathbf{C}_{\hat{\mathbf{x}}}^{\mathrm{T}}(\xi) = \mathbf{C}_{\hat{\mathbf{x}}}(-\xi)$. Note that the subscript of $\mathbf{C}$ represents the axis around which the rotation is carried out and ξ the rotation angle.

The main benefit of a DCM is that it can be used to easily transform one vector from one frame to another:

$$\mathbf{x}|_{\mathcal{B}} = \mathbf{C}_{\mathcal{B}}^{\mathcal{A}}\, \mathbf{x}|_{\mathcal{A}} \tag{3.2}$$

Direction cosine matrices come, however, with a major drawback. Being 3×3 matrices, they contain nine elements, whereas the minimum number of parameters needed to fix the orientation of one frame w.r.t. another is 3 (Schaub and Junkins, 2002). Thus, six extra parameters need to be computed when using DCMs. This is why they are rarely used for attitude representation. In this report, DCMs are in fact only used as intermediate steps for transformations between various other attitude coordinates.

Euler Eigenaxis (Schaub and Junkins, 2002) According to Euler's eigenaxis rotation theorem, a rigid body can be arbitrarily oriented by a rotation about an axis that is fixed and stationary w.r.t. an inertial reference frame. Said axis is called an eigenaxis $\hat{\mathbf{e}}$. A rotation about an eigenaxis by angle ξ can be expressed in terms of DCM by using:

$$\mathbf{C} = \mathbf{I}\cos\xi + (1-\cos\xi)\,\hat{\mathbf{e}}\hat{\mathbf{e}}^{\mathrm{T}} - \hat{\mathbf{e}}^{\mathrm{S}}\sin\xi \tag{3.3a}$$

where $\mathbf{I}$ is the 3×3 identity matrix and $\hat{\mathbf{e}}^{\mathrm{S}}$ the skew-symmetric matrix in terms of $\hat{\mathbf{e}}$, which is defined by:

$$\mathbf{x}^{\mathrm{S}} \triangleq \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \tag{3.3b}$$

Euler Angles (Schaub and Junkins, 2002) Euler angles are used to describe relative orientation of a frame by a set of three successive rotations about the main axes of the body being rotated.

Despite their clear interpretation, Euler angles suffer from significant drawbacks. The parametrization of a DCM in Euler angles is made up of many sine and cosine operators, which can be expensive to compute for a computer. Furthermore, one may need to rotate about multiple sets of Euler angles to achieve the wanted orientation. The DCM changes form based on the selected set of Euler angles: in total there are 12 sets, based on the rotation sequence[2]. Singularities in the equations of motion expressed in Euler angles, arise at different rotation values, depending on the sequence. For a 3-2-1 rotation, for instance, a singularity arises at $\xi_2 = \pm\pi/2$ rad.

Quaternions (Wie, 2008) Quaternions are defined based on Euler's eigenaxis theorem:

$$\eta = \cos\left(\frac{\xi}{2}\right) \tag{3.4a}$$

$$\boldsymbol{\varepsilon} = \hat{\mathbf{e}}\sin\left(\frac{\xi}{2}\right) \tag{3.4b}$$

[2] The 12 rotation sequences are: 1-2-1, 1-2-3, 1-3-1, 1-3-2, 2-1-2, 2-1-3, 2-3-1, 2-3-2, 3-1-2, 3-1-3, 3-2-1, 3-2-3. The numbers 1, 2, and 3 correspond to rotations about axes $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{z}}$.

where η from Eq. (3.4a), represents the real part of the quaternion, and $\boldsymbol{\varepsilon}$ in Eq. (3.4b), the imaginary part. A property of quaternion is that the square of their sum equals unity, i.e.,

$$\eta^2 + \boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\varepsilon} = \eta^2 + \sum_{i=1}^{3} \varepsilon_i^2 = 1\,.$$

This property suggests that quaternions describe a four-dimensional hypersphere, and thus that any rotation is a trajectory on the surface of the three-dimensional sphere. The main limitations of quaternions, are that they are constrained to an eigenaxis rotation with $-\pi \leqslant \xi \leqslant \pi$ and that their magnitude has to equal 1.

In terms of quaternions, the DCM can be expressed as:

$$\mathbf{C} = \mathbf{I}\left(\eta^2 - \boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\varepsilon}\right) + 2\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^{\mathrm{T}} - \boldsymbol{\varepsilon}^{\mathrm{S}} 2\eta \tag{3.5}$$

Modified Rodrigues Parameters (Vittaldev, 2010) The modified Rodrigues parameters (MRP) try to circumvent the limitation of the quaternions, by still using the Euler eigenaxis rotation as reference, but with a slightly tweaked definition:

$$\boldsymbol{\sigma} = \hat{\mathbf{e}} \tan\left(\frac{\xi}{4}\right) \tag{3.6a}$$

Thus, MRP also present a limitation, namely they experience a singularity at $\xi = \pm 2\pi$, i.e., when $\eta = -1$. However, this can be avoided by introducing the shadow modified Rodrigues parameters (SMRP), which are defined as:

$$\boldsymbol{\sigma}_s = \hat{\mathbf{e}} \tan\left(\frac{\xi - 2\pi}{4}\right) \tag{3.6b}$$

Since, SMRP present a singularity at $\eta = 1$, one can switch from MRP and SMRP when $\eta = 0$, thus when $\boldsymbol{\sigma}_s = -\boldsymbol{\sigma}$. In general, transformation between $\boldsymbol{\sigma}_s$ and $\boldsymbol{\sigma}$ is given by:

$$\boldsymbol{\sigma}_s = -\frac{\boldsymbol{\sigma}}{\sigma^2} \tag{3.7}$$

It is good practice to switch between MRP and SMRP whenever the magnitude of $\boldsymbol{\sigma}$ or $\boldsymbol{\sigma}_s$ goes beyond 1 (although there is no constraint on their magnitude).

A disadvantage of MRP is that one needs to keep track of whether the shadow parameters are being used, since the transformation to and from other attitude representations varies. However, the kinematic equations are the same no matter the shadowing.

By using σ to express the magnitude of the MRP vector, the DCM in terms of MRP can be written as:

$$\mathbf{C} = \mathbf{I} - 4\frac{\left(1 - \sigma^2\right)\boldsymbol{\sigma}^{\mathrm{S}} - 8\left(\boldsymbol{\sigma}^{\mathrm{S}}\right)^2}{\left(1 + \sigma^2\right)^2} \tag{3.8}$$

Exponential Mapping (Vittaldev, 2010) Using once again the Euler eigenaxis $\hat{\mathbf{e}}$, one can define the exponential mapping (EM) $\mathbf{e}$ as:

$$\mathbf{e} = \xi\hat{\mathbf{e}} \tag{3.9a}$$

Similarly to MRP, EM also presents a singularity at $\xi = \pm 2\pi$. To circumvent this, the shadow exponential mapping (SEM) is introduced, and is defined by:

$$\mathbf{e}_s = -\left(2\pi - \xi\right)\hat{\mathbf{e}} \tag{3.9b}$$

An advantage of using (S)EM over (S)MRP is that there is no need to keep track of the type of EM used. Thus, one can simply switch from one to the other whenever $\xi \geqslant \pi$. Transformation between $\mathbf{e}$ and $\mathbf{e}_s$ is given by:

$$\mathbf{e}_s = \left(1 - \frac{2\pi}{\xi}\right)\mathbf{e} \tag{3.10}$$

The DCM in terms of EM is expressed as:

$$\mathbf{C} = \exp\left(-\mathbf{e}^{\mathrm{S}}\right) = \sum_{i=0}^{\infty} \frac{1}{i!}\left(-\mathbf{e}^{\mathrm{S}}\right)^i \tag{3.11}$$

The closed form solution of Eq. (3.11) is given by Eq. (3.3a), where $\xi = ||\mathbf{e}||$ and $\hat{\mathbf{e}} = {}^{\mathbf{e}}/_{\xi}$.

3.3.2 | Angular Rates

Angular rates describe how fast the orientation of a body is changing (if it is at all), thus how quickly a vehicle is rotating w.r.t. another body or reference. Angular rates are usually depicted with $\boldsymbol{\omega}$, and are generally expressed as rotational velocities in [rad s^{-1}] about reference axes of the frames shown in Section 3.2. In this report, angular rates are defined as the rotational velocities of the axes of $\mathcal{F}_\mathcal{B}$ w.r.t. the $\mathcal{I}$-frame.

3.3.3 | Position and Velocity

To define the position and velocity of a spacecraft, at least six parameters are needed. The definition of these parameters can vary, based on the case. For this literature study, four sets of parameters are considered: Keplerian, Cartesian, spherical and the unified state model. The first three are widely known and used, and references for these can be found in virtually any astrodynamics book (see e.g., the books by Curtis 2013, Mooij 1994 and the references used in this section). The unified state model, on the other hand, was only recently brought back to the public's attention by Vittaldev (2010), Vittaldev et al. (2012).

Keplerian Elements Keplerian elements, also referred to as orbital elements, where already introduced in Section 3.1 and shown in Fig. 3.1. For sake of completeness, their definitions are repeated here. These elements are widely used in astrodynamics, to define the position and velocity of an orbiting object w.r.t. the $\mathcal{I}$-frame, because of their very straight forward geometrical interpretation, and their slow variation with time (except for true anomaly).

- *Semi-major Axis* or a, with range $a > 0$ [m]
- *Eccentricity* or e, with range $0 \leqslant e < 1$
- *Inclination* or i, with range $0 \leqslant i \leqslant \pi$ [rad]
- *Argument of Periapsis* or ω, with range $0 \leqslant \omega < 2\pi$ [rad]
- *Right Ascension of the Ascending Node* or Ω, with range $0 \leqslant \Omega < 2\pi$ [rad]
- *True Anomaly* or ϑ, with range $0 \leqslant \vartheta < 2\pi$ [rad]

Cartesian Coordinates Cartesian coordinates are widely used for propagation. This is because of their very simple expression of the equations of motion. Unlike the Keplerian elements, the distinction between position and velocity is clear, since the two are defined separately. In this work, they are only defined in $\mathcal{F}_\mathcal{I}$.

- *Position*: x, y, z [m]
- *Velocity*: $\dot{x}$, $\dot{y}$, $\dot{z}$ [m s^{-1}]

Spherical Coordinates Similarly to the Keplerian elements, spherical coordinates also have a straight forward geometrical interpretation, but where the orbital elements describe shape and orientation of the orbit, the spherical elements describe the current position and velocity (i.e., similarly to the Cartesian coordinates, position and velocity are clearly distinguished), w.r.t. a spherical reference. This last part, makes spherical coordinates particularly interesting for the description of the state of a vehicle w.r.t. a planet, and in particular during atmospheric flight.

- *Latitude* or δ, with range $-\pi/2 \leqslant \delta \leqslant \pi/2$ [rad]
- *Longitude* or τ, with range $-\pi < \tau \leqslant \pi$ [rad]
- *Distance* or r, with range $r \geqslant 0$ [m]
- *Flight-path angle* or γ, with range $-\pi/2 \leqslant \gamma \leqslant \pi/2$ [rad]
- *Heading angle* or χ, with range $-\pi < \chi \leqslant \pi$ [rad]
- *Velocity magnitude* or V, with range $V \geqslant 0$ [m s^{-1}]

The elements describing position (i.e., δ, τ and r) are generally defined in $\mathcal{F}_\mathcal{R}$, whereas the ones describing velocity (i.e., γ, χ and V) are given in $\mathcal{F}_\mathcal{T}$.

Unified State Model The unified state model (USM) describes the orientation of the $\mathcal{L}$-frame w.r.t. the $\mathcal{I}$-frame, and the velocity of the spacecraft. The peculiarity of the USM is that the velocity is given in terms of the components of the velocity hodograph. Moreover, the orientation of the $\mathcal{L}$-frame w.r.t. the $\mathcal{I}$-frame can be expressed in various ways. Vittaldev (2010) describes it in terms of quaternions, modified Rodrigues parameters and exponential mapping.

- *Velocity* [m s^{-1}]:

  - C, velocity component perpendicular to the radius vector (parallel to the orbital plane)
  - $\mathbf{R}$, velocity component along semi-latus rectum direction, projected on the $\hat{\mathbf{x}}_{\mathcal{O}}$ and $\hat{\mathbf{y}}_{\mathcal{O}}$ axes, to obtain R_1 and R_2, respectively

- Depending on the state variables chosen, the orientation can be expressed in:
  - *Quaternions* (USM7): $\boldsymbol{\varepsilon}$ and η
  - *(Shadow) Modified Rodrigues Parameters* (USM6): $\boldsymbol{\sigma}$ or $\boldsymbol{\sigma}_s$
  - *(Shadow) Exponential Mapping* (USMEM): $\mathbf{e}$ or $\mathbf{e}_s$

For a comprehensive description of velocity hodographs, the reader is redirected to the work by Vittaldev (2010).

3.3.4 | Considerations

Now that an overview of the mathematics behind these coordinates has been carried out, it is worthwhile to spend a few words on why each variable was introduced.

As it has already been shown, aerobraking can be a very long process. Therefore, what one looks for is a set of coordinates for propagation that despite the fast computation time, provides an accurate result. According to Vittaldev et al. (2012), such a performance is achieved by the USM variables, which are faster and have a higher accuracy than Cartesian coordinates in virtually any tested scenario. Especially, it was shown that USMEM gives the lowest position error and number of function evaluations (for variable step size integration) for an aerobraking maneuver around Mars. Nevertheless, this comparison was only carried out for one case, thus not leading to a conclusive evaluation. Hence, one cannot simply exclude Cartesian coordinates. This is the reason why the state variable performance analysis of Chapter 10 will be undertaken. The results of this chapter, in fact, will be used to select the propagation settings for the aerobraking simulation.

As mentioned in the introduction of this report, the main purpose of using aerobraking is to reduce the eccentricity of the orbit inexpensively. Thus, to notice this effect, the use of Keplerian elements is required. However, since the main interest for aerobraking is the altitude of the apsis extrema, the use of r_p and r_a, i.e., radial distance at perigee and radial distance at apogee, are also used, together with a and e.

Spherical coordinates are widely used in literature for analysis of re-entry and other aeroassisted maneuvers. Specifically, they are used because they inherently provide information on altitude and velocity, which are very important for the definition of the periapsis corridor. Furthermore, latitude and longitude are needed for the calculation of atmospheric properties and gravitational perturbations.

4
Flight Dynamics

Flight dynamics is at the basis for the description of the movement of an object in space. In this analysis, flight dynamics consists of astrodynamics and attitude dynamics. Astrodynamics is what describes the motion of an object due to the main gravitational potential from a main body, and subject to smaller perturbing forces, arising from the surrounding environment and from onboard propulsive systems. Attitude dynamics, on the other hand, describes the rotational dynamics of the object, which also depend on the environment and onboard actuators.

A precise description of the motion and attitude of an object, requires a perfect representation of the environment and the object itself. However, this would require an immense amount of computational power. For this reason, use of assumptions is very common. For instance, some basic assumptions are made for the derivation of the equations of motion. These can be summarized as (Mooij, 1994):

- **Rigid body**: the vehicle is assumed to be a rigid body; thus, no internal changes in configuration of the vehicle (such as propellant sloshing or motion of masses) are modeled
- **Constant mass**: the mass of the vehicle is assumed to remain constant; thus consumption of propellant mass is not taken into account; this is a valid assumption for aerobraking, where the propellant used is in the order of tens of kilograms

With these assumptions, the equations of motion can be simplified, and the resulting expressions can be found in the next sections. In particular, the reader will find a description of the equations for translational motion in Section 4.1, for rotational motion in Section 4.2, and finally a description of the environment in Section 4.3. Here, one can also find more assumptions, which are made to simplify the simulation of the environment.

4.1 | Equations of Translational Motion

Since both Cartesian coordinates and unified state model are going to be used for propagation, the equations of motion are given for both coordinate representations. Cartesian kinematics and dynamics are explained in Subsection 4.1.1 and the equations for USM are discussed in Subsection 4.1.2.

4.1.1 | Cartesian Coordinates

The equations of motion expressed in Cartesian coordinates are given by (Mooij, 1994):

$$\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} = \dot{\mathbf{r}} \tag{4.1a}$$

$$\frac{\mathrm{d}\dot{\mathbf{r}}}{\mathrm{d}t} = \ddot{\mathbf{r}} = \mathbf{a} \tag{4.1b}$$

Here, $\ddot{\mathbf{r}}$ is the acceleration acting on the spacecraft. In general, this term is divided into two main groups, namely the gravitational acceleration and the perturbing accelerations. Thus, considering that the gravitational acceleration is exerted by a point-mass, one can rewrite Eq. (4.1b), i.e., the dynamics, as:

$$\left.\frac{\mathrm{d}\dot{\mathbf{r}}}{\mathrm{d}t}\right|_{\mathcal{I}} = \boldsymbol{\nabla}\left(\frac{\mu}{r}\right) + \ddot{\mathbf{r}}_{\text{pert}} = -\mu\frac{\mathbf{r}}{||\mathbf{r}||^3} + \ddot{\mathbf{r}}_{\text{pert}} \tag{4.2}$$

Equation (4.2) is expressed in the $\mathcal{I}$-frame, thus no apparent forces arise. Furthermore, the term $\ddot{\mathbf{r}}_{\text{pert}}$ represents the perturbing accelerations, which surface because of disturbances or imperfections of the environment. As stated earlier, these are described in Section 4.3.

4.1.2 | Unified State Model

Depending on the unified state model chosen, there are three possible sets of equations of motion to be used. The dynamics, however are the same for all three. Therefore, first the dynamics are treated, and then, depending on the choice of attitude parameters, the kinematics. Reference for the equations shown is Vittaldev et al. (2012), unless otherwise specified.

The dynamic equations of motion for USM are given by (Vittaldev, 2010):

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} C \\ R_1 \\ R_2 \end{pmatrix} = \begin{bmatrix} 0 & -p_1 & 0 \\ \cos\lambda & -(1+p_1)\sin\lambda & -\gamma p_2 \\ \sin\lambda & (1+p_1)\cos\lambda & \gamma p_3 \end{bmatrix} \ddot{\mathbf{r}}_{\text{pert}}\big|_{\mathcal{L}} \tag{4.3}$$

with the accelerations are expressed in the $\mathcal{L}$-frame. Here, γ and p_i are some auxiliary parameters, which are introduced for ease of notation. The value of γ is dependent on the attitude choice, so its equations are given in the respective paragraphs, whereas p_i are given by:

$$\begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = \frac{1}{C - R_1\sin\lambda + R_2\cos\lambda}\begin{pmatrix} C \\ R_2 \\ R_1 \end{pmatrix} \tag{4.4}$$

Similarly to γ, λ also depends on the attitude representation chosen.

Quaternions The equation for kinematics in case of quaternions is given by:

$$\begin{pmatrix} \dot{\eta} \\ \dot{\boldsymbol{\varepsilon}} \end{pmatrix} = \frac{1}{2}\begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix}\begin{pmatrix} \eta \\ \boldsymbol{\varepsilon} \end{pmatrix} \tag{4.5}$$

Here, ω_i are the elements of the rotational velocity vector $\boldsymbol{\omega}$, between the $\mathcal{L}$-frame and the $\mathcal{I}$-frame. Its components are found with:

$$\omega_1 = \frac{\ddot{r}_{\text{pert,3}}}{C - R_1\sin\lambda + R_2\cos\lambda} \tag{4.6a}$$

$$\omega_2 = 0 \tag{4.6b}$$

$$\omega_3 = \frac{C}{\mu}\left(C - R_1\sin\lambda + R_2\cos\lambda\right)^2 \tag{4.6c}$$

Finally, λ and γ can be determined from:

$$\lambda = \arctan_2\left(\frac{2\varepsilon_3\eta}{\varepsilon_3^2+\eta^2}, \frac{\eta^2-\varepsilon_3^2}{\varepsilon_3^2+\eta^2}\right) \tag{4.7a}$$

$$\gamma = \frac{\varepsilon_1\varepsilon_3 - \varepsilon_2\eta}{\varepsilon_3^2+\eta^2} \tag{4.7b}$$

Modified Rodrigues Parameters The kinematic differential equations in terms of MRP, which also holds for SMRP, is (Schaub and Junkins, 2002):

$$\dot{\boldsymbol{\sigma}} = \frac{1}{2}\left[\frac{1}{2}\left(1-\sigma^2\right)\mathbf{I} + \boldsymbol{\sigma}^{\mathrm{S}} + \boldsymbol{\sigma}\boldsymbol{\sigma}^{\mathrm{T}}\right]\boldsymbol{\omega} \tag{4.8}$$

where, $\boldsymbol{\omega}$ is still given by Eq. (4.6). The values for λ and γ, on the other hand, are:

$$\lambda = \arctan_2\left(\frac{4\sigma_3\left(1-\sigma^2\right)}{4\sigma_3^2+\left(1-\sigma^2\right)^2}, \frac{\left(1-\sigma^2\right)^2-4\sigma_3^2}{4\sigma_3^2+\left(1-\sigma^2\right)^2}\right) \tag{4.9a}$$

$$\gamma = \frac{2\left[\sigma_2\left(\sigma^2-1\right)+\sigma_1\sigma_3\right]}{\sigma_1^4+\sigma_2^4+\sigma_3^4+2\left[\sigma_1^2(\sigma_2^2+\sigma_3^2-1)+(\sigma_2^2+1)(\sigma_3^2-1)\right]+3} \tag{4.9b}$$

Once again, Eq. (4.9) also hold for SMRP. However, it is important to keep track of which representation is used to convert to and from other attitude representations. For instance, Vittaldev (2010) uses an extra flag parameter that is added to the state vector.

EXPONENTIAL MAP In terms of EM, the kinematic equations become (Schaub and Junkins, 2002):

$$\dot{\mathbf{e}} = \left\{ \mathbf{I} + \frac{\mathbf{e}^{\mathrm{S}}}{2} + \frac{1}{\xi^2} \left[1 - \frac{\xi}{2} \cot\left(\frac{\xi}{2}\right) \right] \left(\mathbf{e}^{\mathrm{S}}\right)^2 \right\} \boldsymbol{\omega} \tag{4.10}$$

Clearly, this equation presents a singularity at $\xi = 0$, where $\xi = ||\mathbf{e}||$. This singularity can be avoided by approximating Eq. (4.10) with the Taylor series expansion given by:

$$\dot{\mathbf{e}} \approx \frac{1}{2} \left[\left(\frac{12 - \xi^2}{6} \right) \boldsymbol{\omega} - \boldsymbol{\omega} \times \mathbf{e} - \boldsymbol{\omega} \cdot \mathbf{e} \left(\frac{60 + \xi^2}{360} \right) \mathbf{e} \right] \tag{4.11}$$

when ξ is close to zero. It should be noted that both Eqs. (4.10) and (4.11) are valid for both EM and SEM. Furthermore, as already mentioned in Subsection 3.3.1, there is no need to keep track of whether EM or SEM are being used, meaning that no extra flag needs to be added to the state vector.

The values of λ and γ for EM are computed by using the same relations in Eq. (4.7), by first converting the EM to quaternions. This is done with:

$$\eta = \cos\left(\frac{\xi}{2}\right) \tag{4.12a}$$

$$\boldsymbol{\varepsilon} = \frac{1}{\xi} \sin\left(\frac{\xi}{2}\right) \mathbf{e} \tag{4.12b}$$

The singularity in Eq. (4.12b) is tackled by approximating the the term $\sin\left(\xi/2\right)/\xi$ with a Taylor series expansion, when ξ is small. This approximation is:

$$\frac{1}{\xi} \sin\left(\frac{\xi}{2}\right) \approx \frac{1}{2} - \frac{\xi^2}{48} \tag{4.13}$$

4.2 | Equations of Rotational Motion

As mentioned in Subsection 3.3.4, the attitude variables chosen are quaternions, MRPs and the exponential map. Thus, in this section the equations of rotational motion will be expressed in terms of these three state variables. First, the rotational kinematics will be given, in Subsection 4.2.1, then the rotational dynamics, in Subsection 4.2.2. The same assumptions introduced at the beginning of this chapter hold for the rotational motion.

4.2.1 | Kinematics

The equations of rotational kinematics show how the attitude state variables change over time, due to a rotational velocity. For a spacecraft equipped with an IMU, the rotational velocities are measured by the gyroscopes, and then the kinematic equations are integrated to find the current orientation. Otherwise, the velocity can be found by integrating the dynamic equation, and then following the same procedure.

For quaternions, the kinematics are expressed very elegantly by Eq. (4.5). Alternatively, Eq. (4.5) can be written as (Wie, 2008):

$$\dot{\eta} = -\frac{1}{2} \boldsymbol{\omega}^{\mathrm{T}} \boldsymbol{\varepsilon} \tag{4.14a}$$

$$\dot{\boldsymbol{\varepsilon}} = \frac{1}{2} \left(\eta \boldsymbol{\omega} - \boldsymbol{\omega}^{\mathrm{S}} \boldsymbol{\varepsilon} \right) \tag{4.14b}$$

For MRP and EM, the kinematic equations were also given in Section 4.1. They can be found in Eqs. (4.8) and (4.10), respectively.

4.2.2 | Dynamics

The equations of rotational dynamics show how the rotational velocity of vehicle is affected by internal or external perturbations. In the case of rotational dynamics, however, perturbations are expressed as moments $\mathbf{M}$, instead of accelerations as was seen in Section 4.1.

The general form of the dynamic differential equation is given by:

$$\mathbf{J} \dot{\boldsymbol{\omega}} + \boldsymbol{\omega}^{\mathrm{S}} \mathbf{J} \boldsymbol{\omega} = \mathbf{M} \tag{4.15}$$

and is referred to as Euler's rotational equations of motion (Wie, 2008). In Eq. (4.15), $\mathbf{J}$ is the inertia matrix, which combines the moment of inertia (on its diagonal, i.e., J_{ii}) and the products of inertia (the off-diagonal elements, i.e., J_{ij} with $i \neq j$). The inertia matrix is expressed around the body axes, and for an axisymmetric body is a diagonal matrix (meaning that the $\mathcal{B}$-frame corresponds with the principal axes of the vehicle) (Wie, 2008).

In general, the term $\mathbf{M}$ is divided into two components: the control moment $\mathbf{M}_c$ and the disturbance (or perturbing) torque $\mathbf{M}_{\text{pert}}$. The control torque is generated by onboard attitude actuators, whereas the disturbance torque arises from internal and external perturbations. For instance, the spacecraft may be subject to a gravity torque, since the gravitational field acts differently on each element of the spacecraft (center of mass and center of gravity do not coincide), and an aerodynamic torque, in case the aerodynamic center differs from the center of mass (Mooij, 1994).

4.3 | Environment

This section discusses the environment the spacecraft operates in. This environment is, in general, modeled by assuming a main central gravitational acceleration as given by:

$$\ddot{\mathbf{r}}_{\text{grav}} = -\frac{\mu}{r^2}\hat{\mathbf{r}} \tag{4.16}$$

which is perturbed by much smaller accelerations that depend on different sources. The main perturbations are classified as:

- **Gravitational Field**
  Planets are not perfect spheres. The way this imperfection is modeled is with the use of spherical harmonics. These are used to model the gravitational effects of the planet being orbited (or even a third body), which are not the main gravitational acceleration given by Eq. (4.16).
- **Atmosphere**
  This acceleration arises from the interaction of the spacecraft with the highest layers of the atmosphere of the planet. Aerobraking would not be a thing if it were not for this acceleration, thus it is fundamental to model this acceleration. Explanation of the modeling of the acceleration is given in this chapter, whereas details on the modeling of the aerodynamics of the spacecraft are given in Chapter 5.
- **Solar Radiation Pressure**
  Solar radiation pressure arises by the transfer of momentum between photons emitted by the Sun and the spacecraft. It is a non-conservative acceleration and is generally small in magnitude. Since aerobraking can be a very long process, it may be important to take into account even the effect the solar radiation pressure.
- **Third Bodies**
  Similarly to the radiation pressure, it is important to also model the effects of bodies that do not directly affect the aerobraking maneuver.

By analyzing the accelerations acting on a Mars-bound spacecraft, one will find that the main perturbations acting on the spacecraft are given by the Martian gravitational field and the Martian atmosphere. Then, one can find the perturbing accelerations that arise because of the Sun, namely its gravitational attraction and the solar radiation pressure. Finally, the accelerations due to the other solar system bodies are found.

It can be concluded that modeling of the gravity field and atmosphere of Mars is essential, due to their large effects. Following a similar reasoning, the third-body attraction of Venus, Earth, Jupiter and Saturn can be neglected, due to their very low magnitude. Finally, it was decided to also model the gravitational and solar radiation effect of the Sun. This is because over the course of 150 days, correction of these accelerations combined accounts for about 17.5 $\mathrm{m\,s^{-1}}$, which is a considerable amount, when compared to the total the total ΔV used by MRO for aerobraking, i.e., 44 $\mathrm{m\,s^{-1}}$ (Long et al., 2008).

Note that the models discussed, do not necessarily need to be taken into account by the spacecraft onboard computer when making calculations. They are however, important for propagation of the spacecraft state. Models for propagation will be described in Subsection 4.3.2, whereas model for onboard computations in Subsection 4.3.3. First, however, Subsection 4.3.1 will go through of the mathematics behind these perturbations.

4.3.1 | Mathematical Models

The perturbations listed above are discussed below, in the same order. The main reference for this section is Montenbruck and Gill (2005).

Gravitational Field

As was mentioned in the introduction of this section, planets are not perfect spheres, and thus, the assumption that the satellite is orbiting a point-mass is not valid. The equations of motion can thus be generalized for the case of a more realistic gravitational potential. The new equation of motion is expressed as:

$$\ddot{\mathbf{r}}_{\text{grav}}\big|_{\mathcal{R}} = \boldsymbol{\nabla} U \tag{4.17}$$

where U is the gravitational potential and $\boldsymbol{\nabla}$ is the gradient operator. If the planet is assumed to be a point-mass, then $U = {}^{\mu}/_{r}$, and the result is the same as shown in Eq. (4.16). For a real body, however, the potential is described in integral form by:

$$U = G \iiint \frac{\rho\left(\mathbf{s}\right)}{||\mathbf{r}-\mathbf{s}||}\,\mathrm{d}^3\mathbf{s} \tag{4.18a}$$

where $\rho\left(\mathbf{s}\right)$ is the density function of the body and the integral sums over each infinitesimal point $\mathbf{s}$ of the body. The integral in Eq. (4.18a) can be solved by expanding the denominator in a series of Legendre polynomials. After some manipulation, and with the introduction of gravitational moments, this results in (Wakker, 2015):

$$U = -\frac{\mu}{r}\left\{1 - \sum_{n=2}^{\infty}\left[J_n\left(\frac{R}{r}\right)^n \mathcal{P}_n\left(\sin\delta\right) + \sum_{m=1}^{n} J_n^m \left(\frac{R}{r}\right)^n \mathcal{P}_n^m\left(\sin\delta\right)\cos\left(m\left(\tau-\tau_n^m\right)\right)\right]\right\} \tag{4.18b}$$

Here, the terms J_n^m and τ_n^m are the spherical harmonic coefficients and are usually known up to a very high degree n and order m (for Mars, this is up to 120[1]). The terms $\mathcal{P}_n^m$ on the other hand, represents the associated Legendre functions. They are related to the Legendre polynomials $\mathcal{P}_n$ as:

$$\mathcal{P}_n\left(x\right) = \frac{1}{2^n n!}\frac{\mathrm{d}^n}{\mathrm{d}x^n}\left(x^2-1\right)^n \tag{4.19a}$$

$$\mathcal{P}_n^m\left(x\right) = \left(1-x^2\right)^{m/2}\frac{\mathrm{d}^m}{\mathrm{d}x^m}\left(\mathcal{P}_n\left(x\right)\right) \tag{4.19b}$$

The degree n and order m can be used to categorize the coefficients in Eq. (4.18b). The three categories are:

- *zonal coefficients*, when $m = 0$
- *tesseral coefficients*, when $m < n$
- *sectorial coefficients*, when $m = n$

Looking back at Eq. (4.18b), it is clear how the gravitational potential depends on the position of the spacecraft, expressed in spherical coordinates, i.e., δ, τ and r. It is important to notice that these coordinates are expressed in the rotating planetocentric reference frame (the $\mathcal{R}$-frame from Section 3.2). Hence, the acceleration resulting from the gradient of U is also expressed in the $\mathcal{R}$-frame.

As mentioned in Subsection 4.2.2, a gravitational field also gives rise to a gravity torque $\mathbf{M}_{\text{grav}}$. This torque is expressed by (Wertz et al., 2011):

$$\mathbf{M}_{\text{grav}}\big|_{\mathcal{R}} = \frac{3\mu}{r^3}\hat{\mathbf{r}}\times\left(\mathbf{J}\hat{\mathbf{r}}\right) \tag{4.20}$$

Here, one can see that the torque is only assumed to arise from the central gravitational attraction (hence no spherical harmonics are used), and the magnitude depends on the angle between the local vertical and the principal axes of the spacecraft.

Furthermore, it should be pointed out that Eq. (4.20) is a first order approximation to the gravitational torque. More accurate analyses would discretize the spacecraft into small sections and based on the gravitational force experienced by each element, compute the resultant moment. However, such an analysis is only necessary when the spherical harmonics terms have values in the same order of magnitude as the central gravity. This is not the case for Mars, and thus Eq. (4.20) can be used.

Atmosphere

The drag acceleration felt by a spacecraft traveling through an atmosphere is described by:

$$\ddot{\mathbf{r}}_{\text{aero}}\big|_{\mathcal{I}} = -\frac{1}{2}\rho V_A^2 \mathbf{C}\frac{A}{m} \tag{4.21}$$

[1] Goddard Space Flight Center: MRO Gravity Model of Mars – `https://goo.gl/N6ziht` [last accessed on November 18, 2017]

In Eq. (4.21), A, m and $\mathbf{C}$ are the cross-sectional area exposed to the flow, the mass of the spacecraft, and its aerodynamic coefficients (in inertial frame). The value of the latter dimensionless vector is usually expressed in the aerodynamic frame, but it is, nonetheless, not constant. It is in fact affected by both atmospheric and spacecraft conditions. More information of how the aerodynamic coefficients are determined is given in Chapter 5.

The velocity in Eq. (4.21), is the airspeed, i.e., the velocity w.r.t. the atmosphere. In general, it can be assumed that the angular velocity of the atmosphere is the same as the angular velocity of the planet. Thus, if $\mathbf{V}$ is the velocity w.r.t. an inertial reference, and $\boldsymbol{\omega}$ the angular velocity vector of the planet, the airspeed is given by:

$$\mathbf{V}_A = \mathbf{V} - \boldsymbol{\omega}^{\mathrm{S}}\mathbf{r} \tag{4.22}$$

The last variable in Eq. (4.21) to describe is ρ. This is the variable that represents the value of atmospheric density. Various models can be used to describe the value of atmospheric density, and they range from simplified to detailed. The model chosen for this application will be explained in Subsection 4.3.2.

By expressing the aerodynamic force in the $\mathcal{A}$-frame, the usual drag, side and lift coefficients become visible:

$$\ddot{\mathbf{r}}_{\text{aero}}\big|_{\mathcal{A}} = -\frac{\rho A}{2m} V_A^2 \begin{pmatrix} C_D \\ C_S \\ C_L \end{pmatrix} \tag{4.23}$$

Transformation to $\mathcal{F}_{\mathcal{I}}$ is achieved by means of a series of rotations, which can be easily determined based on the discussion in Appendix A.1.

Finally, one can also find the torque generated due to the aerodynamic forces. From aerodynamic analyses, it is known that aerodynamic forces act at the aerodynamic center. Disturbance moments arise in case this center of pressure (CP) were to differ from the center of mass of the spacecraft. In particular, if C_l, C_m and C_n are the moment coefficient about the body axes that arise due to this discrepancy, the aerodynamic torque can be expressed as (Mooij, 1994):

$$\mathbf{M}_{\text{aero}}\big|_{\mathcal{I}} = \frac{\rho A L}{2m} V_A^2 \mathbf{C}_{\mathcal{I}}^{\mathcal{A}} \begin{pmatrix} C_l \\ C_m \\ C_n \end{pmatrix} \tag{4.24}$$

where L is the reference length of the spacecraft.

Solar Radiation

Solar radiation pressure arises from the transfer of momentum of photons, onto the spacecraft. The amount of energy that reaches the spacecraft is expressed as the solar flux Φ, and it is given by (Lissauer and De Pater, 2013):

$$\Phi = \frac{L_{\odot}}{4\pi r^2} \tag{4.25}$$

where $L_{\odot}$ is the luminosity of the Sun, and it is equal to 382.8×10^{26} W[(2)]. The transfer of momentum happens in two ways: reflection (where only specular reflection is considered) and absorption. The effect of these is modeled with:

$$\ddot{\mathbf{r}}_{\text{rad}}\big|_{\mathcal{B}} = -\frac{\Phi}{c}\frac{\hat{\mathbf{r}}_{\odot}}{r_{\odot}^2} C_R \frac{A}{m} \tag{4.26}$$

In the above, c is the speed of light ($299.792\,458 \times 10^6$ m s^{-1}), C_R is the radiation pressure coefficient, and the vector $\mathbf{r}_{\odot}$ is the distance to the sun. The parameter A in Eq. (4.26) represents the cross-sectional area of the spacecraft exposed to the radiation.

Solar radiation pressure, unlike gravity, needs a direct line-of-sight to act. This means that its value is zero if the spacecraft is in a shadow cone. Moreover, the Sun is not the only body of interest to emit radiation. One can also consider the radiation reflected by the planet being orbited. This may is not particularly important for Mars, whose bond albedo (ratio of reflected to received radiation) is about 0.25 (Lissauer and De Pater, 2013). As such, albedo radiation is neglected in the analysis.

Third Bodies

Third bodies are those bodies that are not part of the main system (orbited planet and spacecraft), but whose gravitational acceleration still affects the motion of the spacecraft. Two assumptions are used for this analysis: the spacecraft does not affect the motion of the third body (also valid for the planet being orbited) and the third body is

[(2)] NASA: Sun Fact Sheet – `https://goo.gl/EGhBtj` [last accessed on November 6, 2017] modeled as a point-mass (no spherical harmonics are considered).

The first assumption is valid since the mass of the third body is many orders of magnitude larger (thus the acceleration it feels is negligible), and the second assumption holds because third bodies are usually very far away, such that the $\left({}^{R}\!/_{r}\right)^n$ term from Eq. (4.18b) becomes negligibly small for $n \neq 0$.

The perturbing acceleration due to n third bodies, can be expressed as:

$$\ddot{\mathbf{r}}_{\text{third}}\big|_{\mathcal{B}} = \sum_{i=1}^{n} \mu_i \left(\frac{\mathbf{s}_i - \mathbf{r}}{||\mathbf{s}_i - \mathbf{r}||^3} - \frac{\mathbf{s}_i}{||\mathbf{s}_i||^3} \right) \tag{4.27}$$

where in this situation, $\mathbf{s}$ is the distance from the main body to the third body. The first term in the round brackets of Eq. (4.27) is the gravitational attraction of the spacecraft w.r.t. body i, and the second term represents the attraction of the main body w.r.t. body i.

An important parameter to consider when dealing with many-bodies simulations is which body is the main body. This is determined by using the concept of sphere of influence (SOI). The sphere of influence is defined as that area where the body being analyzed has the highest gravitational attraction on the bodies around it, compared to other third bodies with larger mass. The equation for the SOI expressed as multiples of the radius R of the planet considered is (Curtis, 2013):

$$r_{\text{SOI}} \approx \frac{a}{R} \left(\frac{\mu}{\mu_{3^{\text{rd}}}} \right)^{2/5} \tag{4.28}$$

where a is the semi-major axis of the orbit of the planet, whose central body (i.e., the Sun) exerts an acceleration proportional to $\mu_{3^{\text{rd}}}$. It is important to consider that Eq. (4.28) is only valid if $\mu < \mu_{3^{\text{rd}}}$.

Disturbing torques due to third bodies will be neglected, given their very low gravitational influence.

4.3.2 | Implementation in Propagation

Implementation of the perturbations described in the previous subsection, requires in general, use of environment models. Discussion on models is divided into the same categories as above.

These models are the ones that will be implemented in the software architecture of Chapter 8, and used to represent the environmental conditions the simulated spacecraft encounters. The next section, on the other hand, will describe the models stored on the onboard computer.

Gravitational Field

Regarding the gravitational field, one has to choose the number of terms to include (i.e., the maximum degree and order). It is known that the gravitational field is a conservative force. Thus, if one were to integrate only the effects of the gravitational potential, the result would be a state that is oscillating around a mean value, but no net change in state would be noticed over time. However, since (dissipative) aerodynamic forces are also taken into account, this is no longer true, and one cannot just consider the mean state, but has to take into account a more detailed representation of the field. Murri (2013) used 21 as values for maximum n and m for their simulations of the aerobraking phase of MRO, and without further justification, these same settings will be used for this analysis, as well.

Atmosphere

There exist many types of atmospheric models. The most precise ones are empirical models (i.e., based on measurement made, rather than strict mathematical formulae), but are also very computationally expensive. Simpler methods are usually given by exponential models. An exponential model is described by (Lissauer and De Pater, 2013):

$$\rho = \rho_0 \exp\left(-\frac{h - h_0}{H} \right) \tag{4.29a}$$

which is derived from hydrostatic equilibrium. In the equation, ρ_0 is the density at the reference height h_0 (which for aerobraking missions is usually defined as the pericenter altitude (Tolson and Prince, 2011)), and H is the scale height of the atmosphere, given by:

$$H = \frac{RT}{g} \tag{4.29b}$$

The parameters here are the gas constant R, the temperature T and the gravitational acceleration g. In general, these values are a function of altitude, but to derive Eq. (4.29a), they are considered constant. Appendix C.2, gives an overview of the parameters needed in Eq. (4.29) for Mars.

Exponential models, however, are very approximate. They do not model variations due to day/night, seasonal, zonal and solar effects. These effects are especially important for predicting the atmospheric properties of the highest layers of the atmosphere, which are the ones that are of interest for an aeroassisted maneuver.

Two main models are available, these being Mars-GRAM (Justh et al., 2006), where GRAM stands for Global Reference Atmospheric Model, developed by the Marshall Space Flight Center, and the Mars Climate Database (MCD) (Forget et al., 1999, Millour et al., 2015), developed by the European Space Agency. The Mars-GRAM model was used for and updated with the aerobraking missions performed by NASA around Mars. The MCD, on the other hand, is derived from the general circulation model together with observed data and is publicly available (also provides a web interface[3]). Both models take into account the effects of solar and dust activities, which have very large effects on the density profile.

The main issue with using these very accurate models is indeed their high precision. In Chapter 3, where the $\mathcal{R}$-frame was introduced, it was mentioned that the transformation between the $\mathcal{V}$-frame and $\mathcal{R}$ (expressed by Eq. (A.5)) is simplified. This is the case, because Mars is assumed to be a perfect sphere, and thus the latitude, longitude and altitude used are given w.r.t. a perfect sphere, instead of the areoid (the more accurate representation of Mars' shape). Furthermore, since the transformation between the $\mathcal{I}$-frame and $\mathcal{R}$ (expressed by Eq. (A.4)) is also simplified, the local solar time would also be less accurate than needed for Mars-GRAM or the MCD. Then, one could argue that for this analysis, it is not important to have an exact value of the density, as long as the output profile roughly matches the expected result. After all, the Martian thermosphere is subject to large variations in density, and thus even the most accurate models have trouble with providing a good representation of these layers. It is in the variability of the atmosphere that stands the highest challenge for missions such as aerobraking, since the onboard model needs to be able to predict whether the next periapse altitude needs correcting. Hence, even a tabulated atmosphere with the addition of random perturbations (a constant random offset from the tabulated values during each atmospheric pass) would provide a challenging enough environment for the onboard software, without requiring computationally expensive models.

In fact, the method above is the one chosen. The atmospheric parameters are first determined by averaging the MCD data over time, which results in the tabulated atmosphere model shown in Appendix C.3, and then the density is multiplied by a term that provides some variability to the atmosphere. This term is kept constant for each orbit, and provides a surface that is continuous over altitude, longitude and latitude and generates areas of high/low concentrations. The function used to generate the multiplicative factor is the following:

$$\kappa_\rho = \left| a + \sum_{i=1}^{3} b_i \sin\left[2\pi d_i \left(\frac{x_i - x_{i,\text{min}}}{x_{i,\text{max}} - x_{i,\text{min}}}\right)\right] + c_i \cos\left[2\pi d_i \left(\frac{x_i - x_{i,\text{min}}}{x_{i,\text{max}} - x_{i,\text{min}}}\right)\right]\right| \tag{4.30}$$

where x represents either altitude (x_1), longitude (x_2) or latitude (x_3), and a, b, c and d are random coefficients, whose initial value is determined with the distributions and parameters from Table 4.1. Their value is then varied for the next orbit by adding a random walk to the parameter. This random walk is modeled as follows for b and c:

$$x_{i+1} = x_i + 0.05\, X \tag{4.31a}$$

where w is the value given by the respective distribution in Table 4.1, whereas the values of a and d are adapted with this method:

$$x_{i+1} = \begin{cases} x_i + 0.05\, X & \text{if } \sum_{i=1}^{3} (B + C) \geqslant 0 \\ x_i - 0.05\, X & \text{otherwise} \end{cases} \tag{4.31b}$$

The introduction of the sign variability in Eq. (4.31b), is due to the fact that the values of a and d from the random distribution are never negative, and would otherwise always increase.

Moreover, the values of *max* and *min* represent the maximum and minimum values of the variables x in the tabulated atmosphere (note that for altitude, the maximum value is truncated to 1500 km, such that a higher degree of variability is added to the lower layers of the atmosphere). The absolute value in Eq. (4.30) is added to make sure that no negative values of the multiplier are output. Also, the minimum value supported for κ_ρ is 0.5. One can see the above equation in action, in Appendix C.3.

Third Bodies

Position of third bodies is determined from ephemeris. Ephemeris are usually either computed analytically or, they are taken from tables of past and future positions of the main solar system bodies (some also include spacecraft

[3] Mars Climate Database v5.3: The Web Interface – https://goo.gl/WbaFDt [last accessed on March 8, 2018]

Table 4.1: *Characteristics of the random number distributions used to model the variability of the atmosphere, embodied by the coefficients in Eq.* (4.30).

Coefficient	Distribution	Parameters
a	Uniform	Lower bound: 0.75 Upper bound: 2.0
b and c	Normal	Mean: 0.0 Standard deviation: 0.25
d	Normal	Mean: 1.0 Standard deviation: 0.5

ephemerides). Both formats are provided by the Jet Propulsion Laboratory (JPL) and are publicly available at [4]. In particular, the tabulated ephemeris are made available through a software named SPICE[5], or Spacecraft Planet Instrument Camera-matrix Events. Using tabulated values, gives the advantage of not needing to propagate the solar system bodies. Thus, SPICE will be used in this analysis for planetary ephemeris.

When orbiting Mars, however, one also needs to consider the presence of its two moons: Phobos and Deimos. They orbit at relatively low (average) altitudes (6000 km for the former and 20 000 km for the latter[6]) and as such, they can be a danger to an aerobraking mission (in Table 2.1, it was shown that the apoareion of MRO decreased from 45 000 to 320 km in altitude). Furthermore, one should also consider the other satellites orbiting Mars, whose orbits could potentially intersect the path of the aerobraking spacecraft. Mars Reconnaissance Orbiter used about 1.2 m s^{-1} of ΔV for collision avoidance, which is less than 3 % of the total amount (Long et al., 2008). Therefore, given the low amount of propellant used, collision avoidance is not given a priority in the preliminary design of an aerobraking GNC system. To conclude, these activities will not be modeled.

4.3.3 | Implementation Onboard

For onboard calculations, one needs to keep in mind that OBCs are generally not computationally fast. Thus simplified models are carried onboard, and only for the most important perturbations. In fact, only atmospheric effects and gravitational perturbations are considered in this analysis. Solar radiation pressure and third bodies affect the orbit to a very small degree, and are thus not considered.

The effects of the simplified models on the guidance and navigation will be considered during the thesis work. For instance, selection of the best performing atmosphere model will be made, based on the formulations described in the following paragraphs.

Gravitational Field

For onboard implementation of the gravity field, the degree and order of spherical harmonics is usually limited to a smaller value than for propagation. The actual limiting number depends on the mission and planet. For Earth-bound aeroassisted maneuvers, the J_2 effect is generally the only one to be considered. Since Mars has an even stronger J_2 effect than Earth, it should not be left out. However, due to the long duration of aerobraking, it may be important to model also other effects. For this, a dimensional analysis of the gravitational moments for Mars needs to be done. Table 4.2 shows the values of the first five gravitational moments, i.e., the J_n^m terms with degree and order spanning from two to six.

Clearly, the J_2 (or J_2^0) term is the most prominent, being larger than the other terms by at least two orders of magnitude. However, Jah and Lisano (2004) states that at least a $(4, 4)$ gravity field description is necessary, to keep the navigation error within reasonable amounts. In fact, during the first stages of testing, it was soon noticed that a $(2, 0)$ gravity field is not enough, as the error was rapidly growing to values larger than 50 km in position, within one orbit. To conclude, for onboard estimation, a gravitational field with degree and order up to four will be used.

Atmosphere

The exponential atmospheric model introduced in Eq. (4.29), is a potential candidate for onboard calculations. This is because the empirical models discussed in Subsection 4.3.2 require a lot of knowledge of the spacecraft surround-

[4] Jet Propulsion Laboratory: Solar System Dynamics: Ephemerides – https://goo.gl/mhuFPJ [last accessed on December 6, 2017]

[5] Jet Propulsion Laboratory: Navigation and Ancillary Information Facility: SPICE – https://goo.gl/yPqUN8 [last accessed on December 6, 2017]

[6] National Aeronautics and Space Administration: Mars Fact Sheet – https://goo.gl/TNo8PJ [last accessed on March 7, 2018]

Table 4.2: *Values of the first three spherical harmonics coefficients for Mars (computed from Footnote (1)). All values are multiplied by a factor of 10^3.*

		Order				
		0	1	2	3	4
Degree	2	1.957×10^{0}	7.648×10^{-7}	6.311×10^{-2}		
	3	3.147×10^{-2}	2.750×10^{-2}	6.151×10^{-3}	6.051×10^{-3}	
	4	-1.539×10^{-2}	5.362×10^{-3}	2.019×10^{-3}	3.860×10^{-4}	2.721×10^{-4}

Table 4.3: *Parameters for Martian atmosphere wave model expressed by Eq.* (4.32) *(Dwyer et al., 2001).*

			Wave 1	**Wave 2**	**Wave 3**
Amplitude	A_i	[−]	0.1	0.2	0.1
Phase	τ_i	[deg]	0	50	55

ings and a lot of time to compute. An exponential model, or a variation of it, can be a good first approximation for the calculations done onboard, since they mostly consist of short propagations.

According to Tolson and Prince (2011), a constant scale height (CSH) model would suffice for aerobraking operations, where the atmospheric parameters ρ_0 and H are extracted from analyzing data from previous orbits, by using either a weighted least squares or a simple averaging method. As already mentioned, Cianciolo et al. (2013) recommend using a simple averaging method of the previous seven orbits, since it is a very simple and fast method, and its results still manage to reduce the variability of the atmosphere by 30 %.

Using the CSH model would appear particularly useful for bodies whose atmospheres are not known to a high detail (e.g., Venus and Titan). For Mars, on the other hand, since it has been thoroughly studied in past missions, there exist more sophisticated methods. For instance, Dwyer et al. (2001), Prince et al. (2009) suggest using a wave model described by:

$$\rho\left(h,\delta,\tau\right) = \rho\left(h,\delta\right)\left\{A_0 + S\left(t\right) + \sum_{i=1}^{3} A_i \sin\left[i\left(\tau - \tau_i\right)\right]\right\} \tag{4.32}$$

which depends on altitude, latitude and longitude. The first term on the right-hand side of Eq. (4.32) is the density value that is output by a CSH model that depends on both altitude and latitude, whereas the second term (the one in the curly brackets), represents the longitudinal waves. Prince et al. (2009) added the extra S term to account for Martian dust storms, since they have a tendency of increasing density by more than a factor of two. The term A_0 is given by a Gaussian distribution (unit mean and $0.15 < \sigma < 0.19$) and is used to model for unexpected variations, while the terms A_i and τ_i represent the three stationary longitudinal waves, which are constant and are shown in Table 4.3. Equation (4.32) is especially useful for Monte Carlo simulations, where uncertainties in the atmospheric properties of the body need to be considered.

Yet another representation for the atmospheric density is found in Jah et al. (2008). The authors refer to this model as the five-parameter exponential model, which is given by:

$$\rho = \rho_0 \exp\left\{\kappa_1\left(\frac{h-h_0}{H}\right) + \kappa_2\cos\left[2\pi\left(\frac{h-h_0}{H}\right)\right] + \kappa_3\sin\left[2\pi\left(\frac{h-h_0}{H}\right)\right]\right\} \tag{4.33}$$

Clearly, in case $\kappa_1 = -1$ and $\kappa_2 = \kappa_3 = 0$, the result is the same as in Eq. (4.29a). Gallis et al. (2014) do not give values for the parameters ρ_0, H and κ_i, but estimate them by fitting the acceleration data every orbit.

5 Aerothermodynamics

Aerobraking uses the highest layers of the atmosphere to brake the spacecraft. This braking effect uses friction between the atmosphere and the spacecraft surface, to dissipate kinetic energy. Due to the law on conservation of energy, the dissipated energy is transformed into another type of energy, namely heat. This leads to the generation of high temperatures around the body of the spacecraft, which need to be accounted for in the spacecraft design.

In this chapter, the effects of high-speed and high-temperature effects encountered during aerobraking will be briefly explained. Firstly, in Section 5.1 some similarity parameters will be introduced, which are very useful in simplifying the analysis done in the successive sections. A brief discussion on the thermodynamics and heating of the spacecraft is done in Section 5.2, whereas Section 5.3 gives an overview of the aerodynamics of the spacecraft, which depend on the flow regime encountered.

In Appendix B one will find the chosen spacecraft discretized model that will be used for the aerodynamic analysis. Moreover, Appendix D explains how said analysis was carried out, and Chapter 11 shows a full description of the results, together with validation of the model and results.

5.1 | Similarity Parameters

Similarity parameters are used to reduce the amount of parameters that need to be known, to describe the motion of a vehicle in an atmosphere (Gallais, 2007). In fact, dimensional analysis can be used to define some dimensionless similarity parameters that are very useful in this chapter. They are named after scientists who pioneered the field of aero- and thermodynamics and are the following (Hankey, 1988):

- **Mach number**, or M
  It is defined by:

$$M \triangleq \frac{V}{a} \tag{5.1a}$$

  and its physical interpretation is related to the ratio of ordered to random energy. Here, a is the speed of sound, and it is defined by:

$$a = \sqrt{\gamma R T} \tag{5.1b}$$

  where γ is the specific heat ratio, R the gas constant and T the temperature of the gas. Their average values for Mars are given in Appendix C.

- **Reynolds number**, or Re
  It is defined by:

$$Re \triangleq \frac{\rho V L}{\mu} \tag{5.2}$$

  and it represents the ratio of inertial to viscous forces. In Eq. (5.2), L is the characteristic vehicle dimension and μ is the viscosity of the fluid.

- **Knudsen number**, or Kn
  It is defined by:

$$Kn \triangleq \frac{\lambda}{L} \tag{5.3a}$$

and it represents the ratio of the number of collisions of a molecule with a body over the number of collisions with another molecule. In Eq. (5.3a), λ is the mean free path of the molecules and L is the characteristic vehicle dimension. According to Regan and Anandakrishnan (1993), the mean free path is defined as:

$$\lambda = \frac{1}{\sqrt{2}\pi} \frac{1}{n\sigma^2} \tag{5.3b}$$

where σ and n are the collision diameter and number density of a molecule, respectively. The number density can be found with:

$$n = \frac{N_A}{\mathscr{M}} \rho \tag{5.3c}$$

Here, N_A is Avogadro's number ($N_A = 6.0221 \times 10^{23}$ mol^{-1}) and $\mathscr{M}$ the molar mass of the gas. How to compute the molar mass based on atmospheric composition is explained in Appendix C. The Knudsen number is particularly interesting in this analysis. In fact, it is used to define the flow regimes, which will be discussed in Section 5.3.

- **Stanton number**, or St (Anderson Jr., 2006)
  It is defined by:

$$St \triangleq \frac{C_h}{\rho V C_p} \tag{5.4}$$

and it represents the ratio of heat transferred to the heat capacity of a fluid. In Eq. (5.4), C_h is the heat transfer coefficient and C_p is the specific heat at constant pressure. Equation (5.4) can be re-written as Eq. (5.5) for a body, where q_w is the heat transferred at the wall, $h_{a,w}$ and h_w are the adiabatic wall and wall enthalpies, respectively, and the subscript e represents the conditions at the edge of the boundary layer.

$$St = \frac{q_w}{\rho_e V_e \left(h_{a,w} - h_w\right)} \tag{5.5}$$

5.2 | Heat Transfer

As mentioned in the introduction of this chapter, the dissipation of kinetic energy generates thermal energy, or heat. This heat can become a big limitation for aeroassisted maneuvers, and thus is one of the driving parameters.

Heat transfer during aerobraking can be categorized as convective heat. This is caused by transfer of energy between the hot gas in the boundary layer and the cold vehicle surface (Anderson Jr., 2006). For aeroassisted maneuvers that enter deeper layers of the atmosphere, radiative heating, caused by the radiation emitted by the same hot gas, also plays an important role. Upon review of previous mission aerobraking strategies, however, it was found that this heating effect is neglected for aerobraking maneuvers. In fact, for Mars-bound aerobraking missions, the only heating considered has been the aerodynamic heating, thus neglecting heating due to other sources.

In most aerobraking studies (see e.g., Cichocki et al., 2011, Lyons, 2002, 2012, Smith and Bell, 2005), as a matter of fact, a heat rate indicator is used as reference. Its value does not arise from an analysis of the environment or the spacecraft geometry, rather it only depends on the free-stream conditions (McRonald and Nock, 1977):

$$q_{\text{aero}} = \frac{1}{2} \rho_\infty V_\infty^3 \tag{5.6}$$

As can be seen from the subscript, Eq. (5.6) does not represent the value of the convective heat rate at the stagnation point, but it is used to give an idea of the aerodynamic heating conditions at the spacecraft.

This equation is derived from (Anderson Jr., 2006):

$$q_w = \rho_e V_e \left(h_{a,w} - h_w\right) St \tag{5.7}$$

which is the inverse of Eq. (5.5). By assuming the conditions below (Anderson Jr., 2006), Eq. (5.7) can be simplified to the formula already shown in Eq. (5.6).

- **Parallel Flat Plate**
  Assuming the body is a flat plate parallel to the incoming flow, the conditions at the edge of the boundary layer are the same as the free-stream conditions. Thus, the term $\rho_e V_e$ can be replaced with $\rho_\infty V_\infty$.
- **Unitary Recovery Factor**, $r \approx 1$
  The recovery factor r relates the adiabatic wall enthalpy $h_{a,w}$ to enthalpy the at the edge of the boundary layer h_e and total enthalpy h_0, as:

$$h_{a,w} = h_e + r\left(h_0 - h_e\right)$$

Thus, assuming a unitary recovery rate, leads to the condition $h_{a,w} \approx h_0$, such that no change in enthalpy is experienced through the shock wave and boundary layer.

- **Hypersonic Conditions**, $V_\infty^2 \gg h_\infty$
  By definition, the total enthalpy is given by:
  $$h_0 = h_\infty + \frac{V_\infty^2}{2}$$
  Assuming $V_\infty^2 \gg h_\infty$, the definition of total enthalpy can be simplified to:
  $$h_0 \approx \frac{V_\infty^2}{2}$$
- **Unitary Heat Transfer**, $C_h \approx 1$
  The heat flux at the wall is fully absorbed by the vehicle. Although the average heat transfer of MRO is less than one, its value does locally reach unity at the edges of solar panels, according to results of DSMC simulations by Gallis et al. (2004), Liechty (2007). Thus, using this value allows for accounting of the worst-case scenario.
- **Cold Wall Approximation**, $h_0 \gg h_w$
  This approximation implies that the temperature of the wall is much lower than the temperature of the surrounding gas. This leads to the following simplification:
  $$(h_{a,w} - h_w \approx) \, h_0 - h_w \approx h_0 \left(\approx \frac{V_\infty^2}{2} \right)$$

Given the many simplifications above, one may think that the model from Eq. (5.6) will not perform well in a real-world scenario. It was proven, however, during the previous aerobraking missions, that this indicator does indeed a sufficient job at representing the heating of the spacecraft, and as such, its use is justified.

5.3 | Flow Regimes and Aerodynamics

Based on the Knudsen number introduced in Section 5.1, atmospheric flight can be divided into three regimes (Regan and Anandakrishnan, 1993):

- **Free Molecular Flow**, if $Kn \gg 1$
  In this regime interaction between molecules can be disregarded, since collisions mainly occur between molecules and the vehicle. This regime is also known as rarefied flow and is treated in Subsection 5.3.1 (Regan and Anandakrishnan, 1993).
- **Continuum Flow**, if $Kn \ll 1$
  Intermolecular interactions dominate this regime and the vehicle experiences high thermal load, shock waves and other hypersonic phenomena. This regime is treated in Subsection 5.3.2 (Anderson Jr., 2006).
- **Transition Regime**, if $10^{-3} \leqslant Kn \leqslant 10^{-1}$ and $Re \gg 1$
  The transition regime is what can be found between the free molecular and continuum regimes, and it is dominated by both intermolecular and molecule-surface interactions. This regime is treated in Subsection 5.3.3.

As shown in Appendix C, the Knudsen number reaches the value of 10^{-3} at approximately 75 km, and increases thereafter. Clearly then, the flow conditions experienced in the highest layers of the altitude (above 100 km), are part of the transition regime. Thus, the spacecraft will encounter only free molecular and transition flow. However, since no mathematical representation of the transition regime is present (modeling of this regime is done with bridging functions), the continuum flow still needs to be discussed.

5.3.1 | Free Molecular Flow

When $Kn \gg 1$, the mean free path of the molecules making up the atmosphere is much larger than the characteristic vehicle length, i.e., $\lambda \gg L$. This means that the flow is dominated by collisions between the molecules and the vehicle, that intermolecular interactions can be neglected, and that transfer of momentum and energy between the molecules and the vehicle happens by collisions.

Modeling of molecule-vehicle collisions can be done in two ways:

- **Specular Reflection**
  The particle hitting the surface only transfers momentum in the direction perpendicular to the surface. The velocity component parallel to the surface is unchanged.
- **Diffuse Reflection**
  The particle, after collision, can assume any magnitude and direction of the velocity, given by a Maxwell-Boltzmann distribution.

Assuming that diffuse reflection is the method that represents reality best (which is what experiments show), one can write the pressure and shear stress coefficients acting on an infinitesimal area at an incident angle ϑ, as:

$$C_p = \left\{ \left[\frac{2-\sigma_N}{\sqrt{\pi}} \tilde{s} + \frac{\sigma_N}{2} \sqrt{\frac{T_w}{T_\infty}} \right] \frac{\exp\left(-\tilde{s}^2\right)}{s^2} + \left[(2-\sigma_N) \left(\frac{1}{2} + \tilde{s}^2 \right) + \frac{\sigma_N}{2} \sqrt{\pi \frac{T_w}{T_\infty} \tilde{s}} \right] \frac{1+\operatorname{erf}\left(\tilde{s}\right)}{s^2} \right\} \tag{5.8a}$$

$$C_f = \frac{\sigma_T \cos\vartheta}{\sqrt{\pi} s} \left\{ \exp\left(-\tilde{s}^2\right) + \sqrt{\pi} \tilde{s} \left[1 + \operatorname{erf}\left(\tilde{s}\right)\right] \right\} \tag{5.8b}$$

Note that erf is the error function, and it is defined as (Andrews, 1992):

$$\operatorname{erf}\left(x\right) = \frac{2}{\sqrt{\pi}} \int_0^x \exp\left(-\chi^2\right) \mathrm{d}\chi \tag{5.9}$$

In Eq. (5.8), s is the molecular speed ratio, given by:

$$s = \frac{V_\infty}{\sqrt{2RT_\infty}} = M_\infty \sqrt{\frac{\gamma}{2}} \tag{5.10}$$

and $\tilde{s}$ is the speed ratio perpendicular to the surface, i.e., $\tilde{s} = s \sin\vartheta$. The subscript ∞ refers to free-stream conditions (hence, conditions of the flow unaffected by the vehicle), and T_w is once again, the wall temperature.

Integrating Eq. (5.8) over the surface of the vehicle, results in the values of the forces acting on the body. Alternatively, one can divide the surface of the vehicle into small plates, and compute the force felt by each of these elements, and then sum it to obtain the total. Using either method, one obtains the value of C_D, C_S and C_L, the drag, side and lift force coefficients of the body.

The results that one would obtain with the method explained above are quite approximative. A more reliable and accurate analysis can be done by using the Stochastic Parallel Rarefied-gas Time-accurate Analyzer (SPARTA) software, developed by the Sandia National Laboratories[1] (Gallis et al., 2014). This software is chosen because of its availability (freely-available and open-source, under GNU general public license), and familiarity within the faculty.

SPARTA SPARTA uses a direct simulation Monte Carlo (DSMC) that "*models rarefied gases, using collision, chemistry, and boundary condition models*" (Plimpton and Gallis, 2017). As described by Klothakis and Nikolos (2015), the DSMC method simulates the conditions in a rarefied gas, by using kinetic theory of gases, and in particular the Boltzmann equation, which describes the change in particle distribution, due to collisions. Thus, the DSMC method simulates the motion of millions of particles, each representing a larger number of real particles. The solution obtained with a DSMC method is equivalent to the solution of the Boltzmann equation, when the number of particles tends to infinity and when the mesh and step sizes tend to zero (Bird, 2013). The mesh or grid, is the discretization of the body being analyzed. For SPARTA, for instance, the body is discretized by a triangulated surface mesh (Plimpton and Gallis, 2017). Furthermore, sophisticated versions of the DSMC, just like SPARTA, manage to offer results that closely agree with experimental result, despite the non-ideal conditions (finite number of particles and non-zero mesh and step sizes). However, depending on the geometry of the body, the computation time and accuracy of the results will vary. For instance, a shape that generates large separation regions, will be more computationally challenging than the simulation for a blunt body (Klothakis and Nikolos, 2015).

Theoretically, the DSMC method can be used for both the rarefied and continuum regimes, but for the continuum regime the computation times would be exponentially higher, due to the very large number of intermolecular interactions. Thus, this software is only used for the rarefied regime, and other solutions for the continuum flow will be described in the next subsection.

5.3.2 | Continuum Flow

In situations where $Kn \ll 1$, the flow is dominates by intermolecular interactions, and the relations introduced in Subsection 5.3.1 no longer hold. This regime is referred to as continuum flow, and a body moving through it can

[1] Sandia National Laboratories: SPARTA Direct Simulation Monte Carlo (DSMC) Simulator – https://goo.gl/TqEL8C [last accessed on November 16, 2017]

experience stresses that highly depend on its velocity. In particular, this regime is divided into five categories, based on Mach number (Anderson Jr., 2008):

- **Subsonic**, if $M_\infty < 1$: flow in which the vehicle travels slower than the speed of sound
  - *Incompressible*, if $M_\infty \leqslant 0.3$: flow in which the density is assumed to remain constant
  - *Compressible*, if $0.3 < M_\infty < 1$: flow in which the density can change
- **Transonic**, if $0.8 < M_\infty < 1.2$: flow in which some supersonic effects start to appear locally
- **Supersonic**, if $1 \leqslant M_\infty < 5$: flow in which shock waves and expansion fans are present
- **Hypersonic**, if $M_\infty \geqslant 5$: flow in which high-velocity and high-temperature effects occur

In contrast to a re-entry capsule which experiences all regimes, a spacecraft undergoing aerobraking only experiences hypersonic flow. This is because the spacecraft is always traveling at much higher velocities than the local speed of sound. In Appendix C a simplified analysis is carried out, and it is shown that for a spacecraft in a circular orbit around Mars at a height of 100 km, the Mach number is approximately 13. For this reason, only hypersonic flow is treated.

Based on Anderson Jr. (2006), Regan and Anandakrishnan (1993), the following hypersonic effects can be distinguished:

- **Thin shock wave**: in hypersonic conditions, a thin shock wave will arise close to the body surface; in case of multiple shocks, there may be shock wave interaction; also, the vehicle will experience Prandtl-Meyer expansions
- **Entropy layer**: around the nose of the body, the shock wave is detached and highly curved; high entropy gradients arise, which generate the entropy layer
- **Vorticity interaction**: interaction of entropy and boundary[(2)] layers
- **Viscous interaction**: interaction of boundary layer with outside inviscid flow, and even with the shock wave
- **High-temperature effects**: the kinetic energy of the incoming flow is dissipated in the boundary layer by friction; dissipation generates high temperatures, which cause dissociation and ionization of the molecules and in turn give rise to chemical reactions

Modeling of all these effects requires very precise and computationally expensive software, thus for preliminary design phases, simplified methods are used.

The most simple and famous methods used are grouped under the name of local surface inclination methods. According to these theories, the pressure coefficient felt at a point by an object in a hypersonic flow is only dependent on the inclination of the object at this location w.r.t. the incoming free-stream flow. Thus, $C_p = f(\vartheta)$, and there is no dependence on the free-stream conditions. These methods can be further divided into three categories:

- **Newtonian flow**, where the flow is assumed to transfer all (and only) its normal momentum to the body upon impact
- **Tangent wedge and tangent cone**, where the pressure coefficient is computed by assuming the body is locally approximated with either a wedge or a cone (based on its general shape)
- **Shock expansion**, where the pressure distribution is assumed to be given by a Prandtl-Meyer expansion

Of these, according to Dirkx and Mooij (2017), the best for blunt bodies with a large inclination angle ϑ, is the modified Newtonian method, based on Newtonian flow. The pressure coefficient for this method, is defined by:

$$C_p = C_{p,s} \sin^2 \vartheta \tag{5.11a}$$

where $C_{p,s}$ is given by:

$$C_{p,s} = \frac{2}{\gamma M_\infty^2} \left\{ \left[\frac{(\gamma+1)^2 M_\infty^2}{4\gamma M_\infty^2 - 2(\gamma-1)} \right]^{\gamma/(\gamma-1)} \left[\frac{2\gamma M_\infty^2 - (\gamma-1)}{\gamma+1} \right] - 1 \right\} \tag{5.11b}$$

and it represents the value of the pressure coefficient in the stagnation point behind the normal shock wave, assuming an isentropic deceleration (Dirkx and Mooij, 2017). Clearly, for the modified Newtonian method, C_p is no longer only a function of ϑ, however the $C_{p,s}$ parameter (which depends on M_∞ and γ) needs to be computed only once, since it represents the conditions in the stagnation point.

[(2)] The boundary layer is the region where viscous interaction between airflow and body surface play a major role.

An important factor to consider is shadowing. In case the flow does not directly impact on the surface, the pressure coefficient is assumed to be null. Thus, for a three-dimensional body, where $\hat{\mathbf{n}}$ represents the vector normal to the surface, and $\hat{\mathbf{V}}_\infty$ the direction of the incoming flow, the pressure coefficient in a shadowed zone is given by:

$$C_p = 0 \qquad \text{if } \hat{\mathbf{V}}_\infty \cdot \hat{\mathbf{n}} \geqslant 0 \tag{5.12}$$

where the property $\sin\vartheta = -\hat{\mathbf{V}}_\infty \cdot \hat{\mathbf{n}}$ is used. A slight adjustment was proposed by Gabeaud (1950), to take into account the base pressure, i.e., the pressure felt at the rear of the body. This expression is:

$$C_{p,b} = \frac{2}{\gamma M_\infty^2} \left\{ \left[\frac{1}{M_\infty^2} \left(\frac{2}{\gamma+1} \right) \right]^{1.4} \left[\frac{2\gamma M_\infty^2 - (\gamma-1)}{\gamma+1} \right] - 1 \right\} \tag{5.13}$$

and it represents the correction term, which is only felt by the shadowed parts of the body.

Finally, the pressure coefficient for a three-dimensional vehicle in hypersonic conditions can be approximated by:

$$C_p = \begin{cases} C_{p,s} \left(-\hat{\mathbf{V}}_\infty \cdot \hat{\mathbf{n}} \right)^2 & \hat{\mathbf{V}}_\infty \cdot \hat{\mathbf{n}} < 0 \\ C_{p,b} & \hat{\mathbf{V}}_\infty \cdot \hat{\mathbf{n}} \geqslant 0 \end{cases} \tag{5.14}$$

To find the total aerodynamic force acting on the vehicle, one can thus integrate Eq. (5.14) over the surface. As mentioned in Subsection 5.3.1, an alternative method is to approximate the vehicle in small plates or simple shapes, and sum over the meshed model.

5.3.3 | Transition Flow

No theoretical formulation for the pressure and shear coefficients have been derived for this phase. This is because of the equal importance that intermolecular and molecule-surface interaction play, in this phase of atmospheric flight. In general, DSMC methods are used to estimate the coefficients, but these can be quite computationally expensive (Regan and Anandakrishnan, 1993). On the other hand, simple relationships have been derived from these results, such as the ones detailed by Blanchard and Buck (1986). These relationships act as bridging functions between the results obtained in the continuum and free-molecular regimes. They are usually expressed as a function of Knudsen number, and have the general form of (Regan and Anandakrishnan, 1993):

$$f_x(Kn) = \frac{C_{x,\text{trans}} - C_{x,\text{cont}}}{C_{x,\text{free}} - C_{x,\text{cont}}} \tag{5.15}$$

where x just acts as a place-holder. The subscripts trans, cont and free refer to the transition, continuum and free-molecular flow values, respectively. Thus, one can use Eq. (5.15), to solve for $C_{x,\text{trans}}$, i.e., the coefficient in the transition regime. The function on the left-hand side of Eq. (5.15), is determined by fitting experimental data and DSMC results. The results therefore, vary based on the overall shape of the body. For instance, for the Space Shuttle, Blanchard and Buck (1986) derived the relationship:

$$f(Kn) = \sin^2 \left\{ \frac{\pi}{8} \left[3 + \log_{10}(Kn) \right] \right\} \qquad 10^{-3} \leqslant Kn \leqslant 10^1 \tag{5.16}$$

which can be used for both pressure and shear stress coefficients. Potter (1988), on the other hand, derived two bridging functions for the lift and drag coefficients, which he applied to an aerocapture maneuver with heat shield. In this case, the bridging functions are not a function of Knudsen number, but rather depend on a combination of flow-field and vehicle parameters. The work by Wilmoth et al. (1998), provides a similar solution as Eq. (5.16), but this is also to be applied directly to lift and drag.

Clearly, there exist many possibilities for the expression of the bridging function. However, for sake of simplicity, and for its proven applicability, Eq. (5.16) will be used.

II

Onboard Software

“*Aerobraking: what a breakthrough … the atmosphere!*”

6 Navigation

Moving through space can be a challenging task. There are no roads or signals to follow, and in most cases, spacecraft have no human driver. Therefore, a satellite needs to be able to know where it is and move around without human intervention. In truth, this has rarely been the case for interplanetary probes. Satellites orbiting Earth (especially in low Earth orbit) can use GNSS (Global Navigation Satellite System) to accurately know their position, but in deep space, there are no such systems. Engineers, over the years, have come up with clever ways to overcome this and many other issues that arise from the unusual space conditions. In particular, they developed so-called GNC systems, or guidance, navigation and control systems. These are the part of the spacecraft (both hardware and software) that handles positioning, pointing and path determination problems.

The purpose of the navigation system is to provide the rest of the system with knowledge of the current state of the vehicle. This is generally achieved with the use of sensors and of recursive filtering algorithms (such as the Kalman filter, explained in Appendix E.7). These filters try to reduce the error in position and velocity estimates by combining the data measured by sensors (which is generally corrupted by various error sources), with information obtained by propagating the vehicle state with the environmental models from Section 4.3 and equations of motion from Sections 4.1 and 4.2. The navigation system, finally, also provides information on state-derived parameters, like atmospheric properties.

In this chapter, the navigation system developed for an aerobraking maneuver around Mars will be described, whereas the remaining elements of the GNC system can be found in Chapter 7. The first part of navigation to be discussed are the navigation sensors. These are at the basis of this part of the onboard software, and thus need to be modeled properly. The chosen sensors and their use is explained in Section 6.1. The next part that will be analyzed is the structure of the navigation system. An orbit is divided into various phases, depending on the current position, and on the selected navigation method. Section 6.2 will describe why the need of these partitions arises and how they are defined. Then, in Section 6.3, one will find a description of the state estimation process, based on Kalman filtering, for the various navigation phases of the previous section. Due to the very long duration of an aerobraking maneuver, however, state estimation via filtering may not be enough to keep a low enough position error. Thus, some additional methods need to be discussed. These are explained in Sections 6.4 to 6.6. Finally, Section 6.7 will detail the way that the onboard atmosphere model is adapted every orbit, to account for the varying atmospheric properties.

As will become clear in Chapter 13, the main method investigated in this report, i.e., inertial measurements for aeroassisted navigation (IMAN), did not provide the expected accuracy. The reader will find in this same chapter a detailed description of where and why it is thought that the method in this application fails to reproduce the expected results.

Following from the paragraph above, a secondary method has been implemented. Rather than developing a new navigation system, this method is simply based on the assumption that a position measurement is available, without specifying the particular source of such observation. In Section 6.1, the reader will find a list of possible strategies that could be applied, but they will not be investigated in more detail.

Use of a set of altimeters was also considered in this thesis work, but it soon became clear that altimeters cannot be used as stand-alone external measurements, since their observations do not provide inertial information.

6.1 | Navigation Sensors

Sensors are at the very basis of navigation. They allow for the system to interface with the outside world, by measuring information about the current state and/or conditions of the spacecraft.

For the purpose of IMAN, two main instruments will be used. These are the inertial measurement unit, or IMU, and the star tracker, or ST. To this set of instruments, a generic ranging system, or GRS, is added for the second navigation method investigated in this thesis. One will also find a description of said observation model in this section.

6.1.1 | Inertial Measurement Unit

Inertial measurement units are sensors that measure translational accelerations and rotational velocities. The former is measured by accelerometers, whereas the latter by gyroscopes.

Both of these instruments suffer from errors. In general, these errors can be classified as: bias, scale factor, nonlinearity, hysteresis, noise, etcetera (Jah et al., 2008). They can also suffer from misalignment, which arises both due to placement errors within the IMU itself, and inside the spacecraft. Values for IMU errors are usually given by the manufacturer, however in this case, values from Groves (2013) will be used.

For sake of simplicity, in this analysis not all the errors discussed above will be reproduced. Rather, the simulation will be limited to bias, scale factor, misalignment and noise. With this in mind, if $\mathbf{a}_{\text{non-grav}}|_{\mathcal{I}}$ is the vector of non-gravitational accelerations in the inertial frame, the translational accelerations measured by the IMU are given by (Mooij, 2016):

$$\mathbf{a}_{\text{IMU}}|_{\mathcal{B}} = \mathbf{b}_t + [\mathbf{I} + \operatorname{diag}(\mathbf{s}_t) + \mathbf{M}_t]\, \mathbf{C}_{\mathcal{B}}^{\mathcal{I}}\, \mathbf{a}_{\text{non-grav}}|_{\mathcal{I}} + \mathbf{w}_t \tag{6.1}$$

Here it is assumed that the accelerometers are placed at the CM of the spacecraft, along the axes of the $\mathcal{B}$-frame. Moreover, $\mathbf{b}$ is the bias vector, $\mathbf{s}$ the scale factor, $\mathbf{M}$ the misalignment matrix and $\mathbf{w}$ the uncorrelated white noise vector. The subscript t refers to the translational nature of the acceleration. The definition of these parameters, which are also valid for Eq. (6.3), is:

$$\mathbf{b} = [b_1, b_2, b_3]^{\mathrm{T}} \tag{6.2a}$$

$$\mathbf{s} = [s_1, s_2, s_3]^{\mathrm{T}} \tag{6.2b}$$

$$\mathbf{M} = \begin{bmatrix} 0 & m_{12} & m_{13} \\ m_{21} & 0 & m_{23} \\ m_{31} & m_{32} & 0 \end{bmatrix} \tag{6.2c}$$

Similarly, one can also define the equation for the rotational accelerations. This is given by (Mooij, 2016):

$$\boldsymbol{\omega}_{\text{IMU}}|_{\mathcal{B}} = \mathbf{b}_r + [\mathbf{I} + \operatorname{diag}(\mathbf{s}_r) + \mathbf{M}_r]\, \boldsymbol{\omega} + \mathbf{w}_r \tag{6.3}$$

By inverting Eqs. (6.1) and (6.3), one obtains the non-gravitational accelerations and the rotational velocities, which are the parameters of interest. Finally, transformation from the body frame to the needed frame of reference can be done by applying a combination of the transformations from Appendix A.1. Due to the simplifications done in Subsection 4.3.3, where only Martian gravity and drag are considered for onboard propagation, $\mathbf{a}_{\text{non-grav}}$ virtually represents only the acceleration due to aerodynamic forces.

The modeled pre-calibration IMU errors are given the values in Table 6.1. These values represent the expected errors per axis, where the errors are modeled as random Gaussian variables. Interestingly, the noise depends on the square-root of the sampling rate. Thus, the higher the sampling rate, the higher the noise and in turn, the harder it is to estimate the other IMU errors. In general, according to Groves (2013), the average IMU has a sampling rate of 100 Hz. However, running the state estimator 100 times per second, may be too much of a demand for most space-grade OBCs. Thus, what it is suggested in this case, is to use a sampling rate of 100 Hz for the IMU, but to run the state estimation at a lower rate. For instance, Jah et al. (2008) use a refresh rate for filtering of 20 Hz in their analysis. Choice of this rate will be done in Chapters 12 and 13, where the performance of IMAN and of the generic range-assisted navigation is analyzed.

After talking with the main author of Jah et al. (2008), it became clear that the errors from Table 6.1 were too large to be used for the IMAN process[1]. The method that Jah et al. used to reduce the noise in the data was to bring the IMU data from 100 to 10 Hz, by applying a moving average. This yielded a reduction in noise to about 10^{-5} m s^{-1} at 10 Hz (Jah, 2005). Thus, following this discussion, it was decided to lower the accelerometer noise to account for this post-processing, setting its value to 2×10^{-5} m s^{-2} Hz$^{-0.5}$.

Table 6.1: *Values of main errors (1σ) before calibration in the IMU (Groves, 2013).*

	Accelerometers		Gyroscopes	
Bias	10^{-4}	[m s^{-2}]	5×10^{-9}	[rad s^{-1}]
Scale Factor	10^{-4}	[–]	10^{-4}	[–]
Misalignment	10^{-3}	[–]	10^{-3}	[–]
Noise	2×10^{-4}	[m s^{-2} Hz$^{-0.5}$]	3×10^{-7}	[rad s^{-1} Hz$^{-0.5}$]

[1] Personal communication – October 17, 2018

6.1.2 | Star Tracker

Star trackers are cameras whose purpose is to take pictures of the starry background, and compare the pattern of stars detected with an onboard catalogue. In this manner, one single star tracker can give attitude information around two axes. Hence, for three-axis attitude determination, a pair of two star trackers is needed. The average accuracy of a star tracker can be as high as 1 arcsec (3σ) (Liebe, 1995).

In Table 2.1 it was shown that MRO was equipped two star trackers, which had a refresh rate of 10 Hz[2]. One also needs to model the attitude determination error. According to Lee et al. (2003), the accuracy of the star tracker system onboard MRO was 0.1 mrad (3σ) around its three axes. This value corresponds to about 20 arcsec, and is used as accuracy for the modeling of star trackers.

It is clear that a clash between attitude units is arising. The coordinates chosen to describe attitude both onboard and in propagation are quaternions, whereas the noise is described in radians. A conversion is thus necessary, and is applied with the assumption that the noise is small, and the small angle approximation applies (Sola, 2017):

$$\begin{pmatrix} \eta \\ \boldsymbol{\varepsilon} \end{pmatrix}_{\text{ST}} \approx \begin{pmatrix} \eta \\ \boldsymbol{\varepsilon} \end{pmatrix} \otimes \frac{1}{2} \begin{pmatrix} 2 \\ \mathbf{w}_{\text{ST}} \end{pmatrix} \tag{6.4}$$

where $\mathbf{w}$ expresses a 3×1 Gaussian noise vector with standard deviation given by the MRO data, and $\otimes$ represents the quaternion multiplication operator, whose effect can be expressed as (Sola, 2017):

$$\mathbf{q}_1 \otimes \mathbf{q}_2 = \begin{pmatrix} \eta_1 \eta_2 - \boldsymbol{\varepsilon}_1^{\text{T}} \boldsymbol{\varepsilon}_2 \\ \eta_1 \boldsymbol{\varepsilon}_2 + \eta_2 \boldsymbol{\varepsilon}_1 + \boldsymbol{\varepsilon}_1^{\text{S}} \boldsymbol{\varepsilon}_2 \end{pmatrix} \tag{6.5}$$

where $\mathbf{q}$ is used to represent the full quaternion vector, such that:

$$\mathbf{q} \triangleq \begin{pmatrix} \eta \\ \boldsymbol{\varepsilon} \end{pmatrix}$$

6.1.3 | Generic Ranging System

A ‘generic’ ranging system, or GRS, is a system that provides information on the current position of the spacecraft. Possible candidates for such a system are:

- **Optical Navigation**
  Optical navigation consists of determining position by means of comparing the feed from a camera system, to the expected outcome of this image. For Mars, this can mainly be applied in two scenarios:
  - **High altitudes**: at high altitudes, the navigation system can capture images of the orbiting configuration of Mars, Phobos and Deimos, and compare their relative position to an onboard ephemeris, to determine its own position. This concept has already been tested on MRO[3], but indeed requires a large altitude (since Deimos orbits Mars on average at a distance 20 × 10^3 km[4]).
  - **Low altitudes**: at lower altitudes, feature recognition can be implemented, to match, e.g., craters and other prominent landmarks, to an onboard map. The main disadvantage of this method, however, is its high sensitivity to weather conditions (especially during dust storm season).
- **Relative Satellite Ranging Measurement**
  Alternatively, one could use other satellites orbiting Mars as references w.r.t. which the position can be determined. This type of navigation is usually referred to as liaison navigation (Hill and Born, 2007), and is being investigated for many applications (especially those that classify as three-body problems).
- **Sun-Mars-Star Triad**
  The final method that could be implemented is the combined use of a sun sensor, star sensor and horizon sensor to pinpoint the position of the spacecraft. By assuming that the position of Sun, Mars and the starry background are all known, this would allow the triangulation of the spacecraft position. The accuracy is increased as the angular size of the three objects is small, thus the highest accuracy would be achieved at high altitudes.

[2] Jet Propulsion Laboratory: Mars Reconnaissance Orbiter: Spacecraft Parts: Guidance, Navigation and Control Systems – https://goo.gl/5LfHk6 [last accessed on February 6, 2018]

[3] National Aeronautics and Space Administration: Mars Reconnaissance Orbiter: Optical Navigation Experiment – https://goo.gl/QZUxqc [last accessed on October 2, 2018]

[4] National Aeronautics and Space Administration: Mars Fact Sheet – https://goo.gl/TNo8PJ [last accessed on March 7, 2018]

Of course, use could also be made of the DSN, but this would go against the very purpose of this thesis, as autonomy is what is being sought for.

To make the new method more flexible, it is assumed that position measurements are only available in a certain range of altitudes, and that the accuracy of the measurement is relatively low. The range of altitudes in which no measurements are assumed to be available is from 7500 to 25 × 10^3 km. Moreover, the accuracy of the measurements, when available, is in the order of 500 m (1σ), for each dimension. Thus, modeling of the position observations is done with:

$$\mathbf{r}_{\text{GRS}}\big|_{\mathcal{I}} = \mathbf{r} + \mathbf{w}_{\text{GRS}} \tag{6.6}$$

6.2 | **Navigation Phases**

A navigation phase is defined as a time period, during which a specific navigation technique is used. Setting and switching between different navigation phases is done based on the current time and position. The position clauses will be described later, whereas the time constraint is as follows: no switching between phases shall be allowed within five minutes of the previous phase change. The reason behind this decision is that at times, the position of the spacecraft is only known with a limited accuracy, thus switching between phases may occur at unwanted instances.

Navigation phases have been introduced for two main reasons:

- **Computation Time Constraints**
  Similar to the decision of having two different sampling frequencies inside and outside the atmosphere (which will be explained in Section 8.1), the choice of having multiple navigation phases is also related to the reduction of computational load.
- **Increased Fidelity**
  As mentioned in Subsection 6.1.3, there may be times when the selected navigation method is not available, due to e.g., lack of observations. To increase the degree of realism of the simulation, some of the navigation phases are modeled with a more simplified approach, that does not require external links.

In the next subsections, the reader will firstly find a description of each phase, and then, some considerations on the transition between phases.

6.2.1 | Phase Description

Two main navigation phases have been developed. These are described in the paragraphs below. However, prior to this, a brief introduction is due: one will notice that the first navigation phase is valid for all navigation methods selected (IMAN or GRS-based), whereas the second one describes a general estimation technique that needs to be applied to the chosen navigation method.

Unaided Navigation Phase This navigation phase consists of simply integrating the expected accelerations, computed with the environment models from Subsection 4.3.3, to propagate the state to the next step. Propagation is carried out with Euler integration, i.e., with the formulation from Eq. (E.7). More details about this method will be given in Subsection 6.3.1.

Aided Navigation Phase Assisted navigation is based on the process of sensor fusion: data from an (incomplete) onboard model is combined with information from (error-prone) external sensors, to achieve a more accurate estimate of the spacecraft state. The two main filtering techniques investigated in this report are EKF and UKF, and the two main navigation methods used are IMAN and a generic range-assisted navigation system (or GRS-assisted navigation, or GAN). The algorithms corresponding to these two approaches are explained in more detail in Subsections 6.3.2 and 6.3.3, respectively.

6.2.2 | Phase Transition

As stated in the introduction of this section, phase transitioning is only allowed if the previous phase transition has not occurred in the prior five minutes, and the given reasoning was that at phase interfaces, transition may be triggered multiple times consecutively, due to uncertainty in the spacecraft state. In fact, every phase transition is prompted based on the current altitude of the spacecraft. Table 6.2 shows a detailed overview of the altitude ranges defining the navigation phases for each navigation method: Table 6.2a is dedicated to IMAN, whereas Table 6.2b to GAN.

Table 6.2: *Distribution of navigation phases along each orbit, for both IMAN and GAN.*

(a) *Inertial measurements for aeroassisted navigation.*

Above DAIA	**Below DAIA**
Unaided	Aided

(b) *Generic ranging system-assisted navigation.*

Above	**In-between**	**Below**
25×10^3 km		7.5×10^3 km
Aided	Unaided	Aided

In Table 6.2a, the reader will notice the introduction of a new abbreviation, namely DAIA. This acronym stands for dynamic atmospheric interface altitude, whose definition is given by Eq. (6.20), and will be introduced in Section 6.5.

6.3 | State Estimation

Chapter 4 saw the introduction of the equations of motion for both translational and rotational motion. For the onboard model, the translational motion is expressed in terms of Cartesian coordinates, due to their very simple state derivative expression, whereas the rotational motion is described by a quaternion vector. Note that, since the spacecraft is equipped with a gyroscope (included in the IMU), the rotational velocity is not estimated onboard, but directly taken from this instrument (after being corrected for the bias error, which is indeed included in the state). Thus, the full state for the onboard navigation system is given by:

$$\mathbf{x} = \left[\mathbf{r}^{\mathrm{T}}, \dot{\mathbf{r}}^{\mathrm{T}}, \mathbf{b}_t^{\mathrm{T}}, \eta, \boldsymbol{\varepsilon}^{\mathrm{T}}, \mathbf{b}_r^{\mathrm{T}}\right]^{\mathrm{T}} \tag{6.7}$$

Clearly the state vector also includes some of the inertial measurement unit errors, namely the accelerometer and gyroscope biases. The remaining errors, i.e., scaling and misalignment, are not included, since after a few trials, it was concluded that the filter was not able to converge to their value.

As already stated, the analysis is restricted to the use of Cartesian coordinates as state variables for translational motion. Investigation in the use of the USM elements in filtering was carried out by Vittaldev (2010), with the preliminary conclusion that Cartesian coordinates are better than USM for a ground tracking case. Further enquiry into the comparative performance of these state variables is recommended to properly justify this choice.

In this section, the three main methods used for navigation are explained in more detail. Firstly, in Subsection 6.3.1 the reader will be reintroduced to the concept of unaided navigation. Then, the IMAN method will be laid out in Subsection 6.3.2. Here, the system and observation models will be discussed and the necessary equations will be given. Note that however, this part the analysis will be limited to translational motion. To conclude, Subsection 6.3.3 will explain the generic range-assisted navigation method, by mainly focusing on the differences w.r.t. IMAN.

6.3.1 | Unaided Navigation

As mentioned in Section 6.2, one of the navigation phases is not based on filtering, but simply consists of integrating the expected accelerations. Due to the lack of an estimation process, no system and observation models will be discusses, as they are undefined.

This approach was chosen over integration of the IMU measurements because of the noise in said instrument. Unaided navigation is only activated at high altitudes, where aerodynamic accelerations are negligible. Moreover, the magnitude of the solar radiation pressure is considerably lower (about two orders of magnitude) than the noise level of the accelerometer, thus neglecting the solar radiation leads to a smaller error than including the IMU data.

6.3.2 | Inertial Measurements for Aeroassisted Navigation

One of the main navigation problems with orbiting any astronomical object different from our home world is that knowing the spacecraft position can easily become a challenge. On Earth, one can use one of the many GNSS systems and ground tracking to determine with high accuracy the relative position of a satellite. As soon as one travels too far away from Earth, however, these two methods are no longer available.

For interplanetary missions, a system of large antennas, the deep space network (DSN), is used to pinpoint the spacecraft with great accuracy (see Section 6.6 for more details). Nonetheless, despite its usefulness, this method comes with a major drawback. To continuously know the position of the spacecraft, tracking has to be continuous, as well. This means that the antennas have to be used 24/7, leading to extremely high operational costs.

Here is where IMAN comes in the picture. The spacecraft is equipped with a relatively accurate environment model, which tells the spacecraft which accelerations it is currently experiencing. These are then used as internal measurements for the ever-running onboard filtering process. The external measurements, on the other hand, are given by the accelerometer included in the inertial measurement unit.

As mentioned in the beginning of the section, only translational motion is considered for the IMAN analysis. This is done to limit the analysis to the effects of this method on the root mean square (RMS) position errors. Thus, one can re-write the state vector definition of Eq. (6.7) for the case at hand:

$$\mathbf{x} = \left[\mathbf{r}^{\mathrm{T}}, \dot{\mathbf{r}}^{\mathrm{T}}, \mathbf{b}_t^{\mathrm{T}}, C_D\right]^{\mathrm{T}} \tag{6.8}$$

Other than the lack of the rotational state, Eq. (6.8) also sees the introduction of the drag coefficient C_D in the state vector. This is a parameter that, despite being known with a relatively high confidence for most spacecraft (especially for MRO and MOY[5]), is very useful in the estimation of the error in density modeling.

System Model

The system model for translational motion is simply given by Eq. (4.1a) and Eq. (4.2), where the accelerations are given by the onboard model described in Subsection 4.3.3. Here, one can already see the difference between a conventional navigation system and the system used with IMAN: the acceleration used in the model is not the one given by the IMU, but rather, by an onboard model.

For rotational motion, a more standard method is used. This means that the kinematics of rotational motion are described by Eq. (4.5), and the dynamics are not propagated. In fact, the rotational velocity is directly taken from the IMU, after being corrected for the bias error.

To conclude, the system model equation can be described as:

$$\dot{\mathbf{x}} = \begin{pmatrix} \dot{\mathbf{r}} \\ \ddot{\mathbf{r}} \\ \dot{\mathbf{b}}_t \\ \dot{C_D} \end{pmatrix} = \begin{pmatrix} \mathbf{V} \\ -\mu \frac{\mathbf{r}}{||\mathbf{r}||^3} + \ddot{\mathbf{r}}_{\text{pert}} \\ \mathbf{0} \\ 0 \end{pmatrix} \tag{6.9}$$

where the $\mathbf{0}$ vector is of size 3×1.

Observation Model

The observation model for translational motion is given, as mentioned multiple times in this section, by the IMU measurements. Note that here, these measurements refer to the *external* measurement. The internal (or predicted) observations are given by the accelerations output by the onboard model (where the gravitational acceleration is removed). To make sure that the filter would also estimate the navigation filter errors, the IMU measurements are not corrected for the errors. Rather, the errors are added to the predicted observations themselves. This will be shown in Eqs. (6.10) and (6.11).

In conclusion, the observation model is described by:

$$\mathbf{z} = \mathbf{a}_{\text{non-grav},e}\big|_{\mathcal{I}} \tag{6.10}$$

Of course, depending on when the model from Eq. (6.10) is used, different values need to be input. This is where Table 6.3 comes in handy: it provides a synopsis of where the internal and external observations are taken from.

In case of the internal measurements, the predicted aerodynamic acceleration needs to be corrupted with the accelerometer errors. This is done with a similar expression to Eq. (6.1), where the scaling and misalignment terms are removed (since they are too small and produce a strong non-linear effect that is hard to estimate):

$$\mathbf{a}_{\text{non-grav},e}\big|_{\mathcal{I}} = \mathbf{a}_{\text{non-grav}} + \mathbf{x}^{\mathbf{b}_t} \tag{6.11}$$

Table 6.3: *Summary of internal and external observations for an IMAN-based navigation system.*

	Internal	**External**
Translational	Onboard model, with errors applied	Inertial measurement unit (accelerometer), with errors included

[5] Dr. Jah, M.K., personal communication – October 17, 2018

6.3.3 | Generic Range-assisted Navigation

The approach explained in this subsection is similar to the more conventional approach for satellite navigation. The IMU accelerations, together with an onboard gravity field model, are used as internal measurements, whereas an unspecified ranging system (the GRS from the previous sections), provides the external measurements.

The method of determining the attitude of the spacecraft is also rather conventional, i.e., a set of star trackers is used to figure out the orientation of the spacecraft w.r.t. the starry background (which constitutes a pseudo-inertial reference frame), and these measurements are input to the filtering process as external measurements. The internal measurement is none other than the current estimated quaternion.

System Model

The system model equation is given by:

$$\dot{\mathbf{x}} = \begin{pmatrix} \dot{\mathbf{r}} \\ \ddot{\mathbf{r}} \\ \dot{\mathbf{b}}_t \\ \dot{\eta} \\ \dot{\boldsymbol{\varepsilon}} \\ \dot{\mathbf{b}}_r \end{pmatrix} = \begin{pmatrix} \mathbf{V} \\ -\mu \frac{\mathbf{r}}{||\mathbf{r}||^3} + \ddot{\mathbf{r}}_{\text{grav}} + \mathbf{a}_{\text{non-grav}} \\ \mathbf{0} \\ -\frac{1}{2}\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\varepsilon} \\ \frac{1}{2}\left(\eta\boldsymbol{\omega} - \boldsymbol{\omega}^{\mathrm{S}}\boldsymbol{\varepsilon}\right) \\ \mathbf{0} \end{pmatrix} \tag{6.12}$$

where once again, the $\mathbf{0}$ vectors are both of size 3×1, and the gyroscope errors are removed with:

$$\boldsymbol{\omega}|_{\mathcal{B}} = \boldsymbol{\omega}_{\text{IMU}} - \mathbf{x}^{\mathbf{b}_r} \tag{6.13}$$

Note that the term $\mathbf{x}^{\mathbf{b}_r}$ means that the part of the state vector expressing the bias for rotational motion, i.e., $\mathbf{b}_r$, is to be replaced here. Another important remark is that, due to the definition of the equations of rotational motion, no transformation to inertial frame is needed for Eq. (6.13).

The only difference between Eqs. (6.9) and (6.12) is the split of the term $\ddot{\mathbf{r}}_{\text{pert}}$ into gravitational perturbations (given by the spherical harmonics terms), and the non-gravitational accelerations given by the IMU, after transformation to $\mathcal{F}_{\mathcal{I}}$ and after the bias error, given by $\mathbf{x}^{\mathbf{b}_t}$, has been removed:

$$\mathbf{a}_{\text{non-grav}}|_{\mathcal{I}} = \mathbf{C}_{\mathcal{I}}^{\mathcal{B}}\ \mathbf{a}_{\text{IMU}}|_{\mathcal{B}} - \mathbf{x}^{\mathbf{b}_t} \tag{6.14}$$

which is the equivalent for translational motion of Eq. (6.13).

Observation Model

In the case of range-assisted navigation, the external measurement for translational motion, is simply a vector denoting the current spacecraft position. Thus, the observation model consists of the position and attitude estimated by the onboard computer, i.e.:

$$\mathbf{z} = \begin{pmatrix} \mathbf{x}^{\mathbf{r}}|_{\mathcal{I}} \\ \eta \\ \varepsilon \end{pmatrix} \tag{6.15}$$

For rotational motion, on the other hand, the observation is provided by the star tracker. This means that this part of the observation model is simply a quaternion vector, which, as always, describes the rotation from the $\mathcal{B}$-frame to the $\mathcal{I}$-frame.

One can also rewrite Table 6.3, to adapt the translational observation model, as shown in Table 6.4.

6.3.4 | State Propagation

In some situations, it is necessary to predict the behavior of the spacecraft based on the current or adapted initial conditions. The model used for propagation is the same as the onboard model introduced in Subsection 4.3.3, with the propagation and integration characteristics that follow:

- **Propagator**: unified state model with exponential map (USMEM)
- **Integrator**: Runge-Kutta-Fehlberg 5(6) with 10^{-12} tolerances and minimum and maximum step sizes of 0.1 and 100 s, respectively

Table 6.4: *Summary of internal and external observations for a GRS-assisted Navigation system.*

	Internal	**External**
Translational	Current estimated Cartesian position	Position measurement from one of the systems in Subsection 6.1.3
Rotational	Quaternion in state vector	Star tracker

6.4 | Processing of Accelerometer Measurements

Before proceeding to the remaining navigation software, a brief discussion on how the onboard measurements are analyzed is in order. In fact, before using the accelerometer data, a certain degree of processing is necessary. As mentioned in the previous section, the IMU measurements are affected by a number of errors, namely bias, scaling, misalignment and noise. The first element is estimated by the navigation filter (since it is part of the system model), the second and third ones are too small to be estimated, but for the forth one, something can be done.

The noisy part of the signal can be removed by a very simple smoothing process. As a matter of fact, a moving average (just like the one defined by Eq. (E.2) in Appendix E) using 60 s of sampling data can do a very good job at representing the initial signal. This smoothing process is carried out before running the post-atmosphere processes but after removing the accelerometer errors.

Removal of the accelerometer errors is simply done via the inverse of Eq. (6.1), where scaling and misalignment are disregarded, resulting in the same formulation as Eq. (6.11).

One important fact, however, is which value of $\mathbf{x}^{\mathbf{b}_t}$ to use. In fact, the state estimator keeps updating the estimate of this parameter, based on the current available data. However, this constant refresh leads to some subtle variation in value and consequently to oscillation around a mean value. Hence, the value of $\mathbf{x}^{\mathbf{b}_t}$ for Eq. (6.14) is replaced by the median of the history of estimated errors, where the median is used due to its strong resistance to outliers. Equation (6.14), thus becomes:

$$\mathbf{a}_{\text{non-grav}}\big|_{\mathcal{I}} = \mathbf{C}_{\mathcal{I}}^{\mathcal{B}}\, \mathbf{a}_{\text{IMU}}\big|_{\mathcal{B}} - \tilde{\mathbf{x}}^{\mathbf{b}_t} \tag{6.16}$$

6.5 | Periapse Time Estimator

The function of the PTE is dual: compute the actual periapsis time t_p and compute the change in orbit shape (i.e., semi-major axis and eccentricity offsets) due to drag acceleration. These two functions are performed by analyzing the accelerometer data collected during the atmospheric sweep.

Periapsis Time Correction In a perfect scenario, the periapsis time directly corresponds to the maximum of the drag acceleration curve. Unfortunately, due to accelerometer errors and atmospheric unpredictabilities, this will most likely not be the case. Because of the accelerometer errors, the data will be affected by noise, and because of the ever-changing atmosphere, the data will not be a perfect bell-shaped curve. This makes finding the pericenter a harder task.

The method discussed by Lyons (2001), uses as t_p the time that divides the area of the acceleration curve into two equal parts, i.e., the centroid. To determine this point, a recursive root-finding algorithm is used. Since no information on the derivative of the curve is readily available, the bisection root-finder explained in Appendix E.4 is applied. One of the requirements of the bisection method is that the root-finer function be ever-growing (or ever-decreasing). Were this not to be the case, the algorithm could converge to the wrong answer. Thus, the function used for the root-finding process is given by:

$$f\left(t_{\text{guess}}\right) = \int_{t_{\text{in}}}^{t_{\text{guess}}} ||\mathbf{a}_{\text{aero}}||\,\mathrm{d}t - \int_{t_{\text{guess}}+\Delta t}^{t_{\text{out}}} ||\mathbf{a}_{\text{aero}}||\,\mathrm{d}t \tag{6.17}$$

where t_{guess} is any time between the t_{in} and t_{out} and integration is approximated with the extended Simpson's rule from Appendix E.2.1, since the time steps (Δt) at which the IMU measurements are retrieved are equally spaced. The time bounds of the integral in Eq. (6.17), namely t_{in} and t_{out}, represent the ingress and egress time to and from the atmosphere. Due to the lack of a clear boundary between space and atmosphere, these two parameters are difficult to define. However, one can introduce a fictitious altitude, above which the atmospheric effect is assumed to be negligible. This altitude is called atmospheric interface and for Mars it is defined by REQ-AB-8 at 200 km (this is the altitude where the density reaches 10^{-13} kg m^{-3}, and aerodynamic acceleration for an object like MRO is lower than solar radiation pressure). Finally, the reader can quickly verify that the function above does indeed always match the requirement described before the introduction of Eq. (6.17).

Table 6.5 shows the initial boundary values and the required convergence criteria for the bisection root-finder used for this application, where the relative accuracy represents a difference between two consecutive time guesses of about 0.02 s. The reader will find a more detailed description of the meaning and use of these parameters in the Tudat documentation.

Orbit Shape Correction To fulfill its second function, i.e., the estimation of the change in semi-major axis, a few assumptions need to be made. First, the accelerations due to drag are assumed to only occur at periapsis and secondly (also consequently), they are treated as an impulsive shot. Thus, the simplifications shown in Appendix F.2 apply, and the change in semi-major axis due to an impulsive shot at periapsis is found via:

$$\Delta a = \frac{2}{n}\sqrt{\frac{1+e}{1-e}}\Delta V_{\hat{\mathbf{y}}_{\mathcal{L}}} \tag{6.18a}$$

which is derived (together with Eq. (6.18b)), in Appendix F.3. To complete the correction, the change in eccentricity is also needed, and it is found with:

$$\Delta e = 2\sqrt{\frac{a\left(1-e^2\right)}{\mu}}\Delta V_{\hat{\mathbf{y}}_{\mathcal{L}}} \tag{6.18b}$$

The value of the impulsive shot $\Delta V_{\hat{\mathbf{y}}_{\mathcal{L}}}$, is the velocity change along the y-axis of the $\mathcal{L}$-frame and it is obtained by integrating the drag acceleration. Since the acceleration provided by the IMU is discretized, its value is found by approximating the integral:

$$\Delta V_{\hat{\mathbf{y}}_{\mathcal{L}}} = \kappa \int_{t_{\text{in}}}^{t_{\text{out}}} ||\mathbf{a}_{\text{aero}}||\, \mathrm{d}t \tag{6.19}$$

with a summation, such as the extended method mentioned earlier. Note how this equation is corrected by a constant κ. This constant is used to account for the fact that the drag acceleration is not an impulsive shot at periapsis, but a continuous process that can last several minutes, and is centered at pericenter. The constant κ was determined empirically, and was given the value of 0.955 by Chabot et al. (2013), and of 0.96 by Lyons (2001). The usage of this constant factor becomes especially useful for low period orbits, where the time spent in the atmosphere is a considerable fraction of the orbital period. The value of 0.955 is the one chosen for the present PTE implementation.

Correction Timing In the paragraphs above, it is stated that the estimated velocity change is computed by integrating the measured (smoothed) accelerations from ingress and egress time to and from the atmosphere. One would therefore expect the PTE to be executed right at t_{out}. However, during the first trials of running this correction algorithm, it became clear that running the PTE while leaving the atmospheric interface altitude, is too premature. The main reason for this is the effect of the spherical harmonics of the planet. As one can see from Fig. 6.1, the acceleration arising due to the spherical harmonics terms (hence, with n and m greater than 0) starts acting on the semi-major axis derivative well before the aerodynamic acceleration reaches a considerable value. The same holds for when the spacecraft is leaving the atmosphere. Therefore, one can conclude that the harmonic gravity terms have a longer-lasting impact.

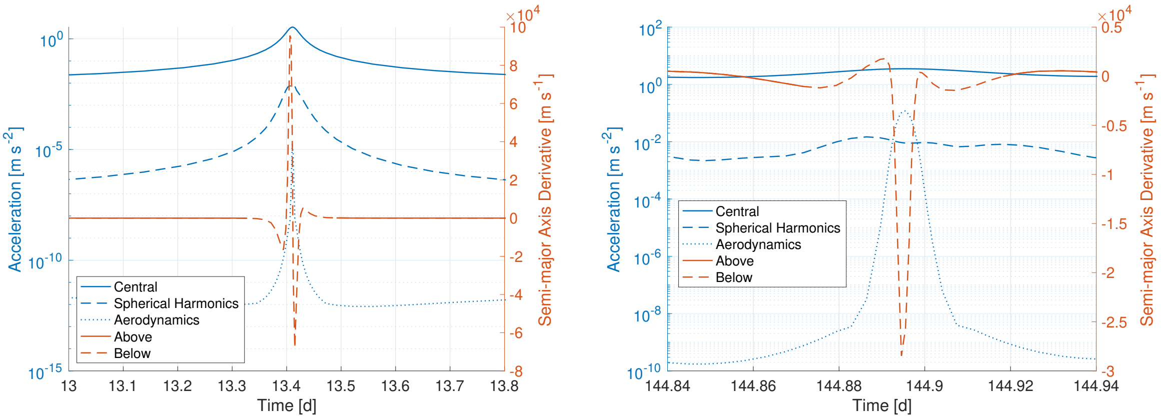
Thus, if the PTE corrections were applied right at the exit of the atmosphere, the value of the semi-major axis would still be feeling the effect of the conservative force terms. Whereas if one were to wait for the spherical harmonics effects to dissipate, the value of a would reach its 'true' post-atmosphere value, and the effect of drag could be quantified more easily.

Based on the discussion above, it was decided to perform the correction whenever the altitude of the spacecraft goes above the threshold set by:

$$h_{\text{DAIA}} = 0.5\left[1+\left(1-\frac{a}{a_0}\right)\right](a-R) \tag{6.20}$$

Table 6.5: *Bisection root-finder settings for the PTE estimation of periapse time.*

Initial Boundaries		**Relative Accuracy**	**Maximum Iterations**
Lower [s]	*Upper* [s]		
t_{in}	t_{out}	8.46×10^{-10}	25



(a) *High eccentricity (*$e \approx 0.869$*). DAIA* $\approx 11.9 \times 10^3$ km.

(b) *Low eccentricity (*$e \approx 0.176$*). DAIA* ≈ 720 km.

Figure 6.1: *Effect of spherical harmonics gravity on the derivative of the semi-major axis as a function of time, centered at periapsis, for two different eccentricity regimes. Note that the solid blue line represents the central gravitational acceleration, and the dashed blue line the higher spherical harmonics degrees and orders (up to 4). In contrast, the solid red line represents the period above the dynamic atmospheric interface altitude.*

i.e., 50 % of the osculating average altitude, with a correction based on how deep into the aerobraking phase the spacecraft currently is. This is an empirically determined formula, which gives the expected result at both high and low values of eccentricity, as it is shown in Figs. 6.1a and 6.1b, respectively. In these figures, one can see that the derivative of the semi-major axis above this dynamic atmospheric interface altitude (DAIA), is roughly zero, and thus, the value of semi-major axis at this condition can be assumed to be constant until the next DAIA.

Note that the same conclusion on DAIA can be achieved if the change in eccentricity over time is used, instead of the change in semi-major axis. Furthermore, it should be stated that the plots from Fig. 6.1 are taken from the propagation of the aerobraking test case (i.e., the fifth test case) from Section 10.3.

CORRECTION APPLICATION The way the three values given by the PTE (t_p, Δa and Δe) are implemented in navigation, is to simply correct the estimated Keplerian elements.

Firstly, the periapsis time needs to be converted to true anomaly to be used by the rest of the navigation system. Thus, the estimated t_p is matched to the corresponding ϑ_p, by matter of simple cubic spline interpolation. Finally, this estimated ϑ_p becomes the correction factor and is simply added to the current Keplerian state.

Then, one can correct the estimated Keplerian state. This is done by using the following elements as starting point for the correction:

- a and e: taken as the values at the previous apoapsis; since the atmosphere is only assumed to act instantaneously at pericenter, the orbit is supposed to be a simple Kepler orbit, with constant elements
- i, ω, Ω and ϑ: taken as the values at t_{out}; despite the fact that the drag acceleration does not largely affect these elements, they are taken as the final available estimate to account for other accelerations, such as spherical harmonics

Thus, the correction is done with the following formulation:

$$[a, e, i, \omega, \Omega, \vartheta]^{\mathrm{T}}_{\text{out}} = \begin{pmatrix} a_a \\ e_a \\ i_{\text{out}} \\ \omega_{\text{out}} \\ \Omega_{\text{out}} \\ \vartheta_{\text{out}} \end{pmatrix} + \begin{pmatrix} \Delta a \\ \Delta e \\ 0 \\ 0 \\ 0 \\ \vartheta_p \end{pmatrix} \tag{6.21}$$

6.6 | Deep Space Network

Despite the discussion done before about the high cost of the DSN, its use is still fundamental. In spite of the correction carried out by the PTE every orbit, at the exit of the atmosphere interface, the error of the navigation system keeps cumulating. Thus, the DSN is used every few days to reset this error and restart the navigation filter.

The way this part of the system is modeled is by feeding the real position of the spacecraft at a previous time

Table 6.6: *Values of 1σ accuracy for DSN tracking (position:* Berner et al. 2007*, velocity:* Iess et al. 2012*).*

Position [m]	Velocity [m s^{-1}]	Light Time [s]
10.0	0.01	10^{-3}

(a time corresponding to the latest DSN tracking measurement) to the OBC, after noise has been introduced. This noise is based on a Gaussian distribution with zero mean and standard deviation from Table 6.6.

One may notice that the noise in Table 6.6 is a factor of ten larger than the errors described in the cited reference. This is to make sure that even in less than optimal conditions, the algorithm that is being tested still functions. Also, the accuracy in Berner et al. (2007), Iess et al. (2012) is mainly achieved after long communication periods, which may not always be available for the case at hand.

Another important effect to consider is the light-time error. This element accounts for the fact that communication between the spacecraft and Earth has a time delay, which depends on the distance between Mars and the DSN antennas. To take this into account, the position at a previous time is fed to the OBC, which then propagates it with the onboard model from Subsection 6.3.4 to the current time, thus introducing a small error.

Some considerations on the use of the DSN network are in order. In reality, the position and velocity of the spacecraft are measured as ranges, where the velocity is only the component parallel to the tracking direction. However, combining this information with the current elevation and azimuth of the spacecraft and a more sophisticated model of the Martian environment, one can determine the full Cartesian state of the spacecraft. This is what is assumed to be sent to the OBC.

6.7 | Atmosphere Estimation

The estimation of the atmosphere is done by first retrieving the atmospheric density from the measured aerodynamic acceleration, and then fitting the data points with one of the least squares methods from Appendix E.6.

To retrieve the density, the smoothed and corrected accelerations measured by the IMU are used as input to:

$$\rho = \frac{2m}{AV_A^2} C_D \left|\left|\mathbf{a}_{\text{aero}}\right|\right| \tag{6.22}$$

to determine the estimated value of density. The main sources of errors in estimating the density, then arise from the noise in the accelerometer measurements. Despite the beneficial effect that smoothing, i.e., the application of a one-minute moving average to the bias-free data, has on the measurements, the effect of noise is still visible in the data, as clearly shown in Fig. 6.2. In fact, it was decided to introduce a second atmospheric interface altitude, called reduced atmospheric interface, which was set to 150 km above ground. The rationale for taking this value is well explained by the same figure, where the IMU acceleration is plotted together with the actual aerodynamic acceleration.

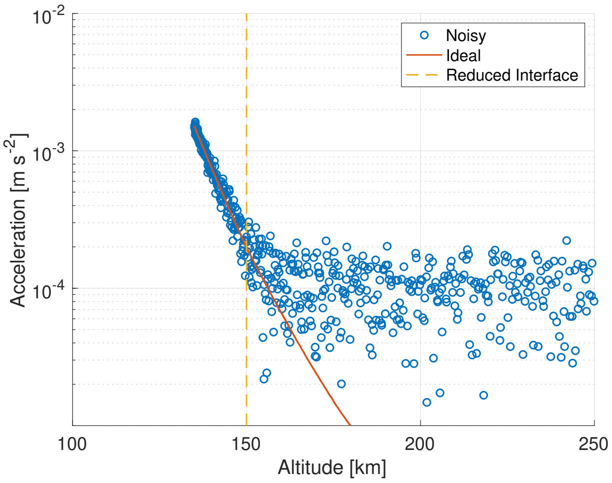


Figure 6.2: *Comparison of the acceleration measured by the IMU and the actual aerodynamic acceleration, used for the definition of the reduced atmospheric interface (represented by the dashed yellow line).*

Fitting of the computed density is then done with least squares. In case of the exponential model from Eq. (4.29), a simple linear least squares can be applied, if this equation is first transformed to logarithmic space (as explained in Appendix E.6.3). For the more complex Eq. (4.33), even after transformation to logarithmic space, linear least square fitting performs very poorly, due to the presence of the sinusoidal terms. Thus, the non-linear Levenberg–Marquardt method (explained in Appendix E.6.2) is applied. To apply either estimation process, however, knowledge of the Jacobian of the system is necessary. This matrix represents the derivative of the density function w.r.t. the density parameters that are to be estimated, and they are derived in Appendix F.1.

After the parameters of the atmosphere model (such as h_0, ρ_0 and H for the exponential model) are computed for the current atmospheric sweep, this information is combined with the data from the previous six orbits, by taking the average of the results. Doing so will allow for a more stable atmosphere estimation, which, on the other hand, is also capable of incorporating higher than expected density fluctuations.

7 Guidance and Control

The remaining parts of the onboard software that need to be discussed are the guidance and control systems. These two elements are discussed in Sections 7.1 and 7.2, respectively.

In general, one would find the description of the components of a GNC system in a different order, namely the order dictated by the initialism itself. However, during the thesis work, it became clear that 1) the amount of work to be dedicated to each of these systems did not match the 'GNC' order, and 2) that the development of a robust navigation system would be a much more critical part of the research, given the various challenges encountered in the Martian environment.

Thus, it should not come as a surprise that it was decided to dedicate a whole chapter to navigation, and to combine the G and C elements in this chapter.

7.1 | Guidance

The role of the guidance system for an aerobraking maneuver is to determine the magnitude and direction of the corrective apocenter maneuvers, to achieve a gradual decrease of the apoapsis altitude, while respecting the heating and lifetime constraints dictated by the spacecraft material properties.

The reader will find that the objective above has been divided in two main tasks for guidance:

- Estimate the **periapsis altitude corridor** based on requirements REQ-AB-4 through REQ-AB-7 and the current **aerobraking phase**
- Target the midpoint of the corridor with an **apoapsis maneuver**

These two tasks are explained in Subsections 7.1.2 and 7.1.3, respectively. Prior to this, however, it is worth spending some time learning about the concept of aerobraking phase. This is done in Subsection 7.1.1.

Based on the requirements listed in Table 2.2, the targeting information and constraints in Table 7.1 have been extracted. These elements act as the main drivers for the guidance system developed in this thesis. One will notice that REQ-AB-4 and REQ-AB-5 differ from the values of maximum allowed heat rate and heat load. This is because the strategy of Lyons (2012) is applied: the maximum allowed guidance values are set as half of the actual maximum values the spacecraft is capable of coping with.

7.1.1 | Aerobraking Phases

In Chapter 6 it was shown how an orbit is divided into a few phases, based on the current spacecraft state. As it turns out, however, to create a robust guidance system, the aerobraking maneuver itself has to be partitioned, as well. In

Table 7.1: *Main drivers for the guidance system developed in this thesis, for an aerobraking maneuver around Mars, for a spacecraft similar in size, shape and initial conditions to MRO.*

Target Periapsis Altitude	Target Apoapsis Altitude	Maximum Allowed Heat Rate	Maximum Allowed Heat Load	Minimum Allowed Dynamic Pressure	Minimum Allowed Predicted Lifetime
$\check{h}_p$ [km]	$\check{h}_a$ [km]	$\check{q}_{\text{max}}$ [W m^{-2}]	$\check{Q}_{\text{max}}$ [kJ]	$\check{\bar{q}}_{\text{min}}$ [N m^{-2}]	$\check{L}_{\text{min}}$ [d]
255	320	1400	250	0.19	2.0

fact, it is divided into five phases: the first four are the same ones that were chosen for MRO, and the final one is introduced to make sure that the final orbit configuration is as expected. The phases can be summarized as:

- **Walk-in**: period in which the periapsis is slowly lowered in the highest layers of the atmosphere, to probe the atmospheric parameters
- **Main**: phase during which most of the braking occurs; no special precautions or actions need to be taken
- **Walk-out**: part during which the lifetime constraints, dictated by REQ-AB-7, take effect
- **Termination**: orbit during which the periapsis is raised outside the atmosphere and the apoapsis is targeted to its final value
- **Aerobraking complete**: final orbit used to check the achieved orbit configuration; no special precautions or actions need to be taken

In the paragraphs below, the reader will find a more detailed description of the aerobraking phases which are deemed most interesting. In Table 7.2, on the other hand, one will find a summary of the conditions that activate each phase.

Walk-in

During walk-in, the main objective is to initialize the atmosphere estimator, in order to minimize the risk of over- and (especially) underestimating the encountered density. This is done by spending the first few orbits (seven orbits to be exact, as recommended by Murri (2013)) probing the atmosphere, by slowly lowering the periapsis. This slow reduction is achieved by multiplying the estimated corridor boundaries (see Subsection 7.1.2 for an explanation on how these are determined), with a simple constant:

$$\kappa_p = \begin{cases} 1.2 - \frac{0.2}{7}\mathcal{N} & \mathcal{N} < 7 \\ 1.0 & \text{otherwise} \end{cases} \tag{7.1}$$

where $\mathcal{N}$ represents the number of orbits performed. Note that the constant factor 1.2 is an empirical value and was chosen to simply introduce an offset between the actual and effective estimated corridor boundaries.

Walk-out

As mentioned at the beginning of this subsection, during walk-out an extra constraint is activated. This is the lifetime constraint, which dictates that the spacecraft lifetime shall never be lower than two days.

The effect of this phase is thus the addition of an extra check on the lower bound of the periapsis corridor. The spacecraft orbit is propagated for three days, and if orbit decay is detected before the two day threshold, the lower corridor boundary is rejected and the estimation process is repeated with the lifetime constraint, instead.

This phase becomes active whenever the value of eccentricity goes below the 0.3 threshold. Similarity to the number of orbits for the walk-in phase, this number was determined empirically. The choice of this value becomes clear when looking at Fig. 7.1, which depicts the evolution of eccentricity and the drop in apoapsis altitude per orbit (expressed as a percentage of the apoapsis altitude itself) during an uncontrolled aerobraking maneuver.

Clearly, the change in apoapsis altitude becomes very extreme towards the end of the maneuver. As soon as the eccentricity goes below roughly 0.4 in value, the drop in apoapsis per orbit accounts for more than 5 % of the apoapsis altitude, itself. One may wonder then, why 0.4 was not selected as walk-out threshold, but one should also keep in mind the uncontrolled nature of the result in Fig. 7.1. In fact, in the last days of aerobraking, the periapsis altitude is in the range of 90 to 100 km, i.e., well below the target periapsis usually output by the maneuver estimator.

Termination

The termination phase lasts for one orbit only, and is initiated when the apoapsis altitude reaches a value within 25 % of the target $\check{h}_a$, conveyed by REQ-AB-2. Here the symbol $\check{\square}$ is used to represent the target value.

Table 7.2: *Summary of aerobraking phases, their activation conditions and the priority used to determine phase order.*

	Walk-in	Main	Walk-out	Termination	Aerobraking Complete
Condition	$\mathcal{N} < 7$	Otherwise	$e < 0.3$	$h_a < 1.25\,\check{h}_a$	Previous phase is *termination*
Priority	4	5	3	2	1

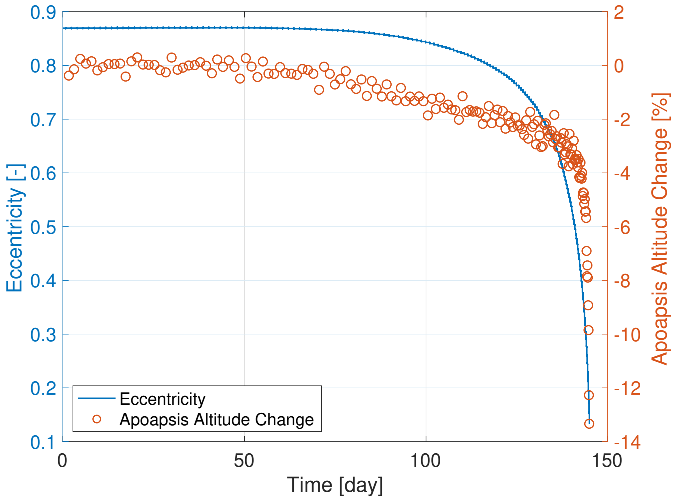


Figure 7.1: *Evolution of the value of eccentricity and apoapsis altitude change, for an uncontrolled aerobraking maneuver, based on the initial conditions of the fifth test case of Table 10.1.*

Once this phase is detected, the spacecraft will not run the corridor estimator, but directly perform an apoapsis maneuver, to raise the pericenter outside the atmosphere. Aim is of course set toward the target periapsis altitude, given by REQ-AB-3. Once said pericenter is reached, the periapsis maneuver is performed. The purpose of this action should be clear by now, and it is to reduce the apoapsis altitude to its target condition.

7.1.2 | Corridor Estimator

Estimation of the corridor boundaries is done by propagating the state of the spacecraft for one orbit, to estimate the heating and pressure conditions endured by the structure.

The estimation itself is carried out with the bisection root-finder of Algorithm 1, in Appendix E.4. The root-finder tries to guess the periapsis altitude, based on the current estimated state for propagation, where the first two Keplerian elements are adapted according to:

$$r_{p,\text{guess}} = h_{p,\text{guess}} + R \tag{7.2a}$$

$$a_0 = \frac{r_a + r_{p,\text{guess}}}{2} \tag{7.2b}$$

$$e_0 = \frac{r_a - r_{p,\text{guess}}}{r_a + r_{p,\text{guess}}} \tag{7.2c}$$

and the others simply keep their latest estimated value. This state is then propagated for two-thirds of the orbital period by using the state propagator of Subsection 6.3.4, and the resulting aerothermodynamics conditions are analyzed.

For the upper altitude bound, the maximum dynamic pressure computed during propagation is compared to the minimum allowed value of Table 7.1. This gives the root-finder function:

$$f_{\text{upper}}\left(h_{p,\text{guess}}\right) = \bar{q}_{\text{max}} - \check{\bar{q}}_{\text{min}} \tag{7.3}$$

For the lower altitude, in contrast, the heating conditions are used. Thus, the root-finder formula becomes:

$$f_{\text{lower}}\left(h_{p,\text{guess}}\right) = \max\left(q_{\text{max}} - \check{q}_{\text{max}}, Q_{\text{max}} - \check{Q}_{\text{max}}\right) \tag{7.4}$$

In case the lifetime constraint is also active, i.e., the current orbit aerobraking phase is set to *walk-out*, the orbit is propagated nominally (thus with the current estimated state as initial conditions) for three days (namely, 1.5 times the minimum lifetime). If a decay is detected, the lower bound estimation is re-run, but this time the propagation is also carried out for three days, and the only parameter that is analyzed is the lifetime:

$$f_{\text{lower}}\left(h_{p,\text{guess}}\right) = L - \check{L}_{\text{min}} \tag{7.5}$$

The root-finding procedures are run with the settings shown in Table 7.3, where the value of 10^{-3} corresponds to a 100 m difference at 100 km altitude.

Table 7.3: *Bisection root-finder settings for the improvement of the corridor boundary estimates.*

Case		**Initial Boundaries**		**Relative Accuracy**	**Maximum Iterations**
		Lower [km]	*Upper* [km]		
Lower Bound	Heat	90	130	10^{-3}	25
	Lifetime	100	140		
Upper Bound		100	130		

Due to the assumption of Eq. (7.2), there will be an offset between the propagated periapsis altitude (resulting from using the conditions in Eq. (7.2) as initial conditions for a $^{2}/_{3}$ orbital period propagation), and the altitude guess $h_{p,\text{guess}}$ used by the bisection algorithm. If one were to plot such difference in form of a ratio between guessed and propagated altitude, one would get what is shown in Fig. 7.2.

From this plot, it is clear that there is a roughly linear relation between the ratio and the altitude guess, in the range of altitudes shown. Thus, one can use this information to correct the root-finder guess, and improve the estimate of the corridor boundaries. Said correction is found by simple linear least squares, and yields the following relation (which is only applied at the end of the root-finding iterative process):

$$\kappa_h = \max\left[1.0,\, \left(0.952 + 2.110 \times 10^{-7}\, h_{p,\text{guess}}\right)^{-1}\right] \tag{7.6}$$

This correction is important, since Fig. 7.2 shows that the altitude guess is lower than the propagated altitude. Thus, the guidance algorithm would be aiming an altitude that is lower than the actual altitude that is being probed for heating conditions.

One can also see from Eq. (7.6), that the maximum between the linear regression and 1.0 is used as correction factor. This is to make sure that the lowest possible value of the factor is indeed unitary.

Since the onboard environment model keeps evolving, so do the ratios in Fig. 7.2. For this reason, Eq. (7.6) is updated every orbit, based on the guessed-to-propagated altitude ratios.

In the end the results are the upper and lower periapsis altitudes, $h_{p,\text{upper}}$ and $h_{p,\text{lower}}$. It may happen, especially during the walk-out phase, that the lower bound is actually higher than the upper bound. In this case, the restriction on dynamic pressure is discarded, and the upper bound is replaced with the lower bound, where a margin of 2.5 km is applied.

Now that the corridor thresholds have been determined, the target periapsis altitude can be found. This will be the value that is fed to the maneuver estimation, in case a maneuver needs to be performed. In fact, if the predicted periapsis altitude already falls between the bounds given by the root-finding process, no maneuver is carried out.

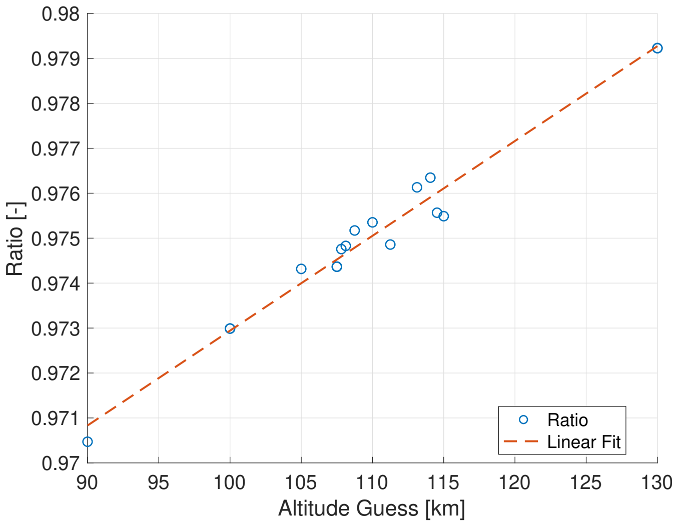


Figure 7.2: *Ratio of altitude guess given as input to the bisection algorithm, and altitude resulting from the propagation of the conditions in Eq. (7.2), for the first corridor estimation performed by onboard computer.*

However, if such condition is not met, the target periapsis value is set to the midpoint of the corridor, i.e.:

$$\check{h}_{p,\mathcal{N}} = \frac{h_{p,\text{upper}} + h_{p,\text{lower}}}{2} \tag{7.7}$$

where, once again, $\mathcal{N}$ denotes the current orbit number.

7.1.3 | Maneuver Estimator

The maneuver estimator is that part of the guidance system responsible for the determination of the magnitude and direction of the apo- and periareion maneuvers. The apoapsis maneuver is the trajectory correction that is applied at apocenter, in case the expected pericenter falls outside the corridor estimated by the algorithm of Subsection 7.1.2. In contrast, the periapsis maneuver is only carried out once, at the very end of aerobraking, and it is used to target the final apocenter altitude. In this subsection, the computation of the magnitudes of both of these corrections is described.

Once the final value of the maneuver magnitude is estimated, it is converted to vector format and added to the apsis state of the spacecraft:

$$\left.\mathbf{V}^{\dagger}\right|_{\mathcal{I}} = \mathbf{V} + \mathbf{C}_{\mathcal{I}}^{\mathcal{L}} \left.\Delta\mathbf{V}\right|_{\mathcal{L}} \tag{7.8a}$$

$$\left.\Delta\mathbf{V}\right|_{\mathcal{L}} = \left[0, \Delta V_{\hat{\mathbf{y}}_{\mathcal{L}}}, 0\right]^{\mathrm{T}} \tag{7.8b}$$

where the marking $\square^{\dagger}$ indicates the post-maneuver value, where it is assumed that no modeling or alignment errors in the thruster system are present, and where the value of $\Delta V_{\hat{\mathbf{y}}_{\mathcal{L}}}$ is determined based on the procedures below. Furthermore, the term $\mathbf{C}_{\mathcal{I}}^{\mathcal{L}}$ represents the rotation from $\mathcal{F}_{\mathcal{L}}$ to $\mathcal{F}_{\mathcal{I}}$, and it is derived at the end of this subsection.

Apoapsis Maneuver

Once the target periapsis altitude is known, the maneuver magnitude can be computed. This task is carried out in two steps:

- **Preliminary Maneuver Magnitude Estimate**
  The first step is to estimate the maneuver magnitude by assuming a Kepler orbit, and an execution at apoapsis. These allow to use the simplifications delineated in Appendix F.2, to obtain the following formulation for the estimated preliminary maneuver magnitude:

$$\Delta V_{\hat{\mathbf{y}}_{\mathcal{L}}} = n \frac{\Delta r_p}{4} \sqrt{\frac{1+e}{1-e}} \tag{7.9}$$

  Here, the symbol $\Delta V_{\hat{\mathbf{y}}_{\mathcal{L}}}$, denotes a maneuver whose magnitude is only along the $\hat{\mathbf{y}}_{\mathcal{L}}$ axis, which at apoapsis, coincides with $\hat{\mathbf{x}}_{\mathcal{T}}$ (assuming a null wind speed). Moreover, Δr_p is the expected difference in periapsis altitude after and before the maneuver, i.e.:

$$\Delta r_p = \check{r}_{p,\mathcal{N}} - r_p$$

  where the value of $\check{r}_{p,\mathcal{N}}$ comes directly from the corridor estimator of Subsection 7.1.2 (after the planetary radius R is added to the target altitude).

- **Improved Maneuver Magnitude Estimate**
  Equation (7.9) was derived after a series of assumptions, which lead to a rather simplified result. Given the sensitivity of maneuver to heat loads, however, it is deemed important to also account for the environment perturbations from Subsection 4.3.3. To do this, the use of another bisection root-finder is employed. This root-finder works by propagating the orbit (see Subsection 6.3.4) with a varying maneuver magnitude and by comparing the propagated periapsis altitude to the target value, to fine-tune the value of $\Delta V_{\hat{\mathbf{y}}_{\mathcal{L}}}$:

$$f\left(\Delta V_{\hat{\mathbf{y}}_{\mathcal{L}},\text{guess}}\right) = h_p - \check{h}_{p,\mathcal{N}} \tag{7.10}$$

  The settings used for this root-finder are displayed in Table 7.4. It is worth mentioning, however, that the root-finder is only run in case the magnitude of the preliminary maneuver is larger than 0.15 N. This cutoff value was introduced because it was noticed that below 0.15 N the improvement was insignificant.

Table 7.4: *Bisection root-finder settings for the improvement of the maneuver magnitude estimate.*

Initial Boundaries		**Relative Accuracy**	**Maximum Iterations**
Lower [m s⁻¹]	*Upper* [m s⁻¹]		
$2\,\Delta V_{\hat{\mathbf{y}}_{\mathcal{L}}}$	$0.5\,\Delta V_{\hat{\mathbf{y}}_{\mathcal{L}}}$	10^{-3}	25

Periapsis Maneuver

The procedure used to determine the maneuver at pericenter is virtually the same as shown above. The main difference is found in Eq. (7.9), but it can be easily demonstrated, following a similar procedure as detailed in Appendix F.2, that the preliminary periapsis maneuver magnitude is given by:

$$\Delta V_{\hat{\mathbf{y}}_{\mathcal{L}}} = n \frac{\Delta r_a}{4} \sqrt{\frac{1-e}{1+e}} \tag{7.11}$$

where the value of Δr_a is found with:

$$\Delta r_a = \check{r}_a - r_a$$

and in turn, $\check{r}_a$ is simply taken from Table 7.1, after the planetary radius R is added to the value of the target altitude, and r_a, on the other hand, is the expected apoapsis height.

Rotation from Local Orbit to Inertial Frame

To express this rotation, two steps are taken. First, the rotation is carried out from $\mathcal{F}_{\mathcal{L}}$ to $\mathcal{F}_{\mathcal{T}}$, then from $\mathcal{F}_{\mathcal{T}}$ to $\mathcal{F}_{\mathcal{I}}$. The first rotation is straightforward and it is shown in Eq. (A.7) of Appendix A.1, where the transformations between the reference frames used in this report are delineated. For the second transformation, a longer discussion is necessary.

It is known that the DCM matrix, namely $\mathbf{C}_{\mathcal{I}}^{\mathcal{T}}$, can be determined by expressing the axes of the $\mathcal{T}$-frame in the inertial frame. In the paragraphs below, this method is used to derive the rotation.

First, the $\hat{\mathbf{x}}_{\mathcal{T}}$ axis can be found, and it is simply equal to:

$$\hat{\mathbf{x}}_{\mathcal{T}}|_{\mathcal{I}} = \hat{\mathbf{V}}_A \tag{7.12a}$$

Then, one can find the relation for $\hat{\mathbf{z}}_{\mathcal{T}}$, which by definition, lies in the orbital plane, pointing toward the planet. Hence, one can use the radial distance, from which the component along the velocity vector is subtracted, i.e.:

$$\hat{\mathbf{z}}_{\mathcal{T}}|_{\mathcal{I}} = -\left[\hat{\mathbf{r}} - \left(\hat{\mathbf{r}} \cdot \hat{\mathbf{V}}_A\right) \hat{\mathbf{V}}_A\right] \tag{7.12b}$$

Finally, the remaining axis is found with the right-hand rule:

$$\hat{\mathbf{y}}_{\mathcal{T}}|_{\mathcal{I}} = \hat{\mathbf{z}}_{\mathcal{T}} \times \hat{\mathbf{x}}_{\mathcal{T}} \tag{7.12c}$$

A quick check of the transformation, is to verify that the resulting $\hat{\mathbf{y}}_{\mathcal{T}}$ does indeed point out-of-plane. To do this, one can expand Eq. (7.12c) (where the subscript A is dropped, for sake of clarity) as:

$$\hat{\mathbf{y}}_{\mathcal{T}}|_{\mathcal{I}} = \left[\left(\hat{\mathbf{r}} \cdot \hat{\mathbf{V}}\right) \hat{\mathbf{V}} - \hat{\mathbf{r}}\right] \times \hat{\mathbf{V}} = \left(\hat{\mathbf{r}} \cdot \hat{\mathbf{V}}\right) \hat{\mathbf{V}} \times \hat{\mathbf{V}} - \hat{\mathbf{r}} \times \hat{\mathbf{V}} = -\hat{\mathbf{H}}$$

Clearly, the $\hat{\mathbf{y}}_{\mathcal{T}}$ corresponds exactly to the unit vector in the opposite direction of the specific angular momentum, which is by definition out-of-plane.

Hence, the full rotation from the $\mathcal{T}$-frame to $\mathcal{F}_{\mathcal{I}}$ is expressed as:

$$\mathbf{C}_{\mathcal{I}}^{\mathcal{T}} = \left[\hat{\mathbf{x}}_{\mathcal{T}}|_{\mathcal{I}}\,,\, \hat{\mathbf{y}}_{\mathcal{T}}|_{\mathcal{I}}\,,\, \hat{\mathbf{z}}_{\mathcal{T}}|_{\mathcal{I}}\right] \tag{7.13}$$

Finally, combining the above relation with Eq. (A.7), leads to the full transformation matrix:

$$\mathbf{C}_{\mathcal{I}}^{\mathcal{L}} = \mathbf{C}_{\mathcal{I}}^{\mathcal{T}} \mathbf{C}_{\mathcal{T}}^{\mathcal{L}} = \left[-\,\hat{\mathbf{z}}_{\mathcal{T}}|_{\mathcal{I}}\,,\, \hat{\mathbf{x}}_{\mathcal{T}}|_{\mathcal{I}}\,,\, -\,\hat{\mathbf{y}}_{\mathcal{T}}|_{\mathcal{I}}\right] \tag{7.14}$$

7.2 | Control

The control system is the element of GNC that is responsible for enacting the commands of the guidance system, based on the current state estimated by the navigation system. Control is not deemed an essential part of this thesis, but it is only developed to make the aerobraking simulation more realistic and to get a benchmark of the GNC system behavior. This will allow for the analysis of how the delayed response of the spacecraft will affect the system.

The development of the control system will thus be limited to a simple approach. This consists of the widely used, and very straightforward, PID control, whose acronym stands for proportional, integral and derivative. This control method is based on the error between commanded and current state and its purpose is to minimize said error, by varying the value of $\mathbf{u}$. For attitude control, this control variable is the control moment $\mathbf{M}_c$ that is generated by the reaction wheels. For position control, the control variable is, on the other hand, the thrust emitted by the thruster system. Of course, the thruster system would also be used for desaturation of the reaction wheels, but design of this effect is considered secondary, and thus it is not modeled.

In this section, both controllers are described. First, one will find, in Subsection 7.2.1 a description of the attitude control, and then later, in Subsection 7.2.2, a discussion on the chosen orbit control method.

7.2.1 | Attitude Control

Attitude control, as the name suggests, deals with the control of the spacecraft orientation. It has already been mentioned above that the control of the rotational state is achieved via PID control. The principle behind this control method is embodied by the formulation (Wie, 2008):

$$\mathbf{M}_c = \mathbf{u} = -\left(\mathbf{K}_p\mathbf{e} + \mathbf{K}_i\int \mathbf{e}\,\mathrm{d}t + \mathbf{K}_d\frac{\mathrm{d}\mathbf{e}}{\mathrm{d}t}\right) \tag{7.15a}$$

$$\mathbf{e} = \mathbf{x}_c - \mathbf{x} \tag{7.15b}$$

where $\mathbf{K}_p$, $\mathbf{K}_i$ and $\mathbf{K}_d$ represent the proportional, integral and derivative gains, and vector $\mathbf{e}$ is the error between the estimated (current) state, and the wanted (commanded) state. The general definition of the error in Eq. (7.15b), however, does not apply in case quaternions, MRPs or exponential map are chosen as attitude representation, since their values do not add or subtract in the conventional manner. In fact, the error vector for quaternions is defined by (Schaub and Junkins, 2002):

$$\boldsymbol{\varepsilon}_e = \left[\begin{array}{cc} -\boldsymbol{\varepsilon}_c & \eta_c\mathbf{I} - \boldsymbol{\varepsilon}_c^{\mathrm{S}} \end{array}\right]\left(\begin{array}{c} \eta \\ \boldsymbol{\varepsilon} \end{array}\right) \tag{7.16}$$

where only the imaginary part is taken as reference (only three parameters are needed for the computation of $\mathbf{u}$).

Then, the derivative term can be expanded as follows, by using the product rule of derivatives:

$$\frac{\mathrm{d}\boldsymbol{\varepsilon}_e}{\mathrm{d}t} = \left[\begin{array}{cc} -\dot{\boldsymbol{\varepsilon}}_c & \dot{\eta}_c\mathbf{I} - \dot{\boldsymbol{\varepsilon}}_c^{\mathrm{S}} \end{array}\right]\left(\begin{array}{c} \eta \\ \boldsymbol{\varepsilon} \end{array}\right) + \left[\begin{array}{cc} -\boldsymbol{\varepsilon}_c & \eta_c\mathbf{I} - \boldsymbol{\varepsilon}_c^{\mathrm{S}} \end{array}\right]\left(\begin{array}{c} \dot{\eta} \\ \dot{\boldsymbol{\varepsilon}} \end{array}\right) \tag{7.17}$$

meaning that the derivative error is defined by two terms: one depending on the commanded quaternion, and one on the commanded derivative. In the next paragraphs, both of these terms will be explained.

Commanded Quaternion

The commanded state is determined based on REQ-C-2, which states that the $\hat{\mathbf{x}}_{\mathcal{B}}$ axis, i.e., the x-axis of the body-fixed frame ($\mathcal{F}_{\mathcal{B}}$) always has to coincide with the direction of the periapsis velocity vector. This means that the both angles of attack and of side-slip shall be null at periapsis. It is clear that one degree of freedom is unconstrained, so it was decided to make sure that the $\hat{\mathbf{z}}_{\mathcal{B}}$ axis at periapsis shall lie in the orbital plane and point away from the planet.

This is the same strategy used by virtually all aerobraking missions, which compromise on the aerodynamic angles at the atmospheric interface, to guarantee the correct attitude at the most critical phase. This is to mitigate the risk of attitude failures.

One may notice that the periapsis attitude condition described above, closely matches the transformation from $\mathcal{T}$-frame to $\mathcal{F}_{\mathcal{I}}$ described in Subsection 7.1.3. In fact, the configuration only differs from the fact that the $\hat{\mathbf{z}}_{\mathcal{B}}$ axis points away from the planet, instead of towards it. Thus, one can use a similar procedure of the one expressed in the aforementioned subsection, to derive $\mathbf{C}_{\mathcal{I}}^{\mathcal{B}}$. For sake of brevity, said derivation is not reported here, but the reader will find a full description of the body-frame axes as expressed in the inertial frame in the paragraphs below.

The $\mathcal{F}_{\mathcal{B}}$ axes can be expressed as:

$$\hat{\mathbf{x}}_{\mathcal{B}}|_{\mathcal{I}} = \hat{\mathbf{V}}_{A,p} \tag{7.18a}$$

$$\hat{\mathbf{z}}_{\mathcal{B}}|_{\mathcal{I}} = \hat{\mathbf{r}}_p - \left(\hat{\mathbf{r}}_p \cdot \hat{\mathbf{V}}_{A,p}\right) \hat{\mathbf{V}}_{A,p} \tag{7.18b}$$

$$\hat{\mathbf{y}}_{\mathcal{B}}|_{\mathcal{I}} = \hat{\mathbf{z}}_{\mathcal{B}} \times \hat{\mathbf{x}}_{\mathcal{B}} \tag{7.18c}$$

and thus, the full rotation from body-fixed to inertial frame can be written as:

$$\mathbf{C}^{\mathcal{B}}_{\mathcal{I}} = \left[\hat{\mathbf{x}}_{\mathcal{B}}|_{\mathcal{I}}\,, \hat{\mathbf{y}}_{\mathcal{B}}|_{\mathcal{I}}\,, \hat{\mathbf{z}}_{\mathcal{B}}|_{\mathcal{I}}\right] \tag{7.19}$$

which corresponds to the following aerodynamic attitude at pericenter:

$$\alpha_p = \beta_p = 0\,\text{deg} \qquad \text{and} \qquad \sigma_p = 180\,\text{deg}$$

Note that the bank angle of 180 deg is representative of the difference in the commanded $\mathcal{B}$-frame and the trajectory frame: $\mathcal{F}_{\mathcal{T}}$ is defined with its $\hat{\mathbf{z}}_{\mathcal{T}}$ axis pointing toward the planet.

Update of the current orbit commanded quaternion occurs after each atmospheric passage, such that the new estimated periapsis attitude can be targeted as soon as pericenter is overtaken.

Commanded Quaternion Derivative

To make sure that REQ-C-2 is achieved, one can also command the derivative of the quaternion, such that the required attitude is achieved at all times.

Since the attitude is to be kept constant, however, the derivative of the commanded quaternion is simply zero, and the following condition holds:

$$\begin{pmatrix} \dot{\eta} \\ \dot{\varepsilon} \end{pmatrix} = \mathbf{0} \tag{7.20}$$

where $\mathbf{0}$ is of size 4×1.

7.2.2 | Orbit Control

In Subsection 7.1.3, the maneuver estimator was introduced. The purpose of this component is to compute the required magnitude and direction of the impulsive shots at apo- and periapsis. In reality, the execution of this maneuver is not instantaneous, but is spread over a short time interval. It can be shown, however, that this time interval is so short, that it can be neglected.

The thrusters used for orbit corrections by MRO had a maximum thrust of 22 N[1]. A periapsis raise of 10 km would then require about 5 s, for an orbit with large eccentricity ($e \approx 0.85$), in case four thrusters are fired in unison. Clearly, the time to perform the maneuver is short enough, and the impulsive shot assumption holds.

Modeling of the maneuver is done as shown in Eq. (7.8a), where the computed $\Delta V_{\hat{\mathbf{y}}_{\mathcal{L}}}$ is simply added to both the current estimated and actual Cartesian states.

[1] Jet Propulsion Laboratory: Mars Reconnaissance Orbiter: Spacecraft Parts: Propulsion – https://goo.gl/9wH4mW [last accessed on February 6, 2018]

8
Software Architecture

Now that the components defining the onboard computer (OBC) have been introduced, it is possible to combine them into one whole system. This is done with the aid of a software architecture diagram, which shows the flow of information between the various GNC elements.

Nonetheless, before an architecture can be outlined, one must also set an order with which the functions of these elements are to be performed. The scheduling sequence defined in Section 8.1 has just this purpose, and it introduces the order with which the various guidance, navigation and control functionalities are called.

To conclude, Section 8.2 shows the full architecture of the system. As one will notice, a new element is included here, which represents the brain of the spacecraft, i.e., the OBC. This is the component that interfaces the various other subsystems, not allowing direct communication between them. Furthermore, one will also see that the sensors from Section 6.1 have been moved out of the navigation system, and are given a stand-alone position. This need arises from the way the simulation is built in Tudat.

As mentioned above, Tudat, or TU Delft Astrodynamics Toolbox, is the software library used to simulate the environment and propagate the state of the spacecraft. One can find documentation for this powerful toolbox at `http://tudat.tudelft.nl`, whereas the full software developed for this thesis can be found on GitHub[1].

8.1 | In-orbit Scheduling Sequence

In both Sections 6.2 and 7.1 the reader was introduced to the concept of navigation and aerobraking phases, respectively. These phases partition the orbit and the maneuver into a series of predetermined behavioral patters, that make the aerobraking operations easier to determine. This, however, is only part of the full aerobraking story.

As a matter of fact, an orbit is split into a few more sections, depending on the current position. In this case, these divisions are used to determine the sequence of actions that are to be carried out by the spacecraft, and they are depicted in Fig. 8.1.

Here, one can distinguish five different stages (the coasting phase is repeated twice). However, in this representation, it is portrayed a slightly different scenario than what is implemented in the software developed for this thesis. The main processes, namely the phases typeset in bold, and the maneuvering phase, are shown as spread over time, whereas they are carried out instantaneously in the simulation developed in Tudat. Thus, both the pre-maneuvering and the maneuvering phases are carried out at the estimated apoapsis condition.

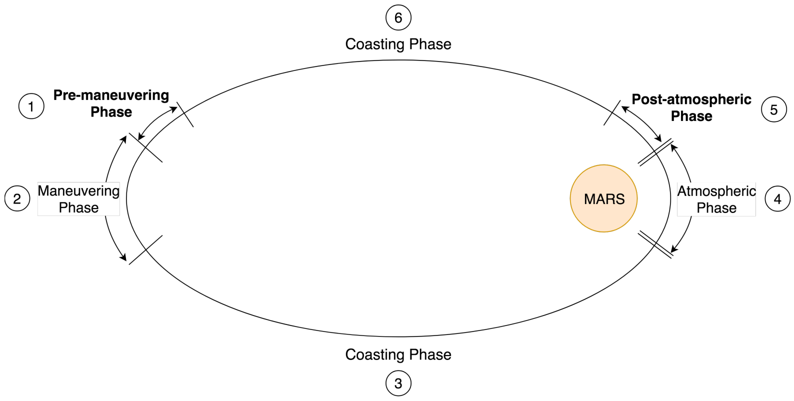


Figure 8.1: *In-orbit scheduling sequence for a spacecraft performing an aerobraking maneuver.*

[1] Repositories: `https://github.com/mfacchinelli/thesisProject` and `https://github.com/mfacchinelli/tudat`. Branches: `masterThesis`, `masterThesisTransOnly` and `masterThesisTransOnlyIMAN`.

A brief description of each of the stages from Fig. 8.1 is given later on. One will notice that for the main processes (as mentioned above, these are the phases typeset in bold in the figure), a list of tasks is shown. In these lists, it may be that one or more of the elements is preceded by a prefix. In total, three prefixes exist, and they are motivated as:

(optional) Tasks marked with this identifier are, as the name suggests, only **carried out under specific conditions**; said conditions are explained on the same line

(IMAN) The discrepancies between the IMAN and GRS navigation methods are highlighted with this identifier; this means that a **task is only part of the IMAN** method; note that all tasks performed as part of GRS are also carried out as part of IMAN

(*) This identifier denotes a special task, that **deserves extra clarification**; where one can find further information is mentioned on the same line

Now that the format of the phase explanation has been introduced, one can find the description of each stage from Fig. 8.1 below, where the order is from the the current-orbit to the next-orbit apoapsis, and where the states or conditions triggering an event are always the ones estimated onboard:

1. **Pre-maneuvering Phase**
   This is the first phase undergone by the spacecraft when starting the aerobraking mission, since it is assumed that the maneuver starts at apoapsis (see the initial conditions from Chapter 12). The functions that are carried out during this phase are the following:
   (a) *(IMAN)* Process deep space network tracking observation (see Section 6.6)
   (b) Determine aerobraking phase (see Subsection 7.1.1)
   (c) Determine periapsis corridor (see Subsection 7.1.2)
   (d) *(optional)* Determine apoapsis maneuver magnitude and direction (see Subsection 7.1.3); only carried out if a maneuver is required
   (e) *(optional)* Update orbit controller (see Subsection 7.2.2); only carried out if a maneuver is required
   (f) Run house keeping routines; used to clear the memory of onboard computer, by eliminating the history of variables only needed for one orbit, such that faster performance can be achieved
   (g) *(*)* Randomize atmosphere perturbations; one can find more information at the end of Section 8.2
2. **Maneuvering Phase**
   This is the part of the orbit where propagation is halted, such that the estimated maneuver can be added to the simulated state of the spacecraft. Of course, the maneuver, and thus the propagation break, is only performed in case it is required by the corridor estimator, i.e., in case the predicted periapsis does not fall in-between the periapsis corridor bounds.
3. **Coasting Phase**
   The coasting phase is nothing but a phase where no actions are taken in terms of control of translational motion, and where no major onboard processing is needed. The actual processing power required depends on the active navigation phase (unaided or aided, IMAN or GRS). This is also the phase when other activities, such as scientific measurements, power collection, communication with Earth, etcetera, can be executed.
4. **Atmospheric Phase**
   The main element of aerobraking is the atmospheric phase. This is when the spacecraft exchanges its orbital energy with friction, i.e., heat, to achieve a reduction in apoapsis altitude.
5. **Post-atmospheric Phase**
   The post-atmospheric phase starts as soon as the spacecraft overcomes the dynamic atmospheric interface. The functions undertaken while in this phase are:
   (a) Update the commanded attitude based on the estimated periapsis state (see Subsection 7.2.1)
   (b) Process current orbit history of accelerometer measurements (see Section 6.4)
   (c) *(optional,IMAN)* Run periapse time estimator (see Section 6.5); only carried out if it is not the first orbit
   (d) Run atmosphere estimator (see Section 6.7)
6. **Coasting Phase**: same as previous coasting phase

Two stages that have not been mentioned above, nor in Fig. 8.1, are the pre-maneuvering and maneuvering phases associated with the **periapsis maneuver**. This is the correction to the spacecraft state that is applied at the very last pericenter, during the termination part of aerobraking. The difference between phases 1 and 2, and the periapsis equivalents is rather limited. One can hence write the tasks for the pre-periapsis maneuvering phase as:

(a) Determine periapsis maneuver magnitude and direction (see Subsection 7.1.3), where target apoapsis is taken directly from Table 7.1

(b) Update orbit controller (see Subsection 7.2.2)

One final part of the onboard software that needs to be discussed is the adaptability of the integration time step (for both onboard processing and propagation). It was initially noticed that running the simulation at the same time rate would lead to a very slow and a very memory-heavy result. Thus, it was decided to run the simulation at a lower rate above 1000 km altitude and at a faster rate below. Table 8.1 shows the step sizes and refresh rates selected for each navigation method developed as part of this thesis.

As one may recall from Section 6.1, and in particular from Table 6.1, the IMU noise depends on the sampling rate of the OBC. Hence, whenever the simulation and onboard step size are change, the standard deviation of the IMU Gaussian noise generators is adapted accordingly.

Table 8.1: *Time steps and refresh rates for propagation and onboard processing, based on the (actual) spacecraft state and on the navigation method being used.*

(a) *Inertial measurements for aeroassisted navigation.*

		Above	**Below**
Time Step	[s]	0.05	0.005
Refresh Rate	[Hz]	20	200

(b) *Generic ranging system-assisted navigation.*

		Above	**Below**
Time Step	[s]	0.1	0.02
Refresh Rate	[Hz]	10	50

8.2 | Architecture

Figure 8.2 shows the top-level architecture of the software developed for this thesis. The structure is clearly divided into five main blocks:

- **Onboard Computer**
  This is the brain of the spacecraft, i.e., the piece of software responsible for the interfaces between Tudat, the instrumentation and the various onboard systems. Its main role is to determine the order of execution of the GNC tasks, based on the orbit scheduling of Section 8.1.
- **Instrument Models**
  Modeling of the instruments is done by implementing an interface between the Tudat environment and the spacecraft model. The main purpose of this interface is to add the required errors and noise, based on the instrument models of Section 6.1.
- **Navigation System**: the role of this section is to carry out the functions of the navigation system; each sub-block roughly corresponds to one of the elements of the navigation system as explained in Chapter 6
- **Guidance System**: the role of this section is to carry out the functions of the guidance system; each sub-block corresponds to one of the elements of the guidance system as explained in Section 7.1
- **Control System**: the role of this section is to carry out the functions of the control system; each sub-block corresponds to one of the elements of the control system as explained in Section 7.2

In this section, the interactions and connections between the various blocks will not be explained, as they should become clear after reading Chapters 6 and 7 and Section 8.1. The focus will be put, instead, on the structure of the architecture.

In Fig. 8.2, one can see that there are two types of arrows, which can be distinguished by their head. The meaning of these two tips is as follows:

- **Main** (open tip): used between the main blocks (defining the systems), and denotes the general flow of information
- **Secondary** (closed tip): used to indicate specific input/output for a sub-block

One may also notice that some of the main input and output arrows do not show all the variables used by a system, and some sub-blocks do not have an output arrow. These questions can be explained with the following reasonings, where the flow of information (in- or output) is always w.r.t. a GNC element:

- if a variable is only on the main input arrow, then it is used internally by every function and it is updated by a specific `set` function (not an explicit input)
- if a variable is only on the main output arrow, it is retrieved by a specific `get` function (not an explicit output)
- all inputs to the GNC systems come from the OBC, hence they are not repeated on the principal input arrows
- all outputs of the GNC systems are fed to the OBC, hence they are not repeated on the principal output arrows

A clear exception to the rules above are the instrument models. In this case, the following rules apply:

- all the secondary input arrows retrieve data from the Tudat environment (and are thus located on the left side of the block, signifying a flow of information from the Tudat suite)
- all the secondary output arrows feed data to the onboard computer (and are thus located on the right side of the block, signifying a flow of information to the spacecraft systems)
- the main in- and output arrows from the onboard computer provide the current time and retrieve the simulated measurements, respectively
- the sub-block named 'Atmosphere Randomizer' will be explained at the end of the section, since it is an exception of its own kind

One will also notice that in the architecture, three new variables have been introduced. These represents concepts that were not readily explained in the previous chapters, and as such, they will be elaborated here (note that however, one will not find them in the list of symbols):

- **Atmosphere initialization indicators**, or $\mathcal{A}$
  These parameters provide information, as the name suggests, on whether the atmosphere estimator of Section 6.7 has been initialized. In case the initialization is complete, then the guidance system can declare the walk-in phase complete, and start the main phase of the aerobraking maneuver.
- **Maneuver indicator**, or $\mathcal{M}$
  The maneuver indicator is nothing but a simple boolean denoting whether the maneuver is to be performed or not. In case the maneuver needs to be performed, then propagation is stopped and the ΔV estimated by the guidance element of Subsection 7.1.3, is fed to the simulation.
- **Phase indicator**, or Φ
  A phase indicator has a rather clear job: it denotes which (navigation or aerobraking) phase is currently active. As is clear from Fig. 8.2, the navigation phase Φ_N (introduced in Section 6.2), is computed every time step, whereas the aerobraking phase Φ_A (introduced in Subsection 7.1.1), is computed only once per orbit, once apoapsis has been reached.

Finally, this last paragraph will be dedicated to a brief explanation of the atmosphere randomizer. This sub-block provides an interface between the computer and the simulated atmosphere. Its job is to call a function that updates the random perturbation coefficients of Eq. (4.30), based on the model explained in Eq. (4.31) and Table 4.1. Thus, the secondary output arrow shown does not supply any variables to the Tudat environment, but rather, it is there to show the flow of *instructions*.

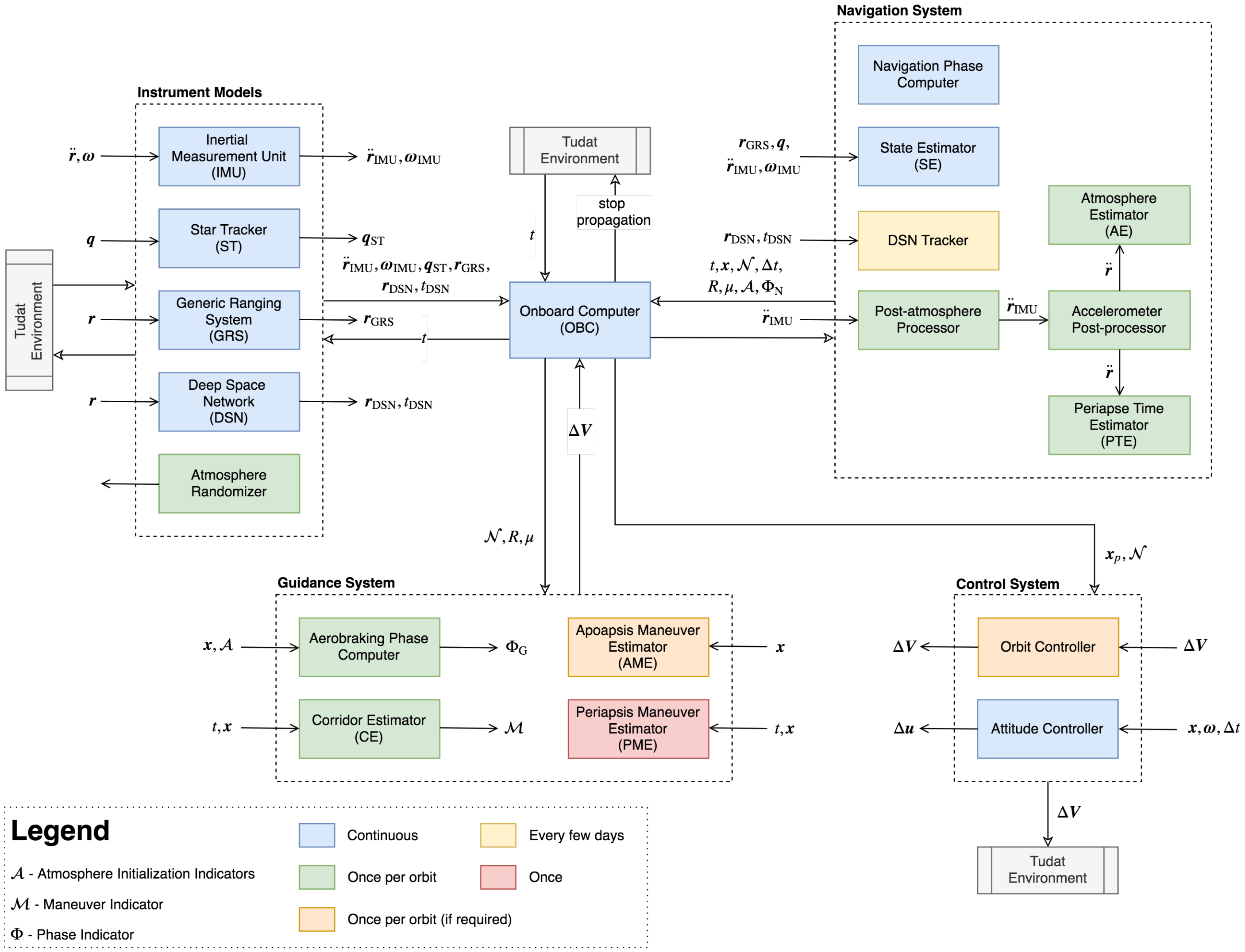

Instrument Models
Inertial Measurement Unit (IMU)
Star Tracker (ST)
Generic Ranging System (GRS)
Deep Space Network (DSN)
Atmosphere Randomizer
Tudat Environment
$\ddot{r}, \omega$
q
r
r
$\ddot{r}_{\mathrm{IMU}}, \omega_{\mathrm{IMU}}$
q_{ST}
$\ddot{r}_{\mathrm{IMU}}, \omega_{\mathrm{IMU}}, q_{\mathrm{ST}}, r_{\mathrm{GRS}}, r_{\mathrm{DSN}}, t_{\mathrm{DSN}}$
r_{GRS}
t
$r_{\mathrm{DSN}}, t_{\mathrm{DSN}}$
Tudat Environment
t
stop propagation
Onboard Computer (OBC)
ΔV
$\mathcal{N}, R, \mu$
Navigation System
Navigation Phase Computer
State Estimator (SE)
DSN Tracker
Post-atmosphere Processor
Atmosphere Estimator (AE)
Accelerometer Post-processor
Periapse Time Estimator (PTE)
$r_{\mathrm{GRS}}, q, \ddot{r}_{\mathrm{IMU}}, \omega_{\mathrm{IMU}}$
$r_{\mathrm{DSN}}, t_{\mathrm{DSN}}$
$t, x, \mathcal{N}, \Delta t, R, \mu, \mathcal{A}, \Phi_{\mathrm{N}}$
$\ddot{r}_{\mathrm{IMU}}$
$\ddot{r}_{\mathrm{IMU}}$
$\ddot{r}$
$\ddot{r}$
$x_p, \mathcal{N}$
Guidance System
Aerobraking Phase Computer
Corridor Estimator (CE)
Apoapsis Maneuver Estimator (AME)
Periapsis Maneuver Estimator (PME)
$x, \mathcal{A}$
t, x
Φ_{G}
$\mathcal{M}$
x
t, x
Control System
Orbit Controller
Attitude Controller
ΔV
Δu
ΔV
$x, \omega, \Delta t$
ΔV
Tudat Environment
Legend
$\mathcal{A}$ - Atmosphere Initialization Indicators
$\mathcal{M}$ - Maneuver Indicator
Φ - Phase Indicator
Continuous
Once per orbit
Once per orbit (if required)
Every few days
Once

Figure 8.2: *Top level architecture for the onboard software of spacecraft performing an aerobraking maneuver.*

Page Intentionally Left Blank

9 Verification

Verification and validation are two fundamental tasks, used to make sure that the algorithms and equations describing the motion and behavior of the spacecraft have been correctly implemented, and to know if the GNC system developed behaves the way it is supposed to.

In this chapter, the verification routines for the onboard software will be presented, whereas one will find the validation of the aerodynamic analysis in the chapter displaying its results, i.e., Chapter 11.

The GNC features of Chapters 6 and 7 are tested by analyzing the performance of their main components under various scenarios:

- By varying the **initial conditions**
- By varying the **accuracy** of the state knowledge
- By analyzing the **results** of Chapter 12

The initial conditions that are tested are taken from three different phases of the orbit: walk-in, main phase and walk-out. In this way, the range of behaviors and scenarios met during an aerobraking maneuver can be analyzed. One can find the initial conditions in Table 9.1, where the first and third states correspond to the ones used in Chapter 13.

Table 9.1: *Initial conditions for the verification and validation of the GNC algorithm. These conditions are such to represent the spacecraft state in the three main phases of the orbit: walk-in (1), main (2) and walk-out (3).*

Initial	a	e	i	ω	Ω	ϑ	t
Conditions	[m]	[−]	[deg]	[deg]	[deg]	[deg]	
1	25 946 932.3	0.865 191 2	93.0	43.6	158.7	180.0	June 30, 2007 at 12:00 Noon
2	5 287 404.0	0.338 564 8					
3	4 710 917.5	0.251 560 4					

Verification is performed by first propagating the orbit of the spacecraft with the same integrator-propagator combination used in Chapter 10 to compute the reference trajectory. Then, depending on the test case and the feature that is being tested, an offset and random noise is added, to reproduce the estimated state. Also, interpolation is applied, to make sure that the estimated elements are generated at a constant step size of 0.1 s.

Table 9.2 shows the types of perturbations used. In the first case, no errors are added, thus the ideal scenario is analyzed. This is done to set a benchmark of the efficacy of the system being analyzed.

The other test case, is used to reproduce the GAN navigation system developed in this report. Simulation of estimated states is rather straight forward: random noise is added to the position and velocity estimates.

In case accelerometer measurements need to be simulated, e.g., to test the PTE and the atmosphere estimator, then the values of noise from Table 6.1 are used to corrupt the actual translational accelerations. Here, as already mentioned, a time step of 0.1 s is used.

9.1 | Navigation

The two main elements of the navigation system are the state estimator, the state propagator, the periapse time estimator (PTE) and the atmosphere estimator (AE).

The first element is validated by comparing the results of the developed filters with a reference book, and can be found in Subsection 9.1.1. The remaining components, on the other hand, were run with the initial conditions of Table 9.1, and with the test cases from Table 9.2 and their performance is documented in the remaining subsections.

Table 9.2: *Test cases for the reproduction of the accuracy of the GNC system developed in this thesis.*

Test Case	**Gaussian Noise STD**		**Aim to Reproduce**
	Position [m]	*Velocity* [m s^{-1}]	
1	—	—	Ideal scenario
2	50	0.1	GAN, i.e., perturbed scenario

9.1.1 | State Estimation

The state estimator is built upon the filtering schemes of Appendix E.7. Both the EKF and UKF filters were implemented in Tudat and therefore needed to be validated. This is done by using one of the examples in Zarchan and Musoff (2013). The chosen application represents a very simple case of a falling object, with the additional challenge of unknown ballistic coefficient.

The filter is set up with the equations of motion and Jacobians (for EKF only) described in the reference. From here come also the initial conditions and the values for the uncertainty matrices. Moreover, the state $\mathbf{x}$ is defined as a vector of size 3×1, whose elements correspond to (in order): position (expressed in ft), velocity (expressed in ft s^{-1}), and ballistic coefficient (expressed in lb ft^{-2}). The measurement $\mathbf{z}$, on the other hand, is only composed of one element, i.e., position itself.

By running the filter for 30 s with a step size of 0.1 s (where integration is carried out with the simple Euler integration, as delineated by Eq. (E.7)), one gets the results shown in Fig. 9.1. The first figure depicts the results obtained with the software developed for this thesis. Here, one can see the convergence of the three state elements over time, for both filtering techniques, and includes the bounds defined by the square root of the diagonal of the covariance matrix, i.e., the standard deviation (STD) of the estimates. In Fig. 9.1b, on the other hand, the respective results of Zarchan and Musoff (2013) are plotted, but only for the EKF filter.

From Fig. 9.1a, it is clear that convergence in both position and velocity occurs quite quickly. This is expected for the position estimate, since this is the value that is directly measured. For the velocity, in fact, one can see some larger oscillations in the estimated value, and a larger and less constant uncertainty, which is embodied by the value of the STD. A slower, but still steady, convergence is witnessed for the ballistic coefficient, which over the 30 s of the simulation shifts from having an error of 60 % (the actual value is 500 lb ft^{-2}), to less than 1 % for both filters.

The same discussion can be done for the results of Zarchan and Musoff (2013). As a matter of fact, the convergence and behavior of the velocity and ballistic coefficient match very closely. A larger difference is noticed for the position, where the Tudat estimate is subject to oscillations with greater amplitude. This can be attributed to the fact that the noise parameter used to generate the position estimates was increased from 25 to 100 m.

9.1.2 | State Propagation

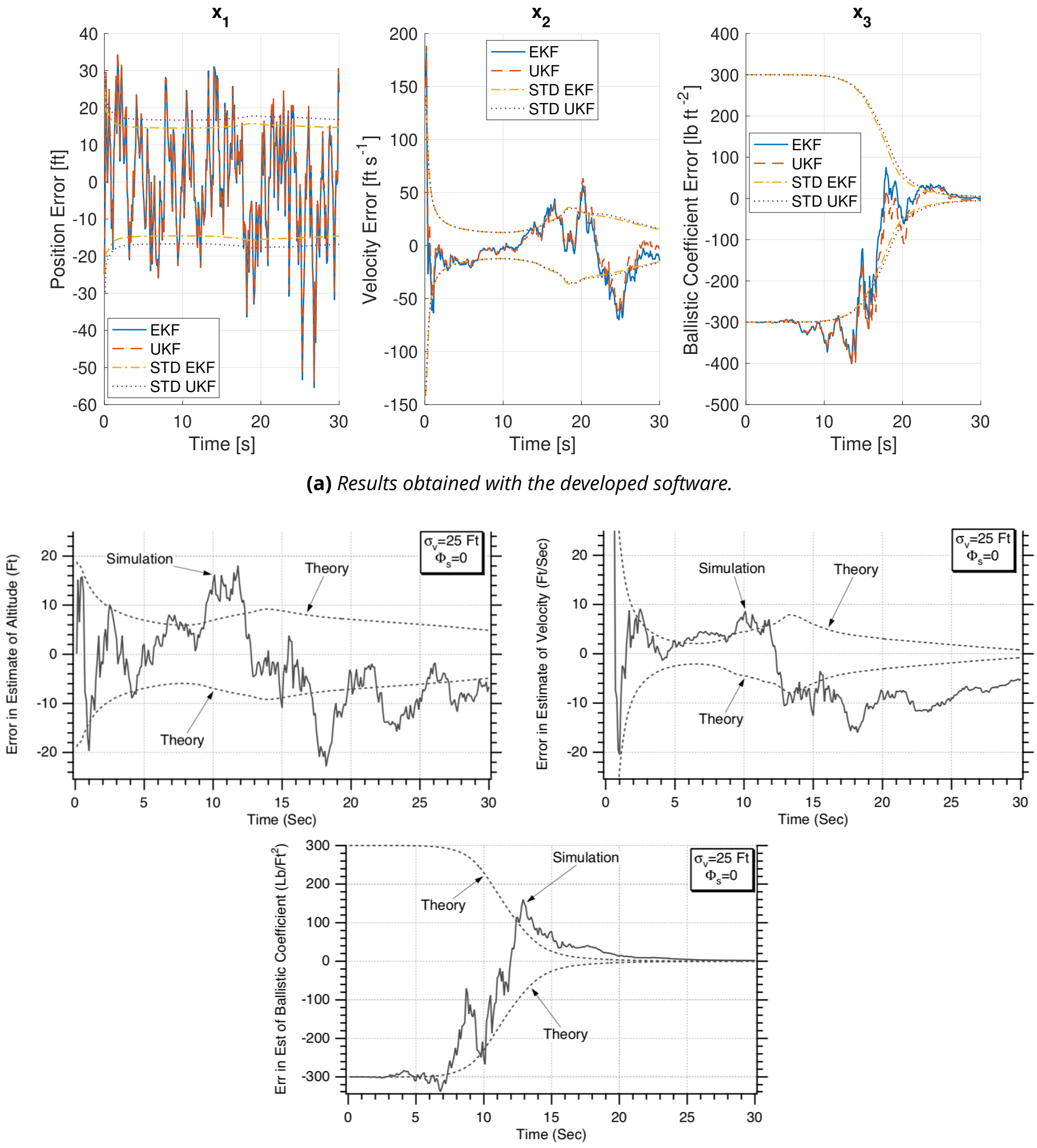
The onboard translational state propagator is part of the Tudat suite, which has been extensively validated. For this section, focus will be placed on the comparison with a more complete propagation environment, to check the differences that arise due to the different modeling choices. To do this, the periapsis altitude and velocity estimated by propagating the initial conditions of Table 9.1 (with both the ideal and perturbed test cases), with both environment models, are compared. Choice of the periapsis conditions as discriminants, are justified by the fact that the pericenter is the most sensitive part of the orbit and these two parameters are used by the rest of the GNC system for various purposes.

The results of this test are shown in Table 9.3, where one can see that the maximum error in altitude is about 3 % (which occurs in the walk-in phase), and that the onboard environment always results in a larger estimated pericenter altitude. For the main and walk-out phases, the error never exceeds 0.6 %, despite the error in initial conditions.

Regarding the velocity, instead, it is clear that the effect of both the error in initial conditions and the difference in environment models, do not cause noticeable offsets. In fact, their difference can be considered negligible, and the source of most uncertainty is observable exclusively in position.

9.1.3 | Periapse Time Estimator

As mentioned in Section 8.1, the periapse time estimator is only used during IMAN navigation. To verify the PTE, one would want the estimated change in Keplerian elements to agree closely enough with the actual values, such that the introduced error in position is less than the one achieved with GRS-assisted navigation.



(a) *Results obtained with the developed software.*

(b) *Results obtained by Zarchan and Musoff (2013).*

Figure 9.1: *Difference between actual and estimated states over time, including standard deviation of the estimates, with both EKF and UKF, for the example of a free-falling object, based on Zarchan and Musoff (2013).*

Table 9.3: *Comparison of periapsis altitude and velocity estimated by the onboard and propagation environments with the initial conditions from Table 9.1 and test cases from Table 9.2.*

(a) *Periapsis altitude.*

Initial Condition	Actual Altitude [km]	Ideal [%]	Perturbed [%]
1	110.5	0.613	3.238
2	113.8	0.021	0.554
3	114.8	0.059	0.525

(b) *Periapsis velocity.*

Initial Condition	Actual Altitude [km s⁻¹]	Ideal [%]	Perturbed [%]
1	4.778	−0.014	−0.048
2	4.045	0.001	−0.009
3	3.917	−0.005	−0.014

The results for running the PTE with the test initial conditions, are shown in Table 9.4. Note that only the results produced with the second test case, i.e., where the uncertainty in state and accelerations are included, are shown.

Table 9.4: *Verification of the results of the PTE element of the navigation system.*

Initial Condition		ΔV [m s^{-1}]	Δa [m]	Δe [−]	$\Delta \vartheta$ [deg]	t_p [s]
1	Computed	−4.270	−641 479.0	−0.003 33	−0.004	63 446.5
	Actual	NA	−635 706.0	−0.003 33	NA	63 436.5
2	Computed	−2.968	−15 685.9	−0.001 96	0.092	5839.69
	Actual	NA	−27 366.9	−0.002 07	NA	5846.82
3	Computed	−2.763	−11 159.1	−0.001 77	−0.108	4890.49
	Actual	NA	−11 096.9	−0.001 88	NA	4890.6

The only requirement on the PTE was REQ-N-2, which stated that the periapsis time shall be estimated with an error of at most 225 s. From Table 9.4, it is clear that this is met in every tested case, with an error of at most 10 s.

For the remaining elements, no accuracy requirements were set, but the change in eccentricity is estimated within a very narrow margin. The same cannot be said, however, for the estimated change in semi-major axis. In fact, the Δa for the second test case, i.e., the one representing the main phase, is almost twice as large. Since the PTE is not used for the main navigation method utilized in for this work, i.e., the GAN method, no further investigation on the discrepancies shown in the table was carried out. Nevertheless, these are recommended as future research the PTE were to be implemented as outlined in Section 6.5. As will be shown in Chapter 13, this may be of particular interest in case the IMAN method were to be chosen.

If one were to only use the PTE to estimate the periapsis time, more testing is also recommended. The main challenge for this part of the software is the processing of the measured acceleration data. Experimentations with more perturbed and representative atmosphere models should be performed before even this part of the algorithm can be considered validated. Furthermore, the current analysis started out by considering a rather small error in initial conditions. As one will see in Chapter 13, the navigation error under IMAN can grow to significantly larger values. Thus, these circumstances should also be analyzed in future testing.

9.1.4 | Atmosphere Estimator

Running the atmosphere estimator for the initial conditions mentioned and with the exponential atmosphere model from Eq. (4.29), yields atmospheric density estimates that are always within −5 to +30 % of the actual value, with the smallest error near pericenter. This is shown in Fig. 9.2 for the worst case examined. In this figure, one can also see the estimate generated by the MATLAB `fit` function, and also for the five-parameter atmosphere model from Eq. (4.33).

Given the very close agreement with the MATLAB fitting functions (the MATLAB results are exactly superimposed to the Tudat ones), the atmosphere model is considered verified, and the discrepancies in the estimate (i.e., the aforementioned +30 %) can be attributed to the noise in acceleration values.

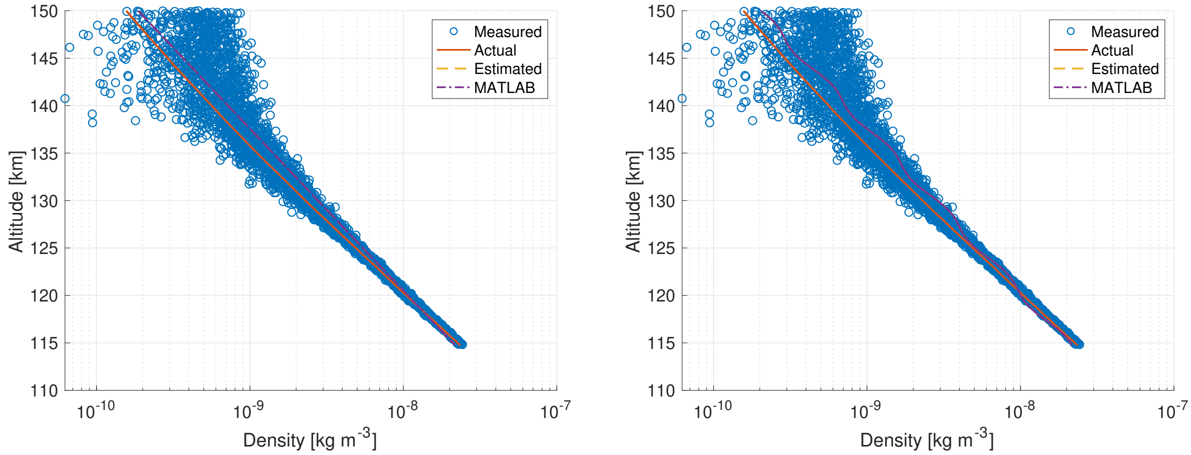
Interestingly, if one compares the actual values of the atmospheric parameters estimated by Tudat and MATLAB, these differ quite significantly. However, as Fig. 9.2 is proof, there exist multiple solutions to the least square problem at hand, and the algorithms simply converge to different minimizing conditions.

9.2 | **Guidance**

The underlying elements of both main components of the guidance system, i.e., the corridor and maneuver estimators, are a bisection root-finder and the onboard propagator from Subsection 6.3.4. Both of these elements have been verified within Tudat, and are thus accepted as-is. Thus, what remains to be discussed is the performance of the combined effect of the guidance system elements.

9.2.1 | Corridor Estimator

The performance of the corridor estimator depends on the accuracy of the propagation and on the efficacy of Eq. (7.6). The onboard propagator was already analyzed in Subsection 9.1.2, and thus, the combination of the altitude correction factor κ_p and the general functioning of the estimator is analyzed by checking that under ideal and perturbed conditions, the threshold aerothermodynamic values defining the boundaries are respected.



(a) *Exponential model from Eq. (4.29).*

(b) *Five-parameter exponential model from Eq. (4.33).*

Figure 9.2: *Comparison of actual and estimated (both by developed software and by MATLAB) density, based on the data measured by the onboard accelerometer.*

Table 9.5: *Pericenter aerothermodynamics parameters at the corridor boundaries estimated by the corridor estimator, with the initial conditions from Table 9.1 and test cases from Table 9.2.*

(a) *Ideal conditions.*

Initial Condition	**Dynamic Pressure** [N m^{-2}]	**Heat Rate** [W m^{-2}]	**Heat Load** [kJ m^{-2}]
1	0.195	1432.2	169.3
2	0.190	1317.5	249.4
3	0.191	1154.6	250.4

(b) *Perturbed conditions.*

Initial Condition	**Dynamic Pressure** [N m^{-2}]	**Heat Rate** [W m^{-2}]	**Heat Load** [kJ m^{-2}]
1	0.191	1406.7	169.2
2	0.190	1317.9	249.4
3	0.191	1154.6	250.4

Indeed, from Table 9.5, it is clear that the estimated aerothermodynamic parameters are targeted properly by the estimator (where the minimum and maximum allowed values are given by Table 7.1). One can even notice the shift in thermodynamic constraint from heat rate to heat load in the second and third initial conditions.

9.2.2 | Maneuver Estimator

Verification of the maneuver estimator is done in two steps:

- compute by hand the result of Eq. (7.8) based on the initial conditions and the required periapsis altitude difference (where a slight discrepancy is expected, due to the use of the maneuver improvement root-finder)
- compare the actual post-maneuver periapsis altitude to the target altitude

The first test results are displayed in Table 9.6, whereas the second ones can be found in Table 9.7. From the former table, one can see how the agreement between the estimated (by the OBC) and the verification (computed by hand) values is very close, both for the ideal and perturbed conditions. This is proof that the evaluation of Eq. (7.8) is carried out correctly, and that the root-finder supplement does have an effect on the final value.

Shifting the focus to the values shown Table 9.7, and especially the ones from Table 9.7a, the reader can notice how the maneuver estimator manages to aim the target periapsis altitude. Thus, it can be concluded that the maneuver estimator performs as expected, and that transformation of the maneuver magnitudes from Table 9.6, to a vector in $\mathcal{F}_\mathcal{I}$ is done correctly. The small differences seen in Table 9.7a are due to the differences in propagation environments (simulation vs. onboard model) and due to the finite accuracy of the root-finder. To these, one has to add the error in initial state, when looking at the results of Table 9.7b.

Finally, without written proof, it is stated that the introduction of the root-finder to improve the maneuver magnitude estimate, leads to a reduction of the error in actual periapsis altitude, ranging from 2 to 35 % in the tested scenarios.

Table 9.6: *Apoapsis maneuver magnitude (expressed as ΔV) to target the midpoint of the pericenter corridor as estimated by the maneuver estimator and as computed by hand, with the initial conditions from Table 9.1 and test cases from Table 9.2.*

(a) *Ideal conditions.*

Initial Condition	**Velocity Increment**	
	Estimated [m s^{-1}]	*Verification* [m s^{-1}]
1	1.1203	1.1233
2	1.1699	1.1662
3	−4.2070	−4.1876

(b) *Perturbed conditions.*

Initial Condition	**Velocity Increment**	
	Estimated [m s^{-1}]	*Verification* [m s^{-1}]
1	0.9916	0.9928
2	1.0557	1.0554
3	−4.3172	−4.3036

Table 9.7: *Periapsis altitude as targeted by the maneuver estimator and actually achieved after the apoapsis maneuver, with the initial conditions from Table 9.1 and test cases from Table 9.2.*

(a) *Ideal conditions.*

Initial Condition	**Periapsis Altitude**	
	Target [km]	*Actual* [km]
1	135.522	134.661
2	113.818	113.867
3	114.873	114.834

(b) *Perturbed conditions.*

Initial Condition	**Periapsis Altitude**	
	Target [km]	*Actual* [km]
1	135.599	132.043
2	113.818	113.235
3	114.873	114.267

III

Results

“*Yesterday is history, tomorrow is a mystery, but today is a gift. That is why it is called the present.*”

Master Oogway, Kung Fu Panda, 2008

10
Unified State Model Performance

As mentioned in Subsection 3.3.4, a comparison of the performance in propagation of the unified state model (USM) is carried out as part of this thesis. The comparison is done w.r.t. Cartesian coordinates, implemented with Cowell's method for propagation. Firstly, however, some considerations on numerical propagation of the USM7 and USMEM coordinates, will be made. They relate to some of the issues that arose while implementing these coordinates in Tudat.

The rest of the chapter is dedicated to the performance analysis of all three USM models in propagation. The reader will find a full explanation of the set up for this comparison. Section 10.2 will describe the settings for the simulation, and Section 10.3 provides the selected test cases. Finally, in Section 10.4, one can found a discussion on the results obtained.

10.1 | Considerations on Numerical Integration

In this section, some general considerations on the numerical integration of the USM and rotational dynamics propagators will be given. These discussions reflect problems that were encountered while implementing the new propagators in Tudat, and are structured as follows. First, a discussion about normalization of quaternions is done. This will be followed by a brief analysis on why the state expressed in exponential map and USMEM has been modified to include a shadow flag, just like the one for MRP (and SMRP).

10.1.1 | Unified State Model with Quaternions

As was seen in Subsection 3.3.1, one of the characteristics of quaternions is that the square of their sum equals unity. In some situations, this property can be viewed as an advantage, leading to simplifications in the equations of motion. During numerical propagation, however, this becomes a disadvantage. Round-off and truncation errors, add small variations in the value of the quaternions, which leads to a norm that differs from one. In general, this problem can be tackled in two different ways, which will be explained below. Note that, for sake of simplicity, the quaternion state, in this section is expressed as $\mathbf{q}$, whose definition is as follows:

$$\mathbf{q} \triangleq \begin{pmatrix} \eta \\ \boldsymbol{\varepsilon} \end{pmatrix}$$

Furthermore, a normalized state is represented with the symbol $\overline{\Box}$.

- **Corbett-Wright Orthogonality Control**
  According to Phillips (2004), the error in the norm can be kept within acceptable limits by modifying the kinematic equations in term of quaternions, i.e., Eq. (4.5) (or Eq. (4.14)), such that the diagonal elements act as controllers for the error in the norm. In fact, taking Eq. (4.5), the kinematics would be expressed as:

$$\dot{\mathbf{q}} = \frac{1}{2}\left(Ke\mathbf{I} + \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} \right)\mathbf{q} \tag{10.1a}$$

  where K is a gain factor, and e the error in the norm, expressed as:

$$e = 1 - \mathbf{q}^{\mathrm{T}}\mathbf{q} \tag{10.1b}$$

  The author also suggests using the magnitude of the rotational velocity vector, i.e., $K = \omega$, as gain. The use of a gain that depends on the current orbit conditions, would increase the accuracy of the normalization, but also increase the computation time.

- **Normalization after Propagation**
  The other strategy, recommended by Fukushima (2008), involves normalizing the state after every propagation step. If an integrator with multiple function evaluations is begin used, normalization should be performed after every evaluation. For instance, for an RK4 integrator, each time step corresponds to four normalizations. Furthermore, Fukushima (2008) also discusses the correction of the derivative of the state, to guarantee its orthogonality condition, by performing the correction:

$$\dot{\overline{\mathbf{q}}} = \dot{\mathbf{q}} - \left(\overline{\mathbf{q}}^{\mathrm{T}}\dot{\mathbf{q}}\right)\overline{\mathbf{q}} \tag{10.2a}$$

  where the normalized state is given by:

$$\overline{\mathbf{q}} = \frac{\mathbf{q}}{\sqrt{\mathbf{q}^{\mathrm{T}}\mathbf{q}}} \tag{10.2b}$$

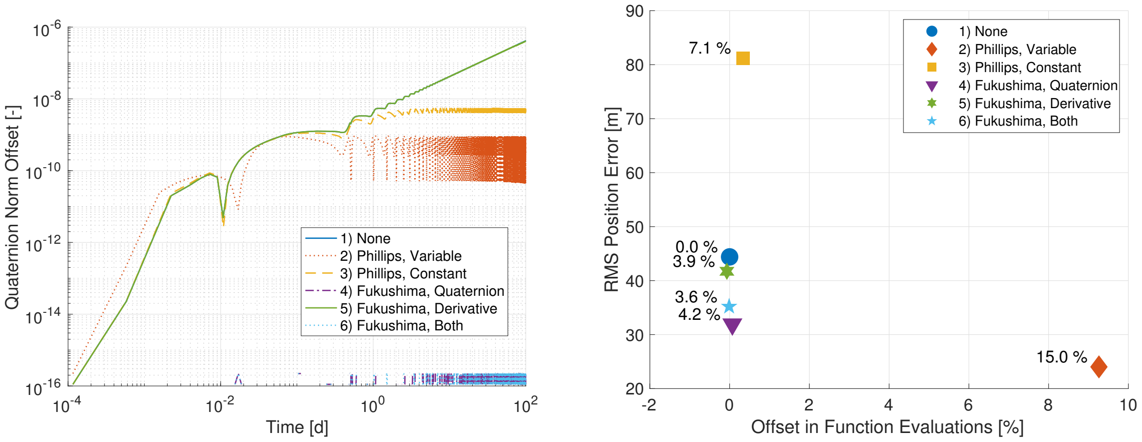
Both methods were implemented in propagation with Tudat to compare their performance. In particular, the following settings were tested:

1. **No normalization**, i.e., the control case of this experiment
2. Phillips method with the magnitude of the current **rotational velocity vector as gain**
3. Phillips method with **constant gain**, equal to 10^{-5} (which after some tuning seemed to give the best compromise between accuracy and number of function evaluations)
4. Fukushima method with **normalization of quaternion** after the integration step
5. Fukushima method with **normalization of quaternion derivative** during the integration step
6. Fukushima method with **normalization of quaternion derivative** during and **normalization of quaternion** after the integration step

The results for each of these methods are as shown in Fig. 10.1. Note that the simulation conditions are the same as for test case 2, from Section 10.3, and row two of Table 10.1. The only difference is that the simulation duration in this case is 100 days.

In Fig. 10.1a, one can see the offset in quaternion norm (w.r.t. unity), for each of the cases in the list above and in Fig. 10.1b one can see the RMS error as a function of function evaluations and computation time. Clearly, the method based on Phillips (2004) with constant offset, i.e., the third method, gives the largest error in position, whereas the second one gives the largest difference in number function evaluations and computation time. As a matter of fact, this method does a very good job at reducing the error, but indeed at the cost of performance.

It is interesting to notice how the RMS position error does not necessarily correlate to the offset in quaternion norm. In fact, the third method does a discrete job at maintaining said offset at a constant value of less than 10^{-8}, despite displaying the largest error.



(a) *Offset of quaternion norm from unity. Note that the first (solid blue) and fifth (solid green) lines are overlapping.*

(b) *RMS position error as a function of offset in number of function evaluations, w.r.t. the control case, i.e., 128 096. The numbers at the upper left of each marker represent the offset in computation time w.r.t. the control case, i.e., 2.705 s.*

Figure 10.1: *Comparison of results of various normalization methods for USM7 during propagation, for a Molniya orbit.*

The lack of correlation can also be seen by looking at the results of the fifth method, hence the method based on Fukushima (2008), where only the derivative of the quaternions is normalized. In this case, the norm of the quaternion steadily increases, following the behavior of the control case, reaching a value of almost 10^{-6} at the end of the propagation. Nevertheless, the error corresponding to this case is slightly less than the control value.

The best performance, however, is seen in the forth and sixth cases, i.e., when the state is normalized after propagation. From Fig. 10.1a, one can see that the offset from the norm barely gets larger than 10^{-16}. This is because of the clause that the error shall be kept below 5×10^{-15}. In fact, normalization is not actually carried out every step, but only when this threshold is overcome. Regardless, these methods manage to reduce the error, while keeping the number of function evaluations and computation time roughly constant.

To conclude, the chosen method is method number four, thus the method that performs the normalization step after integration (if the above-mentioned tolerance is exceeded). The choice is based on the fact that the norm of the quaternion is kept extremely close to unity, and most importantly, that the RMS error in position is reduced and that the performance is not significantly affected.

For sake of completeness, it is stated that performing the above comparison while considering the RMS velocity error, leads to the same conclusions.

10.1.2 | Unified State Model with Exponential Map

In Subsection 3.3.1, the exponential map was introduced, and in Subsection 4.1.2 its derivative. Here, it was explained that both the equations of motion and the conversion equations are the same, regardless of the use of shadow parameters. Thus, there is no need to track whether EM or SEM is currently being propagated.

The above discussion, however, is not always true. As also stated by Vittaldev (2010), in case one needs to look at the full history of the exponential map, for instance for interpolation, then a flag denoting the use of the shadow elements needs to be introduced. This is also true if the elements being interpolated are not the EM themselves, but a set of quaternions derived from the EM history. In fact, conversion from exponential map to quaternions will always lead to a positive η, which as the reader may recall from Subsection 3.3.1, is the condition for the conventional, i.e., not shadow, exponential map (since when $\xi < \pi$, $\eta > 0$). Hence, a shadow flag parameter is added, such that whenever the flag is active, the resulting quaternion is inverted in sign.

One may think that the advantage of exponential mapping over MRPs is then gone, since now both attitude elements are made up of four elements. However, for EM, the flag element is only used in the one occasion explained above. All the relations derived thus far, are in fact valid for both EM and SEM, irrespectively.

10.2 | **Simulation Settings**

The unified state model was analyzed in terms of accuracy and speed by Vittaldev (2010), Vittaldev et al. (2012), where the authors compared the USM7, USM6 and in Vittaldev et al. (2012) also the USMEM state variables, to the very commonly used Cowell propagator. A brief summary of their conclusions was provided in Section 3.3, but for sake of completeness, they will be repeated here.

The main takeaway here was that the USM coordinates perform very well in terms of RMS error and computation time, when compared to a propagation with Cartesian coordinates and similar tolerances. In case of perturbed circular orbits, perturbed highly elliptical orbits, low-thrust orbits and for an aerobraking maneuver (without GNC system) around Mars, the advantage of using USM was always clear and distinct.

The reason why the performance analysis is performed as part of this thesis work is dual:

- To **verify** the correct implementation of the USM models in Tudat.
- To **extend** the test case scenarios to include more challenging environments.

In fact, the reader will find to types of simulations. The first type are based on the same initial conditions as Vittaldev et al. (2012), whereas the second ones, are based on arbitrary conditions.

To make the analysis more comparable to the results by the referred authors, other than initial conditions and environment, also the integration settings are kept as similar as possible. In fact, due to the different simulation software used (Tudat vs. MATLAB), the exact same integration settings could not be used. However, the same principles were kept in mind when choosing the integrators: high-order for the reference case and a lower order for the actual test cases.

INTEGRATORS Three integration settings are used. The first one is the one that is used to generate the benchmark, i.e., the reference trajectory. This integrator uses a variable step size, low tolerances and a high-order integration scheme, to make sure that the integration error is kept to a minimum. The remaining two integrators are used to compare the performance of the propagators in case of constant and variable step sizes, where the accuracy is less than the benchmark, to reduce the computational load.

In particular, the three integrators are:

- **Runge-Kutta-Fehlberg 7(8) variable step size** with both relative and absolute tolerances at 10^{-15} and with free step size; this propagator is used as *benchmark*
- **Runge-Kutta-Fehlberg 5(6) variable step size** with varying tolerances
- **Runge-Kutta 4 constant step size** with varying step size

When the term 'free step size' is used, it is implied that the minimum and maximum allowed step sizes are so small or large, that they are never reached (10^{-5} and 10^{5} s, respectively).

It should be pointed out that the integrators used in Vittaldev et al. (2012) are slightly different. In particular, for variable step size the authors use an RK4(5) integrator, instead of RKF7(8) and RKF5(6), whereas for the constant step size, they also use an RK4.

Other than the integration routine, two other important parameters are the relative and absolute tolerances. Whereas the relative tolerances are depend on the test case, the absolute tolerances are set as follows:

- **Position and velocity**: absolute tolerance set to 10^{-6}
- **Quaternion**: absolute tolerances set to 10^{-15}; such a low value is used due to the normalization of the quaternion state (a larger value would result in excessive errors)

The equivalent of tolerances for constant time step integration, is the step size. Similarly to the relative tolerances, they vary per application.

10.3 | Test Cases

The main reason why the performance of the USM model is carried out yet again, is because of its new implementation in Tudat, the astrodynamics toolbox of the aerospace engineering faculty at TU Delft. Therefore, the results obtained with this new implementation need to be compared to the ones obtained with the implementation by Vittaldev et al. (2012). This justifies the selection of the first three test cases, i.e.:

1. **Circular orbit at LEO** with the initial conditions in the first row
2. **Molniya Orbit** with the initial conditions in the second row
3. **Low-thrust trajectory** with the initial conditions in the third row, and continuous thrust with the following characteristics:
   - Constant magnitude varying from 5×10^{-3} to 50 N with tenfold increments
   - Direction opposite to the current velocity vector

where the row number refers to a row in Table 10.1. Then, some other tests will be performed, to analyze the performance of the propagators in situations similar to the aerobraking conditions that will be met later in this thesis, and for other more general cases:

4. **Aerocapture around Mars** with the initial conditions in the forth row
5. **Aerobraking around Mars** with the initial conditions in the fifth row
6. **Interplanetary orbit** with the initial conditions in the sixth row, corresponding to the Hohmann transfer from Earth to Mars

The full details on the simulation environment can be found in Appendix G.

10.4 | Results

In this section, the results obtained for each test case will be displayed. The way these results are analyzed, depends on the integration method: for variable step size integration the RMS error in position is plotted against the number of function evaluations and the computation time, whereas for constant step size the RMS error is plotted against the constant time step used and the computation time.

The results are split in two subsections. First, Subsection 10.4.1 will look at the results of the first three test cases, i.e., the ones taken directly from Vittaldev et al. (2012). Then, in Subsection 10.4.2, the reader will find a discussion on the last three test cases, thus the ones introduced in this thesis. Finally, Subsection 10.4.3 will conclude the comparison of the propagators.

Table 10.1: *Initial conditions for USM propagators comparison test cases.*

Test Case	a	e	i	ω	Ω	ϑ	Δt	t
	[km]	[−]	[deg]	[deg]	[deg]	[deg]	[d]	
1	6936.0	0.0	28.5	272.3	194.8	0.0	10.0	June 30, 2007 at 12:00 Noon
2	26 559.0	0.7	206.3	281.6	63.2	0.0	25.0	
3	7378.1	0.0	0.0	0.0	0.0	0.0	10.0	
4	−17 305.0	1.2	93.0	23.4	158.7	230.0	0.625	
5	27 228.5	0.869 218	93.0	23.4	158.7	180.0	145.0	
6	188 768 611.5	0.207 506	2.5	147.8	328.6	0.0	260.0	

10.4.1 | Test Cases 1, 2 and 3

Comparing the results shown in Figs. 10.2 to 10.4 to the ones given by Vittaldev et al. (2012), one can see a lot of resemblance. Of course, due to the different integrators and programs used, the results do not coincide exactly, but the general behavior is very comparable.

At equal value of the x-axis parameters, the USM propagators are about one order of magnitude more accurate than Cowell when looking at variable step size integrators, and from two to three orders of magnitude more accurate when constant step size is considered. The USMEM coordinates appear especially interesting for the Molniya orbit case with constant time step. It should be noted however, that as the time step decreases, all propagation methods start to converge to the same accuracy.

Looking at these figures, the following can be said about the two integration methods:

- **Variable Step Size**
  Since the same level of accuracy in USM and Cowell corresponds to a lower number of function evaluations for the former, it would appear that the USM coordinates are inherently less prone to errors than Cowell, thus allowing for a larger time step in integration, and in turn, reducing the total computation time.

- **Constant Step Size**
  Given a constant step size, the USM coordinates provide much more accurate results in terms of RMS error, but with a slightly larger computation time. This is due to the more complicated equations of motion needed for USM (compare e.g., Eqs. (4.3), (4.8) and (4.9) of USM6 to Eq. (4.1) for Cowell) and due to the transformation to/from Cartesian coordinates (needed to determine the accelerations). However, the larger computation time, in virtually all of the step sizes analyzed for test cases 1, 2 and 3, is almost negligible.

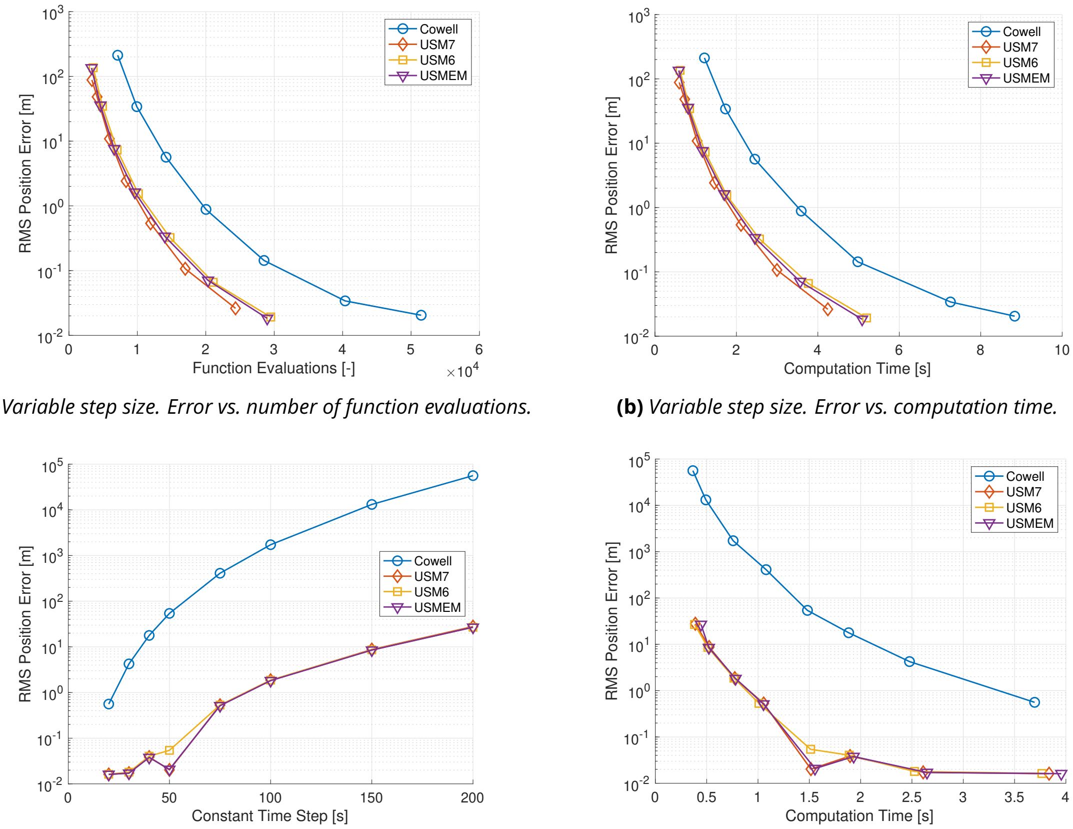
10.4.2 | Test Cases 4, 5 and 6

Some rather different conclusions are reached when looking at Figs. 10.5 to 10.7. Once again, the performance analysis will be split based on the two integration methods, which one can find in the paragraphs below.

Variable Step Size Results for the variable step size integration are inconclusive for the aerobraking and the interplanetary trajectories. In the first case, i.e., when looking at Figs. 10.6a and 10.6b, one can see a similar general trend as for the previous test cases: the RMS position error decreases with increasing number of function evaluations (that is, with decreasing tolerances). However, the benefit of using the USM coordinates is no longer visible. As a matter of fact, the error is smaller for Cowell integration, when the lowest values of tolerances are used (and the computation times are comparable).

For the interplanetary case (see Figs. 10.7a and 10.7b), the aforementioned general trend is also no longer clear. It would appear that the combination of long propagation time and detailed dynamics, overshadow the advantage of the USM, and that the position error grows to the same level, in spite of the value of tolerance.

A different verdict is drawn for the aerocapture test case. Both the overall trend and the performance gain of the USM are visible when examining Figs. 10.5a and 10.5b. However, one will notice that the USM models reach a performance plateau rather early, in terms of function evaluations. This is a similar result as was seen in Fig. 10.4 for the highest value of thrust. It is clear then, that the complex environment does indeed influence the accuracy of the USM propagation coordinates, but only when this is very dynamic, and especially when the propagation duration is large.



(a) *Variable step size. Error vs. number of function evaluations.*

(b) *Variable step size. Error vs. computation time.*

(c) *Constant step size. Error vs. constant time step value.*

(d) *Constant step size. Error vs. computation time.*

Figure 10.2: *Comparison of RMS position error for USM and Cowell propagators, for a circular LEO orbit.*

CONSTANT STEP SIZE In the case of constant step size, one can once again see very different behaviors for the final three test cases:

- **Aerocapture**, or test case 4
  The results shown in Figs. 10.5c and 10.5d depict a situation that is very different from the expectation. It would appear that not only the RMS error is not highly affected by the integration time step (nor propagation coordinates), but most incredibly, that a higher accuracy is achieved with a larger time step. An investigation of the reasons behind this behavior has not been carried out, but it is highly recommended to fully understand the behavior under these conditions.

- **Aerobraking**, or test case 5
  When looking at Fig. 10.6c, the behavior is as expected: the USM models have a higher precision than Cowell propagation for all time steps. Nevertheless, as the time step gets smaller, the RMS curves tend to converge to the same accuracy. The same can be said about Fig. 10.6d. Due to the more complex equations of motion and the extra conversion steps, the USM models are considerably slower than Cowell, especially at small step sizes.

- **Interplanetary**, or test case 6
  Results for the interplanetary trajectory show similar trends as seen for test cases 1, 2 and 3. The position error is smaller for USM, but these coordinates also lead to a slightly larger computation duration. The advantage of USM, then, disappears for larger time steps. One should keep in mind, nonetheless, that the error value does not decrease considerably when USM is chosen. Hence, selection of USM over Cowell is less justifiable in this case.

10.4.3 | Conclusion

When considering variable step size propagation, it would appear, based on the tests carried out, that USM provides a great advantage both in terms of computation time and accuracy, but only if:

1. the environment of the system is not highly dynamic and evolving with time

2. the propagation time is not too large

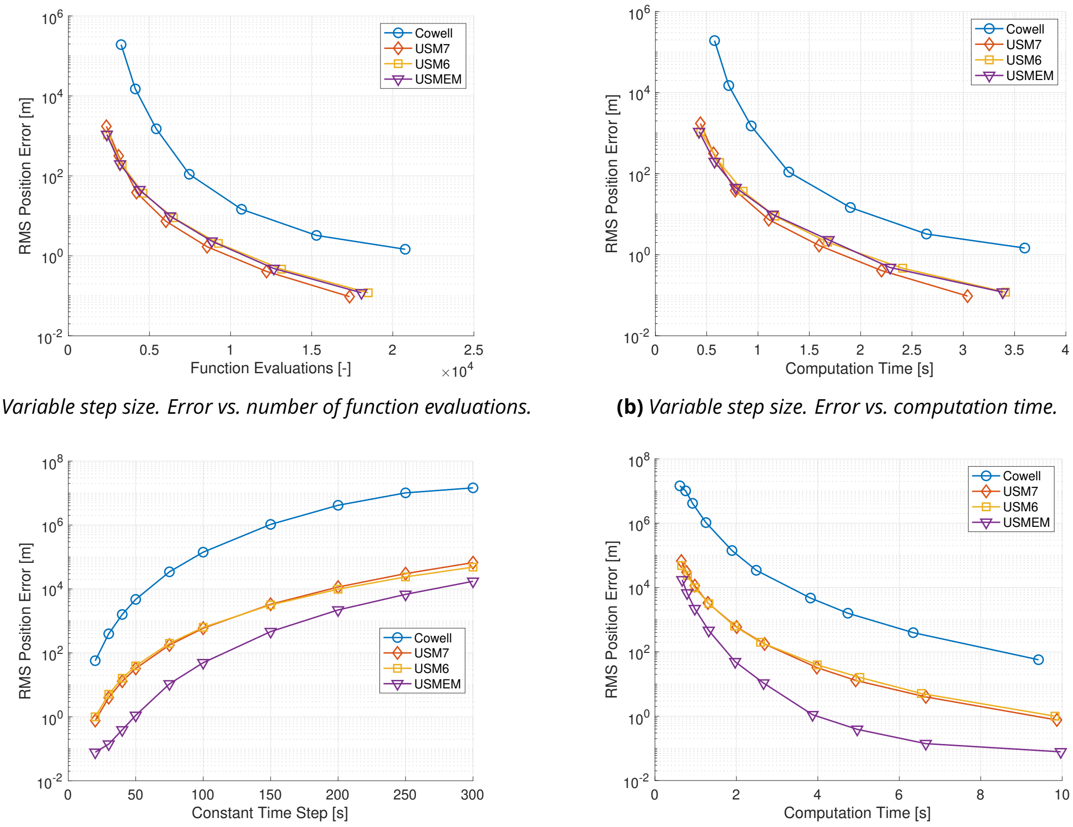
The above conditions define even stricter constraints, if they are used simultaneously. Defining a threshold for each of the elements above, would require the execution of a large number of tests with different dynamics and simulation durations. Said analysis was not deemed necessary for the case at hand, since for an aerobraking mission requires a constant step size to simulate onboard processing.

Regarding the constant step size simulations, on the other hand, Cowell integration is less accurate in virtually every tested scenario. This statement is only challenged when the dynamics of the system are highly dynamic and the time step is small (as is the case for the largest thrust value in Fig. 10.4). The combination of USM and constant time-step, comes nonetheless with one downside: the propagation is slightly more time-consuming.

To conclude, considering the very small onboard time step (in the neighborhood of 10^{-3} to 10^{-1} s), the conditions chosen for integration and propagation settings of the aerobraking maneuver are the following:

- **Propagator**: Cowell (Cartesian coordinates)
- **Integrator**: Runge-Kutta 4 with constant step size depending on the current (actual) state and navigation method (see Table 8.1)

This choice is justified by the fact that when $t \ll 1$ s, the gain in accuracy of the USM model becomes negligibly small (if not negative), and the computation time deficit in computation time, very large.



(a) *Variable step size. Error vs. number of function evaluations.*

(b) *Variable step size. Error vs. computation time.*

(c) *Constant step size. Error vs. constant time step value.*

(d) *Constant step size. Error vs. computation time.*

Figure 10.3: *Comparison of RMS position error for USM and Cowell propagators, for a Molniya orbit.*

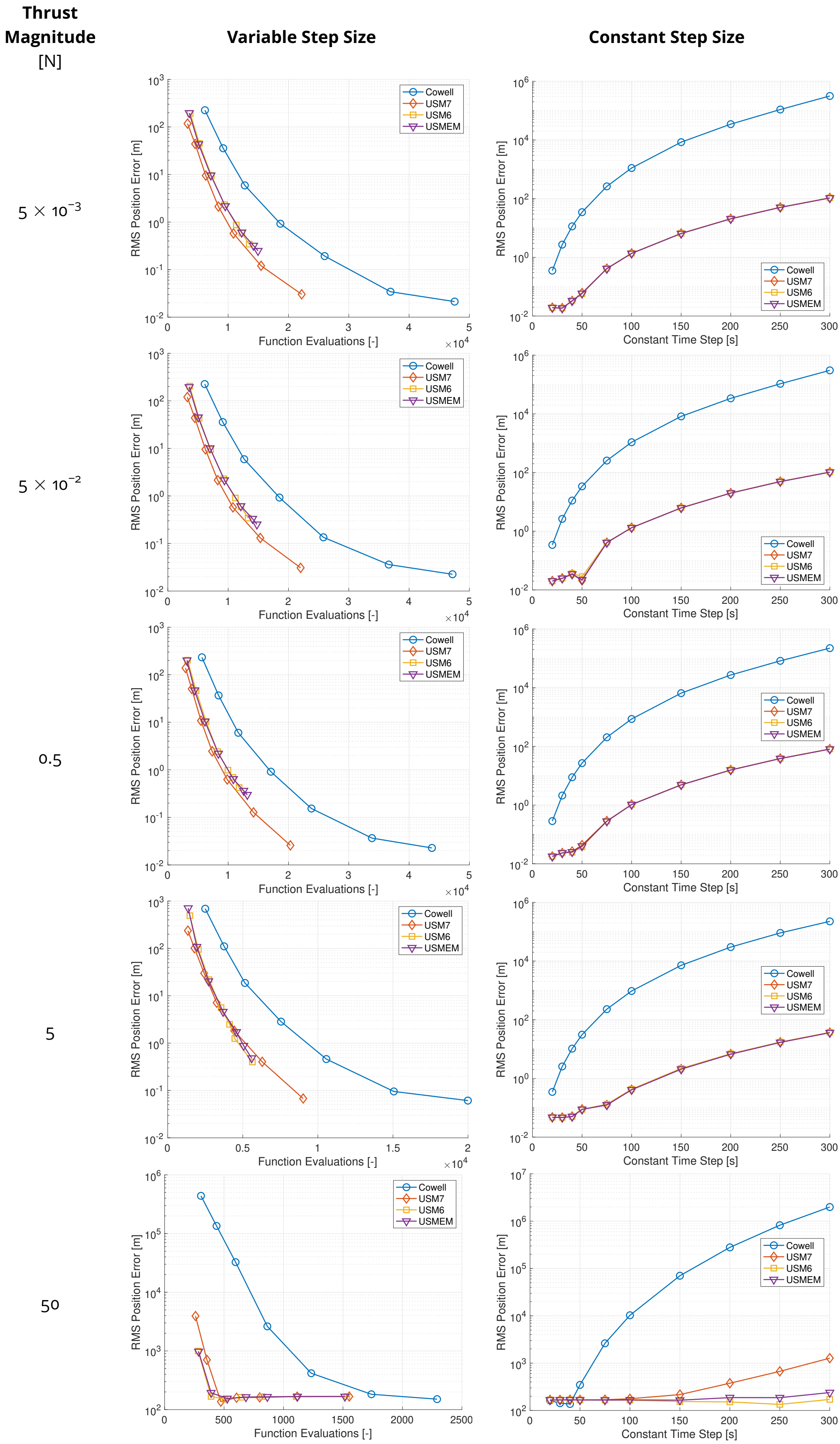
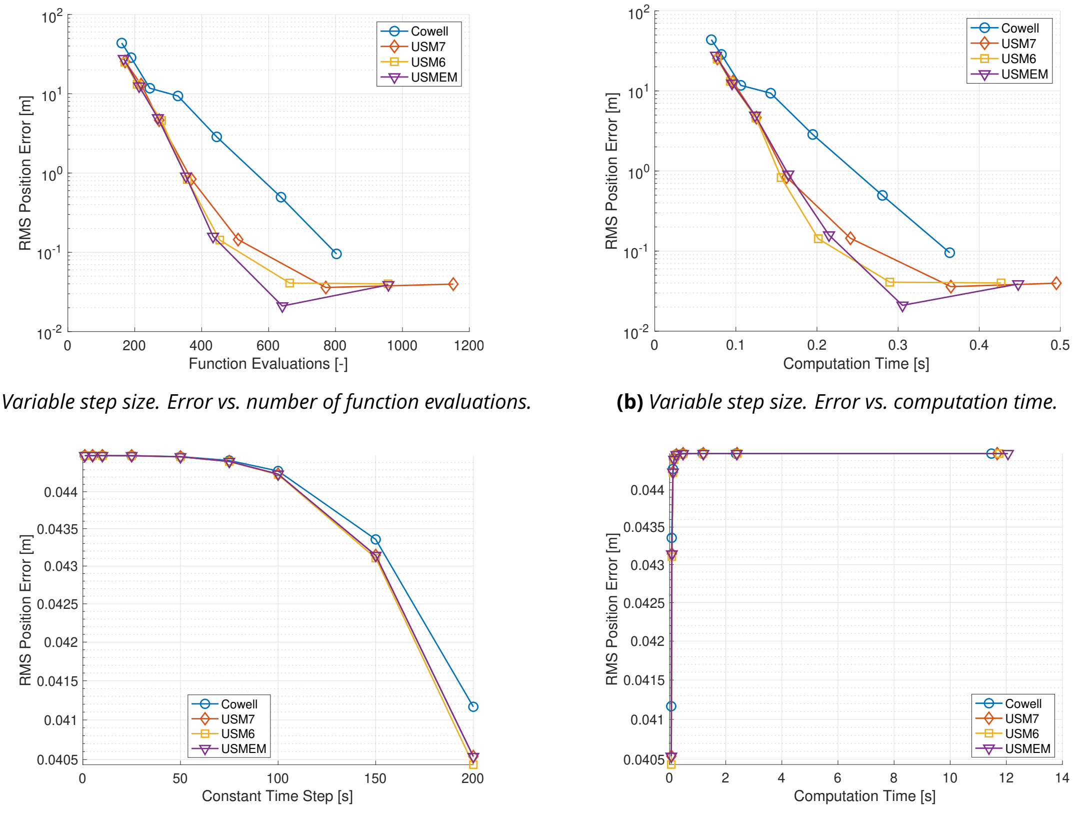


Figure 10.4: *Comparison of RMS position error for USM and Cowell propagators, for a low-thrust trajectory, with varying thrust magnitude.*



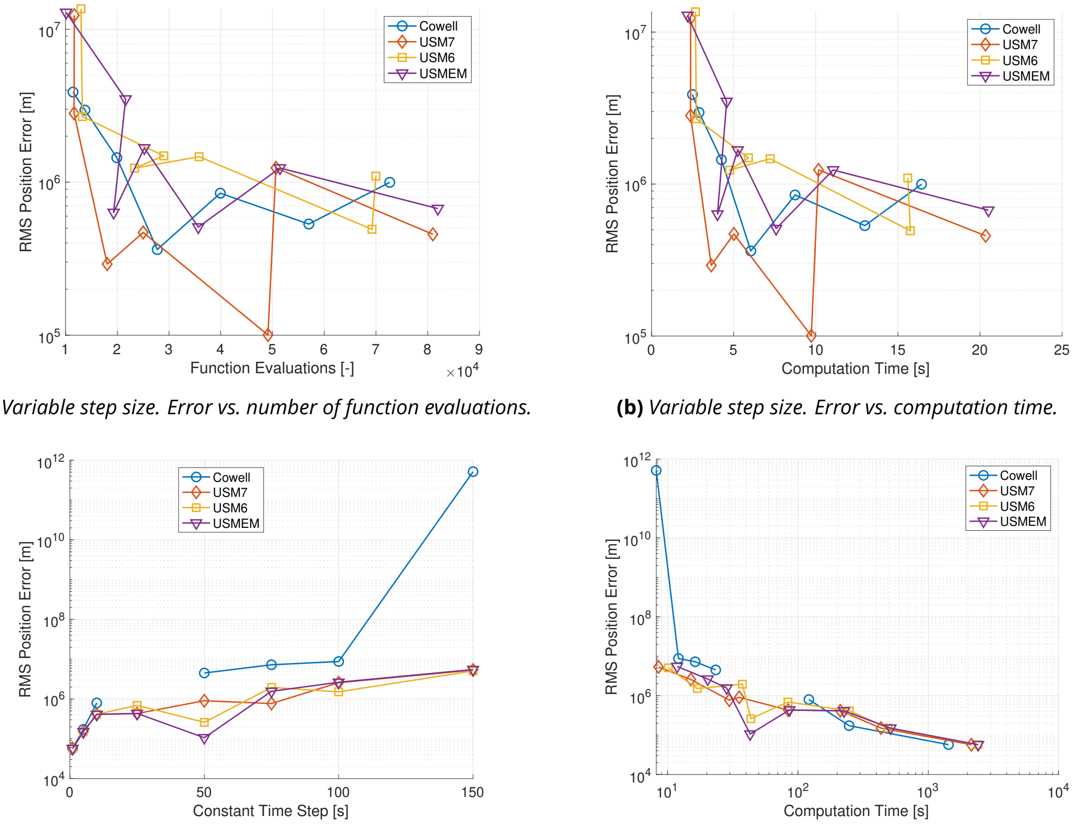
(a) *Variable step size. Error vs. number of function evaluations.*

(b) *Variable step size. Error vs. computation time.*

(c) *Constant step size. Error vs. constant time step value.*

(d) *Constant step size. Error vs. computation time.*

Figure 10.5: *Comparison of RMS position error for USM and Cowell propagators, for an aerocapture maneuver.*



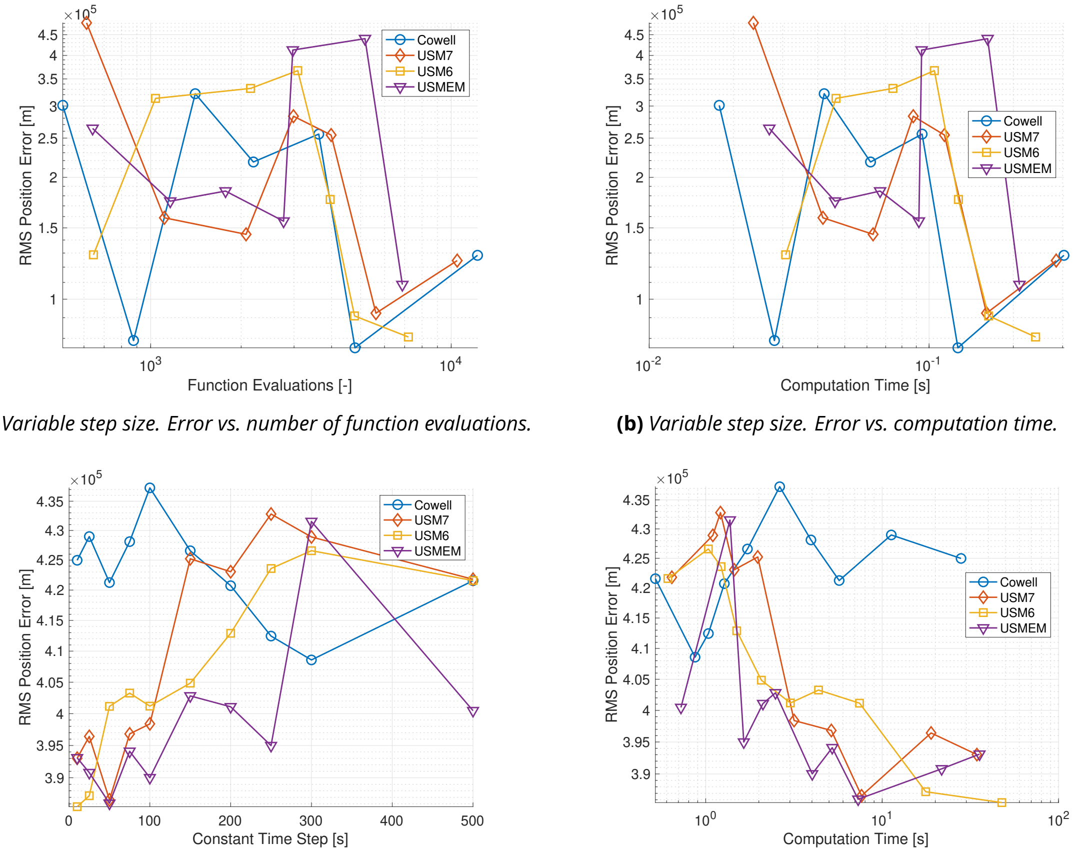
(a) *Variable step size. Error vs. number of function evaluations.*

(b) *Variable step size. Error vs. computation time.*

(c) *Constant step size. Error vs. constant time step value.*

(d) *Constant step size. Error vs. computation time.*

Figure 10.6: *Comparison of RMS position error for USM and Cowell propagators, for an aerobraking maneuver.*



(a) *Variable step size. Error vs. number of function evaluations.*

(b) *Variable step size. Error vs. computation time.*

(c) *Constant step size. Error vs. constant time step value.*

(d) *Constant step size. Error vs. computation time.*

Figure 10.7: *Comparison of RMS position error for USM and Cowell propagators, for an interplanetary orbit.*

11
Aerodynamic Modeling

As mentioned in Section 5.3, the aerodynamics of a vehicle depend on various conditions, not only related to its shape, but also to the environment it operates in. In this chapter, the aerodynamic coefficients of the spacecraft model developed for this thesis (whose description is provided in Appendix B) will be computed. This will be done for both the force and moment coefficients, and as a function of angle of attack, angle of side-slip and altitude.

Modeling of these coefficients is generally an important part of any atmospheric flight, but particularly for an aerobraking maneuver, where the success and duration of the mission are highly dependent on the aerodynamics (and atmospheric parameters). Thus the need of this chapter, and of Appendix C.3.

The chapter is structured as follows. Firstly, Section 11.1 introduces the reader to the force coefficients of the discretized MRO model. Then, in Section 11.2, one will find the description of the aerodynamic moment coefficients, together with an analysis of the solar panel bending moment. Finally, Section 11.3 will present the verification and validation for the current analysis.

Related to the investigation of the aerodynamics portrayed in this chapter is Appendix D. Here, an explanation of the use of the SPARTA software and how its output was processed, will be delineated.

11.1 | Aerodynamic Force Coefficients

In this section, the force coefficients resulting from the SPARTA simulation and the modified Newtonian method will be shown. Also, the bridging function introduced in Subsection 5.3.3 will be applied to this data, to obtain the coefficients for the transition regime.

11.1.1 | Rarefied Regime

Figure 11.1 shows the drag, side and lift coefficients for the rarefied regime, where drag and lift are shown as a function of α, whereas the side force coefficient is plotted against β. Despite not being shown, this same coefficient is almost zero, regardless of the angle of attack. This is in line with what is expected, due to the symmetry of the geometry around the $\hat{\mathbf{x}}_{\mathcal{B}} - \hat{\mathbf{z}}_{\mathcal{B}}$ plane. The reason why there may be slight offset from zero, stands in the fact that the modeling by Monte Carlo simulation leads to some uncertainty.

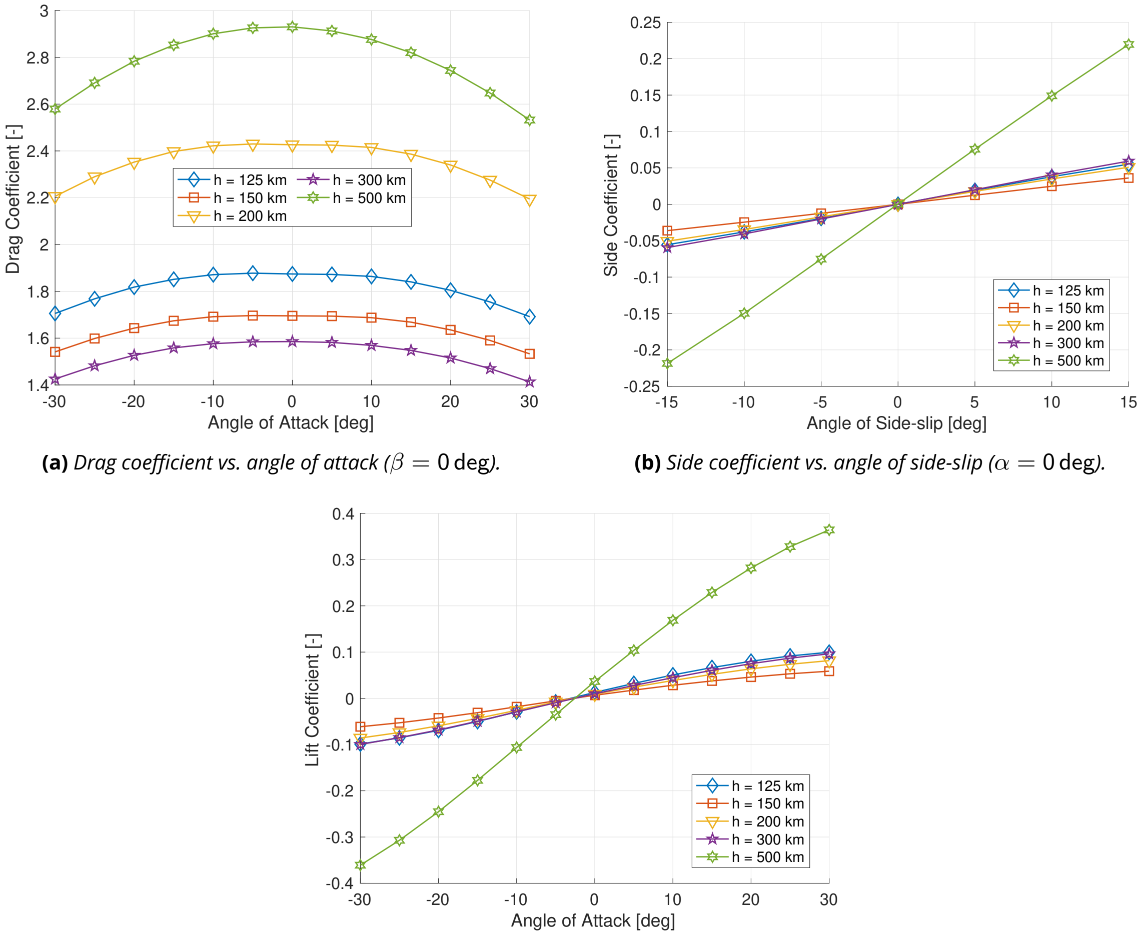
In Figs. 11.1a and 11.1c, the effect of different flow compositions and angle of attack becomes clear. The first effect can be explained by the fact that variation in molecular composition of the incoming flow (which for the altitudes shown in the figure can be found in Table D.2) gives raise to different molecular reactions and interactions with the surface. The latter effect, on the other hand, is due to the change of inclination w.r.t. the flow. In particular, it is clear that at low angles of attack (in magnitude), the drag coefficient is larger, and it slowly decreases with the increasing magnitude of α. In fact, at high angles of attack, the cube with three flat plates which constitutes the spacecraft model, is seen by the incoming gas as inclined. An inclined cube and inclined flat plates are clearly more streamlined than with zero angle-of-attack, thus the reduction in drag.

The lift coefficient, on the other hand, performs as expected: it increases with angle of attack, starts to level off at high α, and it has a low value. One interesting fact is that its value is non-zero at zero angle-of-attack.

The behavior of C_D and C_L as a function of side-slip angle, can be described as being independent, i.e., their values remain approximately constant as β varies. For this reason, their plots are not shown in this report.

11.1.2 | Continuum Regime

The results of the aerodynamic analysis with modified Newtonian method are shown in Fig. 11.2: Fig. 11.2a shows a combined plot of drag (left) and lift (right) coefficients as a function of angle of attack, and Fig. 11.2b on the other hand, shows the side force as a function of side-slip.

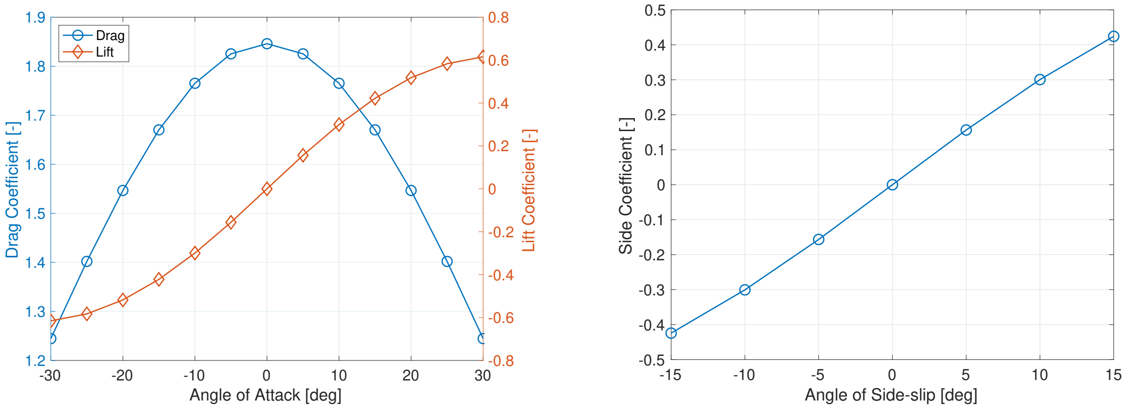


(a) *Drag coefficient vs. angle of attack ($\beta = 0$ deg).*

(b) *Side coefficient vs. angle of side-slip ($\alpha = 0$ deg).*

(c) *Lift coefficient vs. angle of attack ($\beta = 0$ deg).*

Figure 11.1: *Aerodynamic force coefficients for a range of aerodynamic angles and altitudes in the rarefied regime, computed by SPARTA with atmospheric properties based on the Mars Climate Database.*

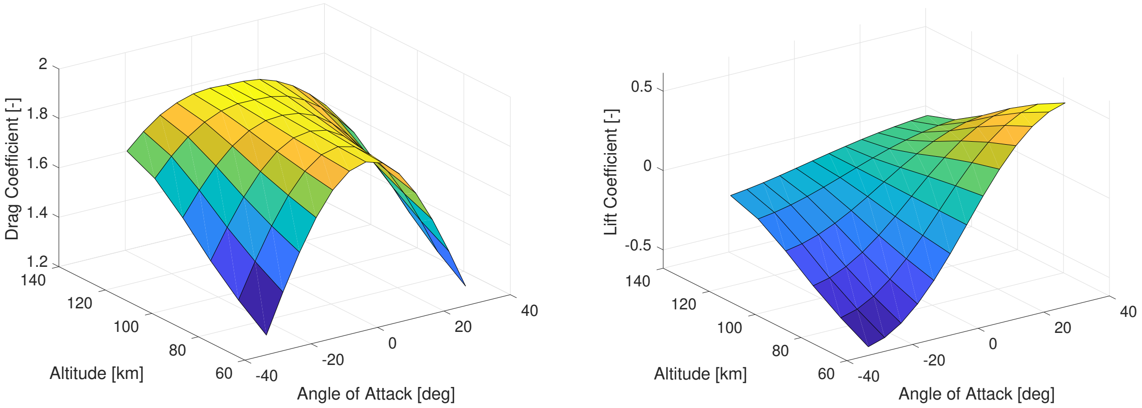


(a) *Drag and lift coefficients vs. angle of attack ($\beta = 0$ deg).*

(b) *Side coefficient vs. angle of side-slip ($\alpha = 0$ deg).*

Figure 11.2: *Aerodynamic force coefficients for a range of aerodynamic angles at the altitude belonging to the continuum regime, computed with the modified Newtonian method and with atmospheric properties based on the Mars Climate Database.*

In general, the behavior of C_D, C_S and C_L in the continuum regime is the same as for the rarefied flow. Some noticeable differences are the larger decrease in C_D with increasing angle of attack (in magnitude), and the higher side and lift coefficients.



(a) *Drag coefficient.*

(b) *Lift coefficient.*

Figure 11.3: *Aerodynamic force coefficients for a range of angles of attack and altitudes in the transition regime, computed by SPARTA with atmospheric properties based on the Mars Climate Database.*

11.1.3 | Transition Regime

For computation of the values in the transition regime, the bridging function from Subsection 5.3.3 is used. The result is a set of aerodynamic coefficients that depend on angle of attack, angle of side-slip and altitude. The boundary values (for altitude) used correspond to:

- 65 km altitude ($Kn \approx 10^{-3}$) for the lower bound (continuum regime)
- 125 km altitude ($Kn \approx 10^{1}$) for the upper bound (rarefied regime)

The resulting coefficients for the transition flow are shown in Fig. 11.3. The values for the side force coefficients are not displayed, since they provide less fundamental information for aerobraking.

11.2 | Aerodynamic Moment Coefficients

In this section, the moment coefficients resulting from the SPARTA simulation and the modified Newtonian method will be shown. Also in this case, the bridging function will be applied to the data, to obtain the coefficient for the transition regime.

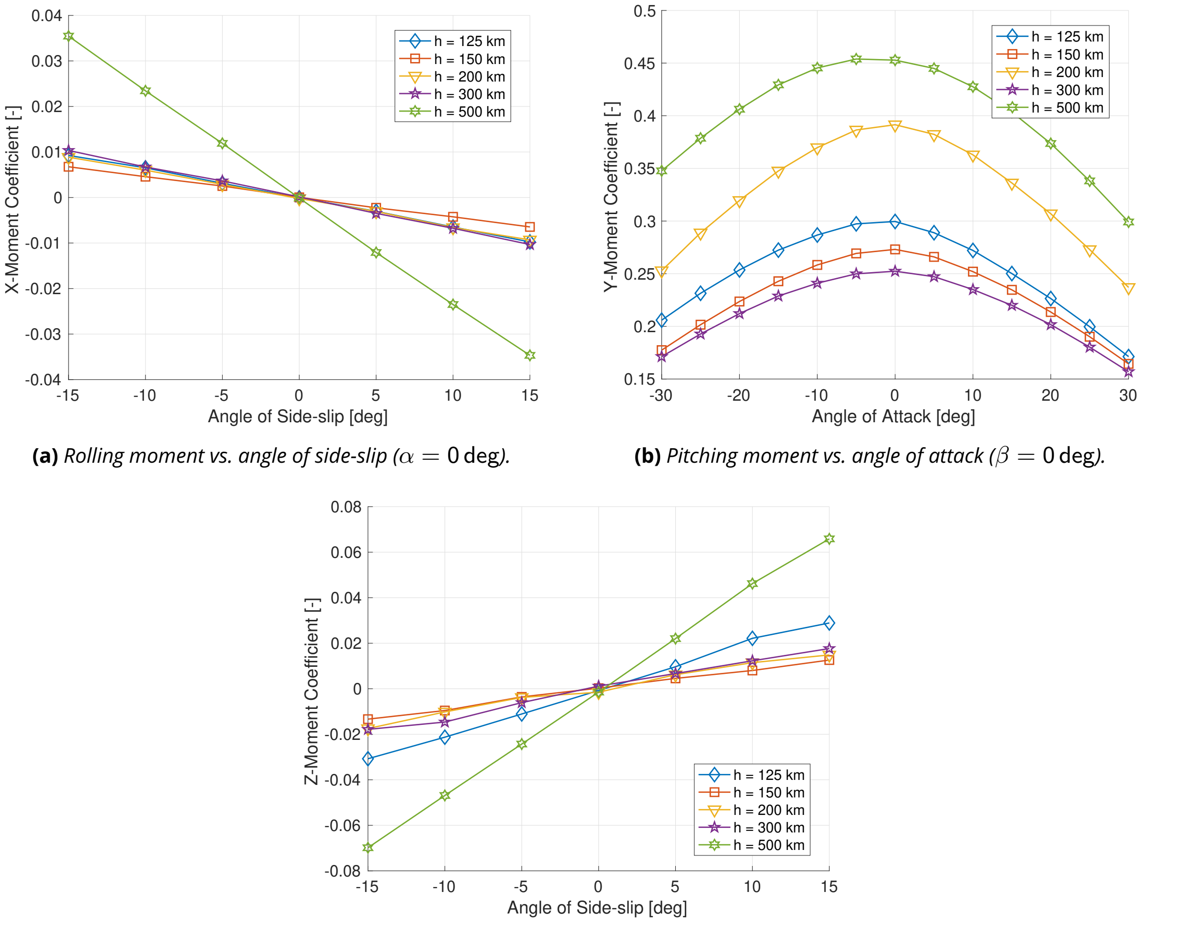
Subsection 11.2.4 will conclude this section, by inspecting the bending moment experienced by the solar panels. It is in fact essential to make sure that at no instant, the structural integrity of the solar array is compromised. A failure in this part of the structure would clearly result in a mission failure.

11.2.1 | Rarefied Regime

Figure 11.4 shows the aerodynamic moments about the $\hat{\mathbf{x}}_{\mathcal{A}}$, $\hat{\mathbf{y}}_{\mathcal{A}}$ and $\hat{\mathbf{z}}_{\mathcal{A}}$ axes for the rarefied regime. The first (C_l) and third (C_n) moment coefficients are shown as a function of angle of side-slip, whereas the second moment, i.e., the one about the y-axis, or C_m, is plotted against angle of attack. The value of C_l and C_n is always zero at zero-angle-of-side-slip (no matter the value of α), whereas C_m shows a low dependency on β, slightly decreasing in value as β increases in magnitude.

Looking at these plots, one can see that the only large moment is the one about the $\hat{\mathbf{y}}_{\mathcal{A}}$ axis. The other two moments are generally low in magnitude at the $\alpha = 0$ deg condition, due to the symmetry of the spacecraft around the $\hat{\mathbf{x}}_{\mathcal{B}} - \hat{\mathbf{z}}_{\mathcal{B}}$ and $\hat{\mathbf{y}}_{\mathcal{B}} - \hat{\mathbf{z}}_{\mathcal{B}}$ planes. In reality, the moment about $\hat{\mathbf{z}}_{\mathcal{A}}$ axis gives some larger fluctuations when compared to the $\hat{\mathbf{x}}_{\mathcal{A}}$ axis, but as will be shown in Subsection 11.3.1, these tend to flatten out when more accurate simulation conditions are used.

One interesting phenomenon that is clear from Fig. 11.4b, is that the curve representing the moment coefficient about the $\hat{\mathbf{y}}_{\mathcal{A}}$ axis is asymmetric w.r.t. 0 deg. The reason behind this shape can be explained by the asymmetry of the vehicle in the $\hat{\mathbf{x}}_{\mathcal{A}} - \hat{\mathbf{y}}_{\mathcal{A}}$ plane, due to the presence of the flat plate representing the antenna. At negative angles of attack, the antenna is fully 'visible' to the flow, and can thus counteract the effect of the lower surfaces of the solar panels and the bus (note how due to the shift in CM due to the antenna, the moment generated by the solar panels and the bus about $\hat{\mathbf{y}}_{\mathcal{A}}$ does not cancel out). On the other hand, at positive angles of attack, the antenna is partially shadowed by the spacecraft body, and can no longer counteract the solar panel-bus imbalance.



(a) *Rolling moment vs. angle of side-slip ($\alpha = 0$ deg).*

(b) *Pitching moment vs. angle of attack ($\beta = 0$ deg).*

(c) *Yawing moment vs. angle of side-slip ($\alpha = 0$ deg).*

Figure 11.4: *Aerodynamic moment coefficients for a range of aerodynamic angles and altitudes in the rarefied regime, computed by SPARTA with atmospheric properties based on the Mars Climate Database.*

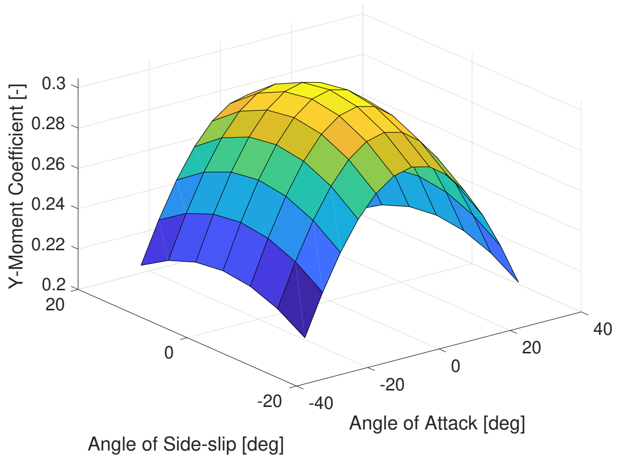


Figure 11.5: *Aerodynamic moment coefficient about $\hat{\mathbf{y}}_{\mathcal{A}}$ for a range of angles of attack and side-slip at the altitude belonging to the continuum regime, computed with the modified Newtonian method and with atmospheric properties based on the Mars Climate Database.*

11.2.2 | Continuum Regime

Figure 11.5 shows the aerodynamic moment coefficient computed with the modified Newtonian method, for continuum flow. Note that only the plot for the moment around $\hat{\mathbf{y}}_{\mathcal{A}}$ is shown, since the other two equal zero.

One will notice two clear differences in this plot w.r.t. the result from the rarefied flow analysis. Firstly, the plot is

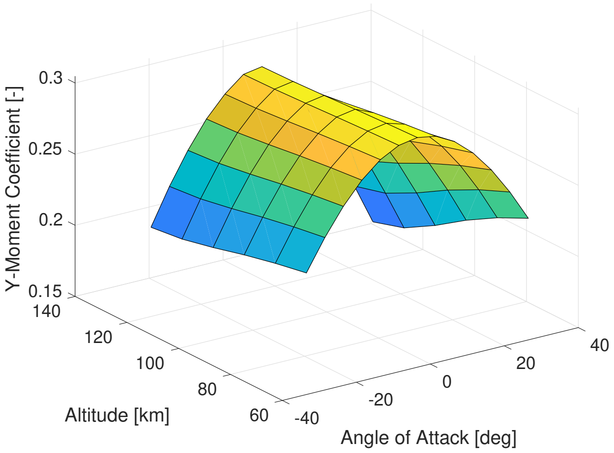


Figure 11.6: *Aerodynamic moment coefficients for a range of angles of attack and altitudes in the transition regime, computed by SPARTA with atmospheric properties based on the Mars Climate Database.*

symmetric w.r.t. the zero-angle-of-attack plane. This is because the model used for the calculation of the coefficient does not take into account shadowing. Thus, the discussion made in the previous subsection about shadowing of the antenna does not apply to this scenario. Secondly, the modified Newtonian method neglects the shear forces, and thus, no 'bump' at an angle of attack of 0 deg is present.

11.2.3 | Transition Regime

The final plot that will be shown for the moment coefficients is the one corresponding to the transition regime. As for the force coefficients, the same method is applied here to find the transitional values. This means that the bridging function in Eq. (5.15) is used, and the altitudes of 65 and 125 km are used as boundaries. The result is displayed in Fig. 11.3.

11.2.4 | Solar Panel Bending Moment

Another factor to consider for the design of a spacecraft performing aerobraking, is the load felt by the structure. In particular, the bending moment at the root of the solar panels may be a limiting factor. Since the design used in this analysis is a pre-existing one (i.e., no re-designing is considered in this thesis work), in this section, the analysis will be limited to laying the theoretical background to estimate the bending moment felt by the solar panels, due to the aerodynamic load.

In case the moment were to exceed a threshold, a different spacecraft configuration will have to be used for the aerobraking phase. For instance, the spacecraft could enter the atmosphere facing the negative $\hat{\mathbf{z}}_\mathcal{B}$ side, rather than the positive $\hat{\mathbf{x}}_\mathcal{B}$ one.

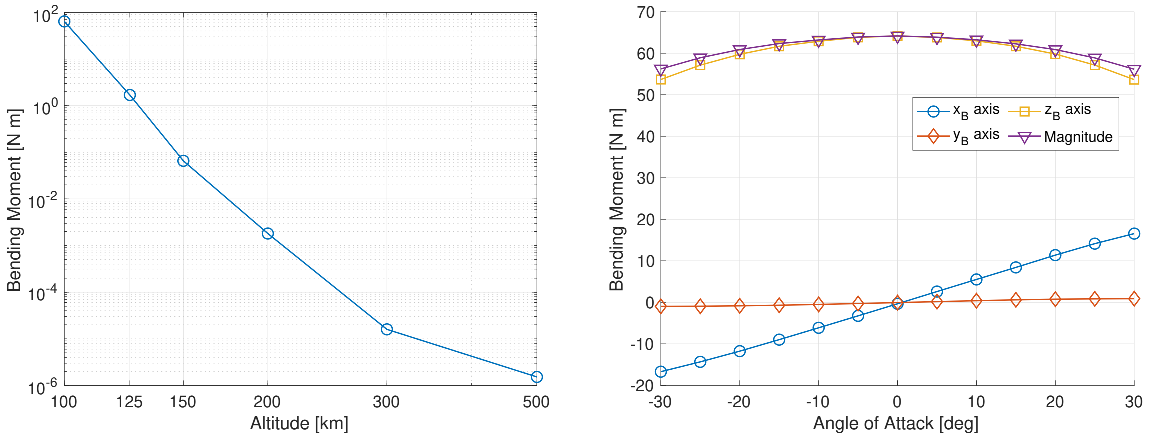
Using the results from the SPARTA analysis, the bending moment is computed with:

$$\mathbf{M}_{\text{SA}} = \sum_i \mathbf{L}_i \times \mathbf{F}_{i,\text{aero}} \tag{11.1}$$

where i runs over all the surface elements comprising the solar arrays, $\mathbf{L}$ is the distance from the centroid of the surface element to the root of the solar panel, and $\mathbf{F}_{\text{aero}}$ is the aerodynamic force, computed by summing up the pressure and shear stress output by SPARTA, with the surface area of the i-th element.

This analysis results in the moments shown in Fig. 11.7a, where the magnitude of the moment is taken for simplicity. Note however, that the moment is largest around the $\hat{\mathbf{z}}_\mathcal{B}$ axis. Moreover, the moment is largest at zero angle-of-attack, which is indeed the condition shown in Fig. 11.7a, for a range of altitudes and −5 deg of side-slip angle.

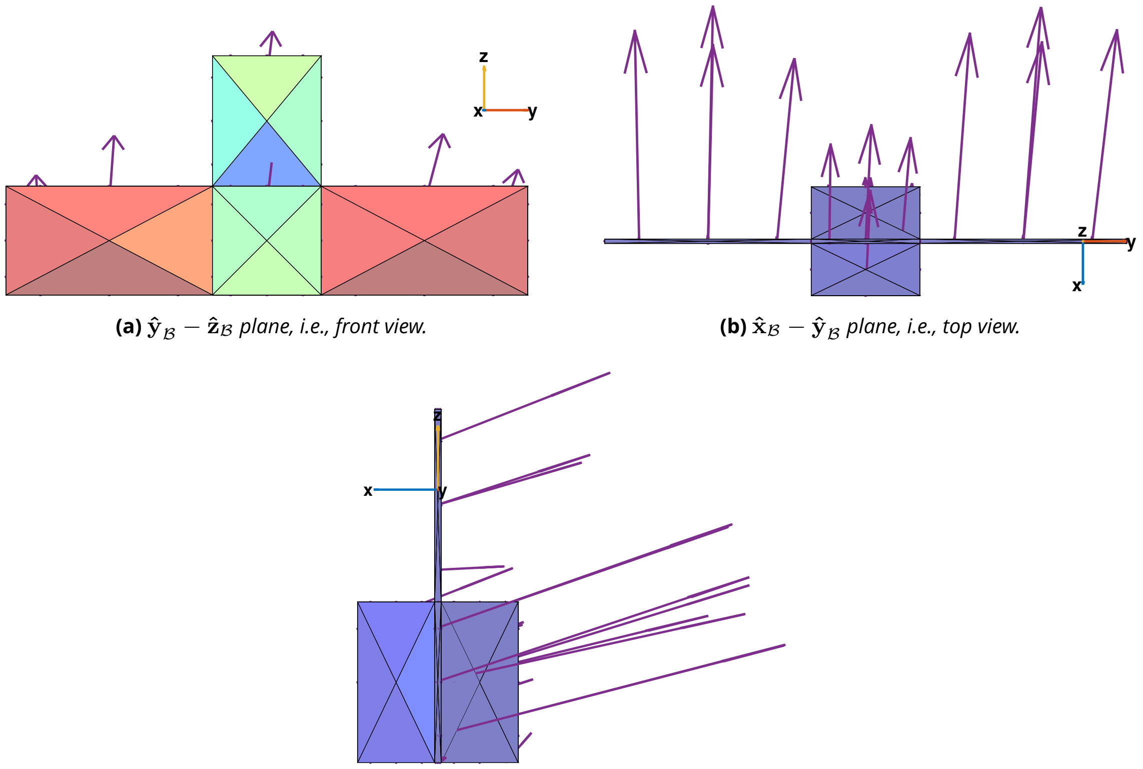
The maximum value of bending moment at the root of the panel, is thus experienced at 100 km, and it rapidly decreases with altitude. Said maximum value is equal to 64.2 N m. For sake of completeness, one can see the values of each component of $\mathbf{M}$ for a range of angles of attack, at a side-slip of −5 deg and 100 km, in Fig. 11.7b. Here, it is clear that, in fact, the moment is largest around the $\hat{\mathbf{z}}_\mathcal{B}$ axis, and that due to symmetry, almost no moment is felt around the $\hat{\mathbf{y}}_\mathcal{B}$ axis. The reason why this component is not exactly zero, is due to the asymmetry in the pressure distribution in the $\hat{\mathbf{y}}_\mathcal{B} - \hat{\mathbf{z}}_\mathcal{B}$ face of the solar panel. The upper part of the solar panel feels a slightly larger pressure force, as one can see from the dark red color of the solar panel in Fig. 11.8a.



(a) *Bending moment magnitude at zero angle-of-attack, and side-slip of −5 deg, experienced at the root of the solar panel, in the transition and rarefied regimes.*

(b) *Bending moment components at 100 km altitude, experienced at the root of the solar panel, for a range of angles of attack, and at an angle of side-slip of −5 deg.*

Figure 11.7: *Aerodynamic bending moment, based on the rarefied flow analysis with SPARTA, with atmospheric properties based on the Mars Climate Database.*



(a) $\hat{\mathbf{y}}_{\mathcal{B}} - \hat{\mathbf{z}}_{\mathcal{B}}$ *plane, i.e., front view.*

(b) $\hat{\mathbf{x}}_{\mathcal{B}} - \hat{\mathbf{y}}_{\mathcal{B}}$ *plane, i.e., top view.*

(c) $\hat{\mathbf{x}}_{\mathcal{B}} - \hat{\mathbf{z}}_{\mathcal{B}}$ *plane, i.e., side view.*

Figure 11.8: *Aerodynamics forces acting on the spacecraft at 30 deg angle of attack, angle of side-slip of −5 deg and 100 km altitude, based on the rarefied flow analysis with SPARTA, with atmospheric properties based on the Mars Climate Database. Note that the axes are used as scale, and they have length 1 N. Also, the spacecraft plates are color-coded based on the magnitude of the force acting on it.*

At first, the reader might be confused by the value of the moment about the longitudinal axis, i.e., $\hat{\mathbf{x}}_{\mathcal{B}}$. As the flow is coming itself from the $\hat{\mathbf{x}}_{\mathcal{A}}$ direction, one would not expect a moment about this axis. However, the subtle difference between $\hat{\mathbf{x}}_{\mathcal{B}}$ and $\hat{\mathbf{x}}_{\mathcal{A}}$ is exactly where the value of the bending moment arises. At non-zero angles of attack, the flow will impact the solar panels at an angle, and thus the total force felt by the panel will be slightly tilted in the same direction as the incoming flow. This can be seen in Fig. 11.8, where three orthogonal views of the spacecraft are provided.

11.3 | Verification and Validation

In the sections above, the aerodynamic modeling of the discretized spacecraft shape of Appendix B was described. This analysis was carried out with SPARTA, whose general use and implementation was described in Appendix D. In this section, the verification and validation of the aerodynamic coefficients obtained will be carried out.

Verification is performed by comparing the results with more accurate settings. As was mentioned in Appendix D, one has to choose some simulation parameters based on the geometry of the spacecraft. In Subsection 11.3.1, the aerodynamic coefficients will be compared with more strict settings, to make sure that the selected values do not lead to inaccurate results. Finally, Subsection 11.3.2 will portray the validation of the results, by comparing them with the data obtained by an accurate DSMC analysis of the actual MRO spacecraft model.

11.3.1 | Comparison with Accurate Settings

As mentioned in Appendix D.1, the settings for ratio of real to simulated particles was selected such that the program would simulate 17.5 particles per cell. Moreover, the grid spacing, i.e., the length of the sides of each cell was chosen to be 0.25 m.

To make sure that such parameters would not give a too simplified result, the simulation was run three more times (for only one altitude, namely 125 km) with the following parameters:

1. **Accurate Parameters**
   - *Number of particles per cell*: 35
   - *Grid spacing*: 0.15 m
2. **Accurate Number of Particles**
   - *Number of particles per cell*: 50
   - Grid spacing: 0.25 m
3. **Accurate Grid Spacing**
   - Number of particles per cell: 17.5
   - *Grid spacing*: 0.1 m

The same procedure as explained in the previous sections were used to analyze the results of the new simulations, and the resulting drag, lift and moment (about $\hat{\mathbf{y}}_{\mathcal{A}}$) coefficients are shown in Fig. 11.9, where the previous results (with baseline, i.e., less accurate, parameters) are also shown. For sake of completeness, values of the computation times are also given, and can be found in Table 11.1.

Table 11.1: *Computation time to run a SPARTA simulation for 13 angles of attack and one altitude, when using the TU Delft servers (Intel® Xeon® Processor E5-2683 v3) with 14 cores.*

Baseline	Accurate Parameters	Accurate Number of Particles	Accurate Grid Size
0.5 h	12 h	12 h	32.5 h

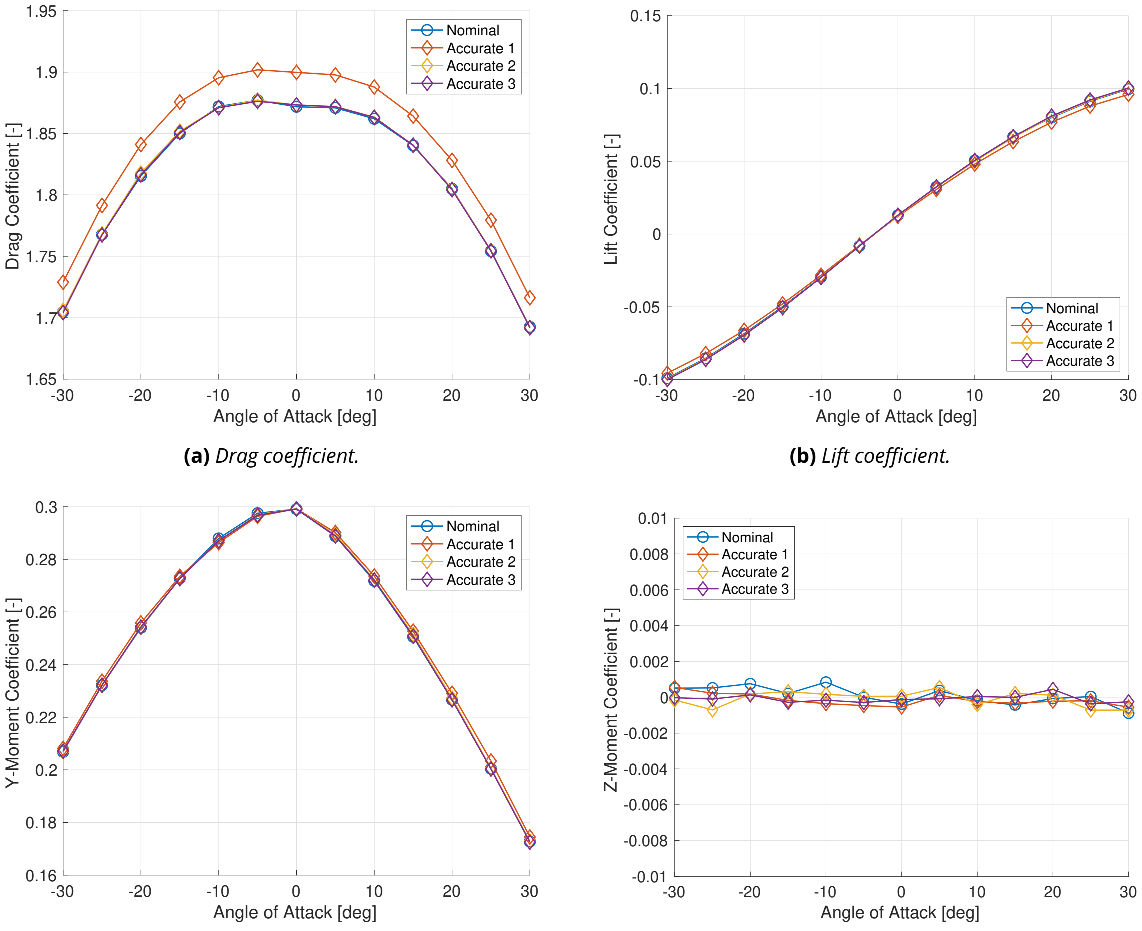
Clearly, the baseline results agree very closely with the simulations using both accurate number of particles and accurate grid spacing. However, such agreement is not found when both parameters are set to more accurate values. As can be seen in Fig. 11.9a, the offset is especially large in the computed drag coefficient. The fact that only when both parameters are changed gives a different result, can be explained by a strong coupling of the two parameters. Due to the preliminary nature of this research, this effect will be disregarded, and the aerodynamic coefficients from Sections 11.1 and 11.2 will be assumed to represent the MRO coefficients.

Moreover, one can see from Fig. 11.9d that the moment coefficient about the $\hat{\mathbf{z}}_{\mathcal{A}}$ axis is indeed zero, when the precision of the simulated environment is increased.

The last considerations made are regarding the computation times from Table 11.1. Clearly, SPARTA can be a very computationally expensive task, and that is why the parameters have been given the values of 17.5 and 0.25 m.

11.3.2 | Validation

From the analysis carried out by Gallis et al. (2004), on the aerothermal characteristics of MRO, the authors extracted a logarithmic fit to the behavior of the drag coefficient w.r.t. the atmospheric density, which was determined by



(a) *Drag coefficient.*

(b) *Lift coefficient.*

(c) *Moment coefficient about $\hat{\mathbf{y}}_{\mathcal{A}}$.*

(d) *Moment coefficient about $\hat{\mathbf{z}}_{\mathcal{A}}$.*

Figure 11.9: *Comparison of aerodynamic coefficients computed by SPARTA, with baseline and all three accurate settings (based on number of particles per cell and grid size), and with atmospheric properties based on the Mars Climate Database, corresponding to 125 km altitude. The number in the legend for the accurate settings, corresponds to the list shown above.*

performing a DSMC . This fitting line is expressed as:

$$C_D\left(\rho\right) = 1.47952 - 0.032578 \log\left(\rho\right) \qquad 10^{-10} \leq \rho < 2 \times 10^{-7} \tag{11.2}$$

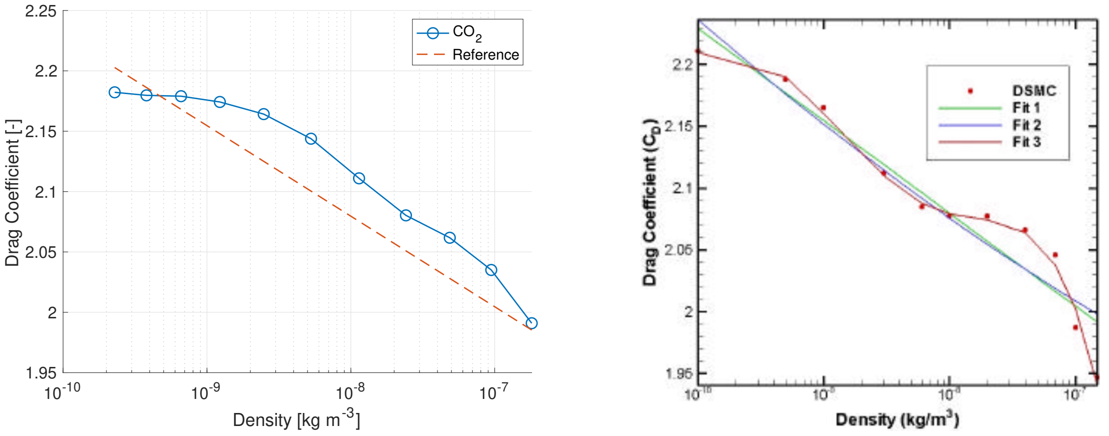
Comparing the values obtained at zero angle-of-attack at various altitudes, with the values output by this function, gives the plot in Fig. 11.10. Note that for this comparison, the SPARTA simulation was run by assuming a constant atmospheric composition, given by 100 % carbon dioxide. It should be specified, however, that the authors do not specify which gases were used in the simulation, apart from mentioning the presence of $\mathrm{CO_2}$.

Table 11.2, shows the same values of drag coefficient as displayed in the figure, and also gives the altitudes corresponding to the densities, and the offset in percentage, w.r.t. the validation data.

Clearly, the results generated by SPARTA agree quite closely to the validation data, by even showing the two 'bumps' at 10^{-7} and 10^{-9} kg m^{-3}, and the flattening after 10^{-10} kg m^{-3}.

One interesting fact, however, is the high dependency of the simulation results, on the atmosphere gases. In fact, looking at Fig. 11.11, one can see the zero angle-of-attack results from Section 11.1, plotted against density. Note that the drag coefficients at conditions corresponding to an altitude of 100 km is added in this figure, where the atmosphere is assumed to be composed solely of $\mathrm{CO_2}$.

The effect of varying atmospheric gases becomes very clear from this graph. Whereas the drag coefficient is about 2.14 at 125 km altitude with 100 % $\mathrm{CO_2}$, the addition of 8 % nitrogen, causes a drop in drag coefficient, which reaches the value of 1.87. The difference becomes even larger at 150 km height, where C_D reaches the value of 1.69. It would therefore appear, that the varying atmospheric composition cannot be disregarded.



(a) *Results from the SPARTA analysis carried out for validation with constant atmospheric composition. The solid line represents the results given by SPARTA, and the dashed line shows the logarithmic fit in Eq.* (11.2).

(b) *Results from DSMC analysis carried out by* Gallis et al. (2004)*. The red dots represent the results at simulated conditions, and the three lines are obtained by fitting the data with three different methods. The light green line (i.e., Fit 1), corresponds to Eq.* (11.2).

Figure 11.10: *Validation of drag coefficient at zero angle-of-attack, as a function of atmospheric density, by comparing to aerothermal analysis of MRO.*

Table 11.2: *Validation of results from the SPARTA simulation described in this chapter, with output of Eq.* (11.2)*, for various atmospheric density conditions. Note that the altitude corresponding to the density is also given.*

Altitude [km]	**Density** [kg m^{-3}]	**Computed** [–]	**Reference** [–]	**Difference** [%]
100	1.8×10^{-7}	1.99	1.99	0.3
105	9.5×10^{-8}	2.04	2.01	1.4
110	4.9×10^{-8}	2.06	2.03	1.7
115	2.4×10^{-8}	2.08	2.05	1.4
120	1.1×10^{-8}	2.11	2.08	1.7
125	5.3×10^{-9}	2.14	2.10	2.1
130	2.5×10^{-9}	2.16	2.13	1.8
135	1.2×10^{-9}	2.17	2.15	1.2
140	6.6×10^{-10}	2.18	2.17	0.5
145	3.8×10^{-10}	2.18	2.19	−0.3
150	2.3×10^{-10}	2.18	2.20	−0.9

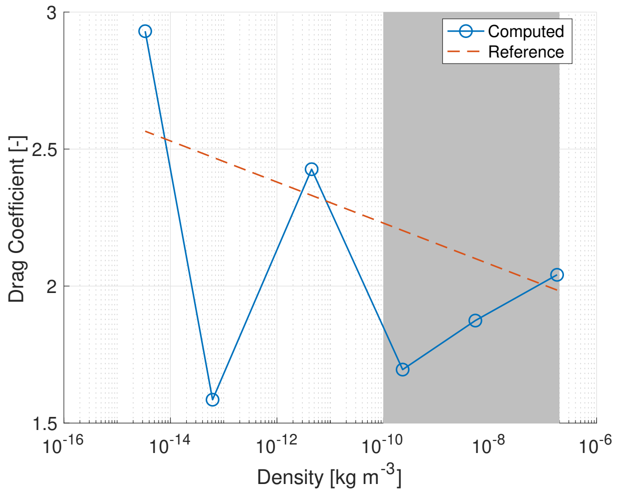


Figure 11.11: *Drag coefficient of MRO at zero angle-of-attack, as a function of atmospheric density, based on the rarefied flow analysis with SPARTA. The points correspond to the simulation altitudes and conditions from Table* D.2*, and the shaded area is the region of validity (domain) of Eq.* (11.2).

PAGE INTENTIONALLY LEFT BLANK

12 Aerobraking

In this chapter, the results of the application of the aerobraking guidance, navigation and control system outlined in Part II will be applied to a maneuver around Mars. Section 12.1 will describe the full simulation settings used for this analysis, by describing the environment and spacecraft characteristics, whereas Section 12.2 will provide a thorough description of the results obtained, by comparing them to the reference MRO mission and by critically examining the guided conditions and their dependent attributes. Finally, Section 12.3 concludes this section, by looking back at the requirements listed in Section 2.3, in order to 1) see which of these have been met, and 2) revisit the ones that were defined as 'to be defined' or 'determined' (TBD). Before continuing to the sections just marked out, a few considerations are in order.

Unfortunately, due to time constraints, the simulation was only run with the exponential model from Eq. (4.29). Usage of Eq. (4.33), i.e., of the five-parameter exponential model, however, in some preliminary testing, did not seem to produce better results. This can be blamed on the fact that the two coefficients of the trigonometric functions were always close to zero, thus yielding an atmospheric model very close to the exponential one.

12.1 | Simulation Settings

The environment settings used for the simulation of the aerobraking mission have been introduced in Section 4.3. In particular, a distinction was made between the propagation (Subsection 4.3.2) and the onboard (Subsection 4.3.3) implementations. For sake of completeness and clarity, a brief summary of the simulation settings is found in Table 12.1. One can obtain the full Tudat settings in Appendix G.

Now that the environment has been defined, one can set the initial conditions. These are the ones shown in Table 12.2, and they represent a highly elliptical orbit, with an inclination and peri- and apoareion altitudes very similar to the ones of MRO at the beginning of aerobraking.

Finally, the last elements to be introduced are the ones that were tuned during the results generation process, i.e, the noise matrices for filtering and the PID gains for control.

The process and measurement noise matrices used for the state estimation of the navigation system (explained in Section 6.3) are given by:

$$\begin{aligned}\mathbf{Q} &= \mathrm{diag}\left(10^2, 10^2, 10^2, 10^{-1}, 10^{-1}, 10^{-1}, 10^{-4}, 10^{-4}, 10^{-4},\right.\\ &\qquad\qquad\qquad \left. 10^{-4}, 10^{-4}, 10^{-4}, 10^{-4}, 5 \times 10^{-9}, 5 \times 10^{-9}, 5 \times 10^{-9}\right)^2\\ \mathbf{R} &= \mathrm{diag}\left(5 \times 10^3, 5 \times 10^3, 5 \times 10^3, 10^{-6}, 10^{-6}, 10^{-6}\right)^2\end{aligned}$$

Table 12.1: *Propagation and onboard environment settings used for the simulation of the GNC system for an aerobraking maneuver around Mars.*

	Spherical Harmonics	Atmosphere	Solar Radiation	Third Bodies	Aerodynamics
Propagation	$(21, 21)$	Tabulated MCD from Appendix C.3	From Sun, Mars as occulting body, and parameters from Table G.1	Sun	Tabulated aerodynamic coefficients from Chapter 11
Onboard	$(4, 4)$	Exponential model from Table G.2	NA	NA	Constant drag coefficient, $C_D = 1.9$

Table 12.2: *Initial conditions for the main test case for the analysis of the performance of the GNC system developed in this thesis for an aerobraking maneuver. These conditions correspond to a highly elliptical orbit, similar to the injection orbit of MRO (adapted from Long et al., 2008).*

a [km]	e [–]	i [deg]	ω [deg]	Ω [deg]	ϑ [deg]	Δt [d]	t
26 021.0	0.859 882	93.0	43.6	158.7	180.0	1.4	June 30, 2007 at 12:00 Noon

and were optimized to reduce the uncertainty in position, velocity and attitude. Moreover, their units are such that they match the units of state and measurement vectors defined by Eqs. (6.7) and (6.15), respectively.

The proportional, integral and derivative gains, on the other hand, were tuned to reduce the error between the commanded and actual attitude, a process which yielded:

$$\mathbf{K}_p = [15.0, 22.5, 12.5]^{\mathrm{T}}$$
$$\mathbf{K}_i = [3.75, 17.5, 5.0]^{\mathrm{T}}$$
$$\mathbf{K}_d = [7.5, 20.0, 10.0]^{\mathrm{T}}$$

12.2 | Results

In this section, the results of the aerobraking maneuver based on the onboard computer software of Part II, the environment and equations of motion of Part I and the initial conditions of the previous section, will be discussed.

The section is divided into three parts. In the first part, the results corresponding to a the first orbit will be analyzed. Here, the filtering results and the rotational motion will be described. In Subsection 12.2.2, on the other hand, the full aerobraking trajectory will be analyzed. Thus, one will find the examination of the final orbit and of the behavior of the Keplerian elements over time. Then, in Subsection 12.2.3, the reader will be shown the performance of the elements of the OBC responsible for the targeting of the pericenter altitude.

It should be noted that the implementation of the rotational motion from Section 4.2 and, consequently, of the attitude control system from Subsection 7.2.1 is only carried out for the single orbit analysis of Subsection 12.2.1. Hence, the results shown in the last two sections describe an aerobraking maneuver with ideal control. Furthermore, the commanded attitude differs from the one shown in Subsection 7.2.1, and instead is the attitude that corresponds to the following condition:

$$\alpha = \beta = 0\,\text{deg} \qquad \text{and} \qquad \sigma = 180\,\text{deg}$$

which is valid throughout the orbit (and not just at periapsis as previously defined). As a result of the omission of control of rotational motion, the state and measurement vector for state estimation of Section 6.3, had to be adapted. The new vectors are described by the following system:

$$\mathbf{x} = \left[\mathbf{r}^{\mathrm{T}}, \dot{\mathbf{r}}^{\mathrm{T}}, \mathbf{b}_t^{\mathrm{T}}\right]^{\mathrm{T}} \tag{12.1a}$$

$$\mathbf{z} = \mathbf{x}^{\mathbf{r}}|_{\mathcal{I}} \tag{12.1b}$$

12.2.1 | Single Orbit

Analysis of a single orbit is carried out to inspect both the behavior of the navigation filter of Section 6.3, and of the attitude control system of Subsection 7.2.1. Note that for this analysis, the periapsis scaling of Subsection 7.1.1 (which is active during walk-in) is switched off, i.e, the pericenter will enter deeper layers of the atmosphere (115 vs. 138 km, to be exact). This is to make sure that the aerodynamic loads used to test the control system are consistent with the worst case scenarios.

The filter convergence plots are shown in Fig. 12.1 for translational motion only. In the plots, one can see both the difference between estimated and actual position of the spacecraft, and the standard deviation (STD) of the estimate. Note that the large gaps in the image refer to those times when the navigation system is in the unaided phase, and thus does not use the onboard filter.

Figure 12.1 shows how the estimate of the translational state matches very closely the actual position and velocity of the spacecraft. In fact, the standard deviation is in general at a constant sub-kilometer value. The only regions where the estimate of the state exceeds the standard deviation is in the transition from unaided to aided navigation,

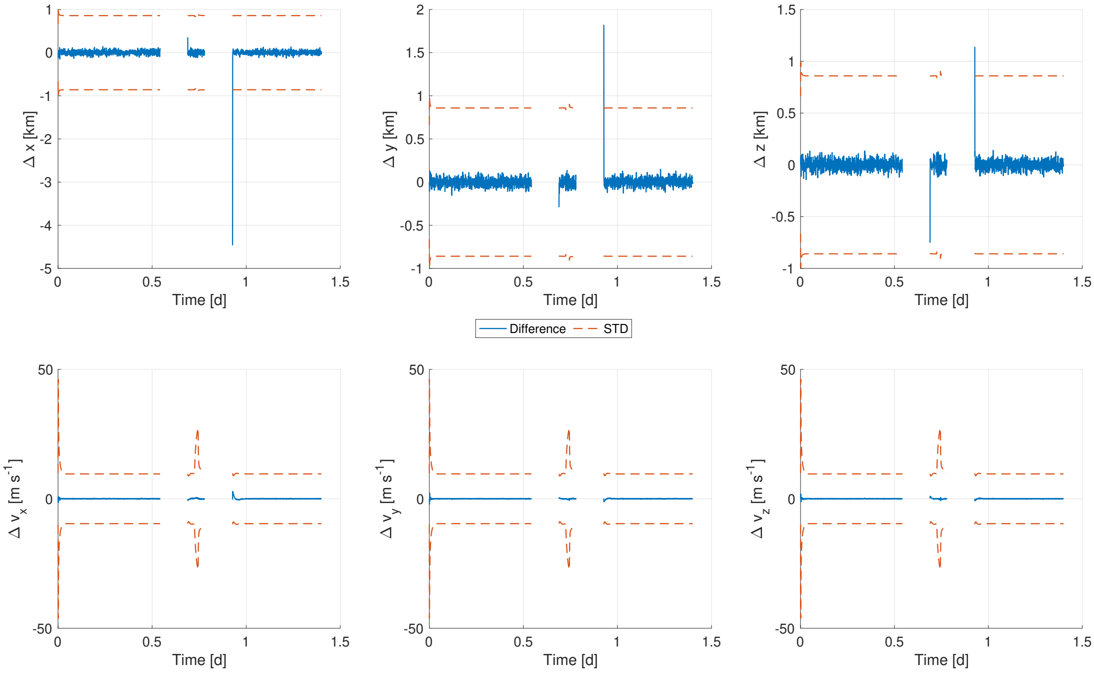


Figure 12.1: *Convergence plots for the Cartesian elements estimated by the navigation filter over the first orbit. The dashed line represents the standard deviation of the estimates.*

i.e., when the OBC has to correct for the error in estimated state that cumulated over time.

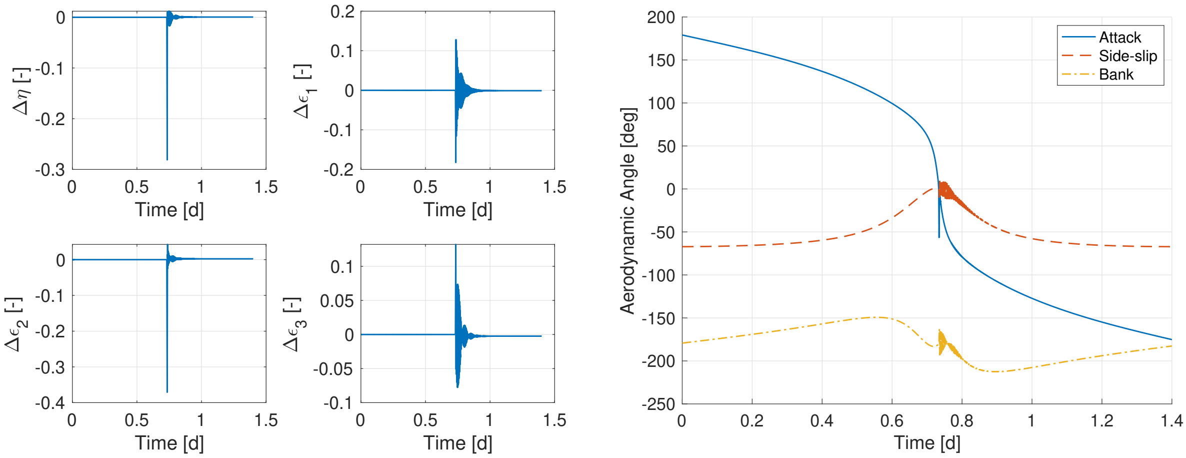
A similar convergence pattern is noticed when looking at the estimation of the IMU bias errors. The filter quickly manages to estimate their values, converging to within 2.5 % of the actual value in the first orbit.

Then, one can also look at the rotational motion. Figure 12.2 shows the offset between actual and commanded quaternions (Fig. 12.2a) and the values aerodynamic angles (α, β and σ) over time (Fig. 12.2b). Note that the value of the commanded quaternion (before the next periapsis) is given by:

$$\mathbf{q}_c = [0.907, -0.102, -0.383, -0.140]^{\mathrm{T}}$$

and corresponds to the condition explained in Subsection 7.2.1.

Before the atmospheric phase, which occurs roughly 0.75 days after the beginning of the maneuver, the actual quaternions match very closely the commanded value. During and right after the atmospheric phase, however, the equilibrium of the spacecraft is perturbed and as a consequence, the attitude angles undertake an oscillation around their default values. Nevertheless, the PID controller manages to constrain the rotational motion of the spacecraft



(a) *Difference between actual and commanded quaternions over the first orbit.*

(b) *Aerodynamic angles over the first orbit. Some angles are plotted beyond −180 deg for sake of clarity.*

Figure 12.2: *Rotational motion over the first orbit.*

to a ±0.1 quaternion range, which can be roughly translated to a variation of ±10 deg in aerodynamic angles, if the small angle approximation is considered, based on (Sola, 2017):

$$\Delta\mathbf{q} \approx \left[1, \tfrac{1}{2}\Delta\boldsymbol{\vartheta}^{\mathrm{T}}\right]^{\mathrm{T}}$$

Soon after, in fact, the oscillations in quaternions (and thus in aerodynamic angles) dissipate and the attitude is restored to the wanted value.

As stated at the beginning of Section 7.2, the development of the controller is only added to establish a benchmark of the GNC system behavior, thus no requirement was set on the attitude error. Looking back to the results obtained, it is clear that the PID controller is enough to achieve the required attitude at the first periapsis, but there is no guarantee that this will hold for following orbits, since the prolonged behavior has not been probed. Therefore, further testing and possibly development of a more sophisticated controller is recommended as future work.

Looking again at the plot of the aerodynamic angles over time, one will notice that the angle of attack spans the full range of possible angles (from −180 to +180 deg). This is to be expected, since the commanded attitude is based on periapsis conditions. It is clear then, that REQ-SC-8 is incompatible with REQ-C-2, and that the full range of aerodynamic angles should have been applied for the aerodynamics analysis of Chapter 11.

From the discussion above, it follows that the values of aerodynamic coefficients used at high altitudes, i.e., where the aerodynamic angles are beyond the modeled values, are a result of extrapolation, and have no physical meaning (at angles of attack of ±180 deg, the spacecraft should present the same aerodynamic coefficients as for 0 deg, due to symmetry, but the values differ significantly in the simulation). Gladly, since the altitudes are indeed large, the aerodynamic loads are negligible and the results of this simulation are not affected, but this should be considered for orbits where the circularization has already occurred, i.e., for later stages of aerobraking.

Finally, one can look at the rotational velocity of the spacecraft during the atmospheric phase, to determine the value of momentum to be generated by the reaction wheels (RWs), in order to stabilize the spacecraft. Figure 12.3 shows just these values: the angular rates about each body axis.

From Fig. 12.3, one can see that the atmospheric phase (at around 0.75 days), causes the spacecraft to start rotating around each axis. These rotations can be transformed to momentum according to (Wie, 2008):

$$\mathbf{H} = \mathbf{J}\boldsymbol{\omega} \tag{12.2}$$

where $\mathbf{J}$ follows from the analysis in Appendix B.2. The result of this equation would then be the required angular momentum to be generated by the reaction wheels. If then one takes the largest of these products, one can set a preliminary minimum threshold of said value. For the current analysis, this is:

$$H_{\text{RW,min}} = 47.5\,\text{N m s}$$

where the assumption of REQ-SC-6 is applied, and the value represents the maximum angular momentum reached around a single axis. Judging of whether achieving this value is feasible, is not deemed part of this research, but rather it should give insight during the sizing of the reaction wheel system. However, for sake of completeness, the reaction wheels of MRO were each capable of delivering a momentum of 100 N m s at the maximum rotational velocity (Lee et al., 2003). Clearly, this would be enough to preform the necessary corrective adjustments. Nonetheless,

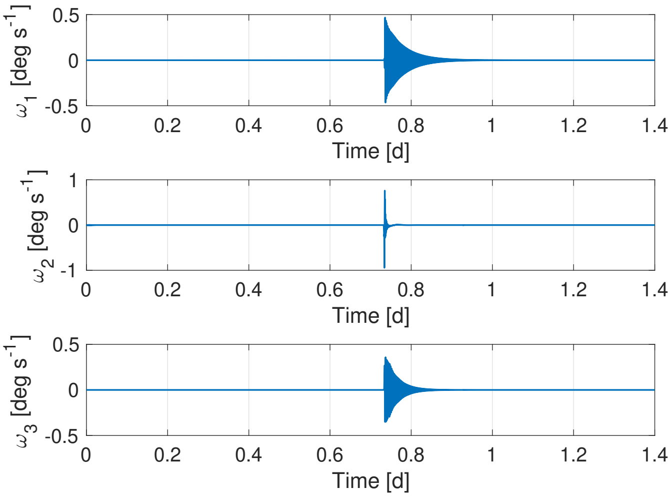


Figure 12.3: *Angular velocity of the spacecraft around each axis, over the first orbit.*

usage of a more robust control algorithm (which also takes into account the maximum RW momenta), could provide an even more efficient and less demanding control scheme. Not to mention that the magnitude of oscillations in quaternions could be reduced both in amplitude and duration.

12.2.2 | Aerobraking Trajectory

To analyze the full aerobraking results, propagation of rotational motion and active control of attitude are turned off. This is in accordance with the investigation of the control system done in the previous subsection, where it became clear that a more complete aerodynamic analysis (in terms of tested aerodynamic angles) is needed.

Figure 12.4 shows the result of 82 days, or 210 orbits, of aerobraking around Mars. Here, one can see the clear braking effect, which arises due to atmospheric drag. The orbit starts off as highly elliptical, and over the course of three months decreases its eccentricity, until it achieves a quasi-circular configuration. In fact, as is reported in Table 12.3, the mean eccentricity during the 'aerobraking complete' phase is approximately 0.016, which together with the other Keplerian elements, gives an average altitude of about 315 km.

FINAL ORBIT CONFIGURATION Comparing the peri- and apocenter altitudes from Table 12.3, with the target values set for the guidance system in Table 7.1, one can compute their relative offset: −0.3 % for periapsis and +15.7 % for apoapsis. Since no requirement on the targeting accuracy was set, one cannot say whether the mission is successful. However, one can compare the accuracy achieved by this simulation, and the one achieved by the actual MRO mission. According to Long et al. (2008), the final orbit configuration of MRO after aerobraking, was one ranging in altitude between 215 and 485 km.

Of course, this comparison is rather unfair. The aerobraking maneuver performed by MRO was a real-life scenario, i.e., a much more challenging and demanding environment. Moreover, it is possible that the reason for the final aerobraking orbit to be different, is unrelated to the aerobraking maneuver itself. Thus, the MRO final configuration simply sets the lowest bar in acceptable accuracy, which in this case, is overtaken with a relatively large margin.

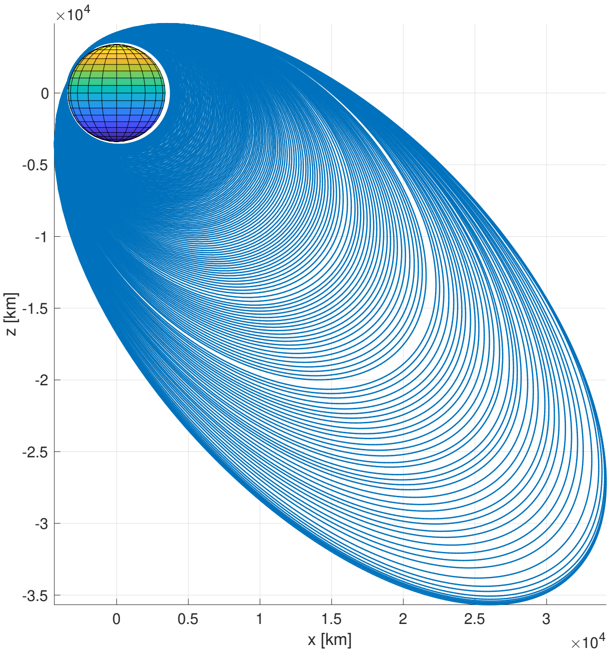


Figure 12.4: *Full aerobraking trajectory for the initial conditions in Table 12.2. The blue lines represent the spacecraft trajectory, whereas the multi-colored sphere represents Mars.*

Table 12.3: *Final orbit configuration after an aerobraking maneuver around Mars, expressed as mean Keplerian elements and mean peri- and apoapsis altitudes.*

$\bar{a}$ [km]	$\bar{e}$ [–]	$\bar{i}$ [deg]	$\bar{\omega}$ [deg]	$\bar{\Omega}$ [deg]	$\bar{h}_p$ [km]	$\bar{h}_a$ [km]	t
3701.7	0.015 657	78.8	3.5	179.8	254.2	370.11	September 18, 2007 at 7:10pm

Keplerian Elements Evolution In Fig. 12.5, the Keplerian elements as a function of time are shown. The plots that are of highest interest are the ones depicting the semi-major axis a, and the eccentricity e, i.e., Fig. 12.5a.

From this image, one can distinguish two main features: a series of *spikes*, followed by a *drop* in value. The spikes are due to the presence of the spherical harmonics gravity model. The consequence of this conservative force, is in fact, to add to the Keplerian elements oscillations centered at periapsis, whose magnitude is especially large for highly elliptical orbits. This was also very clear in the plots shown in Fig. 6.1, where the concept of DAIA was introduced. Arising due to a conservative force, the oscillation only have a temporary impact.

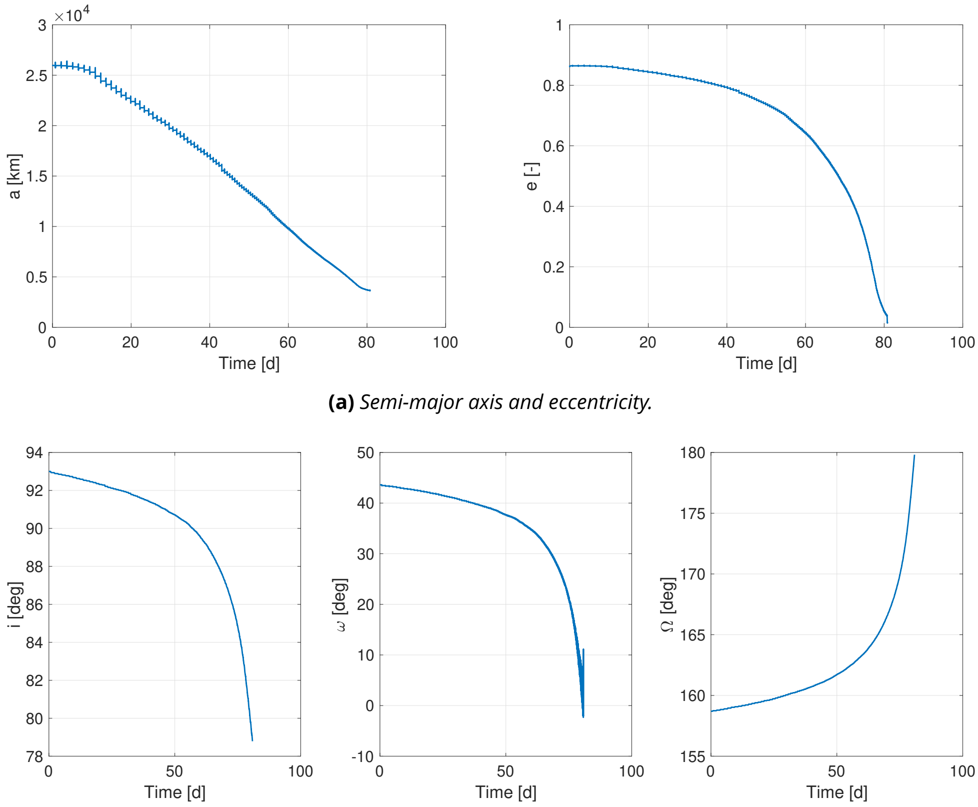
The overall reduction in the value of a and e, is given by the presence of atmospheric drag. This force, on the other hand, is not conservative, and leads to a very significant and permanent impact on their value. The effect can be seen right after the aforementioned oscillations, where a step-like jump can be spotted.

Other step-like kinks are present around some apocenter regions. This is the time when the apoapsis maneuver is performed. Since this maneuver is assumed to be instantaneous, the effect on the Keplerian elements is a sudden jump in a and e. Nonetheless, due to the small thrust magnitude these maneuvers partake, these characteristics are not visible at this scale.

The result of the maneuver on the other Keplerian elements, portrayed in Fig. 12.5b, is less prominent. As a matter of fact, the rate of change of these elements becomes larger only when the eccentricity is low, and the spherical harmonics gravity has a more prolonged influence.

Aerobraking Duration Another parameter of interest that can be extracted from Fig. 12.5 is the duration of the aerobraking maneuver. Clearly, the maneuver lasts short of 82 (Earth) days, which correspond to 210 orbits around Mars. Compared to the duration of the maneuver performed by MRO, this is roughly half (140 days and 425 orbits). This difference can be interpreted in the following ways:

○ *The guidance algorithm implemented in this thesis is more 'aggressive', meaning that the periapsis altitude is lowered into deeper layers of the atmosphere.*
This is clearly not the case when comparing the periapsis altitudes of Fig. 12.6 and Fig. 2.6b. In the latter plot, one can see that the average periareion altitude for MRO was about 105 km, compared to 115 km for ones resulting from the algorithm developed in this thesis. In fact, if anything, it would appear that the heating



(a) *Semi-major axis and eccentricity.*

(b) *Inclination, argument of pericenter and right ascension of ascending node. Some angles are plotted below 0 deg for sake of clarity.*

Figure 12.5: *Full evolution of Keplerian elements for an aerobraking maneuver around Mars. The plot for true anomaly is not reported, due to the low amount of information it provides.*

values allowed are stricter than in the MRO case (remember how REQ-AB-4 and REQ-AB-5 are set based on the values set for TGO).

- *Scheduling of the maneuvering sequence and of the other GNC activities, delayed the aerobraking since they had to be performed on the ground.*
  This would also appear to be a flawed hypothesis, since as shown in both Figs. 2.6b and 2.6c, the pericenter altitude, and consequently the heat rate, remained rather constant throughout the 140-day period.
- *The tabulated atmosphere used to represent the Martian atmosphere described a thicker atmosphere than encountered by MRO.*
  Since the atmosphere values from Appendix C.3 are based on time-average values, the density could have been in a local minimum, in the period and region where MRO was operating. As it turns out, also this proposition may have to be discarded, considering that 2007 was a year that saw the complete enshrouding of Mars in a global dust storm. These type of events always lead to a factor of two increase in density, even in higher layers of the atmosphere. Looking at the values describing the corridor boundaries in Fig. 2.6c, however, one can see that the heat rates encountered are in the same range as set by REQ-AB-4. Thus, one can conclude that the atmosphere model used in this thesis is indeed more conservative than the real atmosphere experienced by MRO.

Looking back at the definition of the aerodynamic acceleration, i.e., Eq. (4.23), the only parameters that have not been mentioned yet, are the spacecraft aerodynamic parameters. These however, have been validated in Section 11.3, where it was shown that the values of drag coefficient match very closely the results of similar studies.

Clearly, there must be other forces at play here. Examination of where these discrepancy arise is significant, and is recommended as a further analysis of the results. The most likely explanation is that not all of the periapses altitudes are shown in Fig. 2.6b, but only the ones where aerobraking was conducted. Therefore, the duration of aerobraking itself, for MRO, is lower.

Filtering Process Regarding the filtering process, the average per-orbit STD position and velocity uncertainties during both aided and unaided navigation are shown in Table 12.4. Clearly, thanks to the GRS system, the spacecraft knows its position within a 1σ certainty of 50 m, in each direction.

Table 12.4: *Value of 1σ uncertainty in position and velocity during aerobraking, for both navigation methods.*

(a) *Aided navigation.*

Position [m]	**Velocity** [m s^{-1}]
50.0	0.1

(b) *Unaided navigation.*

Position [m]	**Velocity** [m s^{-1}]
1500.0	0.04

During unaided navigation, on the other hand, this uncertainty grows considerably larger. This is because of two main reasons: the initial value of the spacecraft state propagated already has a small error (given by the values of STD from Table 12.4a), and the propagation with reduced environment introduces even more errors, due to the simplified onboard environment model. Due to limitations in onboard computational power, the latter cause cannot be tackled without knowing proper OBC processing specifications, but the former could be mitigated: one could take the median of the Keplerian elements estimated in the few minutes before, to reduce the uncertainty in state knowledge. Use of the median would smooth out the noise, and use of Keplerian elements is justified due to their slow variation over time. True anomaly is the only Kepler coordinate that does not show a small time-dependency, and as such may require a more sophisticated fitting model, such as linear or quadratic regression.

Nonetheless, as already shown in Subsection 12.2.1, when it is time to switch between unaided and GRS-assisted navigation, the filtering scheme does a great job at correcting the state estimate, by converging to the actual position (within the STD shown above), within a matter of minutes. This is actually one of the reasons why simulation of aerobraking with uncertainty in initial conditions was not carried out. It is in fact known from the transition between navigation phases, that the state estimation process is perfectly capable of reproducing the initial position value, despite the initial offset. This is true, however, only for the GAN method (since a direct measurement in position is available), but would require further investigation for the IMAN method.

12.2.3 | Periapsis Targeting

Periapsis targeting consists of the combined functionalities of estimating the periapsis corridor and computing the required maneuver to aim for the target altitude. It is arguably the most distinctive feature of an aerobraking software, and also one of its most sensitive.

In Fig. 12.6 the periapsis corridor estimates are plotted against the orbit number. Here, the actual and estimated periapsis altitude are also shown, together with the apoapsis maneuvers (distinguished between up or down, based on whether their purpose is to raise or lower the periapsis).

Aerobraking Phases From the image, one can clearly see the various aerobraking phases described in Subsection 7.1.1. The gradual lowering of the corridor between the first and seventh orbit is a sign of the walk-in phase. The rather constant corridor values between the eighth and the 147th orbits, on the other hand, are part of the main phase. The walk-out phase, finally, can be spotted beyond the 148th orbit, and is also clearly visible in Fig. 12.5a, where one can observe, in the values of a and e, a tendency to plateau. Note that termination and 'aerobraking complete' cannot be seen on this graph, since corridors are not defined for these two cases: the target values of periapsis and apoapsis from Table 7.1 are used directly, without any need of corridor estimation.

Two main features are worth noting from Fig. 12.6: the slow decrease of the periapsis corridor bounds during the main phase, and the zig-zag-looking behavior of the boundaries during walk-out. These behaviors can be explained as follows:

- The negative slope of the corridor threshold in the main phase is related to the lowering of the eccentricity (and semi-major axis) of the orbit. As a matter of fact, as the aerobraking mission evolves, the orbital energy is slowly dissipated. This leads to a slower velocity through the pericenter passage, and thus, lower aerodynamic heat rate. Of course, however, as the eccentricity decreases even further, the time spent in the atmosphere increases considerably, and the cumulative heating becomes more prominent and in turn more constraining. This is (together with the lifetime constraint), the same reason why during the final orbits of the main phase and during walk-out, the pericenter corridor shows an overall positive trend.
- Regarding the other feature, i.e., the zig-zag behavior during the walk-out phase, the blame is in the fact that the guidance algorithm is not future-oriented. The system only thinks about meeting the heating and lifetime requirements for the next orbit, without considering how this will affect the same estimation the next time around. In fact, the zig-zag behavior mainly arises whenever the lifetime constraint is not met. Take the sequence of orbits 188 to 190, for instance. On orbit 188, the onboard computer sees that the lifetime requirement would not be met, if only the heating constraints are used for defining the corridor conditions. Therefore, it decides to raise the periapsis. Now, on orbit 189, the periapsis is rather high (about 126 km), and the next periapsis would result in a rather low dynamic pressure. Consequently, the corridor estimator decides to lower the periapsis, such that more drag can be experienced and aerobraking can last a shorter amount of time. However, now that the pericenter is low again, the lifetime constraint is, once again, no longer met and the system decides to raise the periapsis. One can see how this could (and in fact, it does) lead to a back-and-forth game between lifetime constraint not being met, and low dynamic pressure.

A more optimized guidance, would take into consideration these very contrasting requirements, and possibly provide a more gradual atmosphere-exit strategy. For instance, one could define a simple multiplier function, such as Eq. (7.1) for walk-in, but for the walk-out phase. Removal of the zig-zag behavior would thus considerably reduce the amount of ΔV consumed during walk-out, which, as will be reported in Table 12.5, accounts for roughly 30 % of the total aerobraking velocity increments.

Maneuver Estimator Looking now at the targeting of the pericenter conditions, it can be seen that the periapsis stays within the defined corridor for most of the orbits. However, there are quite some exceptions to this statement, and they are all highlighted with a filled blue dot. The periapsis value at orbit 40 is one example of this. Clearly, the apoapsis maneuver which aims to target the midpoint of the corridor fails to do so, and instead undershoots by a large margin, reaching a value of almost 110, instead of 115 km. The fault is clearly in the maneuver estimator, since the corridor bounds are inline with the previous values. In fact, if one looks closely at the other cases where the corridor is missed, the faulty value is almost always preceded by a maneuver. This is a clear sign that the maneuver estimator of Subsection 7.1.3 does not perform as expected, when combined with the rest of the software. A discussion on the possible sources of errors is provided in the next paragraphs.

Corridor Estimator A similar discussion can be had, however, about the corridor estimator. In fact, when looking at Fig. 12.7, which depicts the peak dynamic pressure, heat rate, heat load and density for each orbit, the maximum values of the two heat parameters are overtaken on a number of occasions. Interestingly, their values are exceeded in different phases of aerobraking (disregarding the outliers, or rather misfires, such as orbit 40):

- **Heat rate**: the value of heat rate is mainly exceeded in mid main phase
- **Heat load**: heat load is mainly exceeded at the end of the main phase and at the beginning of walk-out; this is when the orbit eccentricity is low enough such that the spacecraft spends a longer amount of time in the

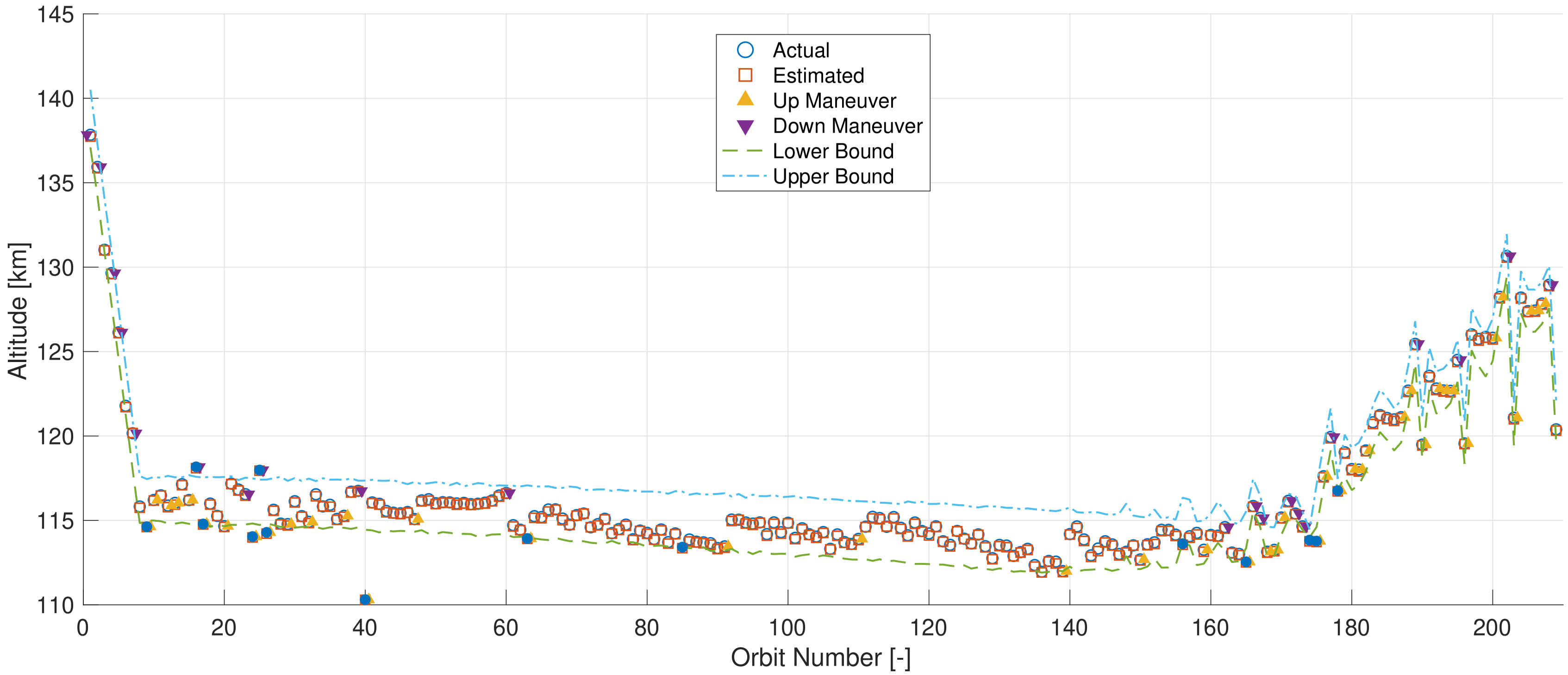
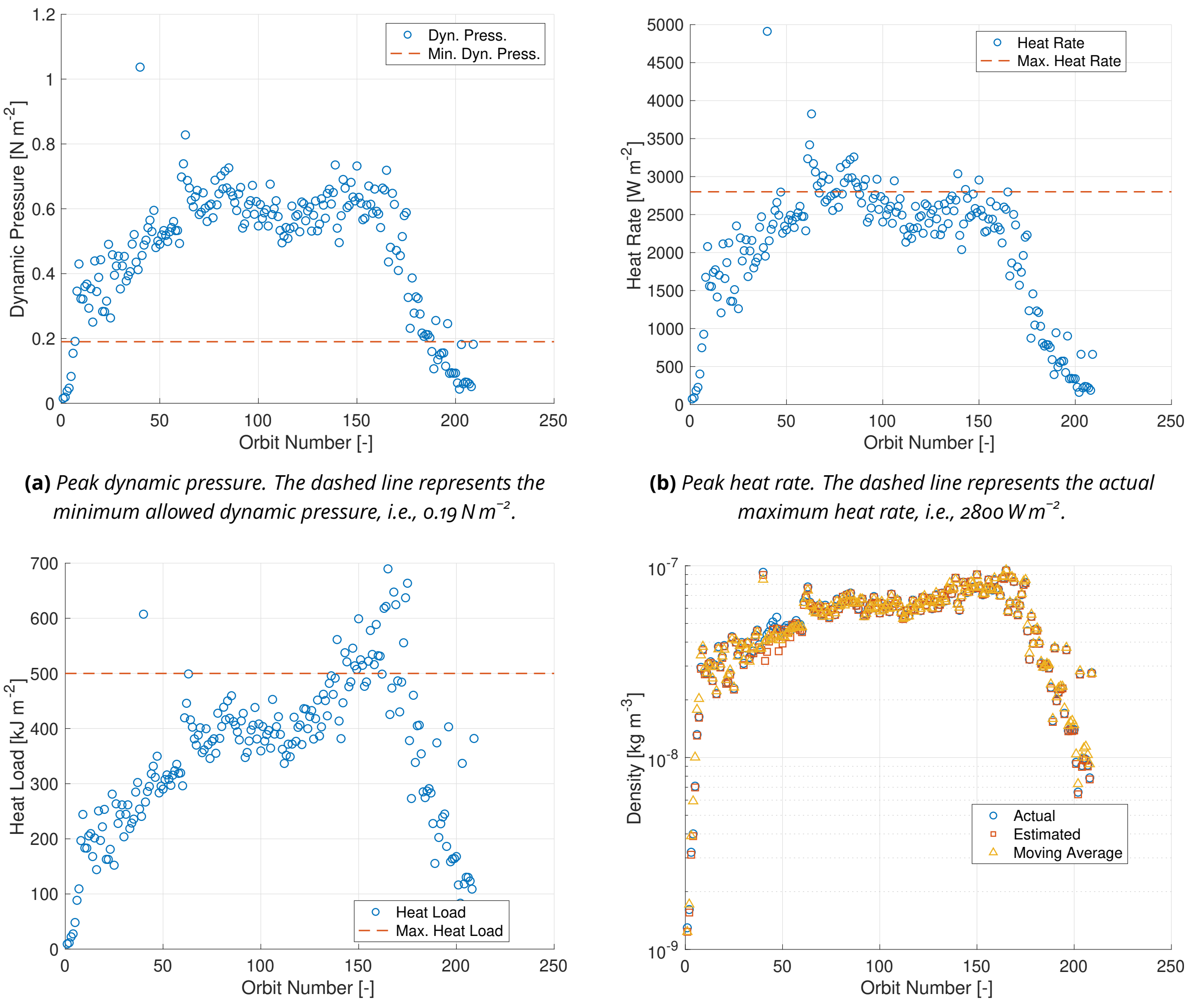


Figure 12.6: *Actual and estimated periapses altitudes for each orbit, plotted together with the corridor boundaries and the apoapsis maneuver instances. The filled dots represent periapses that do not respect the corridor boundaries.*



(a) *Peak dynamic pressure. The dashed line represents the minimum allowed dynamic pressure, i.e., 0.19 N m⁻².*

(b) *Peak heat rate. The dashed line represents the actual maximum heat rate, i.e., 2800 W m⁻².*

(c) *Peak heat load. The dashed line represents the actual maximum heat load, i.e., 500 kJ m⁻².*

(d) *Peak density values. The square dots represent the estimated density based on the current orbit, whereas the triangles show the density based on the moving average of the atmospheric parameters.*

Figure 12.7: *Comparison of peak actual and estimated values of some aerothermodynamic parameters encountered in each orbit.*

atmosphere, and despite the lower velocities, reaches a large cumulative heat; this is the reason why Fig. 12.7c shows the spike around orbit 175, which is not visible in Fig. 12.7b

Note that the maximum values of heat rate and heat load are overtaken, in spite of the strategy of using half of the limiting values as maximum allowed targets (see Section 7.1 and in particular Table 7.1, for a refreshment of these concepts).

The excess in aerodynamic heating can be attributed to four main sources:

- **Low Density Estimate at Pericenter**
  As is shown in Fig. 12.7d, the peak density per orbit is in general underestimated by a few percentage points. As was shown in Section 9.1, this arises due to the fitting of the noisy accelerometer data (converted to estimated density via Eq. (6.22)). A lower estimated density leads to a lower periapsis targeting, which in turn results in higher actual aerodynamic heating. This problem could be solved in three ways (possibly combining multiple together):
  - Giving a larger weight to lower altitudes (by using a weighted least squares as defined in Eq. (E.14b))
  - Further lowering the value of the reduced atmospheric interface altitude, which in Section 6.7 was set to 150 km
  - Applying a more sophisticated smoothing algorithm to the accelerometer data, instead of the simple one-minute moving average described in Section 6.7

- **Error in Initial Position**
  In the previous subsection it was described how the initial error in state knowledge when switching from aided to unaided navigation gives rise to an ever-growing offset between the estimated and actual state (during the unaided phase of navigation). This is also true during periapsis estimation, and as such, a similar strategy can be applied: the first five Keplerian elements could be replaced by the median of their respective one-minute history, and a linear or quadratic regression used to correct the value of true anomaly. In this case, since this adjustment is applied at apoapsis, the correction would lead to an even more accurate result, since the values of Keplerian elements are roughly constant at this height, especially for highly elliptical orbits.

- **Error in Propagation**
  Similar to the discussion above, this hypothesis is linked to the discussion made in the previous subsection when discussing the filtering process. Thus, the simplified environment model used onboard, combined with the low tolerances used for integration and the integration scheme itself, give rise to a propagation errors.

- **Root-Finder Convergence**
  Looking back to Table 7.3, one can see that the root-finder is considered to have converged to a solution whenever two consecutive guesses are within approximately ±100 m of each other (note that the limit on iterations has never been reached in the tested scenarios). This introduces another small yet noticeable error in the corridor values, which could be mitigated by reducing the convergence threshold.

By analyzing the results of those orbits where no maneuver was taken, the following results are obtained. In 80 % of orbits with no maneuver, the estimated periapsis altitude is overestimated by on average 0.6 %, which corresponds to roughly 1 km. This, combined with the lower estimated density (on average the peak density is underestimated by −2 %), leads to the conclusion, for an average main phase orbit (which is the aerobraking stage where most of the surpluses in heat rate and heat load occur), that the heat rate should be underestimated by −2.5 %.

One final consideration is that in case the apoapsis altitude falls between the range of unaided navigation altitudes, then the error in initial conditions is even larger than what is achieved with the GAN method, as detailed in Table 12.4b. In fact, the first apocenter to fall within this range, is in orbit 40. Clearly, one can see the large effect that this has on the corridor and maneuver estimator. In fact, most of the heat rate excesses are achieved while apocenter is in unaided navigation, i.e., until orbit number 102.

However, the violations in heat load in the range of orbits between 150 to 200, are not attained under these conditions. Clearly, there are other elements at play that lead to the excessive values of the aerodynamic heating. Investigation of these sources are recommended as further work for this research.

Velocity Increments Going back to the analysis of the ΔV usage, Table 12.5 shows the distribution of ΔV over each aerobraking phase. As it appears, the main phase is the one that requires the least amount of propellant (assuming the maneuvering burns negligibly affect the spacecraft mass). Despite having a rather large amount of maneuvers (especially in the first part, between orbit 8 and 40), the maneuvers performed in this phase, only raise or lower the periapsis by a few kilometers at most. In the walk-in phase, on the other hand, since the maneuvers lower the periapsis by a cumulative 23 km, the ΔV value reaches a slightly higher value.

However, even larger, is the value of the velocity increment consumed during walk-out. As explained earlier in this section, this is due to the sub-optimal guidance routine developed in this thesis.

Finally, the largest contributions come from the last phase, which uses more than half of the total ΔV. The need of such a large value arises from the fact that, during termination, the periapsis needs to be raised by more than 100 km, going from roughly 120 to (an expected) 255 km, and that the apocenter needs to be lowered from the last value reached, to its target.

Reduction of the this phase's ΔV could also be achieved. The main reason behind the large ΔV value is, in fact, because aerobraking is interrupted whenever the apoapsis falls below 25 % of the final target value. This has been set as a rather arbitrary threshold, at an early stage of the GNC system development. A reduction in this value, combined with a more robust walk-out phase, could give rise to more propellant savings.

Nonetheless, compared to the nominal ΔV needed to reduce the apoapsis for an interplanetary trajectory,

Table 12.5: *Distribution of velocity increments ΔV (expressed in [m s^{-1}]) over each aerobraking phase for both the simulated trajectory and for MRO (Long et al., 2008). The total values sums up to 82 and 43 m s^{-1}, respectively.*

	Walk-in	**Main**	**Walk-out**	**Termination**
Simulation	5.86	4.48	25.9	46.0
MRO	14.5	3.40	25.0	

which is in the range of 1 to 1.5 km s^{-1} (Long et al., 2008), the value described in Table 12.5 would already lead to noticeable cut in propellant mass, and thus to a lowering of the mission cost.

Finally, one can compare the obtained total ΔV estimated for this aerobraking maneuver, with the total used by MRO. As stated in Section 2.1, the total velocity change exhausted by MRO was about 44.1 m s^{-1}. Excluding the 1.2 m s^{-1} used for collision avoidance (which as mentioned in Subsection 4.3.2 is not modeled in this thesis), results in the breakdown shown in Table 12.5, on the second line.

The ΔV discrepancy during walk-in can be explained by comparing the drop in altitude achieved for the simulation developed in this thesis, and the drop attained by MRO: 23 vs. 320 km. The velocity change used during the main phase is rather comparable, denoting the fact that during this phase, no major maneuvers are undertaken by either guidance system. Finally, for walk-out and termination (which for MRO are reported as one) the reader is reminded once again of the sub-optimal guidance described by Section 7.1.

Propellant Mass Looking back to REQ-SC-3, the propellant mass for the maneuver was set to 50 kg. Using the inverse of the Tsiolkovsky equation (Curtis, 2013):

$$m_{\text{prop}} = m_{\text{dry}} \left[\exp \left(\frac{\Delta V}{I_{\text{sp}} g_0} \right) - 1 \right] \tag{12.3}$$

and assuming a specific impulse I_{sp} of 250 s (which represents an average conservative value for a chemical rocket (Ley et al., 2009)), leads to a total used propellant mass of 34 kg. Clearly, there is a large margin w.r.t. the requirement.

Of course, the process used to determine m_{prop} above, is very simplified. For starters, the simulation is run with constant mass, hence the ΔV computed is not representative of the actual ΔV that would be used. Furthermore, the ΔV is not ejected in one maneuver only, but it is spread over 210 orbits. Nonetheless, given the low value of the velocity change, these assumptions hold.

12.3 | Analysis of Requirements

In Section 2.3, a list of requirements for an aerobraking maneuver around Mars for a spacecraft resembling the aerodynamic and physical characteristics of MRO was laid out. Some of these requirements, were not fully constraint, but were left to be specified during the thesis work. In this section, this is what will be done, together with an analysis of whether the other requirements were met.

Requirements TBD The requirements that were marked as either to be *defined* or to be *determined* are shown in Table 12.6. Here, one can see that the requirements have been updated and are now given a numeric value.

The only requirements that were determined as part of this research are REQ-N-3, REQ-N-4 and REQ-C-4, and they all refer to the sampling rate of some of the GNC elements. As it has already been mentioned, the refresh rate of the navigation system depends on two things: the navigation method and the current spacecraft (actual) state. Thus, these requirements are variable, and depend on the conditions of Table 8.1.

The other requirements in Table 12.6 are defined based on the results shown earlier in this chapter. The first two requirements, i.e., the ones denoting the accuracy with which the apo- and periapses altitudes shall be reached, are worth some more considerations. The values shown in the table, are chosen based on the final orbit configuration obtained with the current algorithm. As repeatedly stated, the sub-optimality of the guidance algorithm could be improved to achieve both a lower propellant consumption and better final conditions.

Requirements to Be Met Now that the final requirements have been derived, the reader's attention is brought to the remaining requirements of Table 2.2.

The first requirements to be surveyed are the ones related to the aerothermodynamic characteristics, i.e., REQ-AB-4 and REQ-AB-5. As it has already been seen in Fig. 12.7, these requirements are not met. In fact, the values of heat rate and heat load are exceeded on multiple occasions. Related to these requirements, is REQ-G-1. Despite the inclusion of a simplified model for atmosphere perturbations (given by Eq. (4.30), Eq. (4.31), and Table 4.1), said requirement is not met, due to the excess values of heating.

Note that, technically speaking, REQ-AB-6 was not met on multiple orbits, as is visible from Fig. 12.7a. However, this was necessary 1) to respect the lifetime requirement (REQ-AB-7) and 2) to perform the walk-in maneuver. As such, this requirement is considered applicable only during the main phase.

Another requirement that is not met and is related to the thermal analysis, is REQ-G-4. Indeed, no ability of performing a contingency periapsis maneuver is present in the guidance developed in Section 7.1. Nevertheless, this would be a great addition to the system, and would definitely contribute to the overall safety of the maneuver.

Regarding the rotational motion and the control system, it was already mentioned that REQ-SC-8 was not formulated properly. In fact, it is inconsistent with the commanded attitude. A rectification would lead to the new

Table 12.6: *Finalization of the requirements on the aerobraking phase and on the guidance, navigation and control systems, for a spacecraft orbiting Mars.*

ID	Requirement
	Aerobraking
REQ-AB-2	The aerobraking phase shall reduce the apoapsis altitude to 320 ± 50 km.
REQ-AB-3	The periapsis post-aerobraking phase shall be 255 ± 5 km.
	Navigation
REQ-N-3	The state estimation shall run at a rate dictated by Table 8.1.
REQ-N-4	The inertial measurement unit shall have a sampling rate dictated by Table 8.1.
REQ-N-7	The position error during autonomous navigation shall be less than 50 m.
REQ-N-8	The translational velocity error during autonomous navigation shall be less than 0.1 $\mathrm{m\,s^{-1}}$.
REQ-N-9	The attitude error (expressed as error in quaternion) during autonomous navigation shall be less than 10^{-5}.
	Control
REQ-C-4	The star trackers shall have a sampling rate dictated by Table 8.1.
REQ-C-6	Each reaction wheel shall provide a maximum angular momentum of at least 47.5 N m s.

requirement: the aerodynamic coefficients shall be available for the rarefied and transition regimes, for an angle of attack range from −180 to +180 deg. Moreover, a new requirement would need to be added for the side-slip angle, requesting a range of −90 to +90 deg.

Note that if a requirement has not been named in the paragraphs above, then it can be considered met.

Page Intentionally Left Blank

13
Inertial Measurement for Aeroassisted Navigation

In this chapter, the results obtained with the Inertial Measurement for Aeroassisted Navigation, or IMAN, method will be analyzed. This method was introduced by Jah (2001), and further developed by the same author in more recent publications, such as Jah (2005), Jah and Lisano (2004), Jah et al. (2008). The reader will also find more details about this method, and how it is modeled in this thesis, in Chapter 6. The main purpose of the outlined algorithm, as the name suggests, is to use accelerometer measurements as the sole source of external data for the onboard filtering process. The internal measurement, in contrast, is provided by an onboard description of the simplified environment.

This application differs rather significantly from more conventional navigation softwares, which use the data provided by inertial measurement units as a replacement for the internal model of non-gravitational forces. With this approach, the only uncertainty in the onboard model arises from the difference in modeling of gravitational accelerations (arising from both the central and third bodies), and from the noise present in the accelerometer measurement. In the case of IMAN, instead, despite the adaptation of the onboard atmosphere model (thanks to the atmosphere estimator from Section 6.7) and the addition of C_D to the state vector (see Eq. (6.9)), the onboard model is still an approximation of the real environment and is especially susceptible to unexpected changes in the environment.

The chapter is structured in the following manner. Firstly, the reader will be introduced to the simulation settings used for this analysis. The navigation system was tested by increasing the discrepancies between onboard and simulated environment, a process which in turn increases the complexity of the problem. The various cases analyzed are detailed in Section 13.1. Note that the initial conditions and simulation settings are the same as described in the previous chapter, with the exception that a secondary set of (translational) initial conditions is also considered, to test the implementation of the IMAN method for lower eccentricities. Finally, Section 13.2 will show the results obtained and provide a discussion on the sensitivity to the changes in test case.

13.1 | Test Cases

The term 'test case' in this context is used to describe the various environment and spacecraft characteristics tested in this analysis. Each test case setting is assigned either a *simplified* or a *complete* value, such that the sensitivity of the system to this setting can be investigated.

The parameters that were modified are listed below, where the values for the simplified and for the more complete scenarios are both listed. Note that for the former, the values are equal for both the propagation and onboard environment, whereas for the latter, they differ. These discrepancies are also shown below:

- **Spherical harmonics** model of Mars:
  - Simplified: $(0, 0)$
  - Complete: *propagation*: $(21, 21)$; *onboard*: $(4, 4)$
- **Atmosphere** model of Mars:
  - Simplified: exponential atmosphere with parameters from Table C.2
  - Complete: *propagation*: tabulated atmosphere based on MCD from Appendix C.3 (without random perturbations); *onboard*: exponential atmosphere with parameters from Table C.2
- **Aerodynamics** model of the spacecraft:
  - Simplified: $C_D = 1.9$
  - Complete: *propagation*: full model from Chapter 11; *onboard*: $C_D = 1.9$

A total of eight simulations have been carried out, to achieve the full combination of simulations. Moreover, to confine the analysis to the characteristics above, the simplifications below have been made:

- **No randomized perturbations** in the atmosphere model (see Eq. (4.30))
- **No third body** acceleration from the Sun
- **No solar radiation** pressure from the Sun

It is worth noting, however, that the latter two simplifying steps listed above are also applied in the work by Jah et al. (2008). To simplify the presentation of the results, each simulation was assigned a number. Table 13.1 shows the full list of environment and spacecraft settings for each test case.

One may recall that in Section 6.2, it was mentioned that the IMAN method only becomes active below the DAIA. However, to make sure that the analysis only focuses on the performance of IMAN, rather than the combination of the aided and unaided methods, the latter method is not used, leading to a full IMAN implementation.

Another parameter that is varied during the simulations, is the refresh rate of the onboard computer. This rate is defined as the rate at which the navigation system runs the filtering process, thus corresponding also to the sampling rate of the instrumentation (in particular of the IMU). In Jah et al. (2008), the refresh rate was set to 20 Hz, but for sake of completeness, it was decided to test two more sampling frequencies: 200 and 2 Hz. The way this was simulated was by running the simulation with a time step of 5×10^{-3} s, corresponding to a frequency of 2000 Hz, and running the onboard computer every 10, 100 or 1000 time steps.

Finally, as mentioned in the introduction of this chapter, the results corresponding to two initial conditions are examined. The first initial condition is technically the same as shown in Section 12.1, with the exception that the apoapsis maneuver is already applied, and the periapsis scaling from Eq. (7.1) (which is active during walk-in) is turned off, such that the atmospheric sweep is deeper and IMAN can be more effective.

Following from the discussion above, the adjusted initial conditions, i.e., the conditions after the apoapsis maneuver, are listed in Table 13.2. Here, one can also find the second case that is being analyzed. These Keplerian elements represent an orbit with lower eccentricity and average altitude, such that the efficacy of the IMAN algorithm can also be probed in these conditions (which roughly correspond the orbit configuration of a spacecraft approaching the end of an aerobraking maneuver).

One important fact to mention, is that it soon became clear that the trajectory corresponding to the second conditions of Table 13.2, produced too large and prolonged (time-wise) aerodynamic accelerations, for the onboard navigation refresh rate of 2 Hz to be a viable option. Hence, for this part of the analysis, the decision was made to restrict the analysis to the refresh rates of 200 and 20 Hz.

Last, but not least, the following values for the $\mathbf{Q}$ and $\mathbf{R}$ matrices for the filtering processes were used, which are

Table 13.1: *List of test cases used to analyze the performance of the IMAN algorithm, based on the environment and spacecraft characteristics. A dot, i.e., ●, signifies an active model.*

Settings		**Test Case**							
		1	2	3	4	5	6	7	8
Spherical Harmonics	Simplified	●	●	●	●				
	Complete					●	●	●	●
Atmosphere	Simplified	●	●			●	●		
	Complete			●	●			●	●
Aerodynamics	Simplified	●		●		●		●	
	Complete		●		●		●		●

Table 13.2: *Initial conditions for the analysis of the IMAN algorithm performance. These conditions correspond to a highly elliptical orbit with a periapsis altitude of approximately 110 km, and a less elliptical and lower altitude orbit, similar to what a spacecraft would undergo during the last days of aerobraking.*

Initial Conditions	a [m]	e [–]	i [deg]	ω [deg]	Ω [deg]	ϑ [deg]	Δt [h]	t
1	25 946 932.3	0.865 191 2	93.0	43.6	158.7	180.0	33.6	June 30, 2007 at 12:00 Noon
2	4 710 917.5	0.251 560 4					2.7	

of the same order of magnitude as the ones used by Jah (2005), and are meant to optimize the RMS position error value of test case 8 at 20 Hz:

$$\mathbf{Q} = \mathrm{diag}\left(10^2, 10^2, 10^2, 10^{-1}, 10^{-1}, 10^{-1}, 10^{-4}, 10^{-4}, 10^{-4}, 10^4\right)^2$$
$$\mathbf{R} = \mathrm{diag}\left(10^2, 10^2, 10^2\right)^2$$

13.2 | Results

The results of the IMAN analysis are show in Table 13.3 for the first initial conditions, and in Table 13.4, for the second ones. In these tables, the values of the RMS position error achieved after one orbit is shown for each test case. Table 13.5, on the other hand, displays the errors attained by Jah et al. (2008) for a simulation corresponding to test case 8 at 20 Hz.

In this section, firstly, the results obtained for the one orbit simulation will be analyzed, then Subsection 13.2.2 will extend the simulation time to two orbits, to see how the error grows over time. Finally, Subsection 13.2.3 will collect the considerations produced in the previous subsections, and discuss on the possible realizations of IMAN.

13.2.1 | Single Orbit

From both Tables 13.3 and 13.4 it is clear that by making the environment and system models more complete, the position error increases. This is inline with what is expected, since the larger the difference between the onboard and propagation environments, the less effective the filtering process.

One notable exception is test case 8 at 20 Hz, of Table 13.3. Here, one can see that the RMS error is very small, and undergoes a large decrease in magnitude from the previous scenarios. This is because, as mentioned at the end of Section 13.1, the values of the process and measurement noise matrices, $\mathbf{Q}$ and $\mathbf{R}$, have been optimized for this scenario.

A similar, but much less drastic, decrease in RMS is also noticed for test case 8 at 20 Hz of Table 13.4. Once again, the error is lower for this simulation than for some of the previous ones. In general, however, the error for these initial conditions are small in magnitude, regardless of the tested case.

Comparing the results of Table 13.3 to the ones shown in Table 13.5, it is clear that for a highly elliptic orbit, the error agrees very closely to the one published in Jah et al. (2008). For the lower eccentricity orbit, the errors are rather different. In the reference publication, the errors are larger than what is obtained in this work.

The reason behind such discrepancy likely has to do with the difference in data used in the two research projects. In Jah et al. (2008), the IMU accelerations are taken from real missions. In particular, the results of Table 13.5 are obtained by using accelerometer data from the MOY mission. Thus, these contain real-life effects, such as atmosphere variability and unpredictability, which add uncertainty to the estimation process. In fact, since in the highly elliptical

Table 13.3: *Values of the RMS position error (expressed in [m]) for each combination of simulation environment and refresh rate, achieved after one orbit, with the first set of initial conditions from Table 13.2.*

Onboard Refresh Rate	**Test Case**							
[Hz]	1	2	3	4	5	6	7	8
200	60.24	20.45	109.9	192.1	695.8	765.0	826.9	898.7
20	4.147	16.19	127.5	105.2	695.4	609.6	99.63	34.19
2	476.1	50.45	2065.0	2526.0	785.4	406.3	5018.0	6113.0

Table 13.4: *Values of the RMS position error (expressed in [m]) for each combination of simulation environment and refresh rate, achieved after one orbit, with the second set of initial conditions from Table 13.2.*

Onboard Refresh Rate	**Test Case**							
[Hz]	1	2	3	4	5	6	7	8
200	0.703	14.14	5.772	18.72	57.71	68.77	58.65	69.96
20	10.17	22.09	9.130	20.74	59.49	71.77	59.88	70.91

Table 13.5: *Values of the RMS position error (expressed in [m]), achieved after one orbit, for a large, medium and small orbit configurations (Jah et al., 2008).*

Large Orbit $e \approx 0.774$	**Medium Orbit** $e \approx 0.563$	**Small Orbit** $e \approx 0.054$
30.00	167.0	441.0

orbit condition the atmospheric sweep is relatively short (about 15 min), the error accumulated is low. But as soon as the spacecraft starts spending more time in denser layers of the atmosphere, the error starts growing. Of course, this growth is slowed down by the estimation of the drag coefficient, which as stated in Subsection 6.3.2, incorporates part of the atmosphere modeling and other navigation errors.

Another possible source of difference is the non-optimality of values of the process and measurement noise matrices. As is clear from Table 13.3 there is a large sensitivity of the RMS errors on the noise parameters used.

Nonetheless, the potential of the IMAN navigation method is clear. Despite its simplicity, it manages to reproduce the actual trajectory flown with high accuracy and also provides a way of reconstructing the density value, by exploiting the difference between the estimated drag coefficient, i.e., $\mathbf{x}^{C_D}$, and the actual (known) drag coefficient, which is approximately 1.9 at pericenter altitudes. Before drawing full conclusions on this method, however, its performance over multiple orbits needs to be analyzed. This is done, for two orbits, in the next section.

13.2.2 | Two Orbits

In the previous section, it was shown how the IMAN method implemented in this thesis manages to reproduce the error accuracy of the reference work. This section, however, extends the simulation time to two orbits, to analyze the performance of this navigation system over longer time spans.

Table 13.6 shows the values of RMS position error for a propagation lasting two orbits, for both initial conditions, and a refresh rate of 20 Hz.

Table 13.6: *Values of the RMS position error (expressed in [m]), achieved after two orbits, with both sets of initial conditions from Table 13.2, and with a refresh rate of 20 Hz.*

Conditions 1	**Conditions 2**
7727.0	55.65

In the table above, one can see that the error behaves rather differently in the two conditions. For the highly elliptical orbit case, the error shows a staggering 200 folds increase, whereas in the second case, the error decreases. It would appear that in the latter example, the navigation filter manages to reduce the error by adapting the drag coefficient estimate. However, when looking at the former instance, two main concerns arise, related to this navigation system:

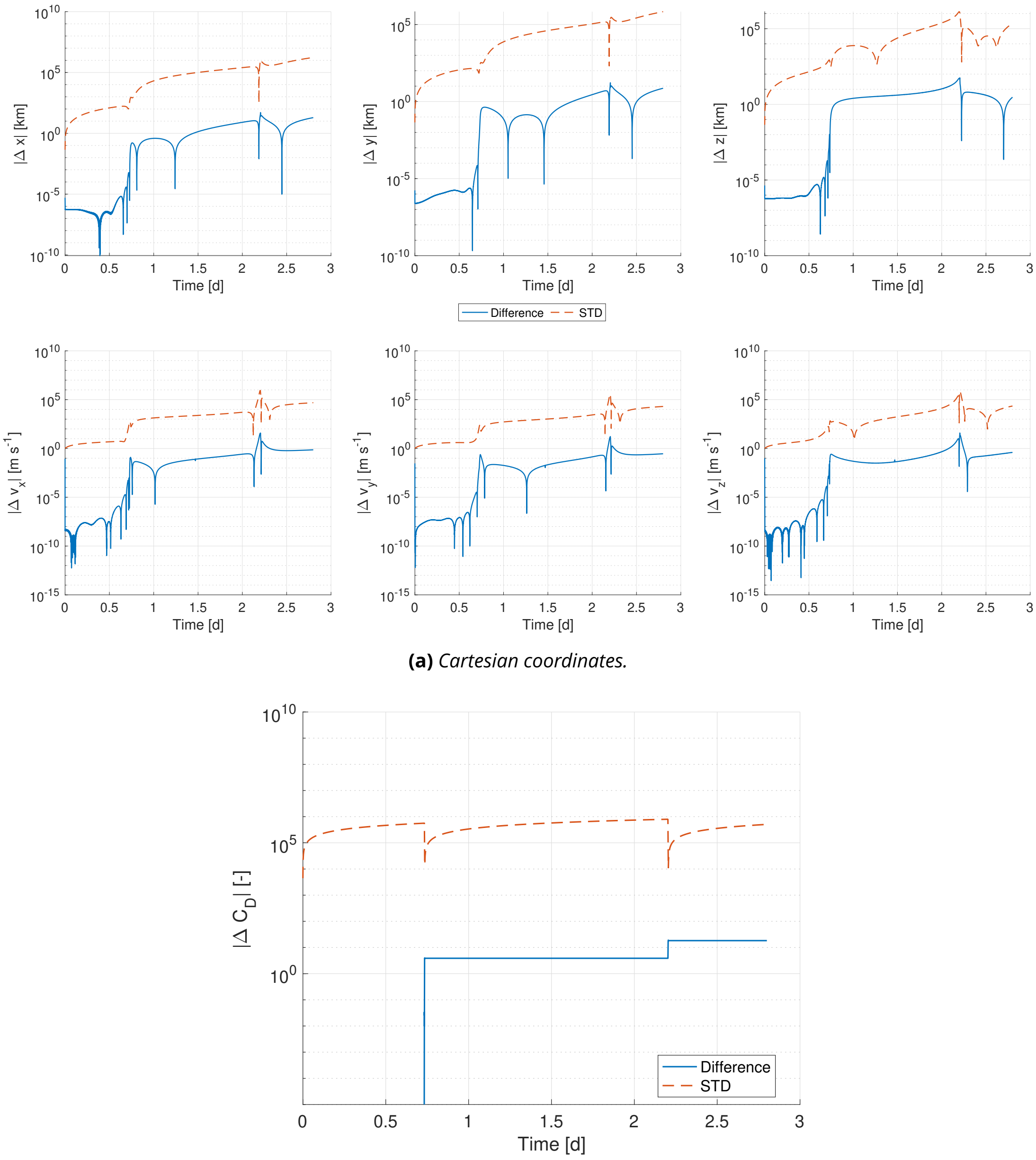
- seemingly unbound growth of the drag coefficient estimate
- large standard deviations of the state (given by the square root of the diagonal of the covariance matrix $\mathbf{P}$)

The growth of the drag coefficient is especially alarming. In fact, as shown in Fig. 13.1b, the drag coefficient reaches a value of almost 20, which causes a large error in the estimation of the aerodynamic drag force, which in turn leads to an underestimation of the first two Keplerian elements, a and e. Such error can also be seen in Fig. 13.1a, and in the RMS value of Table 13.6.

In the same figure, one can also see the values of the standard deviation of the estimates. Clearly, the confidence of the estimate is low, given that the value of STD for position approaches 10^8 and shows no sign of decreasing. The same can be said for the values of σ for velocity.

13.2.3 | Considerations

Despite the very promising results achieved after one orbit, the usage of IMAN for large orbits (i.e., orbits with high eccentricity), leads to a quick divergence in state knowledge. This is of course to be expected, since two orbits for a highly elliptical orbit correspond to almost three days of propagation, whereas the same number of revolution last less than six hours, for an orbit similar to the one denoted by the second initial conditions of Table 13.2.



(a) *Cartesian coordinates.*

(b) *Drag coefficient.*

Figure 13.1: *Absolute value of difference w.r.t. estimated and actual values of the state elements of the navigation filter, with IMAN as navigation method, for two orbits and with the first set of initial conditions from Table 13.2.*

It then follows that, if IMAN were chosen as the preferred navigation method, one would need to update the onboard state estimation with external measurements (such as the ones provided by the DSN of Section 6.6), with a frequency of less than three days (the actual number depending on the required accuracy during the aerobraking phase). This recurrence is inline with REQ-N-1, which sets the threshold at a minimum of one day.

Trying to keep the navigation system as autonomous as possible, instead of using external measurements, one could implement a more robust version of the periapse time estimator of Section 6.5. In this manner, after each atmospheric sweep, the estimated Keplerian state could be corrected based on the measured aerodynamic acceleration.

Page Intentionally Left Blank

14 Conclusions and Recommendations

This thesis saw the introduction of a guidance, navigation and control (GNC) system, specifically designed to perform an aerobraking maneuver. In particular, the focus was placed on a mission centered at Mars, and performed by a spacecraft, whose design and model was based on the Mars Reconnaissance Orbiter (MRO). In this chapter the main conclusions drawn in Chapters 12 and 13 will be summarized and a series of recommendations will be given, based on the discussions made in said chapters.

Onboard Software

First, it is worth spending a few lines in detailing the GNC system functions developed:

- **Navigation System**
  The purpose of this part of the software is to determine the current spacecraft state (position, velocity and possibly attitude). Two navigation systems have been implemented:
  - The first system is called inertial measurements for aeroassisted navigation (IMAN), and was detailed in the work by Jah (2005). This method uses accelerometer measurements as external observations for a Kalman filter, whereas the internal data is provided by an onboard model. This model was limited to the analysis of translational motion.
  - The second system, on the other hand, called GRS-assisted navigation (GAN), is based on the use of a generic ranging system (GRS) as external measurements, whereas the internal observations come from a combination of inertial measurement unit (IMU), two star trackers and an onboard gravitational model.

  These two systems were complemented by a few other elements, whose functions are explained below:
  - *Unaided Navigation*
    The navigation systems above correspond to aided navigation, i.e., navigation using sensors to improve the onboard estimation process. However, another navigation method is introduced to reduce the computational strain and to increase the fidelity of the simulation. This is called unaided navigation, and consists of direct integration of the accelerations estimated by the onboard model.
  - *Periapse time estimator* (PTE): in case IMAN is used as navigation method, this element analyzes the accelerometer measurements to estimate the periapsis time and the changes in velocity, semi-major axis and eccentricity, due to atmospheric drag, to improve the spacecraft translational state estimate
  - *Deep space network* (DSN): once again, this element is only active in case of IMAN navigation, since this method is only based on onboard sensor fusion; processing of the DSN tracking data is performed to update the spacecraft translational state, every few days
  - *Atmosphere estimator* (AE): this element is run regardless of the navigation method and it has the purpose of determining the parameters of the onboard atmosphere model, by fitting the density values derived from the accelerometer data
- **Guidance System**
  The guidance system that was developed has two main task: to estimate the periapsis corridor boundaries and to target the midpoint of said corridor. The first function is achieved by considering the thermodynamic limitations of the spacecraft and the restrictions on aerobraking duration, and by splitting the aerobraking maneuver into five different phases:
  - *Walk-in*: the pericenter is slowly lowered into the highest layers of the atmosphere
  - *Main*: most of the braking effect is achieved
  - *Walk-out*: the periapsis is slowly lifted outside the atmosphere
  - *Termination*: the peri- and apocenters are matched to their target values
  - *Aerobraking complete*: the achieved orbit configuration is compared to said targets

To attain the second purpose, in contrast, use is made of a maneuver estimator. As the name suggests, this element estimates the magnitude and direction of the apo- and periapsis maneuvers, where the latter is only carried out once, during termination, and the former is carried out whenever the estimated periapsis altitude falls outside the intended corridor.

○ **Control System**
Two control systems are employed onboard the spacecraft: one for attitude and the other for orbit control. A proportional, integral and derivative (PID) controller is used to regulate the rotational motion of the spacecraft, by correcting for the errors in current attitude. It should be stated that this element is only modeled to get an idea of the spacecraft response and of its controllability. Conversely, orbit control is realized by assuming an impulsive shot, and is modeled by simply adding or subtracting the velocity change estimated by the maneuver estimator to and from the spacecraft velocity.

Performance of the control system was briefly analyzed for the first orbit. It became clear that a PID controller, combined with the hardware present on MRO, is enough to control the spacecraft attitude, but that a more sophisticated algorithm may have to be designed to constrain the oscillations in aerodynamic angles (which as of the current implementation, show variations from the commanded state of about ±10 deg, after the atmospheric phase) and to reduce the required reaction wheel momentum.

Research Questions

Now that the reader has an overview of the work that has been carried out for this thesis, the research question reported in Chapter 1 can be analyzed, to see if and how it has been answered. The main primary question was:

Can the mass of an interplanetary spacecraft be decreased by using a robust guidance, navigation and control system for aerobraking, without jeopardizing the safety of the spacecraft and maintaining a low navigation error?

and was partly answered in Chapter 12. The use of the adverb ‘partly’ here, is to highlight the fact that indeed, the guidance, navigation and control system developed in this thesis 1) is able to reduce the mass of an interplanetary spacecraft, by requiring the employment of 82 instead of more than 1000 $\mathrm{m\,s^{-1}}$ of velocity increments, and 2) it maintains a low navigation error (for most of the orbit the position error is within 50 m), but, on the other hand, it is not capable of meeting the conditions set by the adjective ‘robust’. In particular, the excessive heating experienced during the atmospheric phases of the mission, poses a large threat to the thermal and overall integrity of the spacecraft, possibly jeopardizing the whole mission.

The lack of robustness will be explained below, under the heading of ‘Guidance’, however, the focus is now brought to the sub-questions for this research.

Constraints on Braking Effect To answer the first question about the constraints on the braking effect, use was made of an extensive mission heritage study. The major constraints encountered by previous missions, were combined with strategies that are currently being developed for future missions, to create the GNC system described in Part II. The main constraint is dictated by the temperature of the spacecraft, which is highly affected by the dissipation of orbital energy into heat, vibrations and noise, during the atmospheric phase of the orbit.

Autonomy of Navigation The second question, namely how long the spacecraft can navigate autonomously, has not been fully answered. As mentioned in the beginning of this chapter, two main navigation methods were made use of. The first method (IMAN) is solely based on the use of accelerometer data to estimate the spacecraft position, via a simple Kalman filtering process. The second method (GAN), also applies filtering techniques to estimate the spacecraft state, but in contrast, uses position measurements as external observations.

The latter navigation system is completely autonomous, in the sense that no information or instructions from Earth need to be received (at least, these are not needed for navigation), thus making this research question inapplicable. For the former method, on the other hand, ground updates are indeed necessary. As was discovered in Chapter 13, these updates should occur with a frequency of at most one every three days, before the position error grows too large. Discussion of more reflections on this method will follow later, under the heading of ‘Navigation’.

Deviations in Initial Conditions The question regarding the sensitivity to deviations from the initial conditions, has a similar answer to what was given above. For the GAN navigation method, deviations in the initial conditions are handled as expected: the filtering algorithm converges to the actual state within minutes of the start of the process.

In case of IMAN, by contrast, the sensitivity of deviations in initial conditions has not been studied. For this navigation method, such an activity would, however, be very insightful, especially since the initial conditions in the current and previous studies, are never perturbed. However, this decision has been made for a very specific reason: IMAN is not an absolute navigation system, and as such it requires the use of quasi-perfect initial state knowledge, such as it is provided by ground station tracking. Nevertheless, even the most precise DSN observations carry with them some uncertainty, and this could be at the basis of the sensitivity analysis.

Uncertainty in Aerodynamic and Atmosphere Modeling The spacecraft aerodynamic coefficients were studied in Chapter 11, where it was shown that the value of drag coefficient (C_D) is in the range of 1.9 at the average periapsis altitude, and for an angle of attack range within ±10 deg, thus showing a rather stable behavior in the most critical phases of the maneuver. In fact, this was the value of C_D used for the onboard estimation. However, no study on the variation of this parameter has been made. Moreover, it was concluded in Chapter 12 that the angles used for the aerodynamic analysis of Chapter 11 consisted of a too limited range.

Regarding the atmosphere modeling, it was explained how the unpredictability and uncertainty in the density is increased by introducing a multiplication factor, which is adapted every orbit, by means of a random walk.

Two more research questions were posed to help guide the thesis development, namely which state variables are best suited for the propagation of the spacecraft state and which sensor configuration provides the optimal navigation strategy.

The question about state variable performance was answered in Chapter 10, where Cartesian coordinates were compared to the very promising unified state model (USM) elements. Despite proving that indeed, in the test cases analyzed by Vittaldev et al. (2012), the USM variables provide a more accurate and faster (for variable step size propagation) solution, as soon as the dynamics of the system and the propagation time are increased, the advantage of USM tends to disappear (even more so, if both characteristics are added simultaneously). Thus, for propagation of the main aerobraking simulation, Cartesian coordinates with constant step size, are selected as propagation and integration settings. For onboard propagation, on the other hand, since both the propagation duration and the number of accelerations modeled (i.e, the system dynamics) are reduced, one of the USM models with variable step size, was chosen.

Finally, the focus can be brought to the remaining question, concerning the optimal sensor configuration. By comparing the two navigation methods analyzed during the thesis work, it is clear that the GAN method provides a smaller navigation error, w.r.t. IMAN. However, the feasibility of such method is debatable. At the time of writing, no existing navigation method would be able to provide the information required by GAN (i.e., the value of position of the spacecraft) at the modeled frequency and accuracy. Therefore, this method cannot be considered optimal. Looking now at the performance of IMAN, it is true that the root mean squared error in position quickly grows after the first orbit, however, inclusion of measurements of the DSN and of the corrective action of the PTE, would significantly increase its capabilities. This justifies the enthusiasm of the author of Jah (2005), who stated that IMAN could potentially be used in future aerobraking missions, to make the GNC routines more (if not fully) autonomous[1].

Nonetheless, before claiming the optimal conditions of IMAN, further testing is required. The reader is referred to the next paragraphs for a discussion on the recommended steps to take.

Navigation

In multiple occasions throughout this chapter, it was explained how two main navigation methods have been analyzed. Recommendations for further development of these systems are found below.

The main assumption behind GRS-assisted navigation, is the availability of position measurements. It was also mentioned earlier, that no real fully functioning system is available to meet the modeled accuracy and frequency of observations. Thus, one of the first steps necessary to continue the development of an aerobraking GNC, would be to finalize this part of the network.

Concerning the development of IMAN, further testing over multiple orbits and over multiple eccentricity values are needed. In particular, focus should be placed on the tuning of the noise parameters of the filtering methods, which as was shown in the results, have a very large influence on the error in estimated position. Moreover, said tuning may even have to be adapted in-orbit, as the shape and thus, the orbital conditions, evolve with time.

Furthermore, the combination of IMAN with the orbit corrections provided by the PTE need to be investigated. The PTE did seem, in most cases, to yield accurate estimates of the effects of drag on the shape-related Keplerian elements; nevertheless, their integration has not been analyzed enough to lead to definitive conclusions, despite the full development of the necessary software.

Finally, since the IMAN analysis was only carried out with an extended Kalman filter, use of more filtering techniques, such as the unscented Kalman filter, could provide more accurate (but possibly more computationally expensive) results. Another insightful addition to the testing regimes, would be the analysis of the sensitivity to initial conditions, as was already stated.

[1] Dr. Jah, M.K., personal communication – October 17, 2018

Guidance

The main takeaways from Chapter 12 regarding the guidance system developed, can be summarized with the following two statements:

- *The aerobraking maneuver reduces by more than a factor of ten the required velocity change needed to circularize the orbit of a Mars-bound spacecraft.*
  As mentioned at the beginning of this chapter, 82 m s^{-1} is the velocity change employed by the maneuver analyzed. However, it was also stated that this value could be lowered by improving the robustness of the guidance design. In particular, from the discussion in Chapter 12, it appeared that the highest reduction in propellant used could be achieved in two manners:
  - by enhancing the walk-out routine: as of the current version, the walk-out phase activates the use of an extra constraint while estimating the lower corridor bound; however, such restriction seems to cause a behavior where the periapsis corridor keeps being raised and lowered in a consecutive maneuvers, leading to a wasteful usage of propellant
  - by setting a lower threshold for the triggering of the termination phase: as soon as the apoapsis reaches a value below 1.25 times the target value, aerobraking is halted and the target apo- and pericenter are aimed; lowering this value would lead to less expensive maneuvers
- *The guidance algorithm requires more accurate translational state estimates to respect the target aerothermodynamic conditions set for the periapsis corridor.*
  It was also shown in Chapter 12, that the error in position estimates during the unaided phase reach values in the order of kilometers. These lead to an inaccurate estimation of the corridor bounds and of the maneuver, which in turn cause excessive values of the heating parameters during the atmospheric sweep. Moreover, it should be stated that during the end of the main phase, the thermodynamic limits were also surpassed, but further investigation in why this behavior arises in necessary.

A recommended step that follows from the last point above, is to undertake the analysis of the sensitivity of the corridor and maneuver estimators to uncertainties in the navigation conditions. As it became clear from, errors in the values of position, have negative consequences on the performance of the guidance software. Hence, a sensitivity analysis would lead to the definition of requirements on the maximum allowed uncertainty in the spacecraft state.

Bibliography

Anderson Jr., J., *Hypersonic and High-temperature Gas Dynamics*, 2nd ed., ser. Education Series. American Institute of Aeronautics and Astronautics, 2006.

——, *Introduction to Flight*, 6th ed., ser. Aeronautical and Aerospace Engineering. McGraw-Hill, 2008.

——, *Fundamentals of Aerodynamics*, 5th ed., ser. Aeronautical and Aerospace Engineering. McGraw-Hill, 2011.

Andrews, L., *Special Functions of Mathematics for Engineers*. SPIE Optical Engineering Press and Oxford University Press, 1992.

Berner, J., Bryant, S., and Kinman, P., "Range measurement as practiced in the deep space network," *Proceedings of the IEEE*, vol. 95, no. 11, pp. 2202–2214, November 2007.

Bird, G., *The DSMC Method*. CreateSpace Independent Publishing Platform, 2013.

——, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. Oxford University Press, 1994.

Blanchard, R. and Buck, G., "Rarefied-flow Aerodynamics and Thermosphere Structure from Shuttle Flight Measurements," *Journal of Spacecraft and Rockets*, vol. 23, no. 1, pp. 18–24, 1986.

Breck, D., *Zeolite Molecular Sieves: Structure, Chemistry and Use*. John Wiley & Sons, Inc., 1974.

Carnevale, E., Carey, C., and Larson, G., "Ultrasonic Determination of Rotational Collision Numbers and Vibrational Relaxation Times of Polyatomic Gases at High Temperatures," *The Journal of Chemical Physics*, vol. 47, no. 8, pp. 2829–2835, October 1967.

Chabot, T., Bonnéry, G., Val Serra, S., and Kemble, S., "Robust Autonomous Aerobraking Strategies," in *5th European Conference for Aeronautics and Space Sciences (EUCASS)*, Munich, Germany, July 2013.

Cianciolo, A., Maddock, R., Prince, J., Bowes, A., Powell, R., White, J., Tolson, R., O'Shaughnessy, D., and Carrelli, D., "Autonomous Aerobraking Development Software: Phase 2 Summary," in *AAS/AIAA Astrodynamics Specialist Conference*, no. 736, Hilton Head, South Carolina, United States, August 2013.

Cichocki, F., Sánchez, M., Clerc, S., and Voirin, T., "Aerobraking Pericentre Control Strategies," in *8th International Planetary Probe Workshop (IPPW)*, Portsmouth, Virginia, United States, June 2011.

Cichocki, F., Sánchez, M., Bakouche, C., Clerc, S., and Voirin, T., "Robust Autonomous Aerobraking Strategies," in *9th International Planetary Probe Workshop (IPPW)*, Toulouse, France, June 2012.

Curtis, H., *Orbital Mechanics for Engineering Students*. Butterworth-Heinemann, 2013.

Damiani, S., Garcia, J., Guilanyà, R., Muñoz, P., and Müller, M., "Flight Dynamics Operations for Venus Express Aerobraking Campaign: A Successful End of Life Experiment," *GMV GmbH at ESOC*, 2015.

Dirkx, D. and Mooij, E., *Conceptual Shape Optimization of Entry Vehicles*. Springer, 2017.

Doody, D., "Aerobraking the Magellan Spacecraft in Venus Orbit," *Acta Astronautica*, vol. 35, pp. 475–480, 1995.

Dwyer, A., Tolson, R., Munk, M., and Tartabini, R., "Development of a Monte Carlo Mars-GRAM Model for Mars 2001 Aerobraking Simulations," *Advances in the Astronautical Sciences*, vol. 109, pp. 1293–1308, 2001.

Forget, F., Hourdin, F., Fournier, R., Hourdin, C., Talagrand, O., Collins, M., Lewis, S., Read, P., and Huot, J.-P., "Improved General Circulation Models of the Martian Atmosphere from the Surface to Above 80 km," *Journal of Geophysical Research: Planets*, vol. 104, no. E10, pp. 24 155–24 175, 1999.

Forget, F., Millour, E., and Lewis, S., "Mars Climate Database v5.2 Detailed Design Document," European Space Research and Technology Center, Noordwijk, Netherlands, Tech. Rep., February 2015.

Fukushima, T., "Simple, Regular, and Efficient Numerical Integration of Rotational Motion," *The Astronomical Journal*, vol. 135, no. 6, pp. 2298–2322, June 2008.

Gabeaud, A., "Base Pressures at Supersonic Velocities," *Journal of the Aeronautical Sciences*, vol. 17, no. 9, pp. 525–526, 1950.

Gallais, P., *Atmospheric Re-Entry Vehicle Mechanics*. Springer, 2007.

Gallis, M., Hermina, W., Johnson, M., and Chapel, J., "Aerothermal Analysis for the Mars Reconnaissance Orbiter," in *37th AIAA Thermophysics Conference*, Portland, Oregon, United States, June 2004, pp. 2684–2695.

Gallis, M. and Harvey, J., "The Modeling of Chemical Reactions and Thermochemical Nonequilibrium in Particle Simulation Computations," *Physics of Fluids*, vol. 10, no. 6, pp. 1344–1358, 1998.

Gallis, M., Torczynski, J., Plimpton, S., Rader, D., and Koehler, T., "Direct Simulation Monte Carlo: The Quest for Speed," in *AIP Conference Proceedings*, vol. 1628, no. 1. Xi'an, China: American Institute of Physics, December 2014, pp. 27–36.

Gerald, C. and Wheatley, P., *Applied Numerical Analysis*, 7th ed. Pearson Education, 2004.

Giorgini, J., Wong, S., You, T., Chadbourne, P., and Lim, L., "Magellan Aerobrake Navigation," *Journal of the British Interplanetary Society*, vol. 48, no. 3, pp. 111–122, 1995.

González-Galindo, F., Forget, F., López-Valverde, M., Angelats i Coll, M., and Millour, E., "A Ground-to-exosphere Martian General Circulation Model: 1. Seasonal, Diurnal, and Solar Cycle Variation of Thermospheric Temperatures," *Journal of Geophysical Research*, vol. 114, no. E04001, 2009.

Groves, P., *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, 2nd ed., ser. GNSS Technology and Applications Series. Artech House, 2013.

Hankey, W., *Re-entry Aerodynamics*, ser. Education Series. American Institute of Aeronautics and Astronautics, 1988.

Hess, J., "Aerogravity Assists," Master Thesis, Delft University of Technology, February 2016.

Hill, K. and Born, G., "Autonomous interplanetary orbit determination using satellite-to-satellite tracking," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 3, pp. 679–686, 2007.

Iess, L., Di Benedetto, M., Marabucci, M., and Racioppa, P., "Improved Doppler Tracking Systems for Deep Space Navigation," in *23rd International Symposium on Space Flight Dynamics*, 2012.

Jah, M., "A Proposed Use of Accelerometer Data for Autonomous Aerobraking at Mars," in *AIAA/AAS Astrodynamics Specialist Conference*, vol. 01, no. 386. Quebec City, Canada: American Astronautical Society, July 2001.

——, "Mars Aerobraking Spacecraft State Estimation by Processing Inertial Measurement Unit Data," Ph.D. dissertation, University of Colorado, 2005.

Jah, M. and Lisano, M., "6-DOF Aerobraking Trajectory Reconstruction by Use of Inertial Measurement Unit (IMU) Data for the Improvement of Aerobraking Navigation," 2004.

Jah, M., Lisano, M., Born, G., and Axelrad, P., "Mars Aerobraking Spacecraft State Estimation By Processing Inertial Measurement Unit Data," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 6, pp. 1802–1812, November–December 2008.

Johnson, M. and Willcockson, W., "Mars Odyssey Aerobraking: The First Step Towards Autonomous Aerobraking Operations," in *Aerospace Conference*, vol. 8. IEEE, 2003, pp. 3503–3510.

Justh, H., Justus, C., and Keller, V., "Global Reference Atmospheric Models, Including Thermospheres, for Mars, Venus and Earth," in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Keystone, Colorado, United States, August 2006.

Kiusalaas, J., *Numerical Methods in Engineering with MATLAB*. Cambridge University Press, 2005.

Klees, R. and Dwight, R., *Applied Numerical Analysis*, Delft University of Technology, Faculty of Aerospace, Astrodynamics and Satellite Missions, November 2014, course: Applied Numerical Analysis - AE2220-I.

Klothakis, A. and Nikolos, I., “Modeling of Rarefied Hypersonic Flows Using the Massively Parallel DSMC Kernel “SPARTA”,” in *8th GRACM International Congress on Computational Mechanics*, Volos, Greece, July 2015.

Lee, S., Skulsky, E., Cwynar, J. C. D., Gehling, R., and Delamere, A., “Mars Reconnaissance Orbiter Design Approach for High-Resolution Surface Imaging,” in *26th Annual AAS Guidance and Control Conference*, vol. 3, no. 67. Breckenridge, Colorado, United States: American Astronautical Society, February 2003, pp. 1–18.

Ley, W., Wittmann, K., and Hallmann, W., Eds., *Handbook of Space Technology*. John Wiley & Sons, 2009.

Liebe, C., “Star Trackers for Attitude Determination,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 10, no. 6, pp. 10–16, June 1995.

Liechty, D., “Aeroheating Analysis for the Mars Reconnaissance Orbiter with Comparison to Flight Data,” *Journal of Spacecraft and Rockets*, vol. 44, no. 6, pp. 1226–1231, 2007.

Lissauer, J. and De Pater, I., *Fundamental Planetary Science: Physics, Chemistry and Habitability*. Cambridge University Press, 2013.

Long, S., Lyons, D., Guinn, J., and Lock, R., “ExoMars/TGO Science Orbit Design,” in *AIAA/AAS Astrodynamics Specialist Conference*, no. 4881. Minneapolis, Minnesota, United States: American Institute of Aeronautics and Astronautics, August 2012.

Long, S., You, T., Halsell, C., Bhat, R., Demcak, S., Graat, E., Higa, E., Highsmith, D., Mottinger, N., and Jah, M., “Mars Reconnaissance Orbiter Aerobraking Navigation Operation,” in *SpaceOps 2008 Conference*, no. 3349. Heidelberg, Germany: American Institute of Aeronautics and Astronautics, 2008.

Lyons, D., “Aerobraking Automation Options,” in *AIAA/AAS Astrodynamics Meeting*, no. 1324, Quebec City, Quebec, Canada, July 2001.

——, “Mars Reconnaissance Orbiter: Aerobraking Reference Trajectory,” in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, no. 4821. Monterey, California, United States: American Institute of Aeronautics and Astronautics, August 2002.

——, “Aerobraking Challenges for a Combined Mars Orbiter and Lander in 2018,” in *AIAA/AAS Astrodynamics Specialist Conference*, no. 5068. Minneapolis, Minnesota, United States: American Institute of Aeronautics and Astronautics, August 2012.

Lyons, D., Saunders, R., and Griffith, D., “The Magellan Venus Mapping Mission: Aerobraking Operations,” *Acta Astronautica*, vol. 35, no. 9-11, pp. 669–676, 1995.

Lyons, D., Beerer, J., Esposito, P., Johnston, M., and Willcockson, W., “Mars Global Surveyor: Aerobraking Mission Overview,” *Journal of Spacecraft and Rockets*, vol. 36, no. 3, pp. 307–313, 1999.

Ma, C. and Feissel, M., “IERS Technical Note 23,” Central Bureau of IERS, Observatoire de Paris, 61, Avenue de l’Observatoire, F-75014 Paris, France, Tech. Rep., June 1997.

Madsen, K., Nielsen, H., and Tingleff, O., *Methods for Non-Linear Least Squares Problems*, 2nd ed., Technical University of Denmark, Faculty of Informatics and Mathematical Modelling, April 2004.

McRonald, A. and Nock, K., “Use of Atmospheric Drag to Circularize an Elliptic Spacecraft Orbit at Venus,” in *AIAA/AAS Astrodynamics Specialist Conference*. Jackson Hole, Wyoming, United States: American Astronautical Society, September 1977, pp. 760–784.

Millour, E., Forget, F., Spiga, A., Navarro, T., Madeleine, J.-B., Montabone, L., Pottier, A., Lefevre, F., Montmessin, F., Chaufray, J.-Y. *et al.*, “The Mars Climate Database (MCD version 5.2),” in *European Planetary Science Congress*, vol. 10, 2015, p. 438.

Montenbruck, O. and Gill, E., *Satellite Orbits: Models, Methods and Applications*. Springer, 2005.

Mooij, E., *The Motion of a Vehicle in a Planetary Atmosphere*. Delft University of Technology, 1994.

——, *Re-entry Systems*, Delft University of Technology, Faculty of Aerospace, Astrodynamics and Satellite Missions, September 2016, course: Re-entry Systems - AE4870B.

Murri, D., "Development of Autonomous Aerobraking - Phase 2," NASA Langley Research Center, Hampton, Virginia, United States, Tech. Rep., August 2013.

Myers, J., "Handbook of Equations for Mass and Area Properties of Various Geometrical Shapes," U.S. Naval Ordnance Test Station, Tech. Rep., April 1962.

Phillips, W., *Mechanics of Flight*. John Wiley & Sons, 2004.

Plimpton, S. and Gallis, M., *SPARTA Users Manual*, Sandia National Laboratories, United States Department of Energy, July 2017.

Potter, J., "Procedure for Estimating Aerodynamics of Three-Dimensional Bodies in Transitional Flow," *Progress in Astronautics and Aeronautics*, vol. 118, pp. 484–492, 1988.

Press, W., Flannery, B., Teukolsky, S., and Vetterling, W., *Numerical Recipes in Fortran 77: The Art of Scientific Computing*, 2nd ed. Cambridge University Press, 1992, vol. 1.

Prince, J., Dec, J., and Tolson, R., "Autonomous Aerobraking Using Thermal Response Surface Analysis," *Journal of Spacecraft and Rockets*, vol. 46, no. 2, pp. 292–298, 2009.

Regan, F. and Anandakrishnan, S., *Dynamics of Atmospheric Re-entry*, 2nd ed., ser. Education Series. American Institute of Aeronautics and Astronautics, 1993.

Schaub, H. and Junkins, J., *Analytical Mechanics of Space Systems*, 2nd ed., ser. Education Series. American Institute of Aeronautics and Astronautics, 2002.

Sivac, P. and Schirmann, T., "The Venus Express Spacecraft System Design," *ESA Special Publication*, vol. SP 1295, pp. 1–22, November 2007.

Smith, J., Jr. and Bell, J., "2001 Mars Odyssey Aerobraking," *Journal of Spacecraft and Rockets*, vol. 42, no. 3, pp. 406–415, 2005.

Sola, J., "Quaternion Kinematics For the Error-state KF," LAAS-CNRS, Toulouse, France, Tech. Rep., 2017.

Spencer, D. and Tolson, R., "Aerobraking Cost and Risk Decisions," *Journal of Spacecraft and Rockets*, vol. 44, no. 6, pp. 1285–1293, 2007.

Spohn, T., Breuer, D., and Johnson, T., Eds., *Encyclopedia of the Solar System*, 3rd ed. Elsevier, 2014.

Sutton, K. and Graves, R., "A General Stagnation-Point Convective-Heating Equation for Arbitrary Gas Mixtures," NASA Langley Research Center, Hampton, Virginia, United States, Tech. Rep., November 1971.

Svedhem, H., "Venus Express Aerobraking and End of Mission," April 2015, ESA/ESTEC. Slides. Available at https://goo.gl/kcCqhc.

Tolson, R. and Prince, J., "Onboard Atmospheric Modeling and Prediction for Autonomous Aerobraking Missions," in *AAS/AIAA Astrodynamics Specialist Conference*, Girdwood, Arkansas, United States, August 2011.

Val Serra, S., Bonnamy, O., Witasse, O., and Camino, O., "Venus Express Aerobraking," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 715–720, 2011.

Vittaldev, V., "The Unified State Model," Master Thesis, Delft University of Technology, May 2010.

Vittaldev, V., Mooij, E., and Naeije, M., "Unified State Model Theory and Application in Astrodynamics," *Celestial Mechanics and Dynamical Astronomy*, vol. 112, no. 3, pp. 253–282, February 2012.

Wakker, K., *Fundamentals of Astrodynamics*, Delft University of Technology, Faculty of Aerospace, Astrodynamics and Satellite Missions, January 2015, course: Astrodynamics I - AE4874.

Wan, E. and Van Der Merwe, R., "The Unscented Kalman Filter for Nonlinear Estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium*. Institute of Electrical and Electronics Engineers, 2000, pp. 153–158.

Wertz, J., *Orbit & Constellation Design & Management*, ser. Space Technology Library. Microcosm Press, 2009.

Wertz, J., Everett, D., and Puschell, J., Eds., *Space Mission Engineering: The New SMAD*, ser. Space Technology Library. Microcosm Press and Springer, 2011.

Wie, B., *Space Vehicle Dynamics and Control*, 2nd ed., ser. Education Series. American Institute of Aeronautics and Astronautics, 2008.

Wilmoth, R., Blanchard, R., and Moss, J., “Rarefied Transitional Bridging of Blunt Body Aerodynamics,” in *21st International Symposium on Rarefied Gas Dynamics*, Marseille, France, July 1998.

Wright, M., Edquist, K., Tang, C., Hollis, B., Krasa, P., and Campbell, C., “A Review of Aerothermal Modeling for Mars Entry Missions,” in *48th AIAA Aerospace Sciences Meeting*, Orlando, Florida, United States, December 2009, pp. 443–480.

Young, C., “The Magellan Venus Explorer’s Guide,” Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, United States, Tech. Rep., June 1990.

Zarchan, P. and Musoff, H., *Fundamentals of Kalman Filtering: A Practical Approach*, 3rd ed. American Institute of Aeronautics and Astronautics, 2013, vol. 232.

PAGE INTENTIONALLY LEFT BLANK

Appendices

“*Like diamonds in the sky.*”

Rihanna, Diamonds, 2012
Erwin, 2018

A Transformations

In this appendix, one will find the description of the frame and coordinate transformations, which are used throughout the report and the developed software. Extensive use of trigonometric identities (both Euclidean and spherical) is done. These identities are summarized in Appendix H.

A.1 | Frame Transformations

Transformations between the reference frames shown in Section 3.2 are expressed using unit-axis rotations. This means that a sequence of one or more rotations is necessary to convert between reference frames. Rotations about the three axes are given by:

$$\mathbf{R}_{\hat{\mathbf{x}}}\left(\xi\right) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\xi & \sin\xi \\ 0 & -\sin\xi & \cos\xi \end{bmatrix} \tag{A.1a}$$

$$\mathbf{R}_{\hat{\mathbf{y}}}\left(\xi\right) = \begin{bmatrix} \cos\xi & 0 & -\sin\xi \\ 0 & 1 & 0 \\ \sin\xi & 0 & \cos\xi \end{bmatrix} \tag{A.1b}$$

$$\mathbf{R}_{\hat{\mathbf{z}}}\left(\xi\right) = \begin{bmatrix} \cos\xi & \sin\xi & 0 \\ -\sin\xi & \cos\xi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.1c}$$

where a positive rotation is carried out in the direction given by the right-hand rule. Intermediate frames are denoted with an apostrophe. Reference for this section is Mooij (1994), unless otherwise stated.

From $\mathcal{I}$ to $\mathcal{O}$ (Vittaldev, 2010) Transformation from $\mathcal{I}$ to $\mathcal{O}$ requires the knowledge of the right ascension of the ascending node Ω and the inclination i of the spacecraft orbit. This transformation is best described by using an Euler eigenaxis rotation, where $\hat{\mathbf{e}}$ is the line of nodes and the angle ξ is the inclination. The line of nodes is defined as

$$\hat{\mathbf{n}} = \left[\cos\Omega, \sin\Omega, 0\right]^{\mathrm{T}} \ ,$$

and the full DCM is:

$$\begin{aligned} \mathbf{C}_{\mathcal{O}}^{\mathcal{I}} &= \mathbf{I}\cos i + \left(1-\cos i\right)\hat{\mathbf{n}}\hat{\mathbf{n}}^{\mathrm{T}} - \hat{\mathbf{n}}^{\mathrm{S}}\sin i = \\ &= \begin{bmatrix} \cos i + \cos^2\Omega\left(1-\cos i\right) & \sin\Omega\cos\Omega\left(1-\cos i\right) & -\sin i\sin\Omega \\ \sin\Omega\cos\Omega\left(1-\cos i\right) & \cos i + \sin^2\Omega\left(1-\cos i\right) & \sin i\cos\Omega \\ \sin i\sin\Omega & -\sin i\cos\Omega & \cos i \end{bmatrix} \end{aligned} \tag{A.2}$$

From $\mathcal{O}$ to $\mathcal{L}$ (Vittaldev, 2010) Transforming from $\mathcal{O}$ to $\mathcal{L}$, also required the knowledge of some of the Keplerian elements of the spacecraft. In particular, one needs Ω, as earlier and the argument of perigee ω and true anomaly ϑ. The full transformation is given by Eq. (A.3a), where λ is given by Eq. (A.3b), and is referred to as right ascension of latitude, The argument of latitude u, instead, is found with Eq. (A.3c).

① $\mathcal{L} \leftarrow \mathcal{O}$ with rotation about $\hat{\mathbf{z}}_{\mathcal{O}} \equiv \hat{\mathbf{z}}_{\mathcal{L}}$ by λ

$$\mathbf{C}_{\mathcal{L}}^{\mathcal{O}} = \mathbf{R}_{\hat{\mathbf{z}}_{\mathcal{O}}}\left(\lambda\right) = \begin{bmatrix} \cos\lambda & \sin\lambda & 0 \\ -\sin\lambda & \cos\lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.3a}$$

$$\lambda = \Omega + u \tag{A.3b}$$

$$u = \omega + \vartheta \tag{A.3c}$$

FROM $\mathcal{I}$ TO $\mathcal{R}$ Since frames $\mathcal{I}$ and $\mathcal{R}$ coincide at the reference epoch, and their $\hat{\mathbf{z}}$ axes always coincide, one only needs to rotate about said $\hat{\mathbf{z}}$ axis by the time t elapsed since the reference period, multiplied by the rotation rate of the planet ω.

1. $\mathcal{R} \leftarrow \mathcal{I}$ with rotation about $\hat{\mathbf{z}}_{\mathcal{I}} \equiv \hat{\mathbf{z}}_{\mathcal{R}}$ by ωt given by Eq. (A.4)

$$\mathbf{C}_{\mathcal{R}}^{\mathcal{I}} = \mathbf{R}_{\hat{\mathbf{z}}_{\mathcal{I}}}(\omega t) = \begin{bmatrix} \cos\omega t & \sin\omega t & 0 \\ -\sin\omega t & \cos\omega t & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.4}$$

Note that this transformation is very simplified. In reality, one would need to take into account many factors such as nutation, precession, sidereal motion and polar motion, together with the variation of the rotation rate of the planet over time (excess length of day).

FROM $\mathcal{R}$ TO $\mathcal{V}$ This transformation is based on the position in spherical coordinates of the spacecraft, and thus depends on its latitude δ and longitude τ. The full transformation is given by Eq. (A.5).

1. $\mathcal{R}' \leftarrow \mathcal{R}$ with rotation about $\hat{\mathbf{z}}_{\mathcal{R}} \equiv \hat{\mathbf{z}}_{\mathcal{R}'}$ by τ
2. $\mathcal{V} \leftarrow \mathcal{R}'$ with rotation about $\hat{\mathbf{y}}_{\mathcal{R}'} \equiv \hat{\mathbf{y}}_{\mathcal{V}}$ by $-\delta - \pi/2$

$$\mathbf{C}_{\mathcal{V}}^{\mathcal{R}} = \mathbf{C}_{\mathcal{V}}^{\mathcal{R}'}\mathbf{C}_{\mathcal{R}'}^{\mathcal{R}} = \mathbf{R}_{\hat{\mathbf{y}}_{\mathcal{R}'}}\left(-\delta - \frac{\pi}{2}\right)\mathbf{R}_{\hat{\mathbf{z}}_{\mathcal{R}}}(\tau) = \begin{bmatrix} -\sin\delta\cos\tau & -\sin\delta\sin\tau & \cos\delta \\ -\sin\tau & \cos\tau & 0 \\ -\cos\delta\cos\tau & -\cos\delta\sin\tau & -\sin\delta \end{bmatrix} \tag{A.5}$$

Note that this transformation is also simplified, due to the assumption that the body is a perfect sphere. Thus, geodetic and geocentric altitude and latitude are assumed to coincide.

FROM $\mathcal{V}$ TO $\mathcal{T}$ This transformation depends on the airspeed velocity vector of the spacecraft. The direction of the velocity is defined in the vertical frame with the heading (angle from $\hat{\mathbf{x}}_{\mathcal{V}}$ to projection of velocity on $\hat{\mathbf{x}}_{\mathcal{V}} - \hat{\mathbf{y}}_{\mathcal{V}}$ plane) and flight-path (angle from projection of velocity on $\hat{\mathbf{x}}_{\mathcal{V}} - \hat{\mathbf{y}}_{\mathcal{V}}$ plane to velocity vector) angles, χ and γ, respectively. The full transformation is given by Eq. (A.6).

1. $\mathcal{V}' \leftarrow \mathcal{V}$ with rotation about $\hat{\mathbf{z}}_{\mathcal{V}} \equiv \hat{\mathbf{z}}_{\mathcal{V}'}$ by χ
2. $\mathcal{T} \leftarrow \mathcal{V}'$ with rotation about $\hat{\mathbf{y}}_{\mathcal{V}'} \equiv \hat{\mathbf{y}}_{\mathcal{T}}$ by γ

$$\mathbf{C}_{\mathcal{T}}^{\mathcal{V}} = \mathbf{C}_{\mathcal{T}}^{\mathcal{V}'}\mathbf{C}_{\mathcal{V}'}^{\mathcal{V}} = \mathbf{R}_{\hat{\mathbf{y}}_{\mathcal{V}'}}(\gamma)\,\mathbf{R}_{\hat{\mathbf{z}}_{\mathcal{V}}}(\chi) = \begin{bmatrix} \cos\gamma\cos\chi & \cos\gamma\sin\chi & -\sin\gamma \\ -\sin\chi & \cos\chi & 0 \\ \sin\gamma\cos\chi & \sin\gamma\sin\chi & \cos\gamma \end{bmatrix} \tag{A.6}$$

FROM $\mathcal{L}$ TO $\mathcal{T}$ The local and trajectory frames are closely related. Despite being centered at different locations ($\mathcal{L}$ is centered at the planet and $\mathcal{T}$ at the SC), they both rotate with the spacecraft, and their axes are always parallel. In fact, it can be shown that the relationship between these two axes is simply given by two 90 deg rotations. The full transformation is given by Eq. (A.7).

1. $\mathcal{L}' \leftarrow \mathcal{L}$ with rotation about $\hat{\mathbf{z}}_{\mathcal{L}} \equiv \hat{\mathbf{z}}_{\mathcal{L}'}$ by $\pi/2$
2. $\mathcal{T} \leftarrow \mathcal{L}'$ with rotation about $\hat{\mathbf{x}}_{\mathcal{L}'} \equiv \hat{\mathbf{x}}_{\mathcal{T}}$ by $-\pi/2$

$$\mathbf{C}_{\mathcal{T}}^{\mathcal{L}} = \mathbf{C}_{\mathcal{T}}^{\mathcal{L}'}\mathbf{C}_{\mathcal{L}'}^{\mathcal{L}} = \mathbf{R}_{\hat{\mathbf{x}}_{\mathcal{L}'}}\left(-\frac{\pi}{2}\right)\mathbf{R}_{\hat{\mathbf{z}}_{\mathcal{L}}}\left(\frac{\pi}{2}\right) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix} \tag{A.7}$$

FROM $\mathcal{T}$ TO $\mathcal{A}$ As mentioned in Section 3.2, the $\hat{\mathbf{x}}$ axes of the $\mathcal{T}$ and $\mathcal{A}$ frames always match. This is because to go from one frame to the other, only one rotation about said axis is needed. This rotation is done by $-\sigma$, the negative of bank angle (angle between the lift vector and the local vertical). The full transformation is given by Eq. (A.8).

1. $\mathcal{A} \leftarrow \mathcal{T}$ with rotation about $\hat{\mathbf{x}}_{\mathcal{T}} \equiv \hat{\mathbf{x}}_{\mathcal{A}}$ by $-\sigma$

$$\mathbf{C}_{\mathcal{A}}^{\mathcal{T}} = \mathbf{R}_{\hat{\mathbf{x}}_{\mathcal{T}}}(-\sigma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\sigma & -\sin\sigma \\ 0 & \sin\sigma & \cos\sigma \end{bmatrix} \tag{A.8}$$

From $\mathcal{A}$ to $\mathcal{B}$ Transformation from $\mathcal{A}$ to $\mathcal{B}$ frames is done by using the angles of sideslip β (angle between the $\hat{\mathbf{x}}_{\mathcal{B}} - \hat{\mathbf{y}}_{\mathcal{B}}$ plane and the velocity vector) and of attack α (angle between the projection of the velocity vector on the $\hat{\mathbf{x}}_{\mathcal{B}} - \hat{\mathbf{y}}_{\mathcal{B}}$ plane and the $\hat{\mathbf{x}}_{\mathcal{B}}$ vector). The full transformation is given by Eq. (A.9).

① $\mathcal{A}' \leftarrow \mathcal{A}$ with rotation about $\hat{\mathbf{z}}_{\mathcal{A}} \equiv \hat{\mathbf{z}}_{\mathcal{A}'}$ by $-\beta$

② $\mathcal{B} \leftarrow \mathcal{A}'$ with rotation about $\hat{\mathbf{y}}_{\mathcal{A}'} \equiv \hat{\mathbf{y}}_{\mathcal{B}}$ by α

$$\mathbf{C}_{\mathcal{B}}^{\mathcal{A}} = \mathbf{C}_{\mathcal{B}}^{\mathcal{A}'} \mathbf{C}_{\mathcal{A}'}^{\mathcal{A}} = \mathbf{R}_{\hat{\mathbf{y}}_{\mathcal{A}'}}(\alpha)\, \mathbf{R}_{\hat{\mathbf{z}}_{\mathcal{A}}}(-\beta) = \begin{bmatrix} \cos\alpha\cos\beta & -\cos\alpha\sin\beta & -\sin\alpha \\ \sin\beta & \cos\beta & 0 \\ \sin\alpha\cos\beta & -\sin\alpha\sin\beta & \cos\alpha \end{bmatrix} \tag{A.9}$$

Any transformation that was not shown above, can be easily found by combining transformation matrices. For instance, the inverse transformation from $\mathcal{B}$ to $\mathcal{I}$ is defined as:

$$\begin{aligned} \mathbf{C}_{\mathcal{I}}^{\mathcal{B}} &= \mathbf{C}_{\mathcal{I}}^{\mathcal{R}} \mathbf{C}_{\mathcal{R}}^{\mathcal{R}'} \mathbf{C}_{\mathcal{R}'}^{\mathcal{V}} \mathbf{C}_{\mathcal{V}}^{\mathcal{V}'} \mathbf{C}_{\mathcal{V}'}^{\mathcal{T}} \mathbf{C}_{\mathcal{T}}^{\mathcal{A}} \mathbf{C}_{\mathcal{A}}^{\mathcal{A}'} \mathbf{C}_{\mathcal{A}'}^{\mathcal{B}} = \\ &= \left(\mathbf{C}_{\mathcal{R}}^{\mathcal{I}}\right)^{\mathrm{T}} \left(\mathbf{C}_{\mathcal{R}'}^{\mathcal{R}}\right)^{\mathrm{T}} \left(\mathbf{C}_{\mathcal{V}}^{\mathcal{R}'}\right)^{\mathrm{T}} \left(\mathbf{C}_{\mathcal{V}'}^{\mathcal{V}}\right)^{\mathrm{T}} \left(\mathbf{C}_{\mathcal{T}}^{\mathcal{V}'}\right)^{\mathrm{T}} \left(\mathbf{C}_{\mathcal{A}}^{\mathcal{T}}\right)^{\mathrm{T}} \left(\mathbf{C}_{\mathcal{A}'}^{\mathcal{A}}\right)^{\mathrm{T}} \left(\mathbf{C}_{\mathcal{B}}^{\mathcal{A}'}\right)^{\mathrm{T}} = \\ &= \left(\mathbf{C}_{\mathcal{B}}^{\mathcal{A}'} \mathbf{C}_{\mathcal{A}'}^{\mathcal{A}} \mathbf{C}_{\mathcal{A}}^{\mathcal{T}} \mathbf{C}_{\mathcal{T}}^{\mathcal{V}'} \mathbf{C}_{\mathcal{V}'}^{\mathcal{V}} \mathbf{C}_{\mathcal{V}}^{\mathcal{R}'} \mathbf{C}_{\mathcal{R}'}^{\mathcal{R}} \mathbf{C}_{\mathcal{R}}^{\mathcal{I}}\right)^{\mathrm{T}} \end{aligned}$$

A.2 | Coordinate Transformations

This section is dedicated to the transformations between the state variables introduced in Section 3.3. Attitude coordinates are treated first, and they are followed by the only remaining set of coordinates, i.e., position and velocity.

A.2.1 | Attitude

In Section 3.3, the relations to obtain the DCM for all attitude states have been shown. In this subsection then, the inverse transformations will be given.

Direction Cosine Matrix to Quaternions (Schaub and Junkins, 2002) Given a DCM $\mathbf{C}$, the real and imaginary part of the quaternions can be found with:

$$\eta = \pm\frac{1}{2}\sqrt{\mathrm{trace}\left(\mathbf{C}\right) + 1} \tag{A.10a}$$

$$\boldsymbol{\varepsilon} = \frac{1}{4\eta}\begin{pmatrix} C_{23} - C_{32} \\ C_{31} - C_{13} \\ C_{12} - C_{21} \end{pmatrix} \tag{A.10b}$$

where $\mathrm{trace}\left(\mathbf{C}\right)$ is the sum of the diagonal elements of $\mathbf{C}$ and C_{ij} represents the element of $\mathbf{C}$ in the i-th row and j-th column. Note that the sign in Eq. (A.10a) can be chosen arbitrarily, since the rotations expressed by $(\boldsymbol{\varepsilon}, \eta)$ and $(-\boldsymbol{\varepsilon}, -\eta)$ are the same. Nonetheless, in general, one should choose the sign based on the quaternion history.

In case the value of η equals zero, a singularity arises in Eq. (A.10). This singularity is bypassed by solving the system of nine equations from Eq. (3.5), where η is replaced with zero.

Direction Cosine Matrix to Modified Rodrigues Parameters (Schaub and Junkins, 2002) If the DCM of the full rotation is known, the MRP can be found by using:

$$\xi = \arccos\left[\frac{1}{2}\left(\mathrm{trace}\left(\mathbf{C}\right) - 1\right)\right] \tag{A.11a}$$

$$\hat{\mathbf{e}} = \frac{1}{2\sin\xi}\begin{pmatrix} C_{23} - C_{32} \\ C_{31} - C_{13} \\ C_{12} - C_{21} \end{pmatrix} \tag{A.11b}$$

to find the rotation angle ξ and the eigenaxis $\hat{\mathbf{e}}$ and then, by substituting their values in Eq. (3.6a). The singularity at $\xi = 0$ rad can be avoided by using a strategy analogous to the one used for quaternions. Hence, one would solve the system of nine equations that arises from Eq. (3.3a), where ξ is given the value of 0 rad.

Direction Cosine Matrix to Exponential Map (Schaub and Junkins, 2002) Similarly to the procedure for MRP, the EM can be found by replacing ξ and $\hat{\mathbf{e}}$ in Eq. (3.9a), with the values from Eq. (A.11).

A.2.2 | Position and Velocity

This subsection summarizes the transformations between state variables for position and velocity. Transformations to and from Keplerian elements will be given for Cartesian coordinates, unified state model, and spherical coordinates.

Note that transformation to and from Keplerian elements is only done by considering elliptical orbits. In fact, singularities arise in these definitions in case circular, parabolic or hyperbolic orbital elements are used, instead.

Keplerian Elements to Cartesian Coordinates (Mooij, 1994)

Transformation from Keplerian to Cartesian is done with the aid of the auxiliary variable $\boldsymbol{\Xi}$:

$$\boldsymbol{\Xi} = \begin{bmatrix} \cos\Omega\cos\omega - \sin\Omega\sin\omega\cos i & -\cos\Omega\sin\omega - \sin\Omega\cos\omega\cos i \\ \sin\Omega\cos\omega + \cos\Omega\sin\omega\cos i & -\sin\Omega\sin\omega + \cos\Omega\cos\omega\cos i \\ \sin\omega\sin i & \cos\omega\sin i \end{bmatrix} \tag{A.12}$$

Then, one can find the Cartesian coordinates with:

$$\mathbf{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = r\,\boldsymbol{\Xi}\begin{pmatrix} \cos\vartheta \\ \sin\vartheta \end{pmatrix} \tag{A.13a}$$

$$\dot{\mathbf{r}} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \frac{\mu}{H}\,\boldsymbol{\Xi}\begin{pmatrix} -\sin\vartheta \\ e+\cos\vartheta \end{pmatrix} \tag{A.13b}$$

where r is given by Eq. (3.1) and the specific angular momentum can be found with:

$$H = \sqrt{\mu a\left(1-e^2\right)} \tag{A.14}$$

Cartesian Coordinates to Keplerian Elements (Wertz, 2009)

To transform from Cartesian to Keplerian coordinates, it is first useful to define the position and velocity vectors, as:

$$\mathbf{r} = [x, y, z]^{\mathrm{T}} \tag{A.15a}$$

$$\dot{\mathbf{r}} = [\dot{x}, \dot{y}, \dot{z}]^{\mathrm{T}} \tag{A.15b}$$

Using the definitions in Eq. (A.15), one can find the specific angular momentum vector, the vector along the line of nodes and the eccentricity vector:

$$\mathbf{H} = \mathbf{r} \times \dot{\mathbf{r}} \tag{A.16a}$$

$$\mathbf{n} = [0, 0, 1]^{\mathrm{T}} \times \hat{\mathbf{H}} \tag{A.16b}$$

$$\mathbf{e} = \frac{\dot{\mathbf{r}} \times \mathbf{H}}{\mu} - \hat{\mathbf{r}} \tag{A.16c}$$

Then, the Keplerian elements are computed with:

$$a = \left(\frac{2}{||\mathbf{r}||} - \frac{||\dot{\mathbf{r}}||^2}{\mu}\right)^{-1} \tag{A.17a}$$

$$e = ||\mathbf{e}|| \tag{A.17b}$$

$$i = \arccos\left(\frac{H_3}{||\mathbf{H}||}\right) \tag{A.17c}$$

$$\omega = \operatorname{sign}\left[(\hat{\mathbf{n}} \times \mathbf{e}) \cdot \mathbf{H}\right] \arccos\left(\hat{\mathbf{e}} \cdot \hat{\mathbf{n}}\right) \tag{A.17d}$$

$$\Omega = \arctan_2\left(\frac{n_2}{\sqrt{n_1^2+n_2^2}}, \frac{n_1}{\sqrt{n_1^2+n_2^2}}\right) \tag{A.17e}$$

$$\vartheta = \operatorname{sign}\left[(\mathbf{e} \times \mathbf{r}) \cdot \mathbf{H}\right] \arccos\left(\hat{\mathbf{e}} \cdot \hat{\mathbf{r}}\right) \tag{A.17f}$$

Keplerian Elements to Unified State Model (Vittaldev, 2010)

The velocity components C and R are found with:

$$C = \sqrt{\frac{\mu}{a\left(1-e^2\right)}} \tag{A.18a}$$

$$\mathbf{R} = \begin{pmatrix} R_1 \\ R_2 \end{pmatrix} = eC \begin{pmatrix} -\sin\left(\Omega+\omega\right) \\ \cos\left(\Omega+\omega\right) \end{pmatrix} \tag{A.18b}$$

where μ is the standard gravitational parameter of the orbited planet.

Quaternions In case of the use of quaternions, the quaternion of the transformation from the $\mathcal{I}$-frame to the $\mathcal{L}$-frame is given by:

$$\begin{pmatrix} \eta \\ \boldsymbol{\varepsilon} \end{pmatrix} = \begin{pmatrix} \sin\left(i/2\right)\cos\left(\frac{\Omega-u}{2}\right) \\ \sin\left(i/2\right)\sin\left(\frac{\Omega-u}{2}\right) \\ \cos\left(i/2\right)\sin\left(\frac{\Omega+u}{2}\right) \\ \cos\left(i/2\right)\cos\left(\frac{\Omega+u}{2}\right) \end{pmatrix} \tag{A.19}$$

where u is the argument of latitude, given by Eq. (A.3c).

Modified Rodrigues Parameters The transformation from orbital elements to modified Rodrigues parameters is done by using:

$$\boldsymbol{\sigma} = \frac{1}{1+\cos\left(i/2\right)\cos\left(\frac{\Omega+u}{2}\right)} \begin{pmatrix} \sin\left(i/2\right)\cos\left(\frac{\Omega-u}{2}\right) \\ \sin\left(i/2\right)\sin\left(\frac{\Omega-u}{2}\right) \\ \cos\left(i/2\right)\sin\left(\frac{\Omega+u}{2}\right) \end{pmatrix} \tag{A.20}$$

Exponential Map Vittaldev (2010) only provided transformations to and from quaternions and MRPs. Hence, the equations relating to EM had to be derived. This process is shown below.

Starting from the quaternion in Eq. (A.19), one can find the exponential map equivalent of the transformation to Keplerian elements, by using:

$$\mathbf{e} = \xi\hat{\mathbf{e}} \tag{A.21a}$$

$$\xi = 2\arccos\left(\eta\right) \tag{A.21b}$$

$$\hat{\mathbf{e}} = \frac{1}{\sin\left(\xi/2\right)}\boldsymbol{\varepsilon} \tag{A.21c}$$

which are the inverses of Eqs. (3.4a) and (3.4b), respectively. Applying the transformation from Eq. (A.21), yields:

$$\xi = 2\arccos\left[\cos\left(\frac{i}{2}\right)\cos\left(\frac{\Omega+u}{2}\right)\right] \tag{A.22a}$$

$$\hat{\mathbf{e}} = \frac{1}{\sin\left(\xi/2\right)} \begin{pmatrix} \sin\left(i/2\right)\cos\left(\frac{\Omega-u}{2}\right) \\ \sin\left(i/2\right)\sin\left(\frac{\Omega-u}{2}\right) \\ \cos\left(i/2\right)\sin\left(\frac{\Omega+u}{2}\right) \end{pmatrix} \tag{A.22b}$$

Clearly, a singularity arises when ξ equals 0 and 2π rad. The second singularity will be avoided, as explained in Subsection 3.3.1, by switching to and from the shadow exponential map, otherwise known as SEM. The first singularity, on the other hand, is resolved by knowing that when ξ is small, the term $\sin\left(\xi/2\right)$ can be replaced with its Taylor expansion, such that:

$$\xi\sin^{-1}\left(\frac{\xi}{2}\right) \approx \frac{48}{24-\xi^2} \tag{A.23}$$

Unified State Model to Keplerian Elements (Vittaldev, 2010)

The semi-major axis, eccentricity, true anomaly and argument of periapsis can be found with:

$$a = \frac{\mu}{C^2\left(1-e^2\right)} \tag{A.24a}$$

$$e = \frac{||\mathbf{R}||}{C} \tag{A.24b}$$

$$\vartheta = \arctan_2\left(\frac{R_1\cos\lambda + R_2\sin\lambda}{||\mathbf{R}||}, \frac{R_2\cos\lambda - R_1\sin\lambda}{||\mathbf{R}||}\right) \tag{A.24c}$$

$$\omega = \lambda - \Omega - \vartheta \tag{A.24d}$$

where the value of λ depends on the chosen attitude representation and is shown in the respective paragraphs below.

Quaternions If quaternions are implemented, the remaining orbital elements are given by:

$$i = \arccos\left[1 - 2\left(\varepsilon_1^2 + \varepsilon_2^2\right)\right] \tag{A.25a}$$

$$\Omega = \arctan_2\left(\frac{\varepsilon_1\varepsilon_3 + \varepsilon_2\eta}{\sqrt{\left(\varepsilon_1^2 + \varepsilon_2^2\right)\left(\varepsilon_3^2 + \eta^2\right)}}, \frac{\varepsilon_1\eta - \varepsilon_2\varepsilon_3}{\sqrt{\left(\varepsilon_1^2 + \varepsilon_2^2\right)\left(\varepsilon_3^2 + \eta^2\right)}}\right) \tag{A.25b}$$

Here, the value of λ is given by:

$$\lambda = \arctan_2\left(\frac{2\varepsilon_3\eta}{\varepsilon_3^2 + \eta^2}, \frac{\eta^2 - \varepsilon_3^2}{\varepsilon_3^2 + \eta^2}\right) \tag{A.26}$$

Modified Rodrigues Parameters The same procedure as explained by Vittaldev (2010) for the determination of the transformation from USM7 will be applied here. Thus, the reader is referred to pages 68 – 69 of this reference for more details.

To start, one needs to express the inverse transformation from the $\mathcal{O}$-frame to $\mathcal{I}$-frame in terms of MRPs. This is done by taking the transpose of Eq. (3.8), i.e.:

$$\mathbf{C}_{\mathcal{I}}^{\mathcal{O}} = \mathbf{I} + \frac{4\left(1 - \sigma^2\right)\boldsymbol{\sigma}^{\mathrm{S}} - 8\left(\boldsymbol{\sigma}^{\mathrm{S}}\right)^2}{\left(1 + \sigma^2\right)^2} \tag{A.27}$$

It is known that Ω is the angle between the $\hat{\mathbf{x}}_{\mathcal{I}}$ axis and the line of nodes, $\hat{\mathbf{n}}$, and from Vittaldev (2010), one knowns that the line of nodes can be expressed in terms of the DCM in Eq. (A.27), as:

$$\mathbf{n} = \left[-C_{23}, C_{31}, 0\right]^{\mathrm{T}}$$

where one should keep in mind that the vector does not have a unit magnitude. Thus, one can then find the sine and cosine of Ω by using the definitions of inner and outer products:

$$\sin\Omega = \frac{||\hat{\mathbf{x}}_{\mathcal{I}} \times \mathbf{n}||}{||\mathbf{n}||}$$
$$\cos\Omega = \frac{\hat{\mathbf{x}}_{\mathcal{I}} \cdot \mathbf{n}}{||\mathbf{n}||}$$

Then, the value of the right ascension of ascending node is found by using the inverse tangent function with quadrant information, that is:

$$\Omega = \arctan_2\left(\frac{2\sigma_1\sigma_3 + \sigma_2\left(1 - \sigma^2\right)}{\sqrt{\left(\sigma_1^2 + \sigma_2^2\right)\left[4\sigma_3^2 + \left(1 - \sigma^2\right)\right]}}, \frac{\sigma_1\left(1 - \sigma^2\right) - 2\sigma_2\sigma_3}{\sqrt{\left(\sigma_1^2 + \sigma_2^2\right)\left[4\sigma_3^2 + \left(1 - \sigma^2\right)\right]}}\right) \tag{A.28}$$

where the denominators are the expansion of:

$$\sqrt{\left[2\sigma_1\sigma_3 + \sigma_2\left(1 - \sigma^2\right)\right]^2 + \left[\sigma_1\left(1 - \sigma^2\right) - 2\sigma_2\sigma_3\right]^2}$$

The value of inclination, on the other hand, can be found by using the following definition, also extracted from Vittaldev (2010):

$$\cos i = C_{33}$$

where, once again, the DCM from Eq. (A.27) is used. Thus, this yields the equation for inclination:

$$i = \arccos\left[\frac{4\left(\sigma_3^2 - \sigma_1^2 - \sigma_2^2\right) + \left(1 - \sigma^2\right)^2}{\left(1 + \sigma^2\right)^2}\right] \tag{A.29}$$

where the value of σ is simply the magnitude of the MRP vector:

$$\sigma = ||\boldsymbol{\sigma}||$$

Recalling what was said at the beginning of this part, one also needs the value of λ, the right ascension of latitude, to successfully convert to Keplerian elements. Its value was derived by Vittaldev (2010), and is given by:

$$\lambda = \arctan_2\left(\frac{4\sigma_3\left(1 - \sigma^2\right)}{4\sigma_3^2 + \left(1 - \sigma^2\right)^2}, \frac{\left(1 - \sigma^2\right)^2 - 4\sigma_3^2}{4\sigma_3^2 + \left(1 - \sigma^2\right)^2}\right) \tag{A.30}$$

Note that in case of shadow modified Rodrigues parameters, the equations shown above differ. For starters, Eq. (A.27) becomes:

$$\mathbf{C}_{\mathcal{I}}^{\mathcal{O}} = \mathbf{I} + \frac{4\left(1-\sigma_s^2\right)\boldsymbol{\sigma}_s^{\mathrm{S}} - 8\left(\boldsymbol{\sigma}_s^{\mathrm{S}}\right)^2}{\left(1+\sigma_s^2\right)^2} \tag{A.31}$$

which was derived by replacing $\boldsymbol{\sigma}$ and σ with the respective definitions of SMRP, i.e.:

$$\boldsymbol{\sigma} = -\frac{\boldsymbol{\sigma}_s}{\sigma_s^2} \qquad \sigma = -\frac{1}{\sigma_s}$$

This, in turn, results in a slightly modified version of Ω and λ:

$$\Omega = \arctan_2\left(\frac{2\sigma_{s,1}\sigma_{s,3} + \sigma_{s,2}\left(\sigma_s^2-1\right)}{\sqrt{\left(\sigma_{s,1}^2+\sigma_{s,2}^2\right)\left[4\sigma_{s,3}^2-\left(\sigma_s^2-1\right)\right]}}, \frac{\sigma_{s,1}\left(\sigma_s^2-1\right) - 2\sigma_{s,2}\sigma_{s,3}}{\sqrt{\left(\sigma_{s,1}^2+\sigma_{s,2}^2\right)\left[4\sigma_{s,3}^2-\left(\sigma_s^2-1\right)\right]}}\right) \tag{A.32a}$$

$$\lambda = \arctan_2\left(\frac{4\sigma_{s,3}\left(1-\sigma_s^2\right)}{4\sigma_{s,3}^2+\left(\sigma_s^2-1\right)^2}, \frac{\left(\sigma_s^2-1\right)^2-4\sigma_{s,3}^2}{4\sigma_{s,3}^2+\left(\sigma_s^2-1\right)^2}\right) \tag{A.32b}$$

where, λ is derived with the same procedure as highlighted by Vittaldev (2010) on page 53, where the quaternion elements are replaced with their SMRP values:

$$\begin{aligned}\boldsymbol{\varepsilon} &= -\frac{2}{\sigma_s+1}\boldsymbol{\sigma}_s \\ \eta &= \frac{\sigma_s^2-1}{\sigma_s^2+1}\end{aligned}$$

Note that, since the term $\sigma_s^2 - 1$ is always squared, there is actually no difference between Eq. (A.30) and Eq. (A.32b). Finally, the equation for i, is not affected, since it only depends on a diagonal element of the DCM.

Exponential Map For the exponential map, the same derivation is followed. Thus, the DCM from $\mathcal{F}_{\mathcal{O}}$ to $\mathcal{F}_{\mathcal{I}}$, in terms of EM is given by the inverse of Eq. (3.11), i.e.:

$$\mathbf{C}_{\mathcal{I}}^{\mathcal{O}} = \mathbf{I}\cos\xi - \left(1-\cos\xi\right)\hat{\mathbf{e}}\hat{\mathbf{e}}^{\mathrm{T}} + \hat{\mathbf{e}}^{\mathrm{S}}\sin\xi \tag{A.33}$$

Then, the right ascension of ascending node and inclination can be found with the same procedure, yielding:

$$\Omega = \arctan_2\left(\frac{\hat{e}_1\hat{e}_3\left(1-\cos\xi\right)+\hat{e}_2\sin\xi}{\sqrt{\left(\hat{e}_1^2+\hat{e}_2^2\right)\left[\sin^2\xi+\hat{e}_3^2\left(1-\cos\xi\right)^2\right]}}, \frac{\hat{e}_1\sin\xi-\hat{e}_2\hat{e}_3\left(1-\cos\xi\right)}{\sqrt{\left(\hat{e}_1^2+\hat{e}_2^2\right)\left[\sin^2\xi+\hat{e}_3^2\left(1-\cos\xi\right)^2\right]}}\right) \tag{A.34a}$$

$$i = \arccos\left[\left(\hat{e}_3^2-1\right)\left(1-\cos\xi\right)-1\right] \tag{A.34b}$$

As for the MRP case, the denominators in Eq. (A.34a) are obtained by expanding:

$$\sqrt{\left[\hat{e}_1\hat{e}_3\left(1-\cos\xi\right)+\hat{e}_2\sin\xi\right]^2+\left[\hat{e}_1\sin\xi-\hat{e}_2\hat{e}_3\left(1-\cos\xi\right)\right]^2}$$

The equations derived here, are in fact in terms of Euler eigenaxis elements. However, the relation between these and EM is very straightforward, being expressed as:

$$\xi = ||\mathbf{e}|| \qquad \text{and} \qquad \hat{\mathbf{e}} = \frac{\mathbf{e}}{\xi}$$

Finally, the value of λ needs to be determined. This can be extracted from the definition of the Euler eigenaxis rotation. In particular, it is known that the third element of the $\hat{\mathbf{e}}$ vector is given by:

$$\hat{e}_3 = \sin^{-1}\left(\frac{\xi}{2}\right)\cos\left(\frac{i}{2}\right)\sin\left(\frac{\lambda}{2}\right)$$

Furthermore, the rotation angle ξ can be expressed as:

$$\cos\left(\frac{\xi}{2}\right) = \cos\left(\frac{i}{2}\right)\cos\left(\frac{\lambda}{2}\right)$$

Thus, combining the two equations yields:

$$\lambda = \arctan_2\left(\hat{e}_3\sin\left(\frac{\xi}{2}\right), \cos\left(\frac{\xi}{2}\right)\right) \tag{A.35}$$

Keplerian Elements to Spherical Coordinates

The first element that is found, is the orbital distance. This is easily found by first computing the semi-latus rectum:

$$p = a\left(1 - e^2\right) \tag{A.36a}$$

$$r = \frac{p}{1 + e\cos\vartheta} \tag{A.36b}$$

Then, the velocity can be found with the vis-viva equation, hence with:

$$V = \sqrt{\mu\left(\frac{2}{r} - \frac{1}{a}\right)} \tag{A.37}$$

Using the two definitions of specific angular momentum H in Eq. (A.38a), one can find the flight-path angle γ, by equating the two equations. The result is:

$$H = \sqrt{\mu p} = rV\sin\gamma \tag{A.38a}$$

$$\gamma = \arcsin\left(\frac{\sqrt{\mu p}}{rV}\right) \tag{A.38b}$$

Since γ is defined between $[-\frac{\pi}{2}, \frac{\pi}{2}]$, Eq. (A.38b) does not lead to quadrant ambiguity. With the aid of Fig. A.1, the latitude δ and longitude τ can be found. Using the law of cosines for arc lengths, with Ω, u $(= \omega + \vartheta)$ and $(\pi - i)$ (the angle highlighted in purple in Fig. A.1), one can find a (the arc length highlighted in red):

$$a = \arccos_2\left(\cos\Omega\cos u - \cos i\sin\Omega\sin u, \pi - i\right) \tag{A.39a}$$

Furthermore, using the same triangle and the cosine rule, the angle α (highlighted in blue) can be found:

$$\alpha = \arccos_2\left(\frac{\cos u - \cos\Omega\cos a}{\sin\Omega\sin a}, u\right) \tag{A.39b}$$

The use of $(\pi - i)$ and u as second terms for the $\arccos_2$ function is extracted from Wertz (2009), where the $\arccos_2$ function computes the arccosine with quadrant information, and is defined in Appendix H by Eq. (H.10). Using Napier's rules for right angled triangles, one can find δ and τ, as:

$$\delta = \arcsin\left(\sin a\sin\alpha\right) \qquad Q\left(\delta\right) = Q\left(\alpha\right) \tag{A.40a}$$

$$\tau = \arctan\left(\tan a\cos\alpha\right) \qquad Q\left(\tau\right) = \begin{cases} Q\left(a\right) & a \leqslant \pi/2 \\ \mathrm{mod}_4\left(Q\left(a\right) + 2\right) + 1 & a > \pi/2 \end{cases} \tag{A.40b}$$

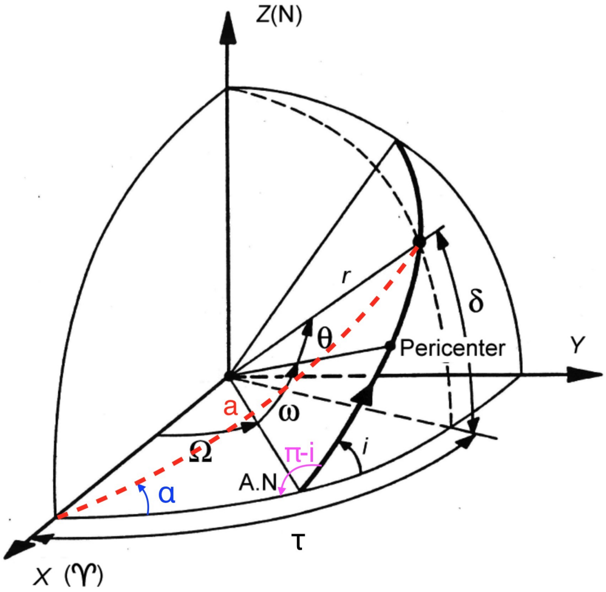


Figure A.1: *Representation of Keplerian Euler angles i, Ω and ω (adapted from Wakker, 2015).*

where Q is the function that determines the quadrant. Finally, the heading angle χ, is found by considering the complimentary spherical triangle of the one shown in Fig. A.1, i.e., the one with sides $\pi/2$, $(\pi/2 - \delta)$ and $(2\pi - a)$. Using the Napier's rule for quadrantal spherical triangles, the result is:

$$\chi = \arctan_2 \left(\frac{\cos i}{\cos \delta}, \frac{\tan \delta}{\tan a} \right) \tag{A.41}$$

Spherical Coordinates to Keplerian Elements (Wakker, 2015)

Transformation from spherical to orbital elements is done with:

$$a = \frac{r}{2 - \frac{rV^2}{\mu}} \tag{A.42a}$$

$$e = \sqrt{1 - \frac{rV^2}{\mu} \left(2 - \frac{rV^2}{\mu} \right) \cos^2 \gamma} \tag{A.42b}$$

$$i = \arccos \left(\cos \delta \sin \chi \right) \tag{A.42c}$$

$$\Omega = \tau - \arctan \left(\frac{\cos i}{\cos \chi} \tan \delta \right) \tag{A.42d}$$

$$\omega = \arctan \left(\frac{\tan \delta}{\cos \chi} \right) - \vartheta \tag{A.42e}$$

$$\vartheta = 2 \arctan \left(\sqrt{\frac{1+e}{1-e}} \tan \frac{E}{2} \right) \tag{A.42f}$$

where the eccentric anomaly E is given by:

$$E = \arctan \left(\sqrt{\frac{a}{\mu}} \frac{rV \sin \gamma}{a - r} \right) \tag{A.43}$$

Page Intentionally Left Blank

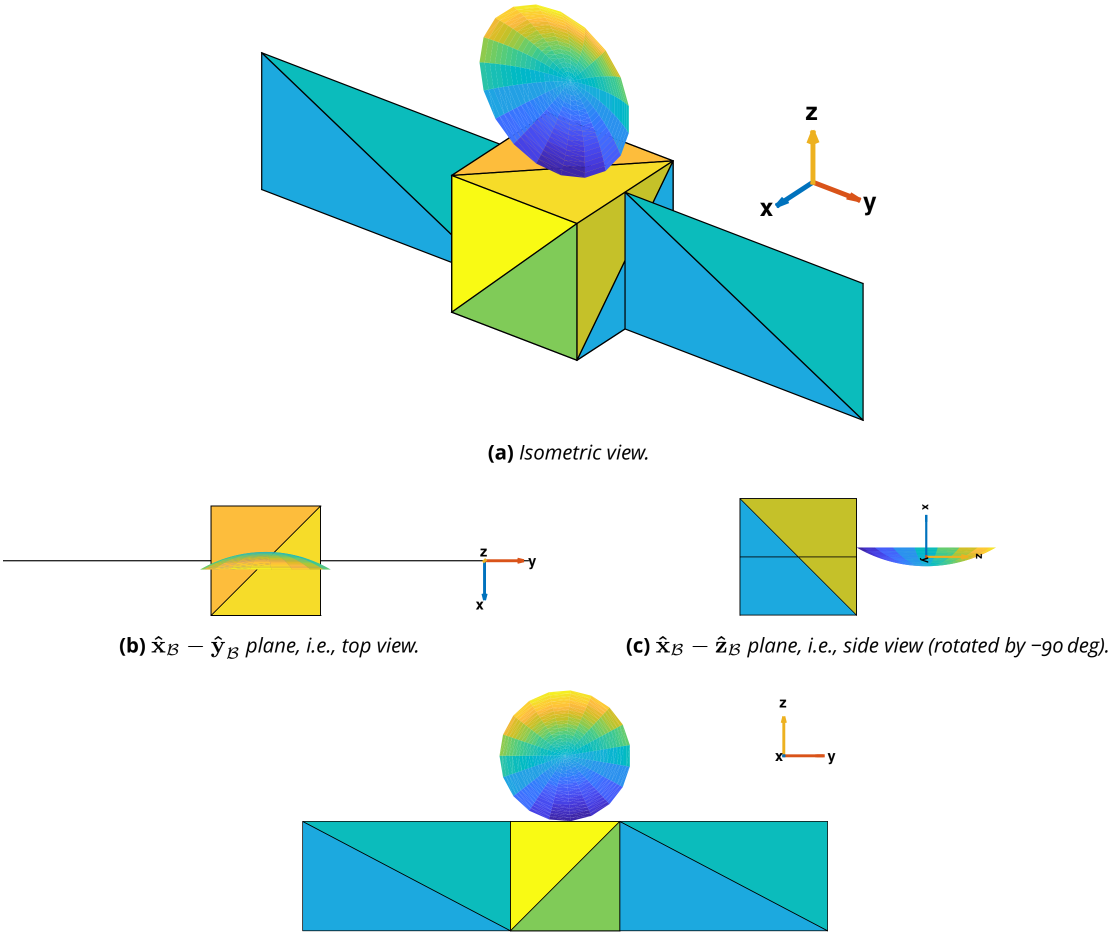
B
Spacecraft Modeling

In the literature study preceding this thesis, a preliminary version of the discretized spacecraft shape was introduced. However, as it turns out, this model was not compliant with the SPARTA shape requirements, and thus needed to be adapted. This process is shown in Appendix B.1.

Appendix B.2, on the other hand, shows the derivation of the mass moment of inertia for the spacecraft, where the design introduced in the previous section is used as reference, with the aid of some further simplifying assumptions.

B.1 | Shape

The spacecraft design developed during the literature study is shown in Fig. B.1, which is based on the design of MRO.



(a) *Isometric view.*

(b) $\hat{\mathbf{x}}_{\mathcal{B}} - \hat{\mathbf{y}}_{\mathcal{B}}$ *plane, i.e., top view.*

(c) $\hat{\mathbf{x}}_{\mathcal{B}} - \hat{\mathbf{z}}_{\mathcal{B}}$ *plane, i.e., side view (rotated by −90 deg).*

(d) $\hat{\mathbf{y}}_{\mathcal{B}} - \hat{\mathbf{z}}_{\mathcal{B}}$ *plane, i.e., front view.*

Figure B.1: *Discretization of the design of Mars Reconnaissance Orbiter, from a 3D view and three perpendicular points of view. Note that the positive direction of the axes is given by the arrows in the figure.*

In these images, one can see that the shape of MRO is simplified with the use of three main components:

- **Solar Array**: the two solar panels are modeled as flat rectangular plates, that extend for 4.75 m and with a height equal to the bus height, such that the total area is 12 m² (representing the 10 m²[1] of MRO's solar array surface, plus 2 m² of extra material to account for hinges and other supports)
- **Antenna**: the antenna is modeled as a spherical cap with a diameter of 3 m[2], which given the small cap extent is a good approximation to a paraboloid
- **Bus**: the bus is represented as a cube, whose edges are 2.5 m long, such that the total frontal area approximates the real MRO value

To summarize, the dimensions of the spacecraft are:

$$L_{\text{bus}} = 2.5\ \text{m} \qquad L_{\text{SA}} = 4.75\ \text{m} \qquad D_{\text{ant}} = 3\ \text{m}$$

This results in a surface area vector of:

$$\mathbf{A}|_{\mathcal{B}} = [37.5, 6.55, 7.20]^{\mathrm{T}}\ \ \text{m}^2$$

where the values correspond to the cross-sectional area along each body axis. Furthermore, the modeled dry mass and mass moment of inertia of the spacecraft are:

$$m_{\text{SC}} = 1000\ \text{kg} \qquad \mathbf{J}_{\text{SC}} = \operatorname{diag}(5750, 1215, 5210)\ \text{kg m}^2$$

respectively, where the inertia matrix is derived in the following section. Another interesting characteristic to look at is the location of the center of mass of the spacecraft. Its position, w.r.t. the center of the bus is:

$$\mathbf{r}_{\text{CM}} = [0, 0, 0.1375]^{\mathrm{T}}\ \ \text{m}$$

which is also derived in the next section, i.e., Appendix B.2.

Given REQ-SC-4, the spacecraft will be equipped with eight vectorial thrusters. These are assumed to be placed at the corners of the bus structure. As mentioned in the introduction of this appendix, however, the model shown in Fig. B.1 had to be modified for it to be used in SPARTA. For starters, SPARTA requires the shape to be watertight, which translates to saying that all the edges defined must be part of exactly two triangles, and that every point needs to be part of at least two triangles. With this in mind, it was clear that the solar array and the antenna needed to be modified. Hence first, the solar array was transformed to a very elongated and thin box (i.e., the flat plate was given some thickness, 5 cm to be precise), and the sides of the bus were divided in three sections to accommodate for the new solar panel design and to make sure that the solar panel edges were connected to the main body.

Furthermore, the antenna had to be modified. This is because the spherical cap from Fig. B.1 was only touching the surface, and there was no real link between it and the bus. To simplify the modeling and the analysis, the antenna was replaced with a thin box the same length of the bus, and with a height of 3 m, to achieve an antenna area of roughly the same value (7.5 m² compared to the previous 7.1 m²).

The modifications explained above, resulted in the final simplified triangular discretization of Fig. B.2.

One will notice another slight difference between Figs. B.1 and B.2, other than the discretized shapes. The triangular panels in the model for the aerothermodynamic analysis have been reduced in size, by adding two extra wedges per plate. This modification was done to more accurately model the aerodynamic moment coefficients. In fact, when the model with only two triangles per face was used in the SPARTA analysis, the moments generated were not representative of the real configuration.

B.2 | Mass Moment of Inertia

In this section, the moment of inertia of the spacecraft will be computed. For this, a few assumptions have been made:

- **Spacecraft Design**
  The design was already introduced in Appendix B.1, and in particular by Fig. B.1. Here it was shown that the spacecraft is discretized with three main shapes: a cube for the bus, two flat rectangular plates for the solar panels and a spherical cap for the antenna.

[1] Jet Propulsion Laboratory: Mars Reconnaissance Orbiter: Spacecraft Parts: Electrical Power – https://goo.gl/bv6Fnk [last accessed on February 15, 2018]

[2] Jet Propulsion Laboratory: Mars Reconnaissance Orbiter: Spacecraft Parts: Antennas – https://goo.gl/rPuYFk [last accessed on February 15, 2018]

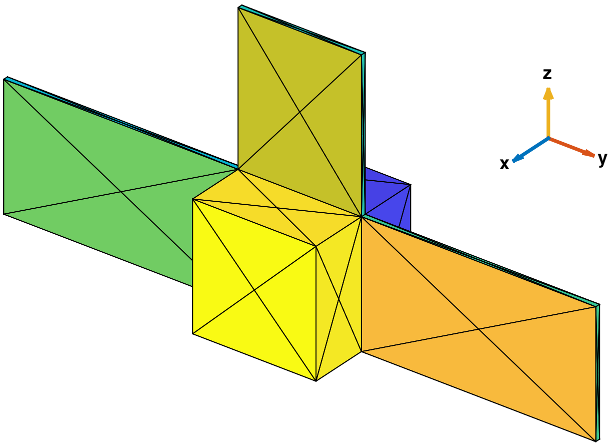


Figure B.2: *Discretization of the design of Mars Reconnaissance Orbiter, for the aerothermodynamic analysis.*

- **Design Simplifications**
  To make this analysis simpler, the spherical cap is approximated with a flat disk with diameter 3 m, located above the centroid of the bus. This assumption simplifies the analysis, since computation of the inertia and center of mass of a spherical cap give rise to more complicated equations.
- **Mass Distribution**
  It is assumed that all four spacecraft components have uniform mass distributions. Furthermore, the mass is spread among the component in the following manner:

$$m_{\text{bus}} = 65\,\%\, m_{\text{SC}} \qquad m_{\text{SA}} = 15\,\%\, m_{\text{SC}} \qquad m_{\text{ant}} = 5\,\%\, m_{\text{SC}}$$

- **Reference Axes**
  At first, the first set of reference axes are centered at the center of mass of the bus. The directions are the same as defined in Fig. B.1. After the center of mass of the whole spacecraft is computed, the reference axis will be moved the this point. This will give rise to the $\mathcal{B}$-frame.

The moment of inertia of a composite object, is found with the parallel axis theorem:

$$J_{ii} = \sum_k \left(J_{ii,\text{CM},k} + m_k L_{ii,k}^2\right) \tag{B.1}$$

where ii is the axis of rotation, $J_{ii,\text{CM}}$ the moment of inertia around the center of mass and L_{ii} the distance to the reference axes. With this in mind, the total inertia can be found. But first, one needs to know the $J_{ii,\text{CM}}$ terms. Using Myers (1962) as reference, the moments of inertia for each shape, around each axis are given by:

- **Cube**, with m_{bus} the mass and L_{bus} the side length:

$$J_{xx,\text{bus}} = J_{yy,\text{bus}} = J_{zz,\text{bus}} = \frac{1}{6} m_{\text{bus}} L_{\text{bus}}^2$$

- **Flat Plate**, with m_{SA} the mass and L_{SA} and L_{bus} the long and short sides, respectively:

$$J_{xx,\text{SA}} = \frac{1}{12} m_{\text{SA}} \left(L_{\text{SA}}^2 + L_{\text{bus}}^2\right) \qquad J_{yy,\text{SA}} = \frac{1}{12} m_{\text{SA}} L_{\text{bus}}^2 \qquad J_{zz,\text{SA}} = \frac{1}{12} m_{\text{SA}} L_{\text{SA}}^2$$

- **Disk**, with m_{ant} the mass and D_{ant} the diameter:

$$J_{xx,\text{ant}} = \frac{1}{8} m_{\text{ant}} D_{\text{ant}}^2 \qquad J_{yy,\text{ant}} = J_{zz,\text{ant}} = \frac{1}{16} m_{\text{ant}} D_{\text{ant}}^2$$

One can then find the position of the center of mass. Since the spacecraft is symmetric about the $\hat{\mathbf{x}}_{\mathcal{B}} - \hat{\mathbf{z}}_{\mathcal{B}}$ and $\hat{\mathbf{y}}_{\mathcal{B}} - \hat{\mathbf{z}}_{\mathcal{B}}$ planes, only the z_{CM} value is non-zero. Its value can be found with:

$$z_{\text{CM}} = \frac{\sum_k m_k L_k}{\sum_k m_k} = \frac{1}{2} \frac{m_{\text{ant}}}{m_{\text{SC}}} \left(L_{\text{bus}} + D_{\text{ant}}\right) \tag{B.2}$$

which leads to:

$$z_{\text{CM}} = 0.1375\,\text{m}$$

Finally, the moment of inertia about each axis is given by:

$$
\begin{aligned}
J_{xx} = J_{xx,\text{bus}} + m_{\text{bus}} z_{\text{CM}}^2 + 2 \left[J_{xx,\text{SA}} + \frac{1}{4} m_{\text{SA}} \left(L_{\text{bus}} + L_{\text{SA}} \right)^2 \right] + \\
+ J_{xx,\text{ant}} + m_{\text{ant}} \left[\frac{1}{2} \left(L_{\text{bus}} + D_{\text{ant}} \right) - z_{\text{CM}} \right]^2
\end{aligned}
\tag{B.3a}
$$

$$
J_{yy} = J_{yy,\text{bus}} + m_{\text{bus}} z_{\text{CM}}^2 + 2 J_{yy,\text{SA}} + J_{yy,\text{ant}} + m_{\text{ant}} \left[\frac{1}{2} \left(L_{\text{bus}} + D_{\text{ant}} \right) - z_{\text{CM}} \right]^2 \tag{B.3b}
$$

$$
J_{zz} = J_{zz,\text{bus}} + 2 \left[J_{zz,\text{SA}} + \frac{1}{4} m_{\text{SA}} \left(L_{\text{bus}} + L_{\text{SA}} \right)^2 \right] + J_{zz,\text{ant}} \tag{B.3c}
$$

Since the axes are chosen such that there is symmetry, the off-diagonal elements of the inertia matrix $\mathbf{J}$ are zero. Finally, the inertia matrix has value:

$$
\mathbf{J}_{\text{SC}} = \operatorname{diag} \left(5750, 1215, 5210 \right) \text{ kg m}^2 \tag{B.4}
$$

C
Mars Characteristics

In this appendix, some of the characteristics of Mars are given. First, Appendix C.1 shows some physical characteristics of the planet and its orbit, then the atmosphere of Mars is analyzed in more detail. As it became clear in Chapters 4 and 5, modeling of the atmospheric parameters is fundamental to achieve a faithful simulation and especially, to know what conditions to expect.

First, a description of the exponential atmosphere of Mars is given (see Appendix C.2). Then, the reader is presented in Appendix C.3, with some reference results of the Mars Climate Database, based on its web interface[1].

C.1 | Physical and Orbital Properties

The main bodies treated in this report are the Sun and Mars. Some of their characteristics are summarized in Table C.1.

Table C.1: *Physical and orbital characteristics of the main solar system bodies treated in this report (Lissauer and De Pater, 2013).*

	Gravitational Parameter [10^{12} m^3 s^{-2}]	**Mean Radius** [10^6 m]	**Bond Albedo** [−]	**Effective Temperature** [K]	**Central Body**	**Mean Orbital Distance** [10^9 m]
Sun	1.327×10^8	695.700	NA	5777	NA	NA
Mars	4.282×10^1	3.390	0.25	210	Sun	227.9

C.2 | Exponential Atmosphere Properties

The atmospheric properties used for the definition of the exponential atmospheric model of Mars, are shown in Table C.2. Moreover, a summary of the composition of the Martian atmosphere is given in Table C.3, where 'traces' is used if a compound is present in less than 1 ppm.

The value of gas constants R from Table C.2, is computed by knowing the atmospheric composition of the planet and the relation of R to the universal gas constant. Using the values of molecular weight of each of the molecules in Table C.3, one can find the gas constant of the atmosphere with:

$$R = \frac{\mathscr{R}}{\mathscr{M}} \tag{C.1a}$$

Table C.2: *Exponential atmosphere parameters (Spohn et al., 2014). The values shown are the average values at the reference altitude $h_0 = 0$ m.*

Scale Height [10^3 m]	**Density** [kg m^{-3}]	**Gas Constant** [J kg^{-1} K^{-1}]	**Temperature** [K]	**Gravitational Acceleration** [m s^{-2}]
11.1	0.020	197	215	3.73

[1] Mars Climate Database v5.3: The Web Interface – https://goo.gl/WbaFDt [last accessed on March 8, 2018]

[2] Lenntech Water Treatment Solutions: Molecular Weight Calculator – https://goo.gl/hnadmn [last accessed on November 7, 2017]

Table C.3: *Atmospheric composition of Mars (Lissauer and De Pater, 2013) and molar masses of respective gases*[2]*, for the exponential atmosphere model.*

		Ar	CH_4	CO	CO_2	H_2O	N_2	O_2	O_3
Composition	[%]	1.90	traces	0.07	96.00	0.01	1.90	0.13	traces
Molar Mass	[g mol^{-1}]	39.95	16.04	28.01	44.01	18.02	28.01	32.00	48.00

where the molar mass $\mathscr{M}$ follows from:

$$\mathscr{M} = \sum_i f_i \mathscr{M}_i \tag{C.1b}$$

In Eq. (C.1b), f_i is the fraction of molecule i (given by e.g., Table C.3) with molar mass $\mathscr{M}_i$. Molar masses of the elements in Table C.3 are shown in the same table. Moreover, $\mathscr{R}$ is the universal gas constant ($\mathscr{R} = 8.3145$ J mol^{-1} K^{-1}).

With atmosphere composition as in Table C.3, the specific heat of Mars becomes:

$$\gamma = 1.3$$

C.3 | Mars Climate Database

In Subsection 4.3.2, the Mars Climate Database, or MCD, was introduced. In this subsection, some of the results of the MCD are shown for density, wind and speed of sound, respectively. These values will be shown for a range of altitudes and times of year. Altitudes are defined w.r.t. the mean Areoid, i.e., above 'mean sea level', and are expressed in kilometers. Time of year is expressed in terms of solar longitude[3]. This is defined as the Mars-Sun angle, and whose zero value corresponds to the spring equinox. Figure C.1 gives a geometrical representation of this angle.

The range of altitudes shown is:

50 to 1500 km

above the Martian mean sea level, with 50 slices and logarithmic spacing, whereas the range of solar longitudes is:

- **0 deg**: northern hemisphere spring equinox and *end* of dust storm season
- **90 deg**: northern hemisphere summer solstice
- **180 deg**: northern hemisphere autumn equinox and *beginning* of dust storm season
- **270 deg**: northern hemisphere winter solstice and *middle* of dust storm season

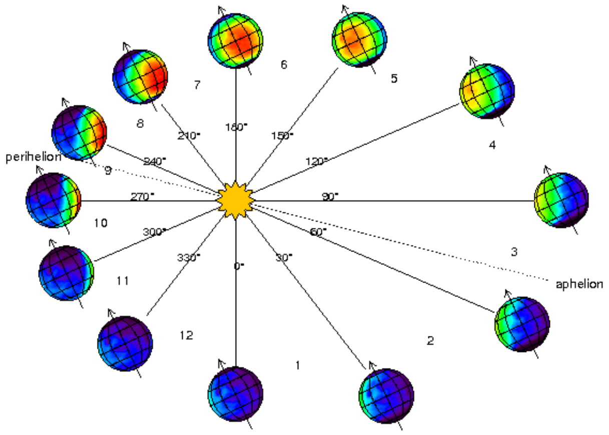


Figure C.1: *Definition of Martian solar longitude and example positions (Footnote (3)).*

[3] The Mars Climate Database Projects: Martian Seasons and Solar Longitude – https://goo.gl/YcDi4e [last accessed on March 7, 2018]

Dust storms are very common during the dust storm season, and they can give rise to large increases in density. More importantly, these increases in density occur quickly, over just a few orbits (Prince et al., 2009).

Further assumptions for the modeling are that the data is extracted at the local solar time of 0 h at the zero-longitude line and with average solar flux conditions.

Note that in the plots, the values of altitudes shown is restricted to these six values:

50, 100, 150, 250, 500 and 1500 km

C.3.1 | Density

Figure C.2 shows density values for the range of altitude and longitudes described above.

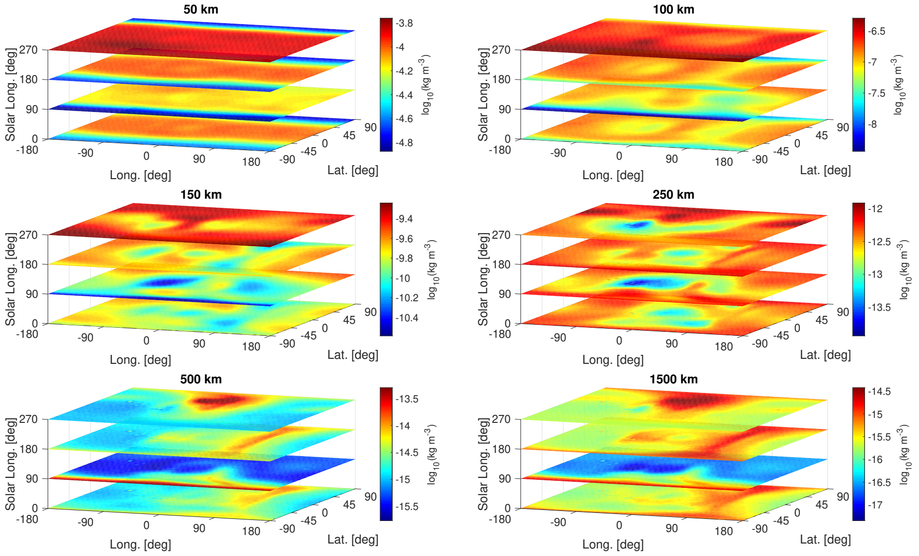


Figure C.2: *Density profile for various altitudes and solar longitudes, based on the Mars Climate Database. Note that the density values are expressed with a logarithmic scale.*

From the plots, it is clear that there are large seasonal changes in density. This is especially clear in the lower layers (between 100 and 150 km), where the poles experience excursions in density of up to two orders of magnitude (note how this is especially true for the poles).

Following the discussion about atmosphere models in Subsection 4.3.2, the chosen atmosphere representation is a tabulated atmosphere averaged over solar longitudes. For density, this gives rise to the plots in Fig. C.3.

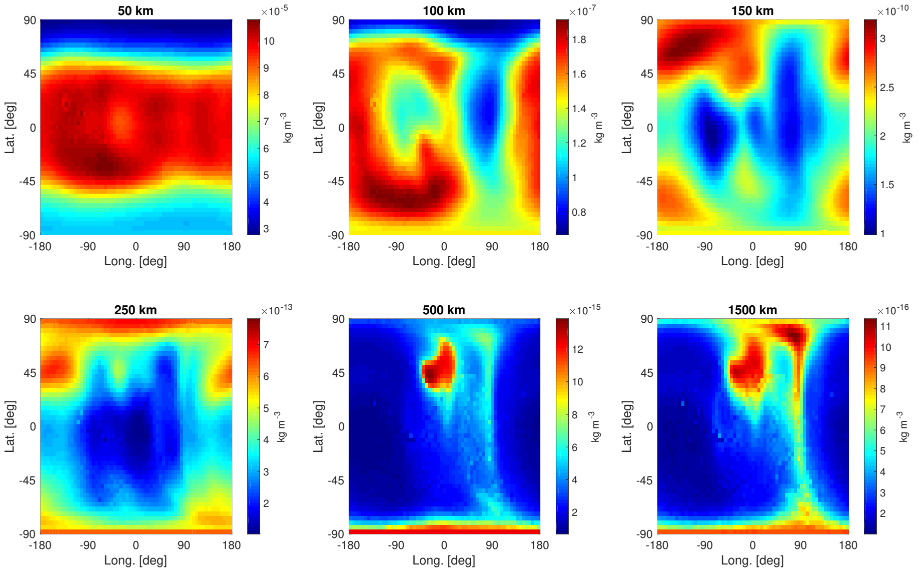


Figure C.3: *Average density values for various altitudes, based on the Mars Climate Database.*

The way this model is used during propagation, is by interpolating to find the density at the desired latitude, longitude and altitude combination. Then, the randomized perturbation from Eq. (4.30) is added (or rather multiplied) to account for variability of the atmosphere over time. The addition of a randomized layer of perturbations to Fig. C.3, results in the plots in Fig. C.4.

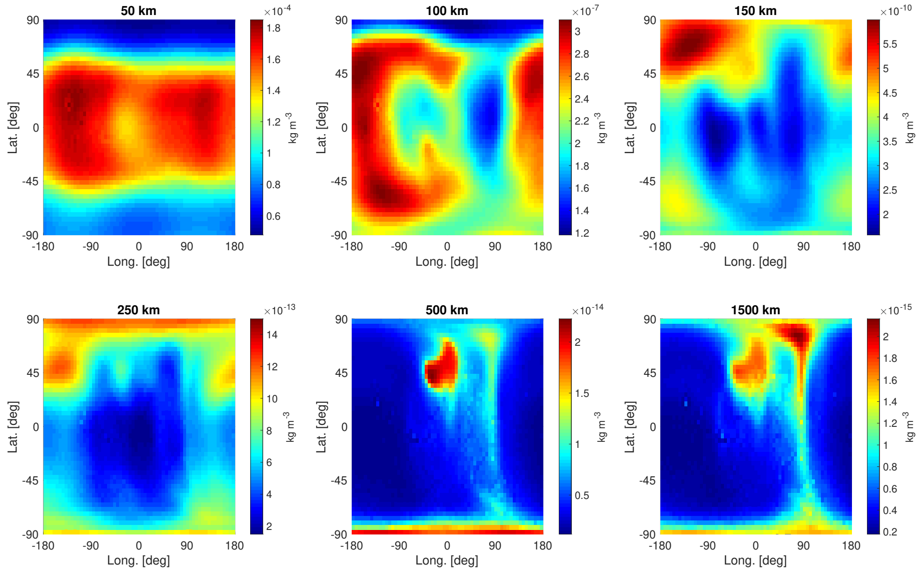


Figure C.4: *Average density values for various altitudes, based on the Mars Climate Database, where a layer of random perturbations is added to account for atmospheric variability.*

For sake of completeness, in the line below, one can find the values of the coefficients a, b, c and d from Eq. (4.30) which give rise to Fig. C.4:

$$a = 1.71415 \qquad \mathbf{b} = \begin{pmatrix} 0.0272539 \\ -0.146923 \\ 0.0475874 \end{pmatrix} \qquad \mathbf{c} = \begin{pmatrix} 0.110777 \\ -0.055571 \\ 0.0165448 \end{pmatrix} \qquad \mathbf{d} = \begin{pmatrix} 1.50714 \\ 0.478782 \\ -0.297889 \end{pmatrix}$$

Thus, one can see that the overall shape of the atmosphere density distribution is roughly the same. The main difference is that some areas have slightly larger or smaller concentrations in density.

One can also look at the evolution of the density multiplication factor over time. This is done in Fig. C.5.

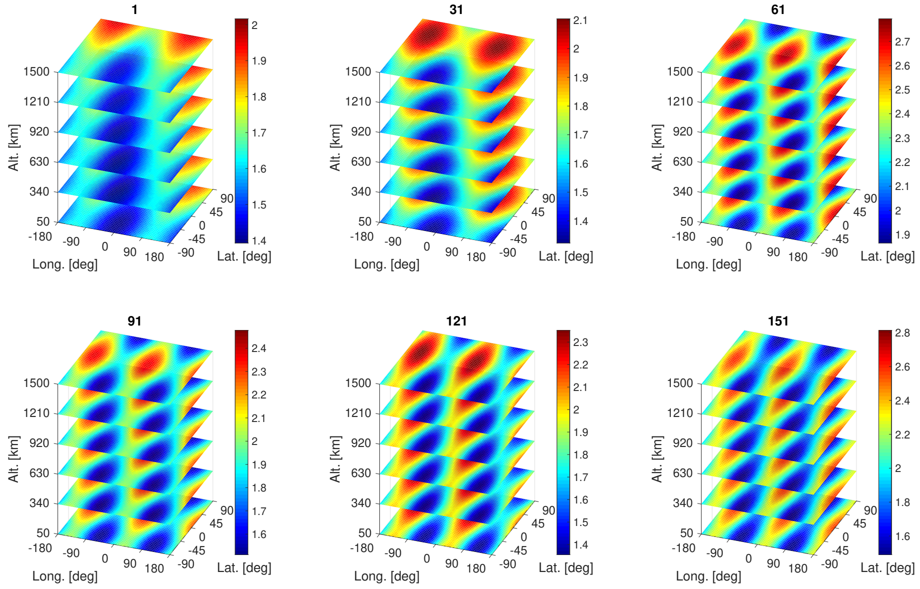


Figure C.5: *Evolution of the density multiplication factor κ_ρ over 150 orbits, based on the random walk described by Eq. (4.31) and Table 4.1, of the coefficients of Eq. (4.30).*

C.3.2 | Wind

Figure C.6 shows wind values for the same range of altitude and longitudes as before. Note that here, unlike the previous example, the various plots show different times, rather than different altitudes.

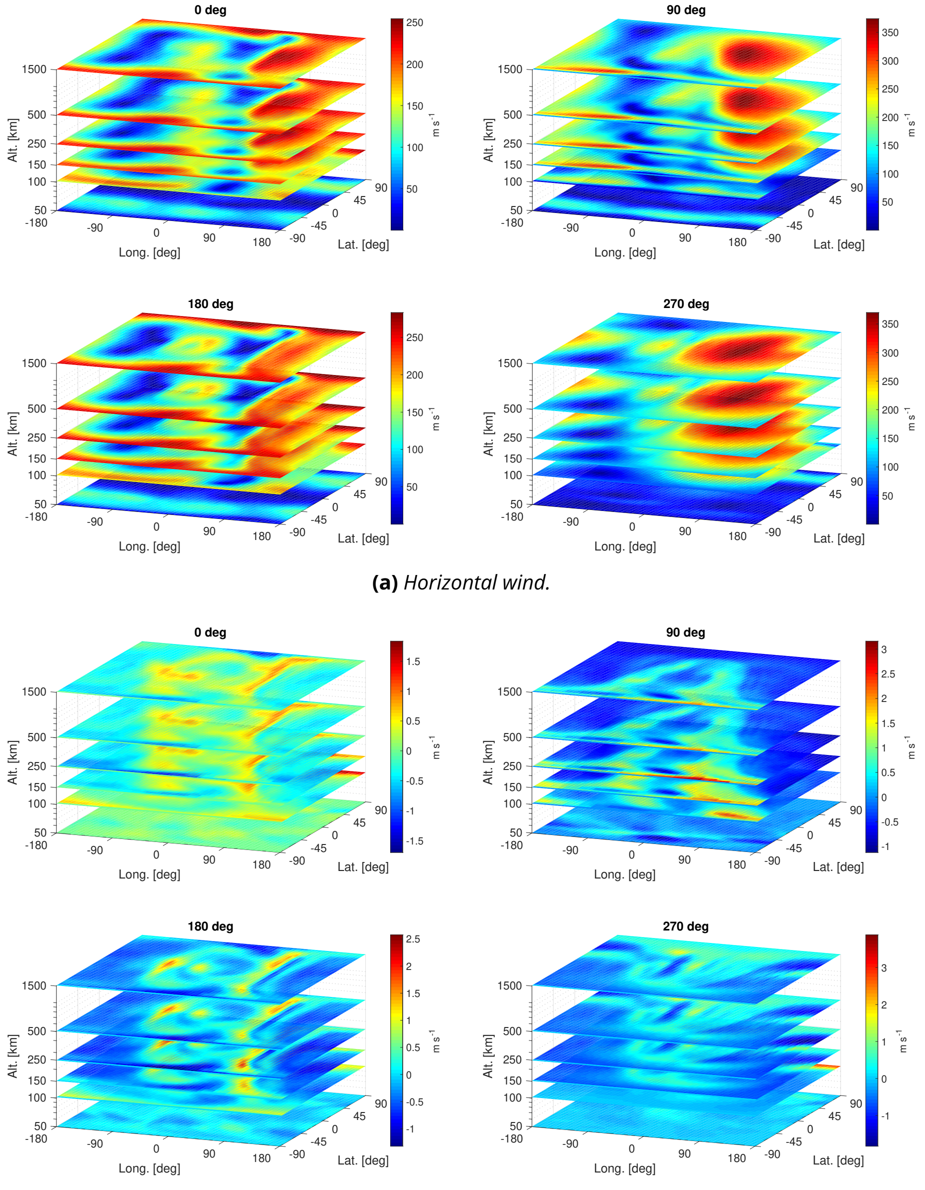
In this figure, in particular, both the vertical and horizontal winds are displayed. As one can see, the vertical winds, shown by Fig. C.6b, have values always smaller than 5 m s^{-1}. As such, they can be neglected.

For the horizontal winds, which can be found in Fig. C.6a, a more detailed analysis needs to be carried out. First of all, one needs to know the magnitude of the (inertial) velocity of the spacecraft at the given altitudes. Assuming circular orbits, the values of the velocities at the altitudes from Fig. C.6 are given by (Curtis, 2013):

$$V = \sqrt{\frac{\mu}{r}} \tag{C.2}$$

with the gravitational parameter from Table C.1. Now, the effect of rotation of the atmosphere and of the winds need to be taken into account. The most likely scenario that will be encountered can be described as:

- **High inclination**: MRO has an inclination of approximately 93 deg (Lyons, 2002), which means that it moves almost vertically, in a slightly retrograde orbit; for sake of simplicity a 90 deg inclination is assumed here



(a) *Horizontal wind.*

(b) *Vertical wind. Note that negative values correspond to upward winds.*

Figure C.6: *Wind profile for various solar longitudes and altitudes, based on the Mars Climate Database.*

- **Mars rotation**: Mars, like Earth, rotates around its axis in a counterclockwise direction, as seen from the north pole; this can be modeled as a wind blowing from west to east
- **East-west winds**: the wind blows from east to west, which is the case for most of the time (although direction is not shown in Fig. C.6a), especially in non-polar regions

Thus, the rotation of the atmosphere (which follows the planet's rotation) and the wind, seem to produce opposing vectors. An analysis of their magnitude can give an idea of how large their influence is. This analysis is carried out by means of Table C.4, where focus is put on altitudes below 500 km.

Table C.4: *Circular velocity for a spacecraft at various altitudes, and corresponding atmosphere and horizontal wind velocities. Note that maximum and root mean squared values are computed over latitude, longitude and solar longitude.*

Altitude	**Circular Velocity**	**Atmosphere Rotational Velocity**	**Maximum Horizontal Wind**	**RMS Horizontal Wind**
[km]	[m s^{-1}]	[m s^{-1}]	[m s^{-1}]	[m s^{-1}]
50	3530	243	190	72
100	3500	247	344	140
150	3480	251	368	177
230	3440	256	374	181
370	3380	266	374	181
500	3320	275	374	181

In this table, one can see the circular velocity, found with Eq. (C.2), the atmosphere rotational velocity, found with:

$$V_{\text{atm}} = \omega r \tag{C.3}$$

and the maximum and root mean square (RMS) values of the wind. The effect of wind and atmosphere rotation, then results in a heading error of at most ± 2 deg and a velocity increase of less than 6 m s^{-1}. Considering the preliminarily of this study, the effect of wind can therefore be neglected.

Note that in case no wind were present (i.e., only atmosphere rotation), the effect could be quantified as a heading error of 4 deg and a velocity increase of less than 10 m s^{-1}.

C.3.3 | Speed of Sound

The value of the speed of sound in Martian conditions is shown in Fig. C.7a. This value cannot be directly output by the web interface of the MCD, but is found via Eq. (5.1b), i.e.,

$$a = \sqrt{\gamma R T}\,,$$

where γ is the specific heat ratio, R is the gas constant, and T is the temperature. Of these three parameters, only the temperature is directly output by the MCD interface. The other two are derived from other model parameters. The value of γ can be found with (Lissauer and De Pater, 2013):

$$\gamma = \frac{1}{1 - \frac{R}{C_p}} \tag{C.4}$$

where C_p is the specific heat at constant pressure. Values of temperature and C_p can be found in Figs. C.7b and C.7c, respectively. In fact, the value of C_p is directly given by the database.

The value of the gas constant, on the other hand, is computed with Eq. (C.1a), where the molar mass is given by the composition from the next section.

From Fig. C.7a, one can see that the maximum value of the speed of sound is about 260 m s^{-1}. Looking back at Table C.4, it is clear that the velocity of a spacecraft in a circular orbit, is at least 13 times as high. This translates to having a Mach number M of at least 13. As was shown in Subsection 5.3.2, this means that the spacecraft is in hypersonic conditions, and thus, the decisions made in Chapter 5 are justified.

C.3.4 | Composition

The Mars Climate Database also provides information on atmospheric composition. All the molecules provided by MCD are shown in Table C.5, where the molecular mass and collision (or kinetic) parameter for each compound are also shown.

Table C.5: *Molar masses and collision diameters of most prevalent molecules in the Martian atmosphere (Sutton and Graves, 1971).*

		Ar	CO[(a)]	CO_2	H	H_2	H_2O[(a)]	He	N_2	O	O_2	O_3
$\mathscr{M}$	[g mol^{-1}]	39.95	28.01	44.01	1.01	2.02	18.02	4.00	28.01	16.00	32.00	48.00
σ	[pm]	354.20	376.00	394.10	282.70	282.70	265.00	255.10	379.80	346.70	346.70	346.70

[(a)] Taken from Breck (1974).

Note that for some of the allotropes of hydrogen and oxygen, no information on the collision diameter could be found. For this reason, the value for the diatomic values are used as reference.

The information provided by Table C.5 will be used to compute the required inputs for SPARTA. As a matter of fact, SPARTA requires the number density of the gas mixture as an input, which is given by Eq. (5.3c), i.e.,

$$n = \frac{N_A}{\mathscr{M}}\rho\,.$$

The value of density is known from the previous subsections and the Avogadro's number is a constant, thus the only unknown is the molar mass of the mixture. This can be found once again with Eq. (C.1b), where the gas fractions can be seen in Fig. C.8a. In this figure, the presence of gases is quantified as a fraction of the total and is averaged over latitude, longitude and solar longitude, giving a variation as a function of altitude. Note that, as shown in Fig. C.8b, the sum of the fractions differs from unity of at most 0.02 at around 250 km, where the composition is in a transition phase from $\mathrm{CO_2}$ to H dominance (and where monoatomic oxygen locally plays an important role).

Clearly, the composition of the atmosphere at the lower layers (below e.g., 100 km) is very different from the composition in the higher ones. If around 50 km the atmosphere is mainly $\mathrm{CO_2}$, at 1000 km it is mainly monoatomic hydrogen. Of course, at 1000 km the density of the air is also about ten orders of magnitude lower than at 50 km, but the change in molecular composition is still noticeable.

This can be seen by looking at the variation of particle number density over altitude, which is plotted in Fig. C.8c. The small kink present at 400 km is indeed caused by the drop of almost two orders of magnitude in molar mass, which on the other hand, is shown in Fig. C.8b.

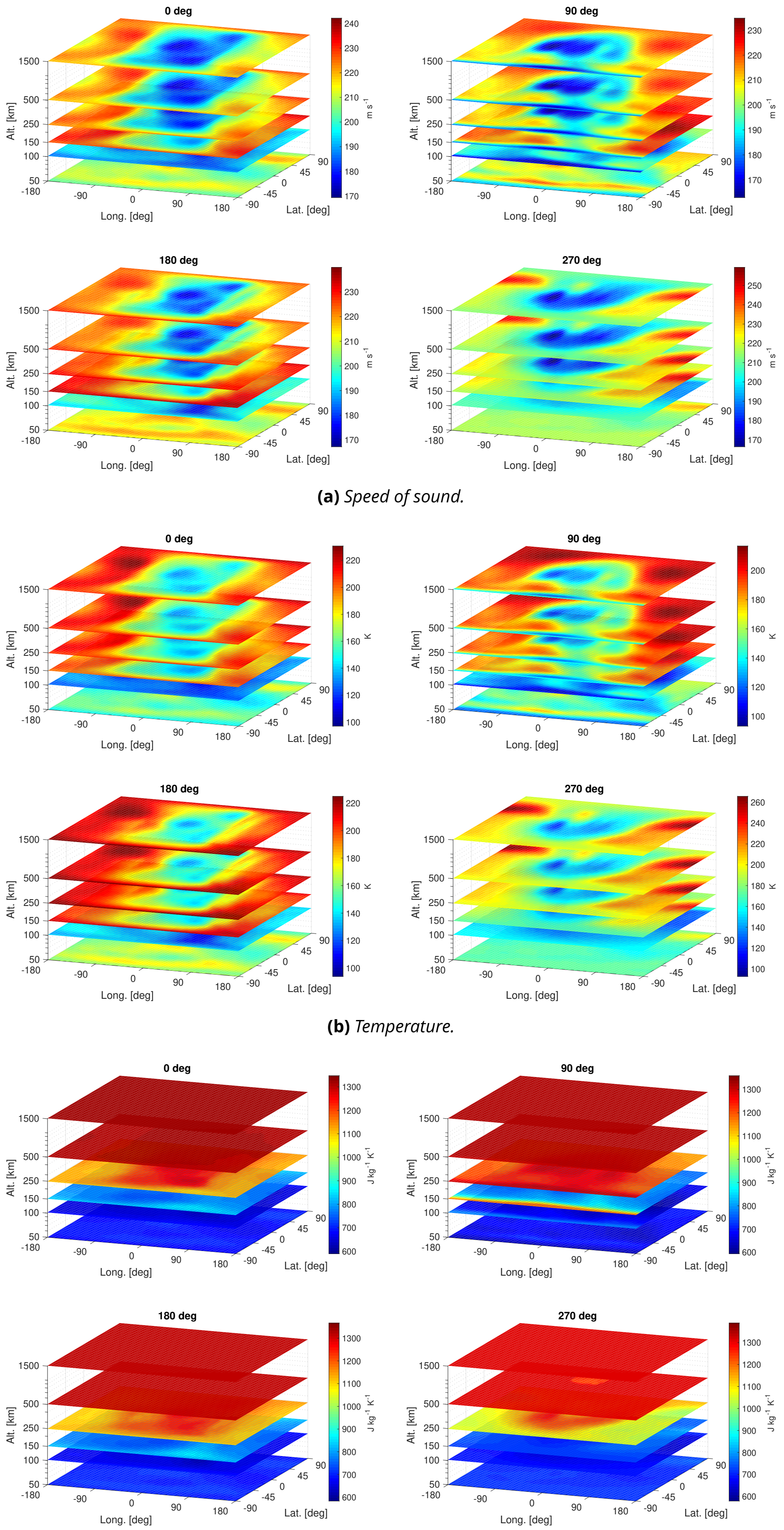
Finally, the value of the Knudsen number is also plotted, for sake of completeness. Clearly, at approximately 100 km the Knudsen number reaches the value of 0.01, meaning that the flow around the spacecraft classifies at most as transition flow. This justifies the modeling of both the continuum and rarefied regimes, since they are both needed to derive the transient aerodynamic coefficients.

C.3.5 | Considerations on Mars Climate Database

The main limitation of the MCD is its altitude restriction. According to González-Galindo et al. (2009), the highest altitude supported is about 250 km, and it corresponds to the beginning of the Martian thermosphere. Above this altitude, the values of density and pressure are extrapolated by assuming a hydrostatic vertical profile (i.e., assuming an exponential atmosphere, and by using Eq. (4.29), where the reference values are the ones corresponding to the lowest altitude in the database) (Forget et al., 2015). On the contrary, temperature is assumed to be constant above this limit (which is also a consequence of hydrostatic equilibrium). Thus, the accuracy of the values above the 250 km threshold is debatable. However, since density is already very small at this altitude, and considering that only a qualitative representation of the atmosphere is needed, the influence of the hydrostatic equilibrium assumption can be disregarded.

Interestingly, no information was found on how the gas composition is extrapolated at higher altitudes.

Another fact to mention is the problem that was encountered with computing the specific heat ratio, i.e., γ. As mentioned in the previous section, the atmosphere is mainly composed of H above 500 km (see Figs. C.8a and C.8b). Since monoatomic hydrogen has a very low molar mass (1.01 g mol^{-1} from Table C.5), its gas constant becomes very large (approximately 8230 J kg^{-1} K^{-1}). However, as was seen in Fig. C.7c, the specific heat at constant pressure is kept constant above 500 km (note how this differs from the altitude threshold from the previous paragraph), by reaching an average value of 1312.5 J kg^{-1} K^{-1}. Hence, when computing γ with Eq. (C.4), the equation outputs nonsensical (negative) results. Therefore, as soon as the concentration of H reaches 75 %, the specific heat ratio is given the ideal value of 1.667 (i.e., $^5/_3$).

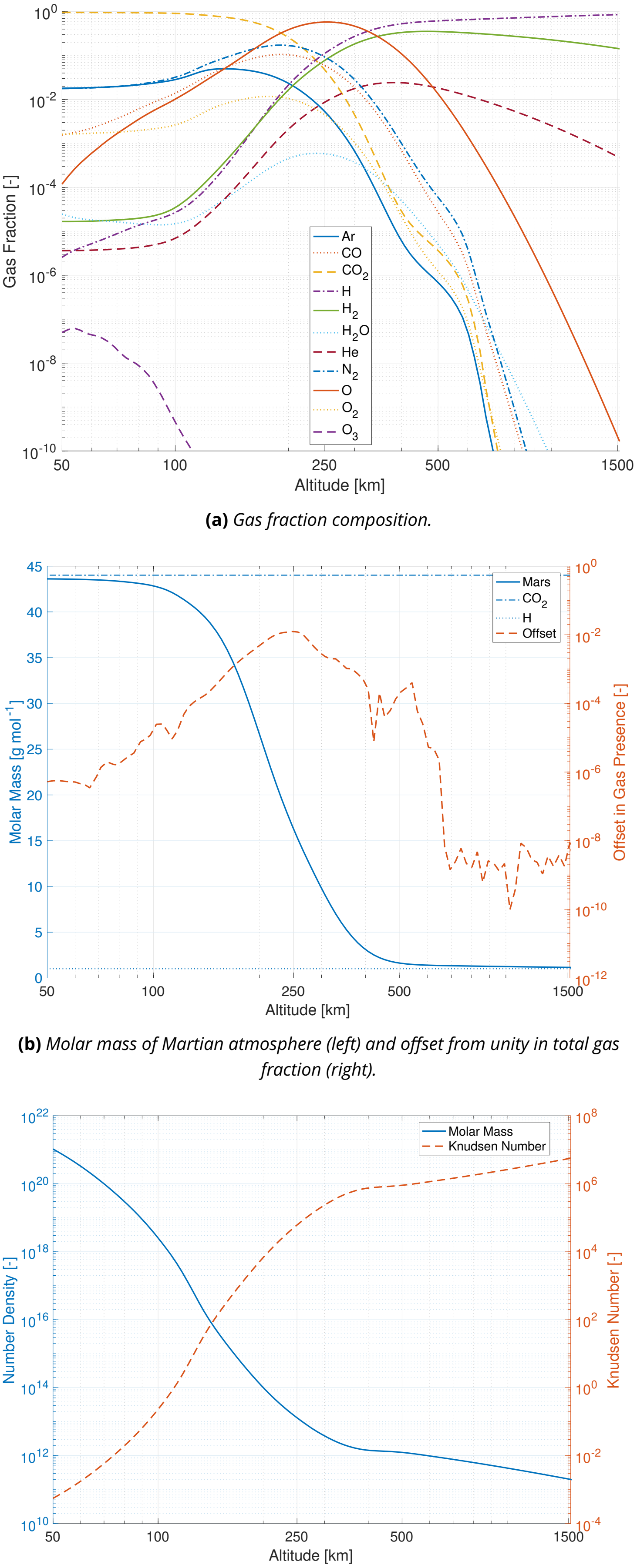


(a) *Speed of sound.*

(b) *Temperature.*

(c) *Specific heat at constant pressure.*

Figure C.7: *Speed of sound, temperature and specific heat at constant pressure values for various solar longitudes and altitudes, based on the Mars Climate Database.*



(a) *Gas fraction composition.*

(b) *Molar mass of Martian atmosphere (left) and offset from unity in total gas fraction (right).*

(c) *Number density (left) and Knudsen number (right).*

Figure C.8: *Time- and spatially-averaged atmospheric composition properties as a function of altitude, based on the Mars Climate Database.*

PAGE INTENTIONALLY LEFT BLANK

D
Aerodynamic Analysis

As mentioned in Subsection 5.3.1, SPARTA is the software used for computation of the aerodynamic coefficients in the rarefied flow. In the same section, the modified Newtonian method is explained, and it is used for the determination of the same coefficients in the continuum flow. In this appendix, details over the gas mixtures used for SPARTA and how the results of both methods were processed, will be given.

D.1 | SPARTA Modeling

To run a SPARTA simulation, four main files are needed:

- **Input File**: file used to define the simulation properties and parameters, and to run the simulation itself
- **Model File**: file where the discretized model (which for this application is described in Appendix B) is stored
- **Gas Species**: two files containing molecular and thermodynamics parameter of the gas species to be used in the simulation of the environment

The first and second files are explained very well in the SPARTA documentation (Plimpton and Gallis, 2017). However, it can be quite the challenge to find values for the gas mixtures needed for the remaining two files. For this reason, in this appendix, a summary of the values found for the main Martian atmospheric gases, are given. First, however, the reader can find a few tips on how to set up the SPARTA simulation, which were first given by Günther March at the Faculty of Aerospace Engineering [(1)].

Some more parameters needed for the simulation, are the wall temperature and accommodation coefficient α of the spacecraft surface. These values were given the values of 300 K and 1.0, respectively, based on the aerothermodynamic analysis by Liechty (2007). In the same study, the author uses 150 K as the temperature of the gas mixture, but in the case of this thesis, the atmosphere temperature at the altitude considered is taken (which is indeed roughly 150 K).

Other simulation parameters are the ratio of real to simulated particles, and the grid spacing. The former was chosen to be 17.5, whereas the latter as 0.25 m. The choice was rather arbitrary for the grid spacing (in Appendix D.1.1, one will see that the ratio of real to simulated particles is based on Günther's recommendation). However, for both variables, another simulation with more accurate parameters was run, to see the effect it had on the result. A discussion on this can be found in Subsection 11.3.1.

D.1.1 | Set Up Tips

As mentioned in Subsection 5.3.1, SPARTA simulates the motion of fictitious particles, which represent a larger number of real particles. A total of 10 to 20 simulated particles per cell is generally used for free molecular flow simulations. Moreover, the simulation time step should be at most 10 % of the time it takes the particle to traverse the simulation box.

Also, when processing the output, make sure not to consider the first couple of output files, since the simulation takes a while to start-up. Other recommendations can be found in Appendix A of Hess (2016).

D.1.2 | Gas Species

Two files need to be input for the modeling of the wanted gas mixture. The first file described the gas species used, i.e., which molecules need to be included. In this file, the following parameters have to be listed: molecular weight and mass, rotational and vibrational degrees of freedom, inverse rotational and vibrational relaxation number and

[(1)] Personal communication – March 26, 2018

Table D.1: *More thermochemical parameters for molecules making up the Martian atmosphere (Bird, 2013).*

		Ar	CO	$\mathrm{CO_2}$	H	$\mathrm{H_2}$	$\mathrm{H_2O}$	He	$\mathrm{N_2}$	O	$\mathrm{O_2}$	$\mathrm{O_3}$
RDOF	[–]	0	2	3	0	2	3	0	2	0	2	NA
IRRN	[–]	0.0	NA	0.4[c]	0.0	NA	NA	0.0	0.2	0.0	0.2	NA
VDOF	[–]	0	2[a]	2[a]	0	NA	NA	0	1	0	2	NA
IVRN[a]	$[10^{-5}]$	0.0	5.78[b]	41.28[b]	0.0	NA	NA	0.0	1.90	0.0	5.59	NA
VT	[K]	0.0	3103.0	3339.0[d]	0.0	6159.0	NA	0.0	3371.0	0.0	2256.0	NA

RDOF: Rotational Degrees of Freedom
IRRN: Inverse Rotational Relaxation Number
VDOF: Vibrational Degrees of Freedom
IVRN: Inverse Vibrational Relaxation Number
VT: Vibrational Temperature
NA: Not Available

[a] Taken from Bird (1994).
[b] Computed with Eq. (D.2) (Bird, 1994). More information is provided at the end of this subsection.
[c] Taken from Carnevale et al. (1967).
[d] Taken from Gallis and Harvey (1998).

vibrational temperature. Description of the meaning of these parameters is deemed outside the scope of this thesis and the reader is referred to the references listed in this section for a description of the parameters.

Table D.1 shows a list of the parameters for the molecules present in the Martian atmosphere, according to the analysis done in Appendix C. Note that the molecular mass of the molecules was already given by Table C.5, whereas the molecular weight can be found by using:

$$m = \frac{\mathscr{M}}{N_A} \tag{D.1}$$

where $\mathscr{M}$ is the molar (or molecular) mass and N_A is Avogadro's number.

As can be seen from Table D.1, not all information for every gas has been collected. This is because of the lack of data on gases that are not predominant in Earth's atmosphere. In fact, most data can be found only about the allotropes of nitrogen and oxygen, which alone make up about 99 % of the Earth's atmosphere. However, recalling Fig. C.8a, the most prevalent gases (meaning those gases with a presence of at least 5 % at altitudes in the range 125 to 500 km, which is bound by the altitude above which $Kn > 10$, and by the atmospheric interface altitude) in the Martian atmosphere are: CO, $\mathrm{CO_2}$, H, $\mathrm{H_2}$, $\mathrm{N_2}$ and O. These are the compounds that will be used for the SPARTA analysis. In particular, the distribution over altitude of gases is found in Table D.2. Note that the values of gas fraction have been normalized to make sure that the sum adds up to 100 %.

Other parameters that are needed for the modeling of intermolecular collisions in SPARTA are based on the variable soft sphere (VSS) collision model. These values are, namely, the collision diameter, the angular scattering

Table D.2: *Distribution of gases over altitudes for SPARTA simulation, based on the Mars Climate Database.*

Altitude [km]	**Gases** [%]					
125	CO	$\mathrm{CO_2}$	H	$\mathrm{H_2}$	$\mathrm{N_2}$	O
	—	92	—	—	8	—
150	CO	$\mathrm{CO_2}$	H	$\mathrm{H_2}$	$\mathrm{N_2}$	O
	8	70	—	—	12	10
200	CO	$\mathrm{CO_2}$	H	$\mathrm{H_2}$	$\mathrm{N_2}$	O
	12	28	—	—	18	42
300	CO	$\mathrm{CO_2}$	H	$\mathrm{H_2}$	$\mathrm{N_2}$	O
	—	—	30	20	—	50
500	CO	$\mathrm{CO_2}$	H	$\mathrm{H_2}$	$\mathrm{N_2}$	O
	—	—	64	36	—	—

Table D.3: *VSS collision model parameters for each species (Bird, 1994).*

		Ar	CO	CO_2	H	H_2	H_2O	He	N_2	O	O_2	O_3
α	[−]	1.40	1.49	1.61	1.35	1.35	NA	1.26	1.36	1.40	1.40	1.40
ω	[−]	0.81	0.73	0.93	0.67	0.67	NA	0.66	0.74	0.77	0.77	0.77

parameter α and the temperature-dependence of viscosity ω. The first values was given by Table C.5, whereas the latter two are given in Table D.3.

Computation of Vibrational Relaxation Number

In Table D.1, some parameters of interest for a DSMC analysis are listed. Here, one of the criteria is the inverse vibrational relaxation number. This value was given by (Bird, 1994) for some of the gas species, but could not be found for the remaining gases.

What was found is the following equation, used to compute the vibrational relaxation number as a function of temperature (Bird, 1994):

$$Z_v = \frac{C_1}{T^\omega} \exp\left(\frac{C_2}{\sqrt[3]{T}}\right) \tag{D.2}$$

where Z_v is the vibrational relaxation number, T the reference temperature, ω the temperature-dependence of viscosity (see Table D.3 for the values) and C_1 and C_2, two coefficients dependent on the gas. The values of the coefficients were also given for some gases by Bird and can be found in Table D.4.

Table D.4: *Values of constant coefficient for a few Martian atmosphere gases, used to compute the vibrational relaxation number (Bird, 1994).*

		CO	N_2	O_2
C_1	[−]	37.7	9.1	56.5
C_2	[−]	175.0	220.0	153.5

Clearly, no coefficients for CO_2 are provided by Bird's book, but they are assumed to be the same as for CO. This assumption is made, since carbon dioxide is a major player in the Martian atmosphere, and proper modeling of this gas is of high importance. Of course, the actual values for this gas will be different than for CO, but at least they are given a non-zero value. Moreover, since it is known from Wright et al. (2009), that both CO and CO_2 have a short vibrational relaxation time, their vibrational relaxation numbers will have a low value. This is indeed what is seen in Table D.1, where the inverse of Z_v is shown, and it indeed shows a faster relaxation that, e.g., for O_2 and N_2.

Note that the value of temperature used in Eq. (D.2), is the value of vibrational temperature from Table D.1. Using the constants provided in this section for N_2 and O_2 does indeed result in the inverse vibrational relaxation numbers listed in the table.

D.2 | Output Processing

According to Anderson Jr. (2011), the process to determine the lift and drag coefficients is to:

1. compute the normal pressure (p) and shear stress (τ) distribution on the surface of the body (which is carried out by SPARTA)
2. find pressure (C_p) and skin friction (C_f) coefficients
3. integrate C_p and C_f to find the normal (C_n) and axial (C_a) force coefficients
4. use angle of attack (α) to project C_n and C_a onto the lift and drag directions, to obtain the lift (C_L) and drag (C_D) coefficients

Since the analysis carried out with SPARTA is done in three-dimensional conditions, the side force coefficient (C_S) will also be found. Moreover, the knowledge of aerodynamic moment coefficients is also necessary. Thus, their computation will also be considered.

As mentioned in the first point of the list above, SPARTA outputs values of normal pressure and shear stress for each surface element, expressed as vectors. The expression of $\mathbf{p}$ and $\boldsymbol{\tau}$ as vectors, simplifies the analysis, since one only needs to add the vector components to get the total normal and shear pressures acting on the spacecraft. However, before adding the vectors, one needs to convert them in pressure and skin friction coefficients (i.e., point two of the list above). This is done with (Anderson Jr., 2011):

$$\mathbf{C}_{p,i} = \frac{\mathbf{p}_i - p_\infty \hat{\mathbf{n}}_i}{q_\infty} \tag{D.3a}$$

$$\mathbf{C}_{f,i} = \frac{\boldsymbol{\tau}_i}{q_\infty} \tag{D.3b}$$

where $\hat{\mathbf{n}}$ is the vector normal to the i-th surface element. Now, the force coefficients can be determined, for each body axis (i.e., for each axis in $\mathcal{F}_\mathcal{B}$). This is done by multiplying the pressure and shear coefficients by the area fraction of the respective element surface, i.e. (Dirkx and Mooij, 2017):

$$\begin{pmatrix} C_a \\ C_t \\ C_n \end{pmatrix} = -\frac{1}{A} \sum_i \left(\mathbf{C}_{p,i} + \mathbf{C}_{f,i} \right) S_i \tag{D.4}$$

where S is the surface area of the i-th surface element, whereas A is the cross-sectional area along the $\hat{\mathbf{x}}_\mathcal{B}$ axis, whose value was given in Appendix B. Note how the transversal force coefficient (C_t) has also been added on the left-hand side of Eq. (D.6), to represent the force in the negative $\hat{\mathbf{y}}_\mathcal{B}$-direction.

Note that the negative sign in Eq. (D.6) is introduced due to the definition of aerodynamic forces. In fact, these forces are defined as positive along the negative $\mathcal{F}_\mathcal{A}$ axes.

Then, the aerodynamic force coefficients can be obtained, by rotating the force coefficient from the $\mathcal{B}$-frame to the $\mathcal{A}$-frame, i.e., by performing the inverse of the transformation from Eq. (A.9). Hence, one can find the drag, side and lift coefficients with:

$$\begin{pmatrix} C_D \\ C_S \\ C_L \end{pmatrix} = \begin{bmatrix} \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \\ -\cos\alpha\sin\beta & \cos\beta & -\sin\alpha\sin\beta \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{pmatrix} C_a \\ C_t \\ C_n \end{pmatrix} \tag{D.5}$$

Assuming the side-slip angle β is zero (which is the case for the SPARTA simulations), the side coefficient C_S equals C_t.

It should be pointed out, that to analyze the results from SPARTA, the frame transformation from Eq. (D.5) is not carried out. This is because SPARTA already outputs the normal pressure and shear stress vectors in the aerodynamic frame.

Finally, one can also compute the moment coefficients. This is done with an equation similar to Eq. (D.6), but where the distance to the center of mass is introduced (Dirkx and Mooij, 2017):

$$\begin{pmatrix} C_l \\ C_m \\ C_n \end{pmatrix} = -\frac{1}{A} \sum_i \left[\mathbf{r}_{\text{CM},i} \times \left(\mathbf{C}_{p,i} + \mathbf{C}_{f,i} \right) \right] S_i \tag{D.6}$$

The aerodynamic moment coefficients are then obtained in the same way as done for the force coefficients. Thus, they are rotated from the body to the aerodynamic frame (except that they are not).

E
Numerical Methods

The purpose of this chapter is to give an overview of the numerical methodologies used throughout this report and subsequent thesis work. Numerical routines are of extreme importance for any simulation-based research, since they can be applied to solve a wide range of problems. These routines are used to reduce the required computational workload and to apply theoretical processes to a real-world scenario.

In this chapter, the reader can find the topics listed below, together with the main reference used.

- Appendix E.1 on **Averaging**
- Appendix E.2 on **Integration**: (Montenbruck and Gill, 2005)
- Appendix E.3 on **Differentiation**: (Kiusalaas, 2005)
- Appendix E.4 on **Root-finding**: (Gerald and Wheatley, 2004)
- Appendix E.5 on **Interpolation**: (Kiusalaas, 2005)
- Appendix E.6 on **Curve Fitting**: (Gerald and Wheatley, 2004)
- Appendix E.7 on **Filtering**: (Vittaldev, 2010)

E.1 | **Averaging**

In Section 6.7, the atmosphere estimator of the navigation part of the GNC system was introduced. Here, it was stated that a moving average is used to combine the estimated parameters of the density formulation from seven consecutive atmospheric sweeps.

A moving average in this case, is defined as the average of the m previous data points. Assuming a data set is made up of n elements, the m-point moving average is given by:

$$\bar{x} = \frac{1}{m} \sum_{i=k}^{k-m} x_i \tag{E.1}$$

where k is the latest point and the symbol $\bar{\square}$ represents the average value. Clearly then, as soon as a new point $k+1$ is added to the list, the moving average is updated.

In case it is possible for the set to have less than m points, the following definition should be used, instead:

$$\bar{x} = \begin{cases} \dfrac{1}{n} \displaystyle\sum_{i=k}^{k-n} x_i & \text{if } m \geqslant n \\ \dfrac{1}{m} \displaystyle\sum_{i=k}^{k-m} x_i & \text{otherwise} \end{cases} \tag{E.2}$$

E.2 | **Integration**

In this section, methods for integration of simple functions in the form of $f(x)$ and of ordinary differential equations (ODE) in the form of $\dot{x} = f(x)$ will be given. One can find the former routines in Appendix E.2.1, and latter in Appendix E.2.2.

E.2.1 | Simple Functions

Numerical integration consists of approximating the definite integral of a function $f(x)$, by evaluating the function at a series of nodes, otherwise called x_i, multiplied by some factor, called weight w_i (Klees and Dwight, 2014):

$$\int_a^b f(x)\,\mathrm{d}t \approx \sum_{i=0}^{n} w_i f(x_i) \tag{E.3}$$

An example of this was shown in Eq. (6.19), where an integration is approximated with a summation called trapezoidal integration:

$$\int_a^b f(x)\,\mathrm{d}t \approx \frac{1}{2}\sum_{i=0}^{n-1} \left[f(x_i) + f(x_{i+1})\right](t_{i+1} - t_i) \tag{E.4}$$

With this formulation, the interval between a and b is divided into n intervals, where no regular spacing is necessary, since the value of h, the step size, is computed with every step.

In Section 7.2, the PID control method was introduced. Here, it was explained that integration of the attitude error is needed to compute the control torque. Since the time step for GNC is constant, Eq. (E.4) can be replaced with a more accurate formulation. Such a formulation is the based on Eq. (E.3), where the weights are given by (Press et al., 1992):

$$\frac{w_i}{h} = \begin{cases} 3/8 & \text{if } i = 0 \text{ or } i = n \\ 7/6 & \text{if } i = 1 \text{ or } i = n-1 \\ 23/24 & \text{if } i = 2 \text{ or } i = n-2 \\ 1 & \text{otherwise} \end{cases} \tag{E.5}$$

Whereas the trapezoidal rule has order $\mathcal{O}\left(h^3\right)$, the extended formulation from Eqs. (E.3) and (E.5), results in an integration with $\mathcal{O}\left(h^4\right)$.

E.2.2 | Ordinary Differential Equations

In most space applications, one needs to integrate a system of ordinary differential equations, which has the form:

$$\frac{\mathrm{d}\mathbf{x}(t, \mathbf{x})}{\mathrm{d}t} = \mathbf{f}(t, \mathbf{x}) \tag{E.6}$$

Here, $\mathbf{f}$ represents the system of differential equations and it is known, $\mathbf{x}$ is the state vector and t is the independent variable, i.e., time.

The most simple numerical integration method for ODEs is Euler integration. Euler integration is based on the first-order Taylor expansion of $\mathbf{x}$, and its mathematical representation is given by:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\dot{\mathbf{x}_n} \tag{E.7}$$

where n represents the current integration step, and h the integration step size. This method is very inaccurate (unless a very small time step is used), and over time other types of numerical methods have been introduced. For sake of simplicity, however, only the Runge-Kutta method was considered in this thesis.

Three Runge-Kutta methods have been used throughout the thesis work. These are:

- **Runge-Kutta 4**, or RK4, used as:
  - main integrator for the aerobraking simulation
  - integrator for the onboard filtering process
  - integrator for constant time step analysis of state variable performance in propagation
- **Runge-Kutta-Fehlberg 5(6)**, or RKF5(6), used as:
  - integrator for the onboard state propagation
  - integrator for variable time step analysis of state variable performance in propagation
- **Runge-Kutta-Fehlberg 7(8)**, or RKF7(8), used as integrator to compute the benchmark for the state variable performance analysis

The only method that will be explained in this section, is the RK4 integrator. The reader will find exhaustive descriptions of the two other integrators (which use a variable step size), in any numerical analysis literature.

The fourth-order Runge-Kutta method, or RK4, is similar to the Euler method, with the distinction that the number of function evaluations is increased to four: two at the extremes of the step and two at an intermediate point. The result is an integrator with global error in the order of $\mathcal{O}\left(h^4\right)$, and it can be expressed as:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{6}\left(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4\right) \tag{E.8a}$$

which is similar to a weighted average of four $\mathbf{k}$ coefficients, which are in turn computed as follows:

$$\mathbf{k}_1 = \dot{\mathbf{x}}\left(t_n, \mathbf{x}_n\right) \tag{E.8b}$$

$$\mathbf{k}_2 = \dot{\mathbf{x}}\left(t_n + \frac{h}{2}, \mathbf{x}_n + \frac{\mathbf{k}_1}{2}\right) \tag{E.8c}$$

$$\mathbf{k}_3 = \dot{\mathbf{x}}\left(t_n + \frac{h}{2}, \mathbf{x}_n + \frac{\mathbf{k}_2}{2}\right) \tag{E.8d}$$

$$\mathbf{k}_4 = \dot{\mathbf{x}}\left(t_n + h, \mathbf{x}_n + \mathbf{k}_3\right) \tag{E.8e}$$

E.3 | **Differentiation**

Numerical differentiation consists of computing the derivative of a function $f\left(x\right)$, by means of $f\left(x\right)$ itself only. Thus, numerical differentiation becomes particularly useful for those functions whose derivative is unknown or too hard to compute. The most common techniques for numerical differentiation are the so-called finite difference approximations. These are the ones treated in this report.

Finite differences are based, just like the Runge-Kutta methods for integration, on Taylor series expansions. In particular, for differentiation, one is interested in looking at both forward and backward expansions about x:

$$f\left(x+h\right) = \sum_{i=0}^{\infty} \frac{h^i}{i!} f^{(i)}\left(x\right) \tag{E.9a}$$

$$f\left(x-h\right) = \sum_{i=0}^{\infty} (-1)^i \frac{h^i}{i!} f^{(i)}\left(x\right) \tag{E.9b}$$

where $f^{(i)}\left(x\right)$ is the i-th derivative of $f\left(x\right)$ w.r.t. x. Based on the definitions above, one can define the three types of differentiation methods: central, backward and forward difference methods, based on which data is used. Considering the application of numerical differentiation in this report, i.e.:

- differentiation of the acceleration felt by the spacecraft for the system and measurements Jacobians from Chapter 6
- differentiation of the control error for computation of the control moment in Section 7.2

it is clear that for the first case, the central difference formulation can be used, since the acceleration function can be evaluated at any point. For the second case, in contrast, the only available information is from previous time steps and thus, the only method that can be used is the backward difference method.

Central Difference By subtracting the two formulas in Eq. (E.9), one can obtain the central difference approximation formula:

$$f^{(1)}\left(x\right) = \frac{f\left(x+h\right) - f\left(x-h\right)}{2h} + \mathcal{O}\left(h^2\right) \tag{E.10}$$

where the accuracy is in the order of $\mathcal{O}\left(h^2\right)$. Here, a constant time step h is used, following the decisions in Appendix E.2.2.

Backward Difference In case only data from x values smaller than the current value are available, one can use backward difference. This method only uses data from previous steps, and is given by:

$$f^{(1)}\left(x\right) = \frac{f\left(x-2h\right) - 4f\left(x-h\right) + 3f\left(x\right)}{2h} + \mathcal{O}\left(h^2\right) \tag{E.11}$$

MULTIPLE DIMENSIONS One might also be interested in derivatives of a system of functions of multiple variables, for instance for the computation of the aforementioned Jacobian. Differentiation, in this case, is no different. This is illustrated by:

$$f_i{}^{(1)}\left(x_j\right) = \frac{f_i\left(x_1, \ldots, x_j + h, \ldots, x_m\right) - f\left(x_1, \ldots, x_j - h, \ldots, x_m\right)}{2h} + \mathcal{O}\left(h^2\right) \tag{E.12}$$

for the central difference equivalent of Eq. (E.10). Note that in Eq. (E.12), the subscripts i and j refer to the i-th equation in the system and the j-th variable in the state vector. Finally, m is the number of state variables.

E.4 | Root-finding

Root-finding is used in this report for determination of the periapsis corridor boundaries and of the improved maneuver magnitude and in the periapse time estimator. The preferred root-finding algorithm is recursive bisection. This decision is motivated by the robustness of the method, the fact that no knowledge of the derivative of the root function is required (unlike for Newton-Raphson) and that it does not lead to divergence (in the cases analyzed), which is not necessarily true for fixed-point iteration.

The concept behind bisection is very simple, but very robust and reliable. Given an interval $[a, b]$, the bisection method can be used as shown in Algorithm 1. This method is based on Bolzano's theorem, which states that if $f(a)$ and $f(b)$ have opposite signs, there exists a $c \in [a, b]$ such that $f(c) = 0$[1]. Note that $f(x)$ needs to be continuous in $[a, b]$.

```
Data: function $f(x)$, interval $[a, b]$ and accuracy $\varepsilon$
Result: root of $f(x)$ inside interval $[a, b]$
if $f(a) f(b) \leqslant 0$ then
    while $|b - a| \geqslant \varepsilon$ do
        compute mid point: $c = \frac{a+b}{2}$;
        if $f(a) f(c) \leqslant 0$ then
            root is in $[a, c]$;
            $b \leftarrow c$;
        else
            root is in $[c, b]$;
            $a \leftarrow c$;
        end
    end
else
    no single root exists in $[a, b]$
end
```

Algorithm 1: Bisection method.

Given a prescribed accuracy ε, one can compute the amount of iterations needed to converge with:

$$n = \log_2(e) \ln\left(\frac{\Delta}{\varepsilon}\right) \tag{E.13}$$

where Δ is the size of the original interval and e is Euler's number.

E.5 | Interpolation

Interpolation is used to estimate a function $f(x)$, based on a series of data points $(x_i, f(x_i))$. Interpolation differs from curve fitting, in the sense that an interpolating function $I(x)$ passes through all the points $(x_i, f(x_i))$, which are used to derive $I(x)$ in the first place. A curve fitting function, on the other hand, does not necessarily fulfill this requirement. For this reason, curve fitting is generally used for noisy data, whereas interpolation is used with exact data. The most famous and widely used curve fitting method is least-squares, and it is briefly discussed in Appendix E.6.

[1] WolframMathWorld: Bolzano's Theorem – https://goo.gl/XTuqau [last accessed on November 10, 2017]

In this thesis, interpolation is used, for instance, to determine the density at a certain location, based on tabulated data. Spline interpolation was chosen as the preferred method, due to its availability and its stability for the cases at hand, meaning that it does not suffer from high oscillations between interpolation nodes. In particular, cubic spline interpolation is the method of choice.

This interpolation technique is achieved by interpolating a cubic polynomial between two consecutive data points, and by setting the first and second derivatives at these two points, equal to the derivatives of the adjacent polynomials. The result is a function of class C^2, in contrast with a function of class C^∞, which would result from polynomial interpolation.

The procedure for cubic spline interpolation is quite lengthy and complex, and for this reason is not treated here. One can find detailed derivations and descriptions in Gerald and Wheatley (2004), Kiusalaas (2005).

E.6 | **Curve Fitting**

Whereas interpolation is used to connect some data points with bridging functions, curve fitting is used to estimate the shape of a function (thus its coefficients), given its general form. Curve fitting is generally useful with noisy data, where interpolation would result in a very oscillatory or zig-zag behavior. For instance, curve fitting is used for the estimation of the atmospheric parameters, by the atmosphere estimation. In fact, an example with exponential functions will be given at the end of this section.

E.6.1 | Linear Least Squares

The most common type of curve fitting is least squares. This method minimizes the sum of the square of the distances between the data points and the fitting curve. The result is an optimal value of coefficients.

The mathematical formulation of the concept detailed above is the following. For a linear problem, the optimal coefficients can be found by solving the system:

$$\hat{\mathbf{x}} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{y} \tag{E.14a}$$

when no weighting parameters have been specified, whereas one can use:

$$\hat{\mathbf{x}} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{W}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{W}\mathbf{y} \tag{E.14b}$$

when a weighing matrix $\mathbf{W}$ is defined. Note that the symbol $\hat{\square}$ represents the estimated value. In Eq. (E.14), $\mathbf{A}$ is the design matrix and $\mathbf{y}$ is the observations vector, i.e., the data points. The design matrix is computed from:

$$\mathbf{A} = \left[\frac{\partial f\left(\mathbf{x}, \mathbf{u}_i\right)}{\partial \mathbf{x}}\right]_{\forall i} \tag{E.15}$$

Here, the vector $\mathbf{x}$ represents the coefficients and the vector $\mathbf{u}$ is the independent variable. The result of Eq. (E.15) is a matrix with the partial derivatives of f w.r.t. the m elements of $\mathbf{x}$ along the columns, where each row represents a different instance of independent variable $\mathbf{u}$. In case n instances of $\mathbf{u}$ are available, the final result is a matrix of size $m \times n$. Furthermore, f is the general form of the function which describes the system.

E.6.2 | Non-linear Least Squares

In case of a non-linear function f, a very similar procedure can be used. In particular, Eq. (E.14a) becomes:

$$\mathrm{d}\hat{\mathbf{x}} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}}\,\mathrm{d}\mathbf{y} \tag{E.16}$$

The main difference here is that the least squares equation is applied to the difference between the observed values $\mathbf{y}$, and the expected values $f\left(\hat{\mathbf{x}}, \mathbf{u}\right)$, as in

$$\mathrm{d}\mathbf{y} = \mathbf{y} - f\left(\hat{\mathbf{x}}, \mathbf{u}\right) \ .$$

Moreover, the output of Eq. (E.16) is the update to the vector $\hat{\mathbf{x}}$, which is to be added and the new vector $\left(\hat{\mathbf{x}} + \mathrm{d}\hat{\mathbf{x}}\right)$ is to be used for the next iteration. This process usually requires two to three iterations to converge, when the initial conditions are close to the optimal solution.

LEVENBERG–MARQUARDT (Madsen et al., 2004) In some cases, the high non-linearity of the system can make the convergence very troublesome. For these cases, the formulation in Eq. (E.16) can be extended to obtain the Levenberg–Marquardt method:

$$\mathrm{d}\hat{\mathbf{x}} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A} + \lambda\mathbf{I}\right)^{-1}\mathbf{A}^{\mathrm{T}}\,\mathrm{d}\mathbf{y} \tag{E.17}$$

The only difference between Eqs. (E.16) and (E.17) is the addition of the parameter λ, which represents a damping coefficient, and is supposed to make the convergence more stable. The value of λ can be adapted throughout the convergence process, depending on the performance of the current iteration. In general, the initial value of the damping coefficient is given by

$$\lambda_0 = 10^{-6}\max\left[\mathrm{diag}\left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)\right]$$

and it is adapted based on a gain factor, whose symbol is usually given the greek letter ρ, and it is determined with:

$$\rho = \frac{\left|\left|\mathbf{y} - f\left(\hat{\mathbf{x}}, \mathbf{u}\right)\right|\right|^2 - \left|\left|\mathbf{y} - f\left(\hat{\mathbf{x}} + \mathrm{d}\hat{\mathbf{x}}, \mathbf{u}\right)\right|\right|^2}{\mathrm{d}\hat{\mathbf{x}}^{\mathrm{T}}\left(\lambda\,\mathrm{d}\hat{\mathbf{x}} + \mathbf{A}^{\mathrm{T}}\,\mathrm{d}\mathbf{y}\right)} \tag{E.18}$$

In case the value of ρ were to be less than zero, then the current iteration (denoted with i) is rejected and Eq. (E.17) is computed once again with the adapted values:

$$\lambda_{i+1} = \nu_i\lambda_i \qquad \nu_{i+1} = 2\nu_i$$

whereas, if $\rho \geq 0$, the iteration is accepted, the value of $\hat{\mathbf{x}}$ is updated and the Levenberg–Marquardt parameters are adapted as follows:

$$\lambda_{i+1} = \lambda_i \max\left[{}^1\!/_3, 1 - (2\rho - 1)^3\right] \qquad \nu_{i+1} = 2$$

E.6.3 | Exponential Case

In other cases, it may be that the non-linearity can be easily avoided. An example used in this thesis is the exponential case, and it is shown below.

In case of an exponential function such as:

$$y = f\left(\mathbf{x}, u\right) = x_1 \exp\left(x_2 u\right) \tag{E.19}$$

for f, one might think that the linear method does not apply, due to the non-linearity of the system w.r.t. $\mathbf{x}$. However, a simple circumvention is to transform the system, by taking the natural logarithm of both sides. Thus, this would result in a new function $\tilde{y} = \tilde{f}\left(\tilde{\mathbf{x}}, u\right) = \log\left[f\left(\mathbf{x}, u\right)\right]$, where also the coefficients have been transformed. The derivation is shown below.

TRANSFORMATION TO LOGARITHMIC SPACE OF EXPONENTIAL LEAST SQUARES PROBLEM Equation (E.19) can be used for a linear least squares case, if the system is first transformed to the logarithmic space. Thus,

$$\log\left[f\left(\mathbf{x}, u\right)\right] = \log\left[x_1 \exp\left(x_2 u\right)\right].$$

Due to the property of logarithms $\log\left(ab\right) = \log a + \log b$,

$$\tilde{f}\left(\tilde{\mathbf{x}}, u\right) = \log\left(x_1\right) + \log\left[\exp\left(x_2 u\right)\right].$$

Since the natural logarithm is the inverse of the exponential function,

$$\tilde{f}\left(\tilde{\mathbf{x}}, u\right) = \tilde{x}_1 + \tilde{x}_2 u\,,$$

which finally means that:

$$\tilde{\mathbf{y}} = \log\left(\mathbf{y}\right) \qquad \tilde{x}_1 = \log\left(x_1\right) \qquad \tilde{x}_2 = x_2$$

and one can simply solve this new linear system, and invert the result for x_1.

E.7 | Filtering

The filtering techniques used in this report are limited to Kalman filtering. Kalman filters, work on the principle that they uses both a model description and measurement data to estimate the state of a system. In case of orbit determination for aerobraking, the model is given by the equations of motion and by the environment (both introduced in Chapter 4), and measurements are provided by the IMU and PTE. IMU data is retrieved very frequently (according to Jah and Lisano (2004), MGS used a refresh rate of 200 Hz), whereas PTE data is available only once per orbit.

A non-linear system is described by:

$$\mathbf{x}_{k+1} = \mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k\right) \tag{E.20a}$$

where $\mathbf{x}$ is the state vector, $\mathbf{u}$ the control vector and $\mathbf{w}$ the noise parameter. Furthermore, measurements are expressed by:

$$\mathbf{z}_{k+1} = \mathbf{h}\left(\mathbf{x}_k, \mathbf{v}_k\right) \tag{E.20b}$$

with $\mathbf{v}$ being the measurement noise. In both Eqs. (E.20a) and (E.20b), k represents the current step, whereas $k+1$ is the next step. The noise parameters from Eq. (E.20) are modeled as uncorrelated white noise, distributed according to

$$\mathbf{w}_k \sim \mathcal{N}\left(0, \mathbf{Q}\right) \quad \text{and} \quad \mathbf{v}_k \sim \mathcal{N}\left(0, \mathbf{R}\right) \ ,$$

where $\mathcal{N}$ is the normal, or Gaussian, distribution, and $\mathbf{Q}$ and $\mathbf{R}$ are the uncertainties present in the modeling of the system and in the measurements, respectively.

E.7.1 | Extended Kalman Filter

As mentioned in the beginning of the section, the EKF works by first computing an estimate of the next step solely based on the system, i.e., Eq. (E.20a). This estimate is called the *a-priori* estimate $\hat{\mathbf{x}}_{k+1}^-$, and is then corrected by the measurement data, to obtain the *a-posteriori* estimate $\hat{\mathbf{x}}_{k+1}$. The symbol $\hat{\square}$ here, signifies estimate.

The a-priori estimate is found by simply using Eq. (E.20a), where the actual state $\mathbf{x}_k$ is replaced with its previous a-posteriori estimate $\hat{\mathbf{x}}_k$,

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{f}\left(\hat{\mathbf{x}}_k, \mathbf{u}_k, \mathbf{0}\right) \ .$$

Similarly, the measurement is then estimated with

$$\hat{\mathbf{z}}_k = \mathbf{h}\left(\hat{\mathbf{x}}_k^-, \mathbf{0}\right) \ .$$

Then, the a-priori estimate of the covariance matrix of the state vector can be found,

$$\mathbf{P}_{k+1}^- = \mathbf{A}_k \mathbf{P}_k \mathbf{A}_k^{\mathrm{T}} + \mathbf{W}_k \mathbf{Q} \mathbf{W}_k^{\mathrm{T}} \ .$$

Here, $\mathbf{A}$ and $\mathbf{W}$ are Jacobian matrices of the system function, w.r.t. the state and noise parameter, respectively:

$$\mathbf{A}_k = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right]_{\left(\hat{\mathbf{x}}_k, \mathbf{u}_k, \mathbf{0}\right)} \tag{E.21a}$$

$$\mathbf{W}_k = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{w}}\right]_{\left(\hat{\mathbf{x}}_k, \mathbf{u}_k, \mathbf{0}\right)} \tag{E.21b}$$

Now, the a-posteriori estimate can be computed. This is dependent on a parameter called Kalman gain, expressed by $\mathbf{K}$, and its value is given by:

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^- \mathbf{H}_k^{\mathrm{T}} \left(\mathbf{H}_k \mathbf{P}_{k+1}^- \mathbf{H}_k^{\mathrm{T}} + \mathbf{V}_k \mathbf{R} \mathbf{V}_k^{\mathrm{T}}\right)^{-1} \tag{E.22}$$

Matrix $\mathbf{P}$ in Eq. (E.22), is the covariance matrix of the state and its value is taken from the previous step. Matrices $\mathbf{H}$ and $\mathbf{V}$, on the other hand, denote the Jacobian matrices of the observations, and their values can be found with:

$$\mathbf{H}_k = \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right]_{\left(\hat{\mathbf{x}}_k, \mathbf{0}\right)} \tag{E.23a}$$

$$\mathbf{V}_k = \left[\frac{\partial \mathbf{h}}{\partial \mathbf{v}}\right]_{\left(\hat{\mathbf{x}}_k, \mathbf{0}\right)} \tag{E.23b}$$

Finally, the a-posteriori value is found with Eq. (E.24a), and the covariance matrix $\mathbf{P}$ for the next step, with:

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1}^{-} + \mathbf{K}_{k+1}\left(\mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1}\right) \tag{E.24a}$$

and

$$\mathbf{P}_{k+1} = \left(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_k\right)\mathbf{P}_{k+1}^{-}\left(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_k\right)^{\mathrm{T}} + \mathbf{K}_{k+1}\mathbf{R}\mathbf{K}_{k+1}^{\mathrm{T}} \tag{E.24b}$$

As initial conditions, one needs a first estimate of the spacecraft state $\hat{\mathbf{x}}_0$ and of the covariance matrix $\mathbf{P}_0^{-}$. In general, $\mathbf{P}_0^{-}$ is set equal to the identity matrix $\mathbf{I}_6$, whereas $\hat{\mathbf{x}}_0$ comes from precise orbit determination, e.g., DSN tracking.

Furthermore, one also needs to values for the uncertainty parameters, i.e., $\mathbf{Q}$ and $\mathbf{R}$. Their values are constant and for $\mathbf{R}$ they are usually given by the manufacturer, while $\mathbf{Q}$ is determined by testing and tuning (Mooij, 2016).

E.7.2 | Unscented Kalman Filter

One of the main advantages of the UKF over EKF, is its lack of linearization and need of Jacobians. Rather, it uses so called sigma points to estimate the covariance of both the state and the measurement. These sigma points are found by introducing small deviations in the initial conditions, and are used to see the effect that propagation does to the system (unscented transformation). Thus, the need to Jacobian matrices may be relieved, but this is replaced by the propagation of the (often many) sigma points.

A key step of UKF is to augment the state vector $\hat{\mathbf{x}}$ with the expected values of the noise parameters $\mathbf{W}$ and $\mathbf{v}$. This results in the new augmented state variable $\hat{\mathbf{x}}^{\mathrm{a}}$, given by:

$$\hat{\mathbf{x}}_k^{\mathrm{a}} = \left(\hat{\mathbf{x}}^{\mathrm{T}}, \hat{\mathbf{w}}^{\mathrm{T}}, \hat{\mathbf{v}}^{\mathrm{T}}\right)^{\mathrm{T}} = \left(\hat{\mathbf{x}}^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}\right)^{\mathrm{T}} \tag{E.25}$$

Consequently, a new augmented covariance matrix $\mathbf{P}^{\mathrm{a}}$ is introduced:

$$\mathbf{P}_k^{\mathrm{a}} = \mathrm{diag}\left(\mathbf{P}_k, \mathbf{Q}, \mathbf{R}\right) \tag{E.26}$$

This will yield a new state and covariance matrix with dimensions $L \times 1$ and $L \times L$, respectively. The $(2L+1)$ sigma points $\boldsymbol{\mathcal{X}}$ can then be found with:

$$\boldsymbol{\mathcal{X}}_{i,k} = \begin{cases} \hat{\mathbf{x}}_k^{\mathrm{a}} & \text{if } i = 0 \\ \hat{\mathbf{x}}_k^{\mathrm{a}} + \gamma\left(\sqrt{\mathbf{P}_k^{\mathrm{a}}}\right)_i & \text{if } i \in [1, \ldots, L] \\ \hat{\mathbf{x}}_k^{\mathrm{a}} - \gamma\left(\sqrt{\mathbf{P}_k^{\mathrm{a}}}\right)_{i-L} & \text{if } i \in [L+1, \ldots, 2L] \end{cases} \tag{E.27}$$

Here, the notation $(\mathbf{A})_i$ refers to the i-th column of matrix $\mathbf{A}$. Furthermore, note that the value of $\sqrt{\mathbf{P}^{\mathrm{a}}} = \mathbf{A}$ is such that

$$\mathbf{A}\mathbf{A} = \mathbf{P}^{\mathrm{a}} .$$

The spread of the sigma points in Eq. (E.27) will depend on the parameter γ, which can be determined with:

$$\lambda = \alpha^2\left(L + \kappa\right) - L \tag{E.28a}$$

$$\gamma = \sqrt{L + \lambda} \tag{E.28b}$$

In Eq. (E.28), the values of α and κ are user-defined, and can be used to tune convergence. Wan and Van Der Merwe (2000) suggests using 0.003 and 0 for their respective values.

The sigma points can then be propagated to obtain a series of estimates:

$$\boldsymbol{\mathcal{X}}_{i,k+1}^{\hat{\mathbf{x}}^{-}} = \mathbf{f}\left(\boldsymbol{\mathcal{X}}_{i,k}^{\mathbf{x}}, \mathbf{u}_k, \boldsymbol{\mathcal{X}}_{i,k}^{\mathbf{w}}\right) \tag{E.29}$$

Here, the superscript to the sigma points is used to indicate which part of the vector needs to be used. For instance $\boldsymbol{\mathcal{X}}_{i,k}^{\mathbf{x}}$ points to the state vector of the i-th sigma point, at instant k.

The a-priori estimate can then be determined by using a weighted average of the results from Eq. (E.29), as done in:

$$\hat{\mathbf{x}}_{k+1}^{-} = \sum_{i=0}^{2L} W_i^m \boldsymbol{\mathcal{X}}_{i,k+1}^{\hat{\mathbf{x}}^{-}} \tag{E.30a}$$

where the weights are given by:

$$W_i^m = \begin{cases} \dfrac{\lambda}{L+\lambda} & \text{if } i=0 \\ \dfrac{1}{2\,(L+\lambda)} & \text{otherwise} \end{cases} \tag{E.30b}$$

Using the values of $\boldsymbol{\mathcal{X}}^{\hat{\mathbf{x}}^-}$ and $\hat{\mathbf{x}}^-$, the a-priori covariance matrix can be estimated. This is done with:

$$\mathbf{P}_{k+1}^- = \mathbf{Q} + \sum_{i=0}^{2L} W_i^c \left(\boldsymbol{\mathcal{X}}_{i,k+1}^{\hat{\mathbf{x}}^-} - \hat{\mathbf{x}}_{k+1}^-\right)\left(\boldsymbol{\mathcal{X}}_{i,k+1}^{\hat{\mathbf{x}}^-} - \hat{\mathbf{x}}_{k+1}^-\right)^{\mathrm{T}} \tag{E.31a}$$

The weights used for covariance are slightly different, and are given by:

$$W_i^c = \begin{cases} W_0^m + 1 - \alpha^2 + \beta & \text{if } i=0 \\ W_i^m & \text{otherwise} \end{cases} \tag{E.31b}$$

In Eq. (E.31b), β is another constant parameter, and its value is 2 for a Gaussian distribution (Wan and Van Der Merwe, 2000). Jah et al. (2008) suggest now, to recompute the sigma points, given the new state estimate $\hat{\mathbf{x}}_{k+1}^-$, to include for effect of process noise. This is done by replacing the first part of the augmented state and covariance in Eq. (E.27), by $\hat{\mathbf{x}}_{k+1}^-$ and $\mathbf{P}_{k+1}^-$. The result is a new set of sigma points $\boldsymbol{\mathcal{X}}_{i,k+1}^{\hat{\mathbf{x}}^-}$.

The measurement values corresponding to the sigma points $\boldsymbol{\mathcal{X}}$ also need to be determined. These yield a set of estimated measurement sigma points, denoted with $\boldsymbol{\mathcal{Z}}$:

$$\boldsymbol{\mathcal{Z}}_{i,k+1} = \mathbf{h}\left(\boldsymbol{\mathcal{X}}_{i,k+1}^{\hat{\mathbf{x}}^-}, \boldsymbol{\mathcal{X}}_{i,k}^{\mathbf{v}}\right) \tag{E.32}$$

Similarly to what was done in Eq. (E.30a), one can find the expected value of the $\boldsymbol{\mathcal{Z}}$ points:

$$\hat{\mathbf{z}}_{k+1} = \sum_{i=0}^{2L} W_i^m \boldsymbol{\mathcal{Z}}_{i,k+1} \tag{E.33}$$

Then, the innovation and cross-correlation matrices can be determined. These represent the covariance of the measurement points and the covariance between state and measurement, and are expressed by:

$$\mathbf{P}_{\mathbf{zz}} = \mathbf{R} + \sum_{i=0}^{2L} W_i^c \left(\boldsymbol{\mathcal{Z}}_{i,k+1} - \hat{\mathbf{z}}_{k+1}\right)\left(\boldsymbol{\mathcal{Z}}_{i,k+1} - \hat{\mathbf{z}}_{k+1}\right)^{\mathrm{T}} \tag{E.34a}$$

$$\mathbf{P}_{\mathbf{xz}} = \sum_{i=0}^{2L} W_i^c \left(\boldsymbol{\mathcal{X}}_{i,k+1}^{\hat{\mathbf{x}}^-} - \hat{\mathbf{x}}_{k+1}^-\right)\left(\boldsymbol{\mathcal{Z}}_{i,k+1} - \hat{\mathbf{z}}_{k+1}\right)^{\mathrm{T}} \tag{E.34b}$$

Finally, the Kalman gain can be found with:

$$\mathbf{K}_{k+1} = \mathbf{P}_{\mathbf{xz}} \mathbf{P}_{\mathbf{zz}}^{-1} \tag{E.35}$$

Its value is the used to update the state vector (by using the same equation as for EKF, i.e., Eq. (E.24a)) and the covariance matrix, by using:

$$\mathbf{P}_{k+1} = \mathbf{P}_{k+1}^- - \mathbf{K}_{k+1}\mathbf{P}_{\mathbf{zz}}\mathbf{K}_{k+1}^{\mathrm{T}} \tag{E.36}$$

As initial conditions, one needs a first estimate of the spacecraft state $\hat{\mathbf{x}}_0$ and of the covariance matrix $\mathbf{P}_0^-$. In general, $\mathbf{P}_0^-$ is set equal to the identity matrix $\mathbf{I}_6$, whereas $\hat{\mathbf{x}}_0$ comes from precise orbit determination, e.g., DSN tracking. Furthermore, one also needs values for the uncertainty parameters, i.e., $\mathbf{Q}$ and $\mathbf{R}$. Their values are constant and for $\mathbf{R}$ they are usually given by the manufacturer, while $\mathbf{Q}$ is determined by testing and tuning (Mooij, 2016).

PAGE INTENTIONALLY LEFT BLANK

F Derivations

In this appendix, the derivations of the non-linear least squares for atmosphere fitting, of the impulsive apocenter maneuver, and of the atmospheric phase drag modeling are given. In the last section, one will find the description of how pseudo-altitude measured by an altimeter is modeled. This part is included to serve as a reference for the instrument model developed in Tudat.

F.1 | Atmosphere Model Fitting

As mentioned in Section 6.7, after each atmospheric phase, the atmosphere model is re-calibrated. This is done by means of non-linear least squares fitting of the measured aerodynamic acceleration. The two main models that are fit are the exponential model, from Eq. (4.29), and the five-parameter exponential model, i.e., Eq. (4.33), which also includes a sine and cosine term.

To perform a least square estimation, the Jacobian of these two functions needs to be computed. In the case of this application, the vector $\mathbf{x}$, representing the system coefficients, can be expressed as:

$$\mathbf{x} = \left[\log\left(\rho_0\right), {}^{1}\!/\!{}_{H}\right]^{\mathrm{T}}$$

for the exponential formulation, and as:

$$\mathbf{x} = \left[\log\left(\rho_0\right), {}^{1}\!/\!{}_{H}, \kappa_1, \kappa_2, \kappa_3\right]^{\mathrm{T}}$$

for the five-term atmosphere. In both cases, the transformation to logarithmic space of Appendix E.6.3 is applied, and the scale height is replaced by its inverse, also known as β. Both of these steps are applied to reduce the non-linearity of the system and simplify the final formulation of the Jacobian. Furthermore, the value of h_0 is simply set to the estimated periapsis altitude.

The final result is for the exponential model the Jacobian:

$$\mathbf{A} = \left[1, h_0 - h_i\right]_{\forall i} \tag{F.1}$$

and for the five-term model, it can be expressed as:

$$\mathbf{A} = \begin{pmatrix} 1 \\ \left(h_0 - h_i\right)\left\{\kappa_1 + 2\pi\left[-\kappa_2 \sin\left[2\pi\beta\left(h_0 - h_i\right)\right] + \kappa_3 \cos\left[2\pi\beta\left(h_0 - h_i\right)\right]\right]\right\} \\ \beta\left(h_0 - h_i\right) \\ \cos\left[2\pi\beta\left(h_0 - h_i\right)\right] \\ \sin\left[2\pi\beta\left(h_0 - h_i\right)\right] \end{pmatrix}_{\forall i}^{\mathrm{T}} \tag{F.2}$$

F.2 | Apoapsis Maneuver

Gauss' form of the Lagrange's planetary equations for semi-major axis a and eccentricity e, are given by (Wakker, 2015):

$$\frac{\mathrm{d}a}{\mathrm{d}t} = 2\frac{a^2}{\sqrt{\mu p}}\left[a_{\hat{\mathbf{x}}_{\mathcal{L}}} e \sin\vartheta + a_{\hat{\mathbf{y}}_{\mathcal{L}}}\left(1 + e\cos\vartheta\right)\right] \tag{F.3}$$

Assuming that there are no perturbing accelerations acting on the spacecraft, one can determine the required ΔV to achieve a periapsis change of Δr_p, where this change is defined as:

$$\Delta r_p = {r_p}^{\dagger} - r_p \tag{F.4}$$

where the parameters post-maneuver are denoted with $\square^\dagger$. The change in periapsis altitude from Eq. (F.4), can be related to changes in semi-major axis and changes in eccentricity. Derivation of these relations is shown in Appendix F.2.1.

Before the ΔV can be computed, one needs to relate ΔV to the accelerations a (Wakker, 2015):

$$\Delta V = \lim_{t \to 0} \int_0^t a \, \mathrm{d}\tau \tag{F.5}$$

Equation (F.5) implies that the maneuver performed at apoapsis is an impulsive shot, and as such it is carried out instantaneously.

In a similar fashion, one can find the expressions for Δa, by applying the limit in Eq. (F.5) to Eq. (F.3). The derivation of the ΔV for the impulsive shot at apoapsis, is shown below in Appendix F.2.2.

F.2.1 | Relation Derived from Periapsis Change

Starting from the definitions of semi-major axis, in terms of peri- and apocenter radii (Wakker, 2015):

$$a = \frac{r_a + r_p}{2} \tag{F.6}$$

one can find the definition of the Δa, due to the apoapsis maneuver. The effect on a due to the maneuver at apoapsis, can be found by computing the difference between $a^\dagger$ and a, where it is assumed that the apocenter altitude remains constant. The derivation is straightforward:

$$a^\dagger - a = \frac{r_a + {r_p}^\dagger}{2} - \frac{r_a + r_p}{2} = \frac{{r_p}^\dagger - r_p}{2}$$

Finally one obtains the relationship:

$$\Delta a = \frac{\Delta r_p}{2} \tag{F.7}$$

F.2.2 | Derivation of Impulsive Shot without Perturbations

In case the motion of the spacecraft is unperturbed, the relation between impulsive shots ΔV and changes in orbital elements can be easily found (Wakker, 2015):

$$\Delta a = 2 \frac{a^2}{\sqrt{\mu p}} \left[\Delta V_{\hat{\mathbf{x}}_\mathcal{L}} e \sin\vartheta + \Delta V_{\hat{\mathbf{y}}_\mathcal{L}} \left(1 + e \cos\vartheta\right) \right] \tag{F.8}$$

Knowing that at apoapsis both ϑ and E equal π rad, one can simplify Eq. (F.8), to obtain:

$$\Delta a = 2 \frac{a^2}{\sqrt{\mu p}} \left(1 - e\right) \Delta V_{\hat{\mathbf{y}}_\mathcal{L}} = \frac{2}{n} \sqrt{\frac{1-e}{1+e}} \Delta V_{\hat{\mathbf{y}}_\mathcal{L}} \tag{F.9}$$

Using Eqs. (F.7) and (F.9), one can relate the magnitude of the required $\Delta V_{\hat{\mathbf{y}}_\mathcal{L}}$, to the change in periapsis altitude:

$$\Delta V_{\hat{\mathbf{y}}_\mathcal{L}} = n \frac{\Delta r_p}{4} \sqrt{\frac{1+e}{1-e}} \tag{F.10}$$

F.3 | **Periapsis Drag Modeling**

In the section above, the assumptions and calculations necessary to compute the maneuver at apocenter have been introduced. In this section, on the other hand, the effect of drag at pericenter will be discussed. As it turns out, however, these two elements are very similar in nature, due to the assumptions made. In fact, since the effect of drag is assumed to occur exactly at apoapsis and as an impulsive shot, the derivation for the change in semi-major axis is exactly the same, with the only difference that the true (and mean) anomaly is zero.

However, for this part of the system, we are also interested in knowing the effect on the eccentricity. Thus, the following equation, describing the change in eccentricity due to external perturbations will come in handy (Wakker, 2015):

$$\frac{\mathrm{d}e}{\mathrm{d}t} = \sqrt{\frac{p}{\mu}} \left[a_{\hat{\mathbf{x}}_\mathcal{L}} \sin\vartheta + a_{\hat{\mathbf{y}}_\mathcal{L}} \left(\cos E + \cos\vartheta\right) \right] \tag{F.11}$$

By assuming an impulsive shot, the same approximations can be made, and Eq. (F.11) can be written as:

$$\Delta e = \sqrt{\frac{p}{\mu}} \left[\Delta V_{\hat{\mathbf{x}}_{\mathcal{L}}} \sin\vartheta + \Delta V_{\hat{\mathbf{y}}_{\mathcal{L}}} \left(\cos E + \cos\vartheta\right) \right] \tag{F.12}$$

Then, knowing that at periapsis both ϑ and E equal 0 rad, one can simplify Eqs. (F.8) and (F.12) to:

$$\Delta a = \frac{2}{n}\sqrt{\frac{1+e}{1-e}} \Delta V_{\hat{\mathbf{y}}_{\mathcal{L}}} \tag{F.13a}$$

$$\Delta e = 2\sqrt{\frac{a\left(1-e^2\right)}{\mu}} \Delta V_{\hat{\mathbf{y}}_{\mathcal{L}}} \tag{F.13b}$$

These are the final relations to be used to model the effect of drag.

F.4 | Pseudo-altitude

Altimeters are instruments whose purpose is to measure distance. This is generally achieved by sending a beam of electromagnetic waves toward a target, to which the distance needs to be computed. These waves can fall into various spectra, such as radio waves or visible light, depending on the application. For the purpose of preliminary navigation system modeling, the type of altimeter is not of high relevance, as long as the following characteristics are defined:

- **Altitude Range**: range of altitudes between which the altimeter is operational
- **Accuracy**: standard deviation of the random error (generally as a function of altitude)
- **Pointing Direction**: direction in the $\mathcal{B}$-frame along which the altimeter is pointing

Note that in the model detailed here, it is assumed that Mars is a perfect sphere, such that no local variations of the radius need to be modeled.

Since the altimeter pointing direction is fixed, the angle between the altimeter line-of-sight and the planet surface might not always be perpendicular. Thus, the distance measured by the altimeter could be affected by the angular distance between the current radial vector and the pointing vector, which is called pointing angle φ, and is given by the inverse of the dot product:

$$\varphi = \arccos\left(\hat{\mathbf{r}}|_{\mathcal{I}} \cdot \hat{\mathbf{p}}|_{\mathcal{I}}\right) \tag{F.14}$$

Moreover, there is a possibility that the planet may not be in sight. This condition is met whenever:

$$\varphi > \arcsin\left(\frac{R}{r}\right)$$

where R is the planetary radius and r the current radial distance. Finally, the measured altitude (or pseudo-altitude), denoted with $\tilde{\square}$, is a function of the actual orbital distance and the pointing direction:

$$\tilde{h} = \left(\mathbf{r}\cdot\hat{\mathbf{p}}\right) - \sqrt{R^2 - r^2 + \left(\mathbf{r}\cdot\hat{\mathbf{p}}\right)^2} \tag{F.15}$$

which is derived below with the use of Fig. F.1, and holds as long as both $\mathbf{r}$ and $\hat{\mathbf{p}}$ are expressed in the same frame of reference.

The known variables from Fig. F.1 are the planetary radius R (which, as already stated, is assumed to be constant), the current orbital distance r and the current pointing angle φ (highlighted in green in the figure). Note that it does not matter which plane the diagram in Fig. F.1 is projected on, since the planet is assumed to be a perfect sphere.

Hence, one can first find the long side a:

$$a = r\cos\varphi$$

from which, the value of b can be subtracted to obtain the pseudo-altitude. The small side b can be found by applying Pythagoras' theorem to the triangle made up by $R-c-b$:

$$b = \sqrt{R^2 - c^2}$$

where c is simply given by:

$$c = r\sin\varphi$$

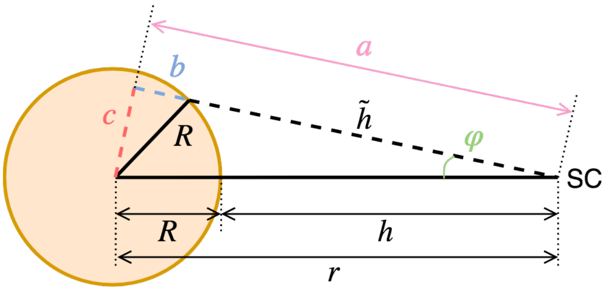


Figure F.1: *Diagram for the derivation of the pseudo-altitude measured by the altimeter. The orange sphere represents Mars, whereas the point SC represents the spacecraft. The four parameters used in the derivation are highlighted with different colors. The green angle, namely φ, represents the pointing angle of the altimeter.*

Thus, one obtains the relationship for $\tilde{h}$:

$$\tilde{h} = a - b = r\cos\varphi - \sqrt{R^2 - r^2\sin^2\varphi} \tag{F.16}$$

which, by replacing the definition of φ from Eq. (F.14) (and by making use of Eq. (H.2)), results in a more computationally efficient form:

$$\tilde{h} = r\left(\hat{\mathbf{r}} \cdot \hat{\mathbf{p}}\right) - \sqrt{R^2 - r^2\left[1 - \left(\hat{\mathbf{r}} \cdot \hat{\mathbf{p}}\right)^2\right]}$$

One can then simplify further, to obtain the formulation shown in Eq. (F.15).

It may be that the above equation needs to be included as observation model for the extended Kalman filter. In such case, one also needs the value of the Jacobian of the measurement function:

$$\frac{\partial\tilde{h}}{\partial\mathbf{r}} = \frac{\partial\left(\mathbf{r} \cdot \hat{\mathbf{p}}\right)}{\partial\mathbf{r}} - \frac{\partial}{\partial\mathbf{r}}\left[\sqrt{R^2 - r^2 + \left(\mathbf{r} \cdot \hat{\mathbf{p}}\right)^2}\right]$$

It can be easily shown that this results in:

$$\frac{\partial\tilde{h}}{\partial r_i} = p_i + \frac{r_i - p_i\left(\mathbf{r} \cdot \hat{\mathbf{p}}\right)}{\sqrt{R^2 - r^2 + \left(\mathbf{r} \cdot \hat{\mathbf{p}}\right)^2}} \qquad i \in [1, 2, 3] \tag{F.17}$$

G Simulation Environments

In this appendix, the full environment models, based on the Tudat software library, are given. They are used to obtain the results of Chapters 10, 12 and 13.

Table G.1 shows the spacecraft physical properties used for the simulations of all the chapters listed above. Note that the mass moment of inertia is only used for the single-orbit results of the aerobraking GNC analysis, i.e., Subsection 12.2.1.

In Table G.2, on the other hand, one will find the initial value of the exponential model parameters used for the onboard environment model. These values are based on Mars Climate Database from Appendix C.3, where the density values are averaged over latitude, longitude and time, and the values at 115 km are used as reference (with the scale height H resulting from Eq. (4.29b), with g from Table C.1). Note that in case the five-parameter exponential model from Eq. (4.33), the values of the three extra parameters are as follows:

$$\kappa_1 = -1 \qquad \kappa_2 = \kappa_3 = 0$$

Finally, Table G.3 shows the full simulation settings for the Tudat analysis of Chapter 10, but Table G.3b also details the settings used for the aerobraking simulation of Chapters 12 and 13. Note that these refer to the settings of the propagation environment, and not the onboard environment, which can be found in the respective chapters.

Table G.1: *Spacecraft physical parameters for comparison of propagators performance carried out in Chapter 10.*

Mass	Reference Area	Mass Moment of Inertia	Aerodynamic Coefficients	Radiation Pressure Coefficient
[kg]	[m²]	[kg m²]		[−]
1000	37.5	From Appendix B.2	From Chapter 11	1.0

Table G.2: *Exponential atmosphere parameters based on the latitude-longitude-time average of the MCD from Appendix C.3, with a reference altitude more resembling of the average periapsis condition.*

Reference Altitude	Scale Height	Density	Gas Constant	Temperature	Specific Heat Ratio
[km]	[m]	[kg m^{-3}]	[J kg^{-1} K^{-1}]	[K]	[−]
115.0	6533.0	2.424×10^{-8}	197.0	215.0	1.3

Table G.3: *Environment characteristics for comparison of propagators performance carried out in Chapter 10.*

(a) *Test cases 1, 2 and 3.*

Gravitational Field	Atmosphere	Solar Radiation Pressure	Third Bodies
GGM02S Model, up to degree and order 2	Earth 1976 standard atmosphere	Cannonball radiation pressure interface, with Earth as occulting body	Central gravity, with gravitational parameter from Table C.1

(b) *Test cases 4 and 5 and the aerobraking simulation.*

Gravitational Field	Atmosphere	Solar Radiation Pressure	Third Bodies
JGMRO120D Model, up to degree and order 21	Mars Climate Database atmosphere from Appendix C, averaged over time	Cannonball radiation pressure interface, with Mars as occulting body	Central gravity, with gravitational parameter from Table C.1

(c) *Test cases 6.*

Gravitational Field	Atmosphere	Solar Radiation Pressure	Third Bodies
Central gravity of Sun	NA	Cannonball radiation pressure interface	Central gravity, with gravitational parameter from Table C.1 or built-in value from Tudat

H
Trigonometric Identities

In this appendix, a list of trigonometric identities is provided. These are used in this report for various derivations. Reference for this appendix is Wertz (2009).

H.1 | Small Angle Approximation

In case the angle α is small, i.e. $|\alpha| \ll 1$, the following approximations hold:

$$\sin\alpha \approx \alpha \tag{H.1a}$$

$$\cos\alpha \approx 1 \tag{H.1b}$$

$$\tan\alpha \approx \alpha \tag{H.1c}$$

H.2 | Euclidean Geometry

Some very useful trigonometric identities for Euclidean geometry, i.e., on a flat surface, are shown in the following subsections.

Inverse Trigonometric Functions Trigonometric functions and their inverse can be used to find the following relationships:

$$\sin\left(\arccos x\right) = \sqrt{1-x^2} \tag{H.2a}$$

$$\cos\left(\arcsin x\right) = \sqrt{1-x^2} \tag{H.2b}$$

Angles-sum Formulas The angle-sum formulas for sine, cosine and tangent can be expanded as:

$$\sin\left(\alpha \pm \beta\right) = \sin\alpha\cos\beta \pm \cos\alpha\sin\beta \tag{H.3a}$$

$$\cos\left(\alpha \pm \beta\right) = \cos\alpha\cos\beta \mp \sin\alpha\sin\beta \tag{H.3b}$$

$$\tan\left(\alpha \pm \beta\right) = \frac{\sin 2\alpha \pm \sin 2\beta}{\cos 2\alpha \mp \cos 2\beta} \tag{H.3c}$$

Half-angle Formulas The half-angle formulas of sine, cosine and tangent are given by:

$$\sin\alpha = 2\sin\frac{\alpha}{2}\cos\frac{\alpha}{2} \tag{H.4a}$$

$$\cos\alpha = \cos^2\frac{\alpha}{2} - \sin^2\frac{\alpha}{2} = 2\cos^2\frac{\alpha}{2} - 1 = 1 - \sin^2\frac{\alpha}{2} \tag{H.4b}$$

$$\tan\frac{\alpha}{2} = \frac{\sin\alpha}{1+\cos\alpha} = \frac{1-\cos\alpha}{\sin\alpha} \tag{H.4c}$$

Sine Rule The sine rule for Euclidean geometry is defined as:

$$\frac{a}{\sin\alpha} = \frac{b}{\sin\beta} = \frac{c}{\sin\gamma} \tag{H.5}$$

where a, b and c correspond to the sides opposite to the angles α, β and γ, respectively.

Cosine Rule The cosine rule for Euclidean geometry is defined as:

$$c = \sqrt{a^2 + b^2 - 2ab\cos\gamma} \tag{H.6}$$

where once again, a, b and c correspond to the sides opposite to the angles α, β and γ, respectively.

H.3 | Spherical Geometry

In this section, Fig. H.1 is used as reference for the nomenclature of angles and sides of spherical triangles.

Sine Rule The sine rule for spherical geometry is defined as:

$$\frac{\sin a}{\sin\alpha} = \frac{\sin b}{\sin\beta} = \frac{\sin c}{\sin\gamma} \tag{H.7}$$

Cosine Rules The cosine rule for arc lengths in given by:

$$\cos a = \cos b\cos c + \sin b\sin c\cos\alpha \tag{H.8a}$$
$$\cos b = \cos c\cos a + \sin c\sin a\cos\beta \tag{H.8b}$$
$$\cos c = \cos a\cos b + \sin a\sin b\cos\gamma \tag{H.8c}$$

One the other hand, the cosine rule for angles is defined as:

$$\cos\alpha = -\cos\beta\cos\gamma + \sin\beta\sin\gamma\cos a \tag{H.9a}$$
$$\cos\beta = -\cos\gamma\cos\alpha + \sin\gamma\sin\alpha\cos b \tag{H.9b}$$
$$\cos\gamma = -\cos\alpha\cos\beta + \sin\alpha\sin\beta\cos c \tag{H.9c}$$

Napier's Rules Napier's rules are only valid for either a spherical triangles with a side equal to $\pi/2$, or with an angle equal to $\pi/2$. The relations that arise from these conditions are best described by Wertz (2009) in Figs. H.2 and H.3, for sides and angles, respectively. Note that here, one can also see the quadrant corresponding to the side or angle.

H.4 | Arccosine

The $\arctan_2$ function is defined in such a way that the quadrant of the resulting angle is correct. Similarly, one can define an $\arccos_2$, such that the quadrant information is not lost. According to Wertz (2009), this new function is defined as:

$$\arccos_2(x, \chi) = \text{sign}(\pi - \chi)\arccos x \tag{H.10}$$

In case the result of $(\pi - \chi)$ is zero, the sign function still gives $+1$ as result.

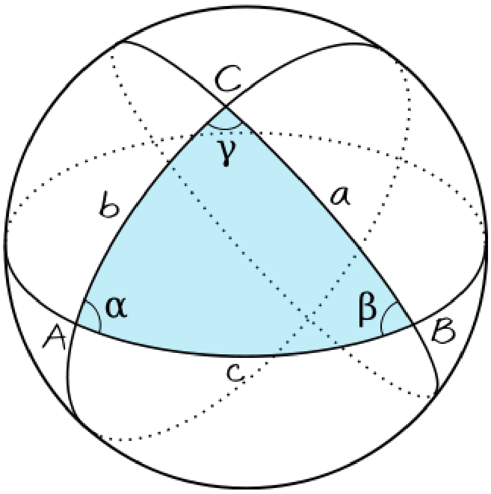
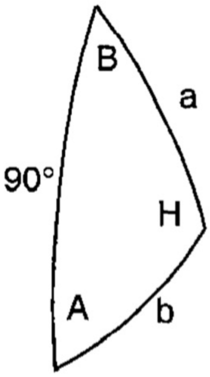
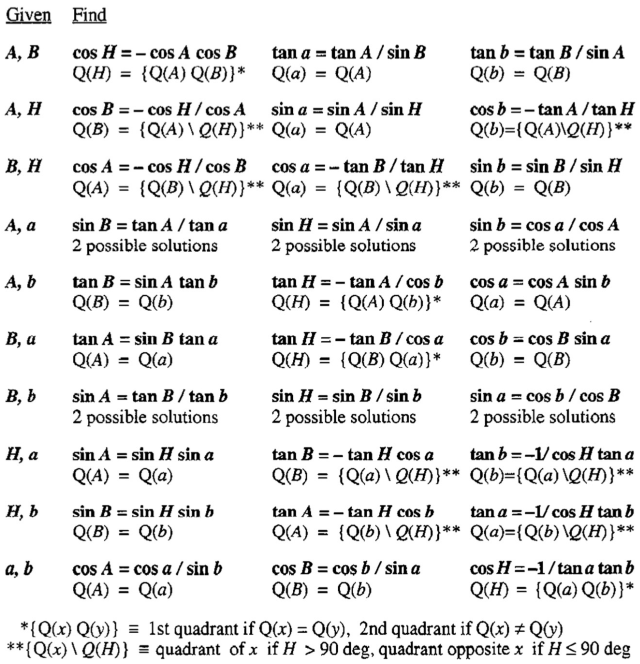


Figure H.1: *Definition of nomenclature for spherical triangles*[1].

[1] Wikibooks: Triangles on a Sphere – https://goo.gl/GyEqKd [last accessed on November 5, 2017]





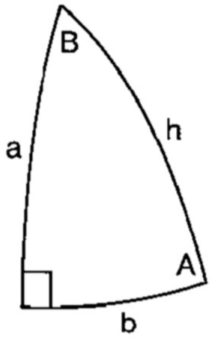
Given	Find		
A, B	$\cos H = -\cos A \cos B$ $Q(H) = \{Q(A)\,Q(B)\}^*$	$\tan a = \tan A / \sin B$ $Q(a) = Q(A)$	$\tan b = \tan B / \sin A$ $Q(b) = Q(B)$
A, H	$\cos B = -\cos H / \cos A$ $Q(B) = \{Q(A) \setminus Q(H)\}^{**}$	$\sin a = \sin A / \sin H$ $Q(a) = Q(A)$	$\cos b = -\tan A / \tan H$ $Q(b)=\{Q(A) \setminus Q(H)\}^{**}$
B, H	$\cos A = -\cos H / \cos B$ $Q(A) = \{Q(B) \setminus Q(H)\}^{**}$	$\cos a = -\tan B / \tan H$ $Q(a) = \{Q(B) \setminus Q(H)\}^{**}$	$\sin b = \sin B / \sin H$ $Q(b) = Q(B)$
A, a	$\sin B = \tan A / \tan a$ 2 possible solutions	$\sin H = \sin A / \sin a$ 2 possible solutions	$\sin b = \cos a / \cos A$ 2 possible solutions
A, b	$\tan B = \sin A \tan b$ $Q(B) = Q(b)$	$\tan H = -\tan A / \cos b$ $Q(H) = \{Q(A)\,Q(b)\}^*$	$\cos a = \cos A \sin b$ $Q(a) = Q(A)$
B, a	$\tan A = \sin B \tan a$ $Q(A) = Q(a)$	$\tan H = -\tan B / \cos a$ $Q(H) = \{Q(B)\,Q(a)\}^*$	$\cos b = \cos B \sin a$ $Q(b) = Q(B)$
B, b	$\sin A = \tan B / \tan b$ 2 possible solutions	$\sin H = \sin B / \sin b$ 2 possible solutions	$\sin a = \cos b / \cos B$ 2 possible solutions
H, a	$\sin A = \sin H \sin a$ $Q(A) = Q(a)$	$\tan B = -\tan H \cos a$ $Q(B) = \{Q(a) \setminus Q(H)\}^{**}$	$\tan b = -1/ \cos H \tan a$ $Q(b)=\{Q(a) \setminus Q(H)\}^{**}$
H, b	$\sin B = \sin H \sin b$ $Q(B) = Q(b)$	$\tan A = -\tan H \cos b$ $Q(A) = \{Q(b) \setminus Q(H)\}^{**}$	$\tan a = -1/ \cos H \tan b$ $Q(a)=\{Q(b) \setminus Q(H)\}^{**}$
a, b	$\cos A = \cos a / \sin b$ $Q(A) = Q(a)$	$\cos B = \cos b / \sin a$ $Q(B) = Q(b)$	$\cos H = -1 / \tan a \tan b$ $Q(H) = \{Q(a)\,Q(b)\}^*$

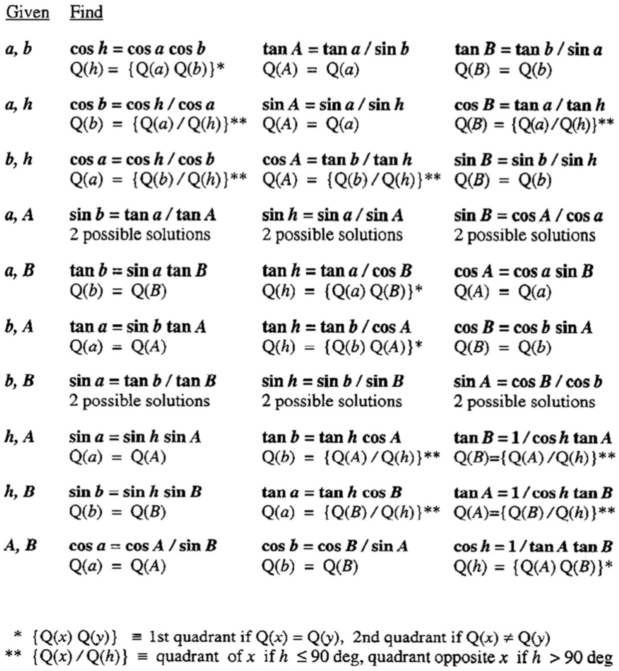
$^*\{Q(x)\,Q(y)\} \equiv$ 1st quadrant if $Q(x) = Q(y)$, 2nd quadrant if $Q(x) \neq Q(y)$
$^{**}\{Q(x) \setminus Q(H)\} \equiv$ quadrant of x if $H > 90$ deg, quadrant opposite x if $H \leq 90$ deg

(a) *Nomenclature.*

(b) *Napier's rules.*

Figure H.2: *Napier's rules for a spherical triangle with a quadrantal side (Wertz, 2009).*





Given	Find		
a, b	$\cos h = \cos a \cos b$ $Q(h) = \{Q(a)\,Q(b)\}^*$	$\tan A = \tan a / \sin b$ $Q(A) = Q(a)$	$\tan B = \tan b / \sin a$ $Q(B) = Q(b)$
a, h	$\cos b = \cos h / \cos a$ $Q(b) = \{Q(a) / Q(h)\}^{**}$	$\sin A = \sin a / \sin h$ $Q(A) = Q(a)$	$\cos B = \tan a / \tan h$ $Q(B) = \{Q(a) / Q(h)\}^{**}$
b, h	$\cos a = \cos h / \cos b$ $Q(a) = \{Q(b) / Q(h)\}^{**}$	$\cos A = \tan b / \tan h$ $Q(A) = \{Q(b) / Q(h)\}^{**}$	$\sin B = \sin b / \sin h$ $Q(B) = Q(b)$
a, A	$\sin b = \tan a / \tan A$ 2 possible solutions	$\sin h = \sin a / \sin A$ 2 possible solutions	$\sin B = \cos A / \cos a$ 2 possible solutions
a, B	$\tan b = \sin a \tan B$ $Q(b) = Q(B)$	$\tan h = \tan a / \cos B$ $Q(h) = \{Q(a)\,Q(B)\}^*$	$\cos A = \cos a \sin B$ $Q(A) = Q(a)$
b, A	$\tan a = \sin b \tan A$ $Q(a) = Q(A)$	$\tan h = \tan b / \cos A$ $Q(h) = \{Q(b)\,Q(A)\}^*$	$\cos B = \cos b \sin A$ $Q(B) = Q(b)$
b, B	$\sin a = \tan b / \tan B$ 2 possible solutions	$\sin h = \sin b / \sin B$ 2 possible solutions	$\sin A = \cos B / \cos b$ 2 possible solutions
h, A	$\sin a = \sin h \sin A$ $Q(a) = Q(A)$	$\tan b = \tan h \cos A$ $Q(b) = \{Q(A) / Q(h)\}^{**}$	$\tan B = 1 / \cos h \tan A$ $Q(B)=\{Q(A) / Q(h)\}^{**}$
h, B	$\sin b = \sin h \sin B$ $Q(b) = Q(B)$	$\tan a = \tan h \cos B$ $Q(a) = \{Q(B) / Q(h)\}^{**}$	$\tan A = 1 / \cos h \tan B$ $Q(A)=\{Q(B) / Q(h)\}^{**}$
A, B	$\cos a = \cos A / \sin B$ $Q(a) = Q(A)$	$\cos b = \cos B / \sin A$ $Q(b) = Q(B)$	$\cos h = 1 / \tan A \tan B$ $Q(h) = \{Q(A)\,Q(B)\}^*$

* $\{Q(x)\,Q(y)\} \equiv$ 1st quadrant if $Q(x) = Q(y)$, 2nd quadrant if $Q(x) \neq Q(y)$
** $\{Q(x) / Q(h)\} \equiv$ quadrant of x if $h \leq 90$ deg, quadrant opposite x if $h > 90$ deg

(a) *Nomenclature.*

(b) *Napier's rules.*

Figure H.3: *Napier's rules for a spherical triangle with a right angle (Wertz, 2009).*