

POLITECNICO DI TORINO

Master's degree in Aerospace Engineering



Master's degree thesis

**Design of Guidance Algorithms for
Mars Landing Applications**

Academic year 2021/2022

Supervisors:

Elisa Capello

Mauro Mancini

Author:

Federico Coccia

Abstract

Sixty years ago, the first landing attempt on the Red Planet was made by the lander that was part of the Soviet Unions mission called Sputnik 24. From that day to the present, several strides have been made in the field of precision landings on large planetary bodies having an atmosphere. This progression has permitted to reverse the trend that saw most of these manoeuvres fail and also to reach levels of efficiency that even allow rovers and drones to land. In addition, the great interest in the exploration of planets in the solar system, that has been recently ignited, has helped research in this field to become extremely fertile. Consequently, the world of Guidance, Navigation and Control (GNC) systems has also been facing a great development due to the fact that they are a key component for the success of these manoeuvres. Indeed, a great variety of proposals for new control logics and studies as far as these are concerned have emerged in recent years from universities and public agencies.

The future of GNC systems, however, seems to have taken a preponderant direction, converging toward the use of Zero Effort Miss - Zero Effort Velocity (ZEM-ZEV) type guidance algorithms, developed and now established for missile applications. These have numerous advantages that make them particularly attractive for powered descent phases in Entry, Descent and Landing (EDL) manoeuvres. ZEM-ZEV guidance algorithms also turn out to be particularly well suited to be hybridized with Sliding mode control laws, thus achieving excellent results as far as performance and robustness, both of which are important features for this applications. The purpose of this work is therefore to develop a landing simulator in the Matlab-Simulink environment by implementing in it a guidance law for position control and a control law for attitude control. The guidance law implemented for the trajectory is a hybrid algorithm between ZEM-ZEV and Sliding Mode, called Optimal Sliding Guidance and proposed by Daniel R. Wibben and Roberto Furfaro (Department of Aerospace and Mechanical Engineering, University of Arizona). On the other hand, the control law that deals with tracking the desired attitude is the Super-Twisting Sliding Mode. The performance of the aforementioned control systems is then observed by simulating a realistic Mars landing mission. Specifically, the mission referred to is NASA's Phoenix mission, which took place in May 2008, during which the Phoenix Mars Lander successfully landed on the Martian surface.

Keywords: Optimal Sliding Guidance, Closed-loop Guidance, Mars landing.

Acknowledgement

I would like to thank Professor Elisa Capello and doctoral student Mauro Mancini for assisting me with passion and dedication during the realization of this work. It is also appropriate to thank my family and the people who have stood by my side and have always supported me. Without them, this achievement would not have been possible.

Contents

Contents	IV
List of Figures	VI
List of Tables	VIII
1 Introduction	1
1.1 Research objectives	1
1.2 Mars, planet characteristics overview	3
1.3 Mars missions overview	5
1.3.1 1960-1969: The Space Race of the US and the USSR	5
1.3.2 1970-1989: First lander successfully landed on the Red Planet	5
1.3.3 1990-1999: The beginning of the rover era	6
1.3.4 2000-2009: The first European-flagged Mars mission	6
1.3.5 2010 - present: New protagonists join the exploration of the Red Planet .	7
1.4 Entry, Descent and Landing technique	10
1.4.1 Approach	10
1.4.2 Atmospheric Entry	11
1.4.3 Parachute Phase	12
1.4.4 Powered Descent	13
2 Mathematical models	15
2.1 Mars Surface model	15
2.2 Reference frames	17
2.2.1 M: Inertial Mars centered reference frame	17
2.2.2 T: Target centered reference frame	18
2.2.3 B: Spacecraft CoM centered frame	19
2.3 Spacecraft dynamics models	20
2.3.1 Spacecraft position dynamics	20
2.3.2 Spacecraft attitude dynamics	22
2.4 Target dynamics models	24
2.4.1 Target position dynamics	24
2.4.2 Target attitude dynamics	24
2.5 Spacecraft actuation system model	25
2.5.1 Thruster model	25
2.5.2 Reaction wheel model	26
2.5.3 Control logic of thrusters	27
2.5.4 Unification of thruster and reaction wheel models	29
2.6 Fuel consumption model	29
2.7 Mars disturbance	30
2.7.1 J_2 effect	30
2.7.2 Wind, atmospheric and aerodynamic drag models	31

2.7.3	Gravity gradient effect	37
3	Guidance algorithms	39
3.1	Position dynamics control	39
3.1.1	Zero-Effort-Miss/Zero-Effort-Velocity	39
3.1.2	Sliding Mode Control	41
3.1.3	Union of the two control logics.	43
3.2	Attitude dynamics control	44
3.2.1	Calculation of the reference quaternion	44
3.2.2	Super-Twisting Sliding Mode Controller	46
4	Simulations	47
4.1	Spacecraft and parameters selected	47
4.2	Missions and landing constraints definition	50
4.3	Nominal Mission	54
4.3.1	Initial conditions and mission overview	54
4.3.2	Guidance algorithm settings	55
4.3.3	Results and comments	56
4.4	Perturbed Mission	61
4.4.1	Initial conditions and mission overview	61
4.4.2	Guidance algorithm settings	62
4.4.3	Results and comments	62
5	Conclusions, future works and possible development	67
5.1	Conclusions	67
5.2	Future works and possible development	68
	Bibliography	69

List of Figures

1.1	To Mars and back: company's Space X project involving a round-trip mission to and from Mars using the Starship spacecraft and Falcon Heavy launcher.[13]	2
1.2	True color image of Mars taken by the OSIRIS instrument on the ESA Rosetta spacecraft during its February 2007 flyby of the planet. [2]	3
1.3	Infographic of all Mars missions.[3]	8
1.4	Mars launch windows over the last 20 years.[4]	9
1.5	Phoenix EDL sequence of events.[6]	10
1.6	Phoenix Spacecraft in Cruise Configuration.[8]	11
1.7	Phoenix parachute configuration.[8]	12
1.8	Phoenix Lander in deployed state on surface of Mars.[8]	13
2.1	Maps of Mars Global Topography. [22]	15
2.2	3D representation of Mars surface model implemented.	16
2.3	3D representation of Target longitude and latitude at time zero.	18
2.4	3D representation of Body reference system viewed from the Target reference system.	19
2.5	3D representation of positions of thruster in Body reference system.	25
2.6	Gravity field of Mars surface.	30
2.7	Average wind in South/North direction on Mars at 1000m altitude. [26]	32
2.8	Average wind in West/East direction on Mars at 1000m altitude. [26]	32
2.9	Average wind in vertical direction on Mars at 1000m altitude. [26]	33
2.10	Anemometer data measured in a 50 s long record and spectral density analysis in frequency domain. [28]	33
2.11	Value components of vector $\overline{V_{wind}}$ as seen from reference system T.	34
2.12	3D model of the spacecraft body implemented in the simulator.	36
2.13	Gravitational forces acting on a dumbbell-shaped satellite.	37
3.1	State-space representation in which reaching phase and sliding phase are clearly distinguished. [35]	41
3.2	Zoom of Figure 3.1 in which chattering effect is noticeable.	42
3.3	Two-dimensional representation of how $\widehat{z_B}$ versor and $\widehat{z_{cmd}}$ versor act.	45
4.1	Concept depicts NASA's Phoenix Mars Lander during touchdown on the arctic plains of Mars. [22]	47
4.2	Dimensionless drag coefficients derived from both CFD simulations and wind tunnel experiments.[41]	48
4.3	Map of Mars in which the landing sites of rovers and landers that have reached the surface of Mars are indicated. [42]	50
4.4	Position of reference system T in the Mars model implemented in the simulator.	51
4.5	A 2D representation of some Monte Carlo runs carried out by Jet Propulsion Laboratory. [43]	52
4.6	3D representation of the trajectory followed by the lander during the mission.	56

4.7	Evolution in time of the components of the spacecraft position and velocity vectors viewed from reference system T.	57
4.8	Evolution in time of the components of the Euler angles and attitude rates that describe the spacecraft attitude.	58
4.9	Evolution in time of the forces and moments actuated viewed from reference system B.	59
4.10	Attitude rates registered during a simulation of a powered descent phase of the Phoenix lander.[46]	60
4.11	3D representation of the trajectory followed by the lander during the mission.	62
4.12	Evolution in time of the components of the spacecraft position and velocity vectors viewed from reference system T.	63
4.13	Evolution in time of the components of the Euler angles and attitude rates that describe the spacecraft attitude.	64
4.14	Evolution in time of the forces and moments actuated viewed from reference system B.	65

List of Tables

1.1	Table comparing characteristics between planet Earth and Mars.	4
1.2	Initial Inertial Velocity and Flight Path Angle for MPF and MER missions.	10
2.1	Table comparing harmonic coefficents for Earth and Mars.	31
4.1	Table summarizing the characteristics of the Phoenix-like spacecraft implemented in the simulator.	49
4.2	Table summarizing requirements imposed for the simulated missions.	52
4.3	Table summarizing the initial conditions of the nominal mission.	54
4.4	Summary table of parameters related to the OSG and Super-Twisting Sliding Mode algorithms implemented in the simulator.	55
4.5	Table summarizing the initial conditions of the nominal mission.	60
4.6	Table summarizing the initial conditions of the perturbed mission.	61
4.7	Table summarizing parameters involved in touchdown.	66

Chapter 1

Introduction

1.1 Research objectives

The current historical period has seen a resurgence of public interest in space exploration comparable to that of the 1960s. Exploring and understanding what lies beyond the imaginary boundary marked by the Kármán line has returned mankind to momentous discoveries and has enabled an exponential technological evolution that continues to this day, i.e. the hundreds of scientific experiments and research conducted by astronauts in the International Space Station (ISS). In short, the space sector is currently rather fertile. Proof of this is that, in recent years, the private sector has entered into this field, an event that even just ten years ago was considered as impossible. The most shining example is that of tycoon Elon Musk's Space X company, which has greatly contributed to some of the latest achievements, such as the first successful landing and recovery of a first stage of a launcher.

Guidance, Navigation and Control (GNC) systems turn out to play a leading role in many types of missions and the ability to aspire to increasingly ambitious goals in the space field depends on their development. Indeed, these are capable of autonomously steering and guiding spacecrafts and other space modules by bringing them to a predetermined point. In fact, GNC systems have been used in missions involving intercept maneuvers, such as docking between two orbiting modules and landing or reentry maneuvers on planets with or without an atmosphere; as far as these are concerned, to be mentioned is the US Apollo 11 mission, which was one of the first to successfully exploit this technology.

The first types of GNC systems were open-loop type controls based on the implementation of predetermined commands on the ground. In fact, these commands do not depend on state feedbacks and are calculated off-line before the mission starts. The advantage brought by this control is computational lightness, which made them reliable and compatible with the computing power available in the 1970s. Some examples of this type of GNC system are *Off-line Optimization* and *On-board Tracking* and *Gravity Turn*. Later, algorithms capable of implementing commands calculated in real time during the mission and based on feedbacks received from sensors installed on the vehicle appeared. These can be divided into two main categories: explicit and numerical. The former define acceleration profiles based on solving a closed-form expression defined by the state variables measured by the sensors. These are computationally efficient, but the formulation of the closed-form expression requires that necessary assumptions be made and it is impossible to implement logic of collision avoidance and imposition of constraints. Moreover, they are non-optimized as far as fuel consumption is concerned. *E-Guidance* and *Apollo Guidance* are examples of this type of GNC. Numerical methods, on the other hand, are based on solving closed systems similar to those used by explicit methods and to which, however, fewer approximations are applied and whose limitations and collision avoidance logic can be implemented. Even so, the fact that they are computationally heavier can be mentioned. Some examples of this typology are *Convex Guidance* and *Gradient-based Optimization* [6].

There are many reasons and needs that push the development of new powered descent guidance algorithms forward. Certainly, among the most important ones, the fact that their efficiency is crucial to being able to transport payloads of increasing size and weight to the Red Planet at the expense of lower fuel consumption can be mentioned. This goal is critical both for exploration missions performed through increasingly complex and therefore heavier landers and rovers and also following recent statements by investors such as aforementioned Elon Musk, speaking of a future in which round-trip missions to and from Mars are possible, as can be noticed in Figure 1.1 taken from his company's website [13]. Such an objective requires the use of GNC systems capable of accomplishing high-precision landings both in the standpoint of final position and in the final velocity on which the violence of impact with the ground depends.

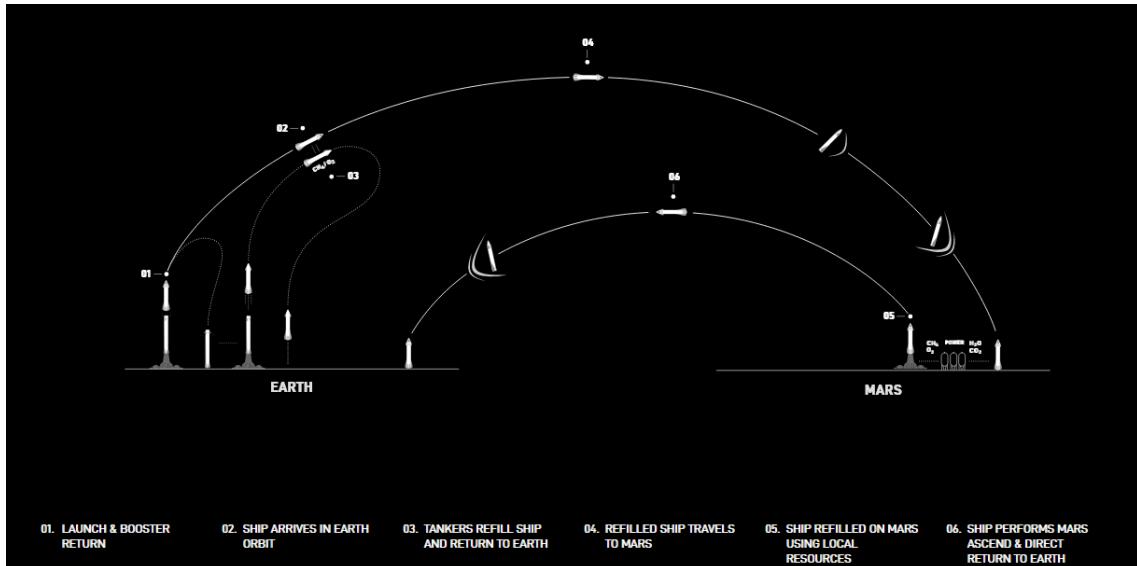


Figure 1.1: To Mars and back: company's Space X project involving a round-trip mission to and from Mars using the Starship spacecraft and Falcon Heavy launcher.[13]

Another crucial aspect that requires the use of increasingly complex and accurate algorithms in this type of mission is the distance from planet Earth. This detail makes remotely controlling the S/C impossible; therefore, it must be equipped with a self-sufficient and robust GNC system that allows it to meet landing requirements even despite disturbances and contingencies that cannot be calculated in advance, such as a poor landing site conditions that would force the S/C to evaluate a safer landing site. For these reasons, there has been a steady evolution in this field over the years.

This paper will discuss the Optimal Sliding Guidance, that is one of the latest Powered Descent Guidance Algorithms developed and proposed for this type of mission[15] [16]. The algorithms will then be implemented in a simulator developed in the Matlab-Simulink environment and applied to a case similar to NASA's Phoenix mission [17] [18] [19]. Then, some simulations of EDL missions will be run with different boundary conditions in order to refute the stability of the implemented control system. Thus, the purpose of this paper is to refute the robustness and effectiveness of the guidance algorithms based on ZEM/ZEV logic and Sliding Mode applied to the case of landings on planet Mars.

1.2 Mars, planet characteristics overview

Mars is the fourth planet by distance from the Sun in the solar system and it is positioned between Earth and the Asteroid belt that precedes Jupiter. It owes its name to the god of war in Roman mythology. The Martian surface is covered with a large amount of iron oxide, which, as can be seen from the photo taken by the Rosetta probe in 2007 and shown in Figure 1.2, gives it a reddish color, from which its nickname "Red Planet" derives.

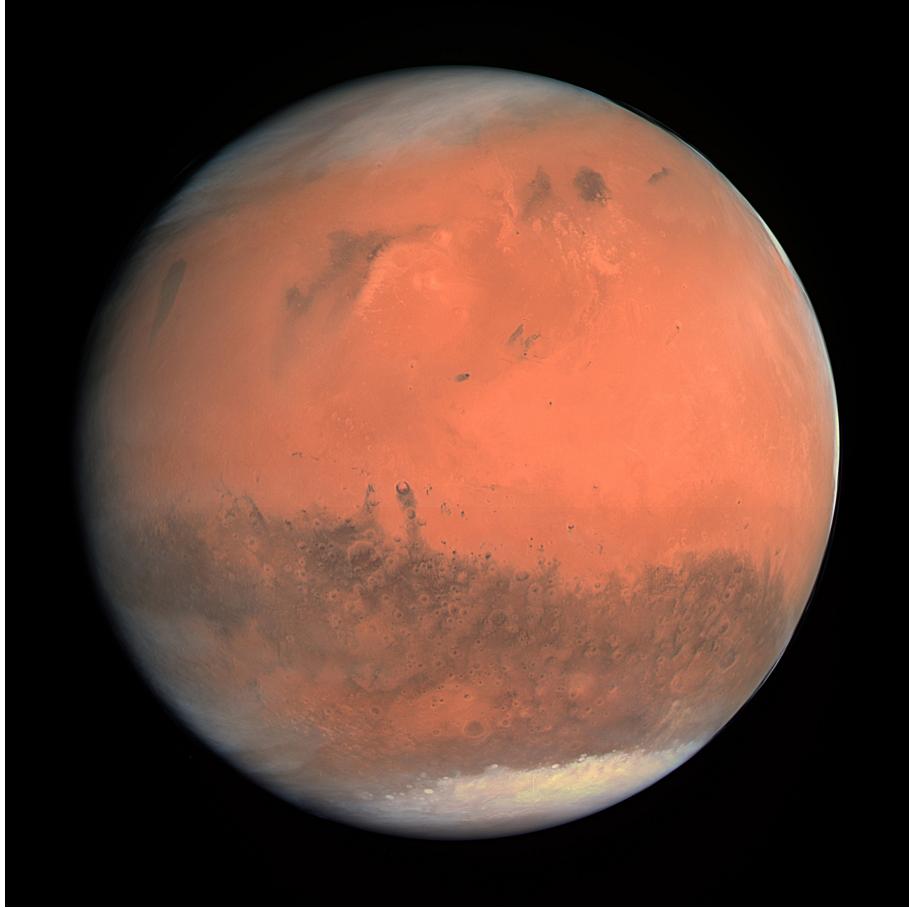


Figure 1.2: True color image of Mars taken by the OSIRIS instrument on the ESA Rosetta spacecraft during its February 2007 flyby of the planet. [2]

The atmosphere of Mars is composed of 95% carbon dioxide, whereas the remaining 5% is divided into nitrogen, argon, oxygen, carbon monoxide and, in a minority, of other elements among which water appears. The average density is about a quarter lower than the Earth's despite temperatures being markedly lower. Viking 1 lander recorded maximum daytime temperatures of -31°C and minimums of -89°C during its mission . In addition, the very low atmospheric pressure, about one hundredth of the Earth's, is prohibitive for the formation of liquid water. The winds that can occur have speeds ranging from 2m/s to 10m/s but can reach values of up to 30m/s during sandstorms. A Martian day is about as long as a day on Earth, with the former being longer than the latter by only about 37 minutes. The size of the Red Planet is smaller than that of Earth; in fact, the equatorial radius of the former is about half that of the latter. The mass of Mars follows this trend as well and it is therefore one-tenth of Earth's, resulting in a gravitational acceleration on the surface of 3.71m/s^2 [1].

The aforementioned values are captured in Table 1.1 along with other values of interest and which have been used in the implementation of the simulator that is going to be discussed in this paper.

Parameters	Units	Mars	Earth	Ratio Mars/Earth
Mass	$10^{24} kg$	0.64169	5.9722	0.107
Volume	$10^{10} km^3$	16.318	108.321	0.151
Equatorial radius	km	3396.2	6378.1	0.532
Polar radius	km	3376.2	6356.8	0.531
Ellipticity	<i>Flattening</i>	0.00589	0.00335	1.76
Surface gravity	m/s^2	3.71	9.80	0.379
Mean density	kg/m^3	3934	5513	0.714
Surface acceleration	m/s^2	3.69	9.78	0.377
Escape velocity	km/s	5.03	11.19	0.450
Standard gravitational parameter	$10^6 km^3/s^2$	0.042828	0.39860	0.107
Solar irradiance	W/m^2	586.2	1361.0	0.431
Black-body temperature	K	209.8	254.0	0.826
Moment of inertia	I/MR^2	0.366	0.3308	1.106
J_2 Harmonic coefficients	10^{-6}	1960.45	1082.63	1.811
Number of natural satellites	—	2	1	—

Table 1.1: Table comparing characteristics between planet Earth and Mars.

1.3 Mars missions overview

1.3.1 1960-1969: The Space Race of the US and the USSR

The history of Mars exploration began in the 1960s and the cause of emergence of this interest in exploring other planets can be attributed to the post-WW2 space race which occurred between the United States and the Soviet Union. In particular, this rivalry prompted the Soviet Union to attempt the first Mars flybys with the Marsnik 1 and Marsnik 2 probes in October 1960; however, both failed by not even reaching the orbit of the Earth. This failure did not deter the USSR, which attempted again flybys of the Red Planet in 1962 with Sputnik 22, Mars 1 and Sputnik 24 probes. Again, these were unsuccessful due to failures and breakdowns.

Later, in November 1964, the United States launched Mariner 3 and Mariner 4 probes within days from one another. Both reached Mars, but only Mariner 4 managed to deploy its solar panels and complete the mission; therefore, it is remembered as the first probe to have reached Mars and have sent 21 photos of the Martian surface back to Earth.

Some days later, the Russians launched the Zond 2 probe, which successfully made the flyby but was unable to send any data back to Earth due to a radio failure.

In the spring of 1969, the Americans launched two more probes, Mariner 6 and Mariner 7, which successfully reached Mars and managed to take more than 200 photos of the Martian soil.

1.3.2 1970-1989: First lander successfully landed on the Red Planet

1971 was a watershed year for both sides, as the goals became so ambitious that the Americans launched probes not only capable of flybying the planet, but also capable of setting up a stable orbit around the planet. Again, two orbiters, Mariner 8 and Mariner 9, were launched by the Americans; however, only the latter succeeded in completing the mission by being able to remain in orbit around the Red Planet for almost a year and to send back a large amount of images. The Russians went one step further by making the first attempts to land on the planet with landers. They launched the Lander Cosmos 419, which however never succeeded in leaving planet Earth, and two orbiters equipped with two landers; the names of these programs were Mars 2 and Mars 3. Of the latter, only Mars 3 successfully completed its mission by making the first landing on Mars. However, just 15 seconds after landing had succeeded, radio communication with the lander broke down. Two years later there were several launches by the Soviet Union, which undertook the Mars 4, 5, 6 and 7 missions. The first two were orbiters, whereas the last two were landers. Only the obiter successfully reached Mars correctly fitting into the orbit planned by Russian engineers, but losing communication with Earth after only 22 days due to a failure.

After a four-year hiatus, the Americans attempted a Mars docking for the first time in their history with the Vikings program. This involved the launch of two twin spacecrafsts equipped with both orbiter and lander modules developed by the US research center Jet Propulsion Laboratory (JPL). This was a success as both orbiters entered their intended orbits and both landers managed to land while remaining operational for more than six years, sending back a large amount of data about the Martian terrain and atmosphere.

In the summer of 1988, the Russians launched their last probes in history as USSR, that were part of the Phobos program and had the goal of exploring Mars and its two moons, Fobos and Deimos, through its two lander and obiter modules. Both probes ended their missions prematurely due to failures and an incorrect end-of-mission command received by the computer of the Phobos 1 probe.

1.3.3 1990-1999: The beginning of the rover era

The 1990s began with the launch of the Mars Observer orbiter made by Lockheed Martin and part of NASA's Planetary Observer program. The orbiter successfully reached Mars, but after a year of operation ceased communication with Earth, only partially achieving the mission's intended goals.

In 1996, an advantageous launch window encouraged Russia to attempt the Mars landing again with the Mars 96 mission, which involved the launch of a probe equipped with both a lander and an orbiter module. During the ascent phase, the Russian Proton launcher had some problems with the fourth stage. As a result, both the Russian orbiter and lander, which also contained several European experiments, were destroyed by crashing into the Pacific Ocean.

The fate of the two American programs Mars Global Surveyor and Mars Pathfinder was different. The Mars Global Surveyor mission involving the launch of an orbiter was so fully successful that the probe, having completed its main mission of mapping the Martian soil, was assigned secondary missions. The probe did not stop responding to commands until the winter of 2006. The Mars Pathfinder mission was a true revolution in space exploration as it was the first to have involved the use not only of a lander module but also of a rover called Sojourner. The latter was operational for about three months analyzing the physical properties of several sites near the lander.

In the late 1990s, three other probes were launched to Mars, two US ones being part of the Mars Climate Orbiter and Mars Polar Lander programs and a Japanese one being part of the Nozomi mission. All three missions were failures; still, the Nozomi had historical significance as it was the first mission not to have American or Russian origin.

1.3.4 2000-2009: The first European-flagged Mars mission

At the beginning of the new century and precisely in April 2001, NASA initiated the 2001 Mars Odyssey mission in collaboration with Lockheed Martin. It consisted in the launch of a probe orbiting Mars with the task of analyzing various features of the Red Planet. The 2001 Mars Odyssey probe is by far the longest-running probe ever launched to Mars; in fact, it has been operating as a repeater between Earth and several US rovers since having completed its primary analysis mission in 2007. It is expected to be able to get enough fuel to continue operating until 2025.

Two years later, the European Space Agency carried out its first launch of an orbiter and Beagle 2 lander that was part of the Mars Express program. Beagle 2 managed to land, but failed to deploy all of its solar panels, which prevented ground stations from communicating with the lander, that therefore remained inactive. On the other hand, the orbiter is to date still operational.

Within days of the launch of Mars Express, the United States also initiated the Mars Exploration Rover program. This involved sending two twin rovers called Spirit and Opportunity to Mars. The mission proved to be a great success as the two rovers, which were designed to operate for a duration of just 90 Martian days, remained active until March 2010 and June 2018 instead.

Then, NASA placed an orbiter in orbit around Mars again in 2005 with the Mars Reconnaissance Orbiter program. This aimed at scanning the planet's surface in order to find suitable landing sites for landers. The Mars Reconnaissance Orbiter probe is today still operational and orbiting the Red Planet.

In 2007, the US initiated the Phoenix mission having as its goal to bring a lander developed by the University of Arizona in collaboration with NASA and several other universities to Mars. The mission ended successfully and it was the first to have implemented a powered descent landing guided by a non open-loop GNC system .

1.3.5 2010 - present: New protagonists join the exploration of the Red Planet

The decade from 2010 to 2020 was a prosperous one for Mars exploration as many missions and programs by multiple nations have begun and are still in progress today. The first mission, which took place in 2011, was the Russian-originated Fobos-Grunt, which involved landing a lander on Fobos and placing the Chinese probe Yinghuo-1 in orbit around Mars. This mission failed as the probe and lander never reached the Red Planet, but crashed into the Pacific Ocean instead.

A few days after the launch of Fobos-Grunt, the launch of the US Curiosity rover, part of the Mars Science Laboratory program, occurred as well. The rover's landing, in addition to being successful, took place using the Entry, Descent, Landing (EDL) technique, which brought to the achievement of a level of accuracy which was not reachable in previous missions. The rover is still operational today and it is tasked with investigating Mars's past and present ability to sustain life.

In 2013, the first Indian-originated mission called Mars Orbiter Mission began. The program involved placing a probe in orbit around Mars. This mission was successful and the probe is still operational today.

In November 2013, the Americans placed a probe in orbit around the Red Planet as well with the task of analyzing the chemical composition of the Martian atmosphere. To date, the probe is still operational as a repeater between Earth stations and the rovers and landers on the Martian surface.

In March 2016, the ESA began the operational part of the ExoMars program, which involved landing a lander and placing an orbiter in orbit around Mars. The orbiter part of the mission was a success, whereas the lander failed to land properly and resulted in inoperability.

Two years later, in May 2018, NASA launched a lander and two CubeSats toward Mars, the latter having the goal of performing a flyby of the Red Planet. The flyby and lander landing parts of the mission were successful and, to date, the lander is still operational and sending data back to Earth.

The month of July 2020 was marked by the start of the operational phases of three programs by different nations. The first was the Dhabi-originated Emirates Mars Mission program, which involved placing the Hope probe in orbit around Mars. This entry was successful and to this day the probe is reported to be operational. The second space program was of Chinese origin and it was named Tianwen-1. This mission involved landing a rover and placing an orbiter in orbit around the Red Planet, both of which were successful. The latest program is NASA's Mars 2020. This mission turned out to be particularly interesting as it involved not only the landing of the Perseverance rover but also that of a small demonstration moving-wing drone called Ingenuity. Ingenuity is to date operational, and so is the Perseverance rover.

Below, in Figure 1.3, is an infographic produced by ESA summarizing all Mars missions. It is interesting to notice that from 1960 to the present, the success rate has significantly increased, despite the fact that more than 50 percent of Mars missions have failed.

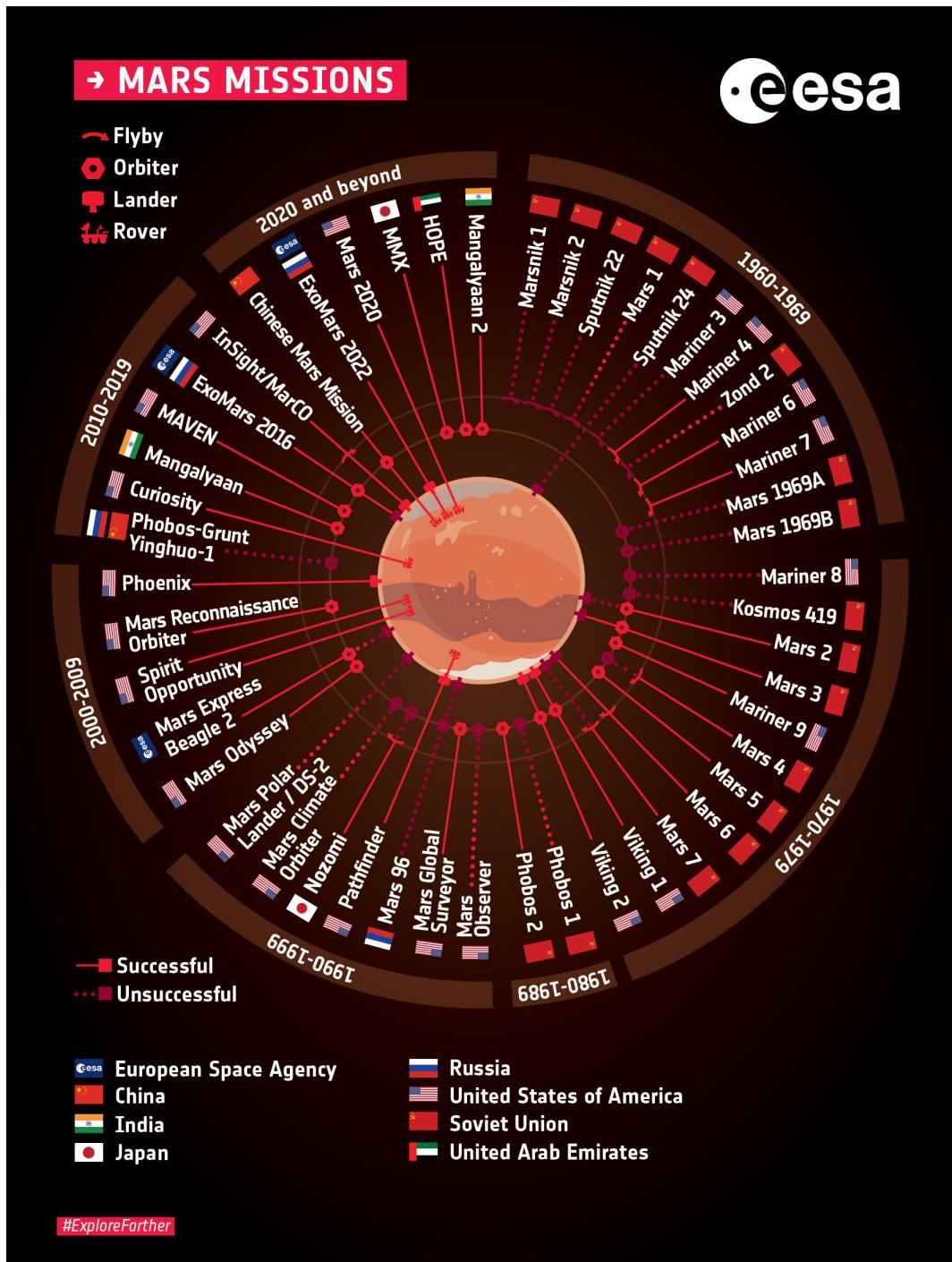


Figure 1.3: Infographic of all Mars missions.[3]

An interesting aspect that can be noticed by looking at the timeline of the launches to Mars is that these are made with an interval of roughly two years and two months. This is due to the fact that, as can be seen from Figure 1.4, the Red Planet enters in opposition with Earth every 2.1 years and it is at these times that launches becomes particularly advantageous in terms of distance between the two planets and, consequently, in terms of energy and timing as well.

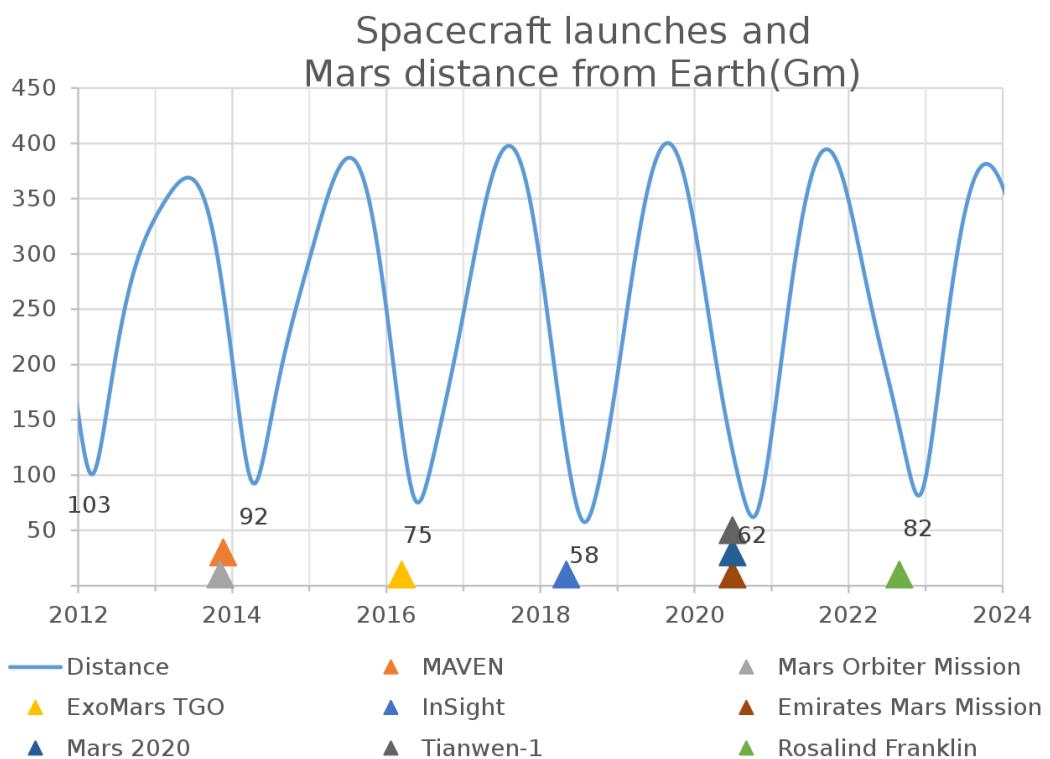


Figure 1.4: Mars launch windows over the last 20 years.[4]

1.4 Entry, Descent and Landing technique

With the succession in Mars exploration missions over time and the emergence of modules such as landers and rovers, high precision in landing expected for Martian exploration missions has become increasingly essential. It is for this reason that landing techniques and related Guidance, Navigation and Control (GNC) systems have been evolving in recent years. The state of the art of these techniques is represented by Entry, Descent and Landing (EDL), which provides a division of the landing phase into 4 maneuvers: Approach, Atmospheric Entry, Parachute Phase and Powered Descent. These will be analyzed in more detail in sections 1.4.1, 1.4.2, 1.4.3 and 1.4.4. In NASA's Phoenix mission [6], for instance, an EDL landing took place and its phases are depicted in Figure 1.5 and used as an example in the following subsections.

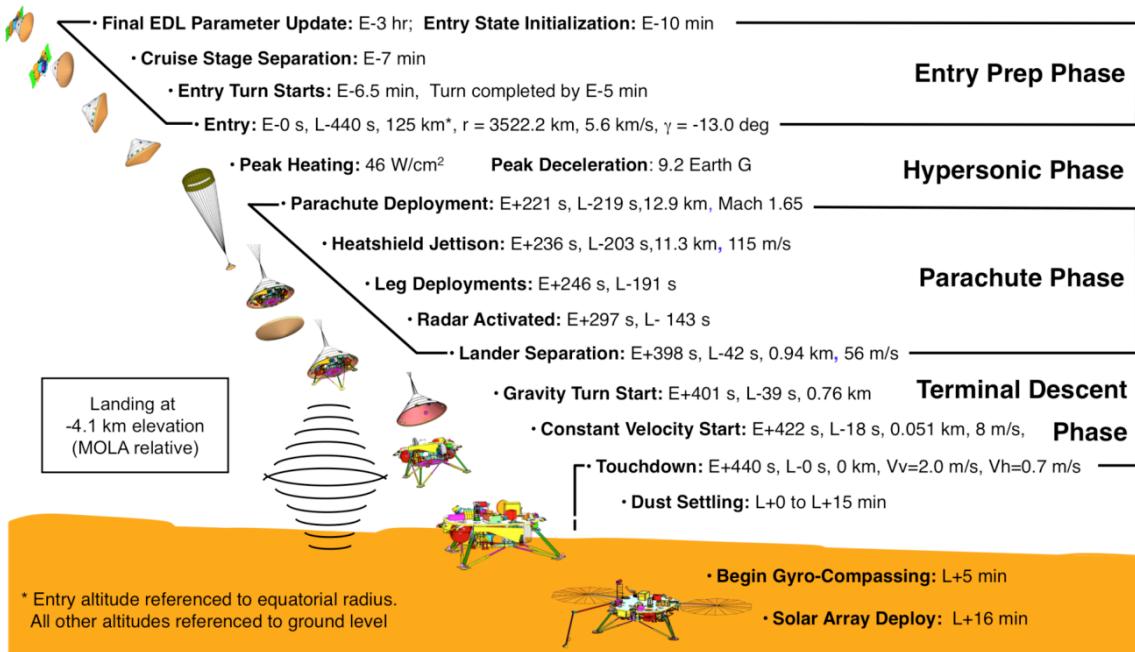


Figure 1.5: Phoenix EDL sequence of events.[6]

1.4.1 Approach

This first phase consists of the spacecraft maneuvers previous to entering the planet's atmosphere and, therefore, before passing the atmospheric interface which is conventionally located at an altitude of 125 km. This is because, at this altitude, the aerodynamic drag becomes significant in determining the trajectory of the vehicle. This phase is preceded by the cruise phase, that starts as soon as the S/C leaves Earth and that is the longest part of the mission. In the approach phase, the spacecraft must shed the solar panels, tanks and thrusters used in the previous phase and orient itself by acquiring the attitude that determines the precise initial Flight Path Angle (FPA) expected for the atmospheric entry.

Parameters	Units	MPF	MER-A	MER-B
Initial Inertial Velocity	km/s	7.26	5.63	5.70
Flight Path Angle	deg	-14.06	-11.5	-11.5

Table 1.2: Initial Inertial Velocity and Flight Path Angle for MPF and MER missions.

Typical FPAs chosen for this type of mission are usually between -10 deg and -15 deg. The right choice of this vestment is critical as the successful capture of the S/C by the target planet's atmosphere depends on it. In fact, too high a FPA can induce prohibitive decelerations for the structural strength of the shield. On the other hand, too small a FPA may not generate enough deceleration and, consequently, the vehicle may fail to stay in the planet's atmosphere and may regain altitude. At this stage, the vehicle possesses very high velocities, usually between 5.5 km/s and 8 km/s. Table 1.2 shows the FPA and Inertial Velocity values of the Mars Pathfinder (MPF) and Mars Exploration Rovers (MER) missions [5].

1.4.2 Atmospheric Entry

At this point, the lander safely remains inside a cone-shaped container called Aerobase. As can be seen from Figure 1.6a and 1.6b, the Aerobase is mainly made up of two parts: an upper one, called the Backshell, which is not directly affected by the flow, and a protective shield, called the Heatshield, which is located at the bottom [9]. The latter is able to withstand the very high temperatures, about 1300 degrees Celsius, that are reached due to atmospheric friction, which in this phase allows the S/C to drastically reduce its speed. During the descent, the Heatshield is oriented towards the flow so that it is loaded with the largest thermal loads. At first there was no control system for this stage, so a ballistic type entry that only depended on the initial conditions was used. Later, through experience, studies and research, it became clear that most of the uncertainty errors on the landing point came from this phase. It was therefore decided to implement control systems that would make the atmospheric entry guided. GNCs improved the atmospheric entry phase by implementing attitude changes from the equilibrium condition that result in changes in the orientation and modulus of the resultant aerodynamic force acting in the aerodynamic center of the vehicle. These types of GNCs usually act by either directly controlling the Bank angle through switching logic implemented during descent or through the offset of the S/C center of gravity that changes the equilibrium condition resulting in an attitude change [14]. Thus, by controlling the orientation of the aerodynamic forces, chasing a reference trajectory during ascent has become possible, drastically reducing the uncertainty errors on the final position.

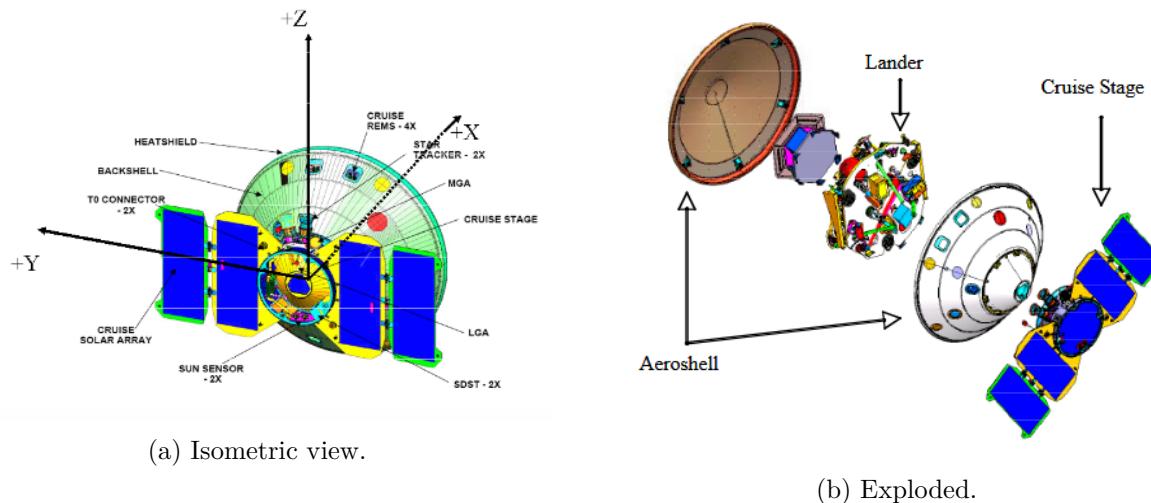


Figure 1.6: Phoenix Spacecraft in Cruise Configuration.[8]

1.4.3 Parachute Phase

The third phase begins with the deployment of parachutes, as shown in Figure 1.7 [10]. This operation, in order to be successfully performed, must be done at an altitude between 5 and 10 km, as in this range of height speeds are low and dynamic pressures are high. In this phase, there is no control over position and attitude and, in case it were preceded by a guided entry, this is the point in which the worst errors occur, because the vehicle is subject to disturbances that can't be contrasted. Usually, a few seconds after parachute deployment, the heat shield is released and allowed to plummet to the ground, so that the module's antennas contained in the Aeroshield remain uncovered, allowing the internal systems to calculate the position of the landing site relative to the S/C. The descent phase ends with the detachment of the Backshell to which the parachute is anchored; from this point on, the lander is totally uncovered.

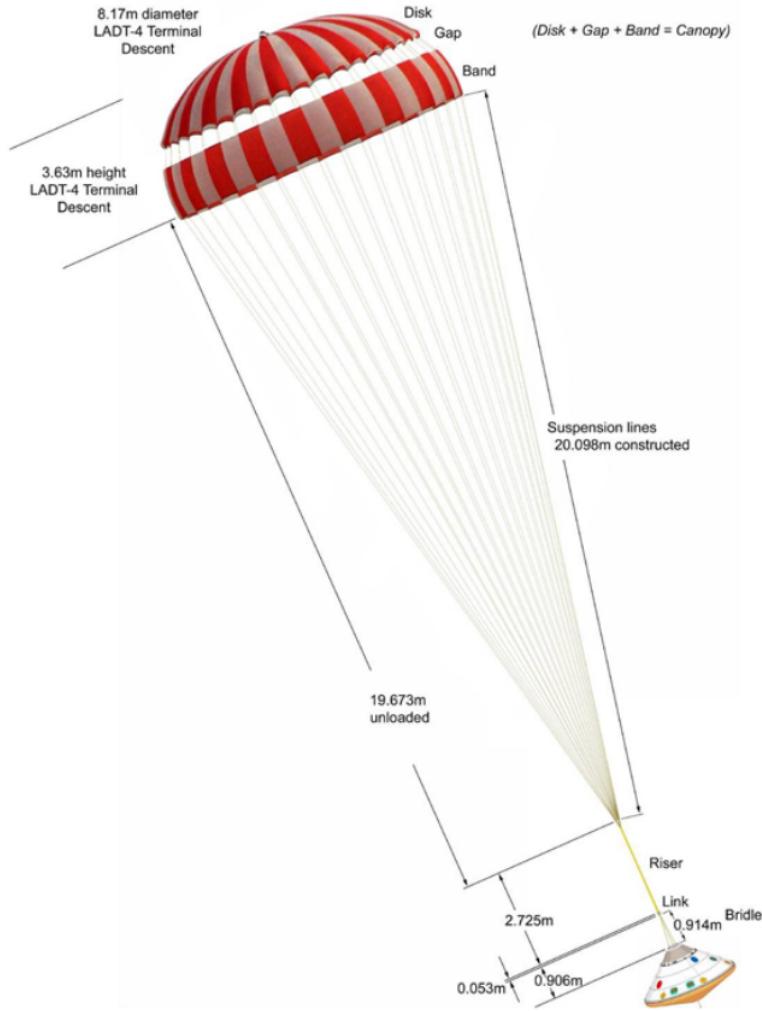


Figure 1.7: Phoenix parachute configuration.[8]

1.4.4 Powered Descent

The landing phase begins when the S/C, left in free fall after parachute release, fires up the engines at about 1 km altitude at a speed that is usually around 50 m/s. Thrusters are controlled by the GNC system, which is responsible for bringing the S/C to the predetermined point for landing with the smallest deviation possible in terms of meters. In addition, during touchdown there are constraints on the speed that the spacecraft must have. For instance, in the case of the Phoenix mission, shown in Figure 1.8 [11], the vertical velocity at the time of touchdown had to be about 2 m/s, whereas the velocity in the horizontal plane had to be lower than 0.7 m/s. This phase is crucial and, therefore, the algorithm must be able to make up for the errors accumulated in the previous phases and, more importantly, it must exhibit such a robustness that allows the lander to complete the mission despite disturbances or changes in initial conditions and to maintain good fuel efficiency [12]. The US Phoenix's Powered Descent phase was designed by dividing it into two subphases: the first one, called tip-up, which allowed the righting of the spacecraft and the second one, called Backshell Avoidance Maneuver (BAM), having the purpose of maximizing the distance between the landing point and the one in which the protective shield, released at the beginning of the maneuver, would crash. The maneuver performed by the Phoenix Terminal Descent lander lasted approximately 42 seconds and about 37.4kg of fuel was consumed in performing it.

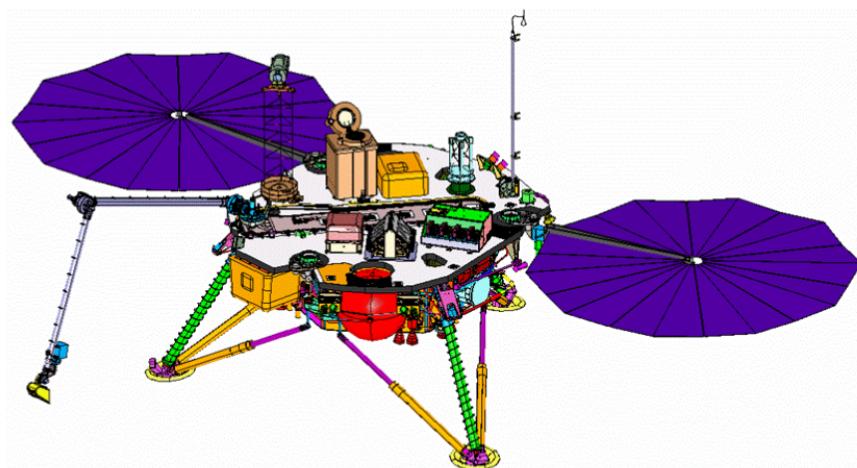


Figure 1.8: Phoenix Lander in deployed state on surface of Mars.[8]

Chapter 2

Mathematical models

2.1 Mars Surface model

The numerous missions undertaken to map the surface of Mars have stated that the surface of the planet is very irregular. Proof of this is the fact that it can boast the highest mountain as well as the widest volcano in the solar system, called Olympus Mons and having an height of more than 21 km, and also Valles Marineris, one of the largest canyons known so far [4]. In addition, the shape of the Red Planet is far from being perfectly spherical; in fact, it possesses a fairly pronounced ellipticity that is greater than that possessed by planet Earth. This detail makes the J_2 effect non-negligible, which will be discussed more in detail in section 2.7.1.

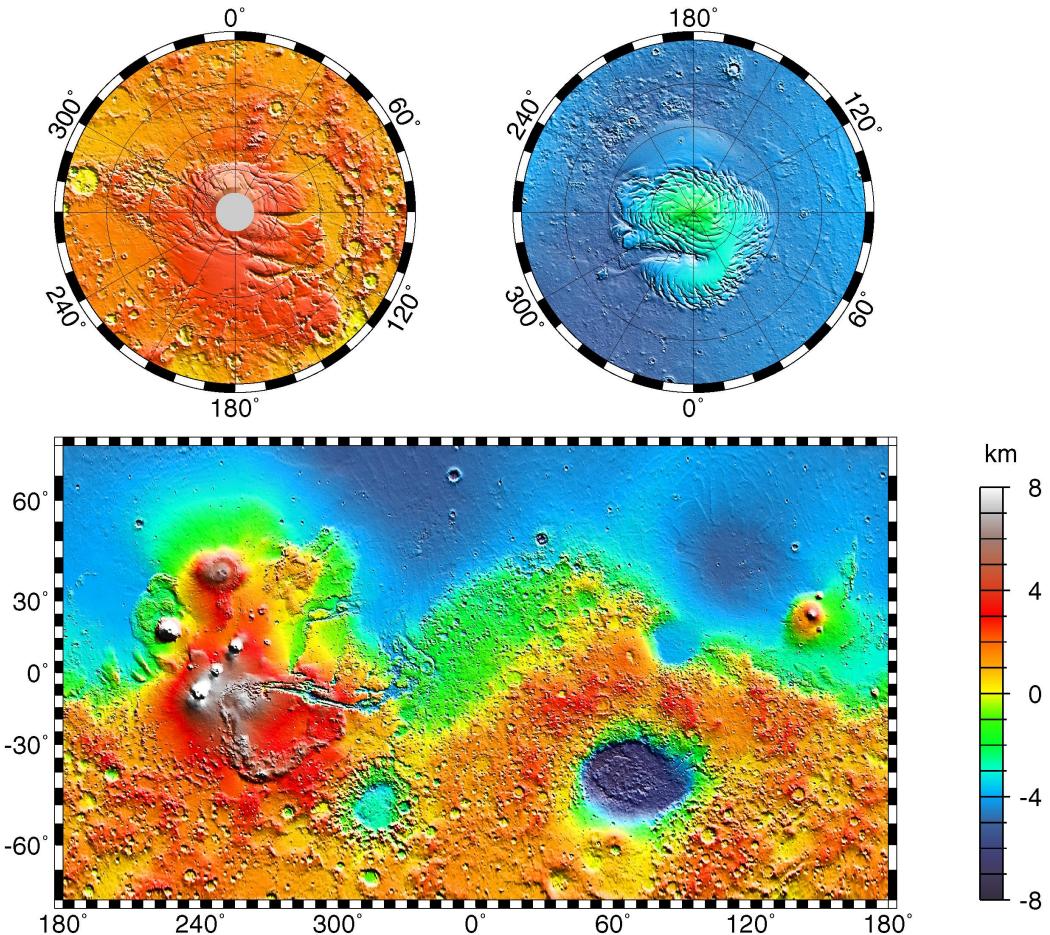


Figure 2.1: Maps of Mars Global Topography. [22]

However, the purpose of the simulator is to evaluate and prove the performance of guidance algorithms which have no obstacle identification and avoidance logic implemented. Due to this reason and in favour of greater computational lightness, the Martian surface has been approximated in the simulator as a perfect sphere with a radius equal to that in the equatorial plane. The assumption of perfect sphericity will generate errors and discrepancies in the determination of altitude but will not affect the gravitational model, which will also take into account the J_2 effect given by the planet's ellipticity.

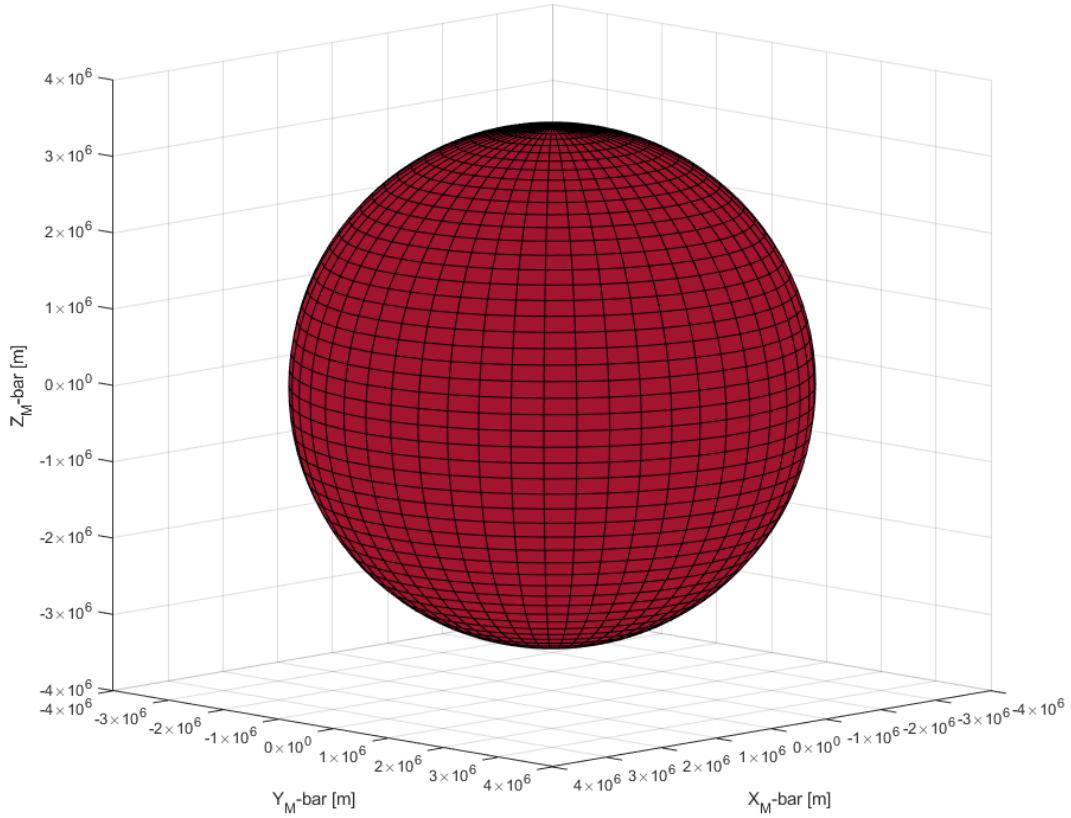


Figure 2.2: 3D representation of Mars surface model implemented.

2.2 Reference frames

This section will discuss in detail the reference systems that have been used in the simulator and with respect to which the equations of dynamics have been stated.

Therefore, the following reference systems are defined:

- M - Inertial Mars centered reference frame, drawn in black.
- T - Target centered reference frame, drawn in yellow.
- B - Spacecraft CoM centered frame, drawn in blue or, when necessary in order to better understand how the spacecraft is oriented, in more than one color.

2.2.1 M: Inertial Mars centered reference frame

The Inertial Mars centered reference frame, shortened M, is a reference frame placed in the center of mass (CoM) of Mars. Its axes are considered as fixed in space; this detail allows it to be taken like an inertial reference frame and therefore makes the formulation of the dynamics equation easier. This approximation leads to ignoring all the apparent accelerations coming from the fact that the CoM of Mars does not actually have a fixed position in space because it orbits around the Sun, which in turn is in motion within the Milky Way. Nevertheless, this approximation is acceptable for the physics of the mission to be simulated.

The z-axis is placed by making its orientation equal in direction but with opposite verse respect to vector $\overline{\omega_M}$ that describes the angular velocity of the planet. The opposite verse is due to the fact that the rotation of Mars is counterclockwise viewed from the top respect to the elliptical plane of the solar system. Consequently, vector $\overline{\omega_M}$ viewed from reference system M has its first two components null and its last one negative, taking the form that is given in Equation 2.1:

$$\overline{\omega_M} = \begin{bmatrix} 0 \\ 0 \\ \omega_{Mars} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -7.078e-5 \end{bmatrix} \frac{rad}{s} \quad (2.1)$$

Accordingly, the x-axis and y-axis lie on the plane, which is also the equatorial plane, perpendicular to the previously defined axis. In the end, the reference system is defined in its entirety by imposing that the x-axis, lying on the equatorial plane, is oriented so that it intersects the planet's surface at the point having zero latitude and zero longitude at the beginning of the simulation. Later, as time passes in the simulation, this reference system being defined as fixed in space and Mars rotating around the z-axis, this alignment will be lost.

In Figure 2.3, a 3D representation of an example of relative orientation between reference system M and reference system T at the initial time of a simulation can be observed. The initial position of the target was set as that identified by the following coordinates: 40 East - 45 North.

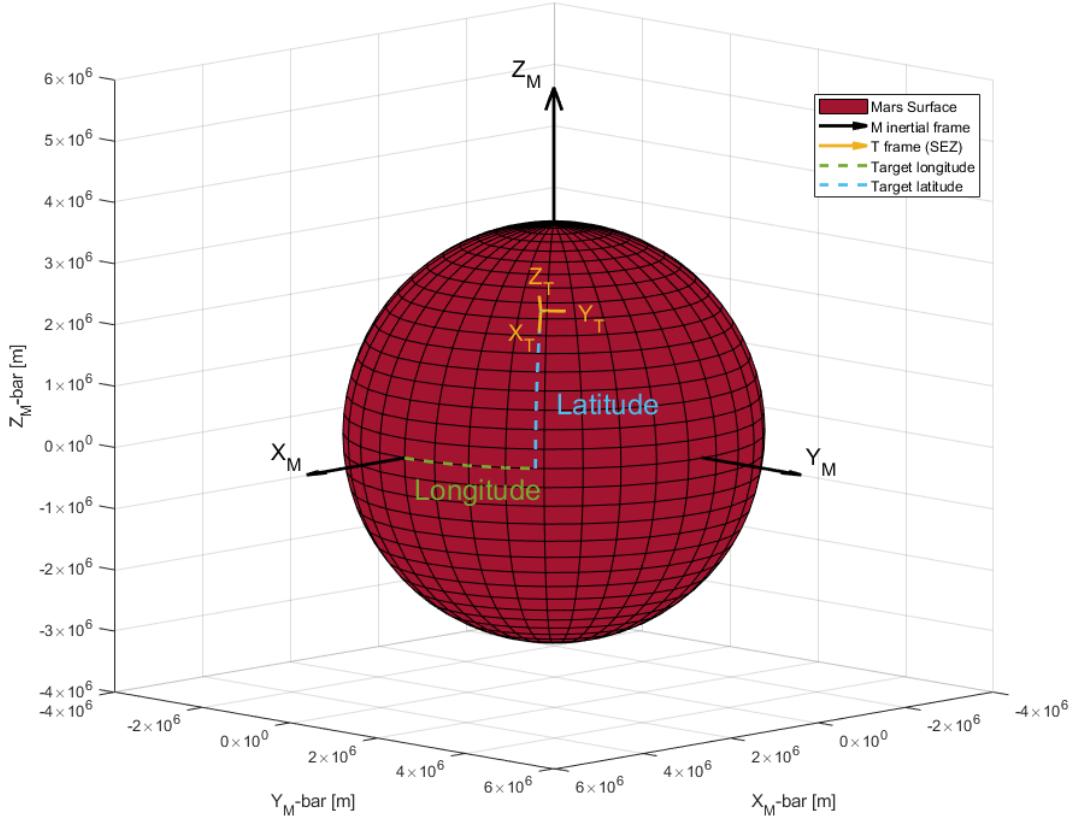


Figure 2.3: 3D representation of Target longitude and latitude at time zero.

2.2.2 T: Target centered reference frame

The target centred reference frame, shortened T, is centered at the point having the coordinates in terms of longitude and latitude equal to those of the S/C's intended landing point. The type of reference system chosen for the orientation of the axes of reference system T is South-East-Altitude (SEZ). This requires the x-axis to be oriented so that it points towards the local North direction, the y-axis to point towards the local East direction and, finally, the z-axis to be perpendicular to the previous two axes respecting the rule of the right hand. Reference system T is anchored to the Martian surface; therefore, during the simulation, it will rotate around the z-axis of the Martian Inertial reference system, varying the orientation of its axes in such a way that the characteristics of SEZ-type systems are always respected.

2.2.3 B: Spacecraft CoM centered frame

The Spacecraft CoM centred frame reference system is shortened in formulae to letter B. This abbreviation recalls the fact that this reference system is centred in the centre of mass of the spacecraft and is anchored in orientation to the spacecraft body. This way, defining the attitude of the S/C with respect to the M and T reference systems becomes possible by exploiting the rotation metrics from which the relative Euler angles and quaternion are derived. In particular, in the simulator, each operation and calculation involving the attitude dynamics of the various reference systems are expressed in terms of quaternions in order to avoid the problem of gimbal lock that arises in the case of rotation matrices derived from Euler angles. More in detail, this phenomenon appears when the pitch angle reaches 90 deg.

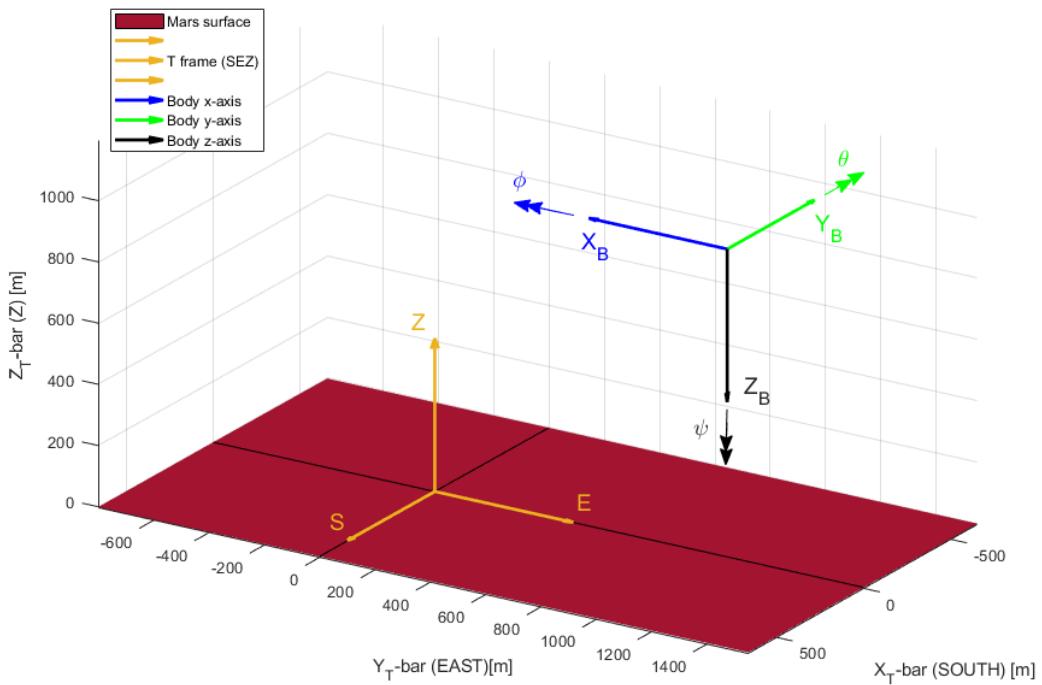


Figure 2.4: 3D representation of Body reference system viewed from the Target reference system.

As far as the orientation of the axes of the CoM-centered reference system of the spacecraft is concerned, it was defined in analogy with the conventions related to the aviation world. Such convention is that the x-axis is oriented towards the nominal direction of movement, the z-axis is oriented downwards and contained in the plane of symmetry of the aircraft and the y-axis is oriented accordingly to the right-hand rule. The Cartesian triplet oriented this way allows the three typical Euler angles to be used, as in the aviation field, in order to describe the attitude of the S/C. These three angles are:

ϕ - Roll angle

θ - Pitch angle

ψ - Yaw angle

An additional angle called heading angle is defined, which is the angle measured in the local horizontal plane between the x-axis of the body, anchored to the aircraft, and the local North direction. This angle is represented by the Greek letter Φ and it is useful in order to describe the direction of the S/C viewed from reference system M. In Figure 2.4 a 3D representation of an example of relative orientation between reference system T the reference system B at the initial time of a simulation can be observed.

2.3 Spacecraft dynamics models

This chapter discusses the models implemented in the simulator that allow the temporal evolution of spacecraft state variables to be simulated. This is done by discretizing time into constant intervals and solving the differential equations on which the above models are based through the use of numerical methods for solving ordinary differential equations (ODEs). The numerical method chosen for the simulator that is dealt with in this paper is the explicit Euler method. The reasons for this choice are several and include the fact that the typical duration of a Powered Descent phase is of the order of a minute, it is therefore permitted to choose an integration step that is very small; due to this reason, the use of a more complex method such as the implicit or variable integration step type turns out to be not so advantageous. The formulation of the explicit Euler method, on which the operation of the implemented simulator is based, is given below at equation 2.2 where this is used, as an application example, in order to calculate the value of a general variable y at a later step than the instantaneous simulation step:

$$y_{k+1} = y_k + \Delta t \cdot f(t_k, y_k) \quad (2.2)$$

Where y_{k+1} is the state variable to be calculated at integration step $k + 1$ and y_k is the known value of the state variable at integration step k . This value is known because it was calculated in the previous step at time t_k . Δt is the time interval between two integration steps. This parameter is of great importance because too large an integration step could lead to the emergence of numerical instabilities. On the other hand, if too small an integration step is chosen, the computational burden could be so high that it would result in very long times for computing the solution. $f(t_k, y_k)$ is the derivative of the state variable of interest computed at time t_k , that is known. Thus, the concept is to approximate the problem by dividing it into time intervals and, by linearizing, to successively compute the time evolution of the variables.

At last, the explicit Euler method with constant integration step of 0.001 seconds was chosen for the simulator. Below, in Sections 2.3.1 and 2.3.2, the implementation of the models that interface with this numerical method for time integration of the variables describing the spacecraft state are discussed in detail.

2.3.1 Spacecraft position dynamics

The model of spacecraft position dynamics in the simulator was implemented based on the universal gravitational law formulated by Isaac Newton [20], given in vector form at equation 2.3. In the application of the gravitational model thus expressed, some approximations have been made. Indeed, the gravitational actions generated by Mars's two natural satellites, Deimos and Phobos, have been ignored. Moreover, as a consequence of the fact that the model assumes the mass of the attractive planet and the attracted body as concentrated in the center of mass, the J_2 effect is ignored. This has later been implemented as a disturbance and will be dealt with in detail in Section 2.7.1. The model therefore receives it as input embedded in the $\overline{F_{J_2}}$ vector.

$$\overline{F_g} = -\frac{G \cdot Mass_{Mars} \cdot Mass_{SC}}{R_{SC}^2} \cdot \frac{\overline{R_{SC}}}{R_{SC}} \quad (2.3)$$

Where $\overline{F_g}$, expressed in Newtons, is the gravitational force acting on the mass $Mass_{SC}$, representing the spacecraft. G is the universal gravitational constant, which equals approximately $6.674 \frac{N \cdot m^2}{kg^2}$ and $Mass_{Mars}$ is the mass of the attracting body, which in this case is planet Mars and equals approximately $6.4169 \cdot 10^{23} kg$. For convenience, the mass of the main body and the universal gravitational constant are often multiplied together defining a new term: the standard gravitational parameter μ , which in the case of the Red Planet is equivalent to $4.282837 \cdot 10^{13} \frac{m^3}{s^2}$. The mass of the attracted body, which is the spacecraft, is represented by parameter $Mass_{SC}$, measured in kg . The last term, $\frac{\overline{r}}{r}$, is the versor that allows to give direction to the force $\overline{F_g}$, which turns out to be directed towards the center of mass of planet Mars.

Before proceeding with integration and thus determining the velocity and position of the spacecraft, it is necessary to derive the resultant of all the acting forces applied to the vehicle's center of mass. This operation results in a simple vector sum of all the forces, which must be expressed with respect to the same reference system. The reference system chosen for this application is the inertial one centered in the center of mass of planet Mars. Below, at equation 2.4, is the second-degree differential equation implemented in the simulator.

$$\overline{A_{SC}} = \overline{\dot{V}_{SC}} = \overline{\ddot{R}_{SC}} = \frac{\overline{F_{Act}} + \overline{F_{Drag}} + \overline{F_g} + \overline{F_{J2}}}{Mass_{S/C}} \quad (2.4)$$

with:

$$\overline{F_{Act}} = \overline{\overline{L}_{MB}} \cdot \overline{f_{Act}} \quad (2.5)$$

$$\overline{F_{Drag}} = \overline{\overline{L}_{MB}} \cdot \overline{f_{Drag}} \quad (2.6)$$

Where $\overline{A_{SC}}$, $\overline{V_{SC}}$ and $\overline{R_{SC}}$ are respectively the acceleration, velocity and position vectors seen from reference system M and measured with the units proposed by the International System of Measurement. The acceleration vector and the velocity vector act in the center of mass of the spacecraft, whereas the position vector is the vector starting from the origin of reference system M and pointing towards the origin of reference system B. $\overline{\overline{L}_{MB}}$ is the rotation matrix that is able to convert the components of a vector seen from B reference system to M reference system. The vector containing the contribution in terms of forces generated by the spacecraft propulsion system is $\overline{F_{Act}}$. This vector turns out to be expressed with the coordinates related to reference system B, so it is necessary to convert it by multiplying for $\overline{\overline{L}_{MB}}$. $\overline{F_{Drag}}$ is the vector containing the aerodynamic forces acting on the spacecraft generated by the difference in velocity between itself and the flow in which it is immersed. This vector is expressed with the coordinates relative to the body's reference system as well, so it should be converted as previously done to $\overline{F_{Act}}$. $\overline{F_g}$ is the gravity force vector defined by the universal gravitational law, defined earlier in equation 2.3. $\overline{F_{J2}}$ is the vector containing the contributions given by the J_2 effect, to be added to the gravity force vector $\overline{F_g}$. All four parameters previously mentioned and referring to forces are measured in Newtons. $Mass_{SC}$, as mention above, is the mass of the spacecraft. It is dealt with as a state variable since it is received as input and can decrease during the simulation, thus simulating fuel consumption.

2.3.2 Spacecraft attitude dynamics

The attitude dynamics of the spacecraft, similarly to what has been done for that of position, was implemented by taking system M as reference. In fact, it was decided to use the Euler equations in order to derive the angular velocities by working with vectors expressed in the Body reference system since, by referring to it, the moments generated by the propulsion system and any reaction wheel turn out to be easily expressed. As far as kinematic closure equations are concerned, on the other hand, the use of quaternions was opted for, thus making avoiding the phenomenon of indetermination of attitude angles given by the gimbal lock possible. Secondly, Euler angles are still derived from the quaternions by exploiting special blocks present in the Simulink environment. The following subsections will discuss in detail how these concepts have been implemented in the simulator.

Euler's equation

Euler's equation is a differential equation allowing, knowing the moments and angular moments applied to the spacecraft, to derive the angular velocity vector $\overline{\omega_{SC}}$. In the simulator, this equation has been implemented in order to work with vectors and matrices expressed with coordinates referenced to reference system B. By doing so, a much more streamlined formulation is obtained, since both control actions generated by the actuators and the inertia tensor can be used without having to perform additional actions such as multiplication with a rotation matrix. Following the differential Equation 2.8 is the vector equation implemented in the simulator, based on and developed from the more streamlined Equation 2.7:

$$\dot{\overline{H_{SC}}} = [\overline{M_{TH}} + \overline{M_D} - \overline{\omega_{SC}} \wedge (\overline{L_{SC}} + \overline{H_{RW}})] \quad (2.7)$$

$$\dot{\overline{\omega_{SC}}} = \overline{\bar{J}}^{-1} \cdot [\overline{M_{TH}} + \overline{M_D} - \overline{\omega_{SC}} \wedge (\overline{\bar{J}} \cdot \overline{\omega_{SC}} + \overline{H_{RW}})] \quad (2.8)$$

Where $\overline{\omega_{SC}}$ is the vector describing the angular velocity expressed in rad/s and possessed by the spacecraft. $\overline{\bar{J}}$, on the other hand, is the inertia tensor that has the form of a square matrix of dimension 3×3 containing the moments of inertia measured in $kg \cdot m^2$. This matrix carries information about how the mass of the spacecraft is distributed around the axes of the Body reference system and allows, knowing the angular velocity vector $\overline{\omega_{SC}}$, to calculate the angular momentum $\overline{H_{SC}}$, measured in $kg \cdot m^2 \cdot rad/s$ and possessed by the spacecraft. In Equation 2.9, the aforementioned correlation between these two quantities is given. $\overline{M_{TH}}$ and $\overline{M_D}$ are, respectively, the moments applied by the actuators, which can be either thrusters or reaction wheels, and the external disturbance moments. Both are measured in $N \cdot m$. Lastly, there is term $\overline{H_{RW}}$, which has the typical unit of measurement for angular moments and represents the angular velocity possessed by any body internal to the spacecraft, such as reaction wheels may be.

$$\begin{bmatrix} H_{SC_x} \\ H_{SC_y} \\ H_{SC_z} \end{bmatrix} = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{xy} & J_{yy} & J_{yz} \\ J_{xz} & J_{yz} & J_{zz} \end{bmatrix} \cdot \begin{bmatrix} \omega_{SC_x} \\ \omega_{SC_y} \\ \omega_{SC_z} \end{bmatrix} \quad (2.9)$$

Kinematic closure equations

These first-order differential equations allow to determinate the time evolution of the attitude of one reference system with respect to another one. In this case, the system with respect to which the dynamics of the Body reference system have to be made propagate in time is reference system M. There is a variety of ways to describe the orientation of the axes of one reference system with respect to one another; in the simulator implemented in this paper, Euler angles and quaternions have been chosen for this purpose. The former will only serve as output since they are much more intuitive than quaternions. As can be seen from equation 2.10, the latter are more complex since, unlike the contenders which involve defining only three parameters, a quaternion turns out to be a vector consisting of four elements. These parameters carry information that makes moving from one reference system to another possible by making a single rotation α around the Euler axis \hat{e} . This way, the indeterminacy given by the gimbal lock phenomenon that might be encountered by using Euler angles is avoided; however, complexity and computational weight are increased.

$$\bar{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos \frac{\theta}{2} \\ e_1 \cdot \sin \frac{\theta}{2} \\ e_2 \cdot \sin \frac{\theta}{2} \\ e_3 \cdot \sin \frac{\theta}{2} \end{bmatrix} \quad (2.10)$$

The next step is to solve the first-order differential equation given in Equation 2.11 and, lastly, to calculate the rotation matrix $\overline{\overline{L}}_{BM}$ by solving Equation 2.12 [21]. The rotation matrix thus found is of fundamental importance when multiplied by a vector expressed with coordinates referring to the Mars reference system because it allows the aforementioned vector to be translated into coordinates expressed in the Body reference system.

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \cdot \begin{bmatrix} 0 & -\omega_{XB} & -\omega_{YB} & -\omega_{ZB} \\ \omega_{XB} & 0 & \omega_{ZB} & -\omega_{YB} \\ \omega_{YB} & -\omega_{ZB} & 0 & \omega_{XB} \\ \omega_{ZB} & \omega_{YB} & -\omega_{XB} & 0 \end{bmatrix} \cdot \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (2.11)$$

$$\overline{\overline{L}}_{BM} = \begin{bmatrix} 2 \cdot q_0^2 - 1 + 2 \cdot q_1^2 & 2 \cdot q_1 \cdot q_2 + 2 \cdot q_0 \cdot q_3 & 2 \cdot q_1 \cdot q_3 + 2 \cdot q_0 \cdot q_2 \\ 2 \cdot q_1 \cdot q_2 + 2 \cdot q_0 \cdot q_3 & 2 \cdot q_0^2 - 1 + 2 \cdot q_2^2 & 2 \cdot q_2 \cdot q_3 + 2 \cdot q_0 \cdot q_1 \\ 2 \cdot q_1 \cdot q_3 + 2 \cdot q_0 \cdot q_2 & 2 \cdot q_2 \cdot q_3 + 2 \cdot q_0 \cdot q_1 & 2 \cdot q_0^2 - 1 + 2 \cdot q_3^2 \end{bmatrix} \quad (2.12)$$

2.4 Target dynamics models

Since the dynamics of the spacecraft are simulated by referring to reference system M, developing simulator components capable of determining the position, velocity and attitude that system T assumes during the mission is necessary. In short, reference system T is dealt with as if it were a second moving body viewed from an inertial point of view. Following Sections 2.4.1 and 2.4.2 will deal with the equations used in the simulator.

2.4.1 Target position dynamics

Reference system T turns out to have its origin coincident with the intended landing point, which is a fixed point relative to the Martian surface. This dictum allows to implement in a different way the subsystem that will have the task of determining the time evolution of the position of the origin of reference system T as seen from reference system M. In fact, the followed procedure was starting from the determination of the position and then deriving in order to obtain velocity and acceleration. This has been done because the initial latitude and longitude of the landing point, which in the formulae are defined as lat_T and lng_T , had previously been defined. Having applied the approximations mentioned in Section 2.2.1, the time evolution of the position of T turns out to be defined in its entirety by vector equation 2.13.

$$\overline{R_T} = \begin{bmatrix} X_T \\ Y_T \\ Z_T \end{bmatrix} = \begin{bmatrix} R_M \cdot \cos lat_T \cdot \cos t \cdot \omega_M + lng_T \\ R_M \cdot \cos lat_T \cdot \sin t \cdot \omega_M +{lng_T} \\ R_M \cdot \sin lat_T \end{bmatrix} \quad (2.13)$$

Where R_M is the radius of planet Mars measured in meters, whereas t is the simulation time measured in seconds. Finally, for convenience, the derivation to find velocity and acceleration, that is reported at Equation 2.14, has been done by using blocks in the Simulink library.

$$\overline{A_T} = \overline{\dot{V}_T} = \overline{\ddot{R}_T} \quad (2.14)$$

2.4.2 Target attitude dynamics

The setup of reference system T is simulated similarly to that of the Body reference system, but by applying some simplifications. In fact, there is no need in this case to employ the Euler equations because the angular velocity of this system is known as it is equal to the inverse of the rotational velocity of the planet Mars. This is due to the fact that, as defined in 2.2.2, reference system T turns out to have x and y axes tangent to the Martian surface, which has been defined as perfectly spherical. Consequently, directly going to the resolution of the equations for kinematic closure and determination of the rotation matrix is possible. The aforementioned matrix is called $\overline{\overline{L}_{MT}}$ and makes translating a vector expressed in coordinates referring to the T reference system into one expressed in coordinates referring to the M reference system possible. The equations are thus the same as those used in section 2.3.2 and are a repetition of equations 2.11 and 2.12.

2.5 Spacecraft actuation system model

Below, in Sections 2.5.1 and 2.5.2, formulae and models implemented in the simulator to replicate the performance of actuators expected to be used for missions involving landing landers on planets with an atmosphere are discussed. This model was devised by having control system, receiving as input vector \bar{U}_{cmd} containing forces and moments commanded by the control system, apply to it the various physical constraints typical of actuators. Thus, vector \bar{U}_{Act} containing forces and moments that are actually applied to spacecraft structure and thus to reference system B is provided as output.

2.5.1 Thruster model

The main actuators used for this type of mission are thrusters powered by fuels that can generate large thrusts while minding to have high specific pulses. Speaking about these, in fact, the higher they are the more they allow, for the same thrust generated, a lower fuel consumption. This way, the weight of the lander at the moment of launch is reduced as much as possible. A great variety of this type of spacecraft possesses thrusters installed on the bottom of the structure and oriented in concordance with the normal of the latter, which is oriented in concordance with the z Body axis, according to aeronautical conventions. This is because the main purpose of these actuators is to slow the spacecraft down. Expected velocity deltas for landing maneuvers of this type are on the order of $60m/s$, which roughly corresponds to the velocity possessed by the vehicle at the time of parachute release. These actuators are usually about ten in number and are arranged axis-symmetrically, so that through modulation of the thrust produced by each of these, generating yaw and roll moments is possible, thus allowing the spacecraft's trajectory to be controlled.

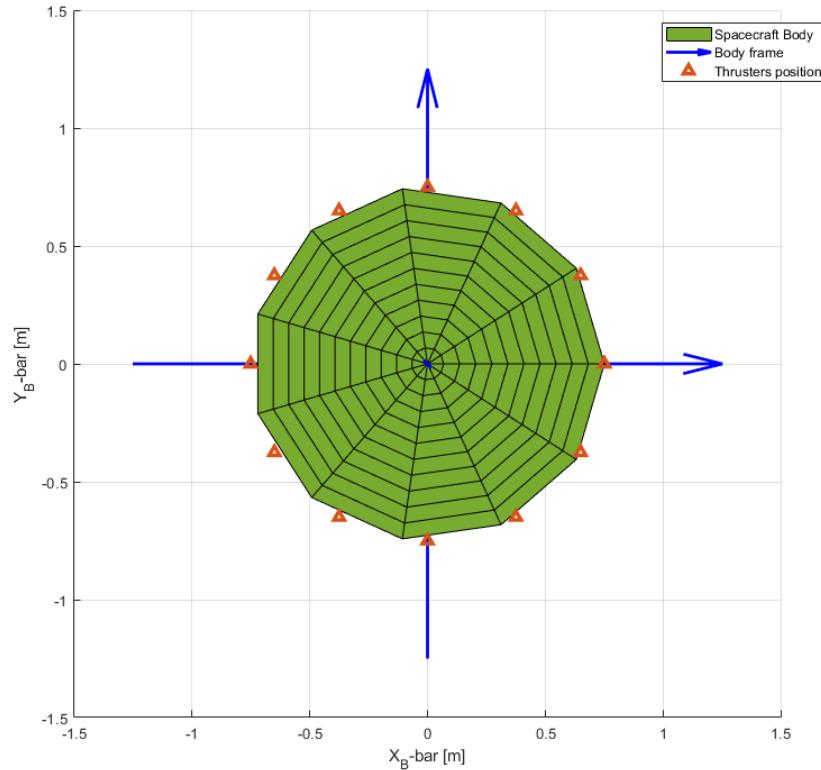


Figure 2.5: 3D representation of positions of thruster in Body reference system.

In the simulator, the propulsion system model was designed following this concept and inspired by the propulsion system of NASA's Phoenix Lander [23]. In fact, as can be seen from Figure 2.5, twelve thrusters were implemented and arranged equidistant from each other at a distance of r_{TH} to the spacecraft's center of mass and oriented in concordance with the z-axis of the Cartesian tern of the body reference system. A 12×2 matrix containing the x and y coordinates in body axes of each of the thrusters called \overline{ThPosM} was then defined.

$$\overline{ThPosM} = r_{TH} \cdot \begin{bmatrix} \cos(0) & \sin(0) \\ \cos\left(\frac{\pi}{6}\right) & \sin\left(\frac{\pi}{6}\right) \\ \cos\left(\frac{2\pi}{6}\right) & \sin\left(\frac{2\pi}{6}\right) \\ \vdots & \vdots \\ \cos\left(\frac{11\pi}{6}\right) & \sin\left(\frac{11\pi}{6}\right) \end{bmatrix} \quad (2.15)$$

The type of thrusters usually employed for this type of mission, which in the case of the Phoenix Lander is the Aerojet monopropellant hydrazine, cannot provide more than a maximum thrust value and cannot shut down completely. This aspect was implemented in the simulator by inserting saturations that would limit the thrust produced by each individual thruster. In addition, at the aim of best reproducing the performance and evaluating the stability of the implemented algorithms, first-degree transfer functions reported at equation 2.16 were also added.

$$G_{TH} = \frac{1}{\tau_{TH} \cdot s + 1} \quad (2.16)$$

Where τ_{TH} is the time constant that governs the delay effect in thruster actuation.

2.5.2 Reaction wheel model

The model of this actuator was implemented in the simulator by simply implementing Equation 2.17. In fact, the latter equalizes the $M_{Z_{cmd}}$ component of the $\overline{U_{cmd}}$ vector, which represents the moment that is required by the guidance algorithm that controls the attitude, with $M_{Z_{act}}$, which instead represents the yaw moment that the actuating system actually managed to generate. Obviously, the latter was saturated by preventing it from generating moments greater than those allowed by the type of reaction wheel chosen.

$$M_{Z_{act}} = M_{Z_{cmd}} \quad (2.17)$$

In addition, it was also necessary to calculate the angular momentum $\overline{H_{RW}}$ possessed by the reaction wheel put into rotation in order to account for it in the equation of attitude dynamics 2.8. This was done simply by implementing differential equation 2.18.

$$\dot{H}_{RW} = M_{Z_{act}} \quad (2.18)$$

Similarly to what was done for thrusters, it was decided that the output momentum generated by the reaction wheel should be filtered through the use of a first-order transfer function analogous to equation 2.16, but using a different time constant τ_{RW} , which, however, since the reaction capacity of the reaction wheels is higher than that of the thrusters, will be more restrained.

$$G_{RW} = \frac{1}{\tau_{RW} \cdot s + 1} \quad (2.19)$$

2.5.3 Control logic of thrusters

Before explaining how the contributions of thrusters and reaction wheels are summed, explicating the control logic devised in order to determine the value of thrust that each of the twelve thrusters must generate to satisfy the demands contained in the vector of commands received as input is necessary. It was thought, for simplicity, to interpret thrust modulus as consisting of two contributions, the first one called T_{avr} , being common to all thrusters and calculated based on the force along the z Body axis required by the control system and the second one, called dT_{Th_i} , being defined in such a way as to generate the required roll and yaw moments without generating additional thrust beyond that required.

$$\overline{U_{cmd}} = \begin{bmatrix} F_{X_{cmd}} \\ F_{Y_{cmd}} \\ F_{Z_{cmd}} \\ M_{X_{cmd}} \\ M_{Y_{cmd}} \\ M_{Z_{cmd}} \end{bmatrix} \quad (2.20)$$

$$\overline{T} = \begin{bmatrix} T_{Th_1} \\ T_{Th_2} \\ T_{Th_3} \\ T_{Th_4} \\ T_{Th_5} \\ T_{Th_6} \\ T_{Th_7} \\ T_{Th_8} \end{bmatrix} = \begin{bmatrix} dT_{Th_1} \\ dT_{Th_2} \\ dT_{Th_3} \\ dT_{Th_4} \\ dT_{Th_5} \\ dT_{Th_6} \\ dT_{Th_7} \\ dT_{Th_8} \end{bmatrix} + \begin{bmatrix} T_{avr} \\ T_{avr} \end{bmatrix} \quad (2.21)$$

As for the definition of contribution T_{avr} , having the goal of generating an amount of thrust equal to that required through the term $F_{Z_{cmd}}$, it is therefore derived as given in Equation 2.22.

$$T_{avr} = \frac{F_{Z_{cmd}}}{n_{Th}} \quad (2.22)$$

Where n_{Th} is the number of installed thrusters, that in this case is equal to twelve.

The calculation of the vector of contributions dT_{Th_i} is however more complex since these must generate the moments around the x and y Body axes, which are called $M_{X_{cmd}}$ and $M_{Y_{cmd}}$. This requirement was met by implementing a matrix calculation that bases the determination of thrust contributions on exploiting the position of the thrusters with respect to the CoM of the spacecraft, which coincides with the arm that determines the generated momentum. It is in fact imposed that there are linear type multiplicative relationships between thrust contributions dT_{Th_i} and the distance that the thrusters have with respect to the axes of the Body reference system. These multiplicative ratios are called p_x and p_y ; they are measured in N/m and obtained, as can be seen from Equation 2.23, by multiplying the moments required by the control system by a square matrix of dimension 2×1 called $\overline{Mc2p}$. These are then multiplied by matrix \overline{ThPosM} , defined earlier in Section 2.5.1 to Equation 2.15, which contains the coordinates of the thruster positions measured in meters. Doing so provides with matrix dT_{Th_i} containing the contributions in terms of thrust measured in Newtons and having dimension 12×1 :

$$\bar{p} = \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{(\sum_{i=1:n} y_{Th_i})^2} \\ \frac{1}{(\sum_{i=1:n} x_{Th_i})^2} & 0 \end{bmatrix} \cdot \begin{bmatrix} M_{cmd_x} \\ M_{cmd_y} \end{bmatrix} \quad (2.23)$$

$$\begin{bmatrix} dT_{Th_1} \\ dT_{Th_2} \\ dT_{Th_3} \\ dT_{Th_4} \\ dT_{Th_5} \\ dT_{Th_6} \\ dT_{Th_7} \\ dT_{Th_8} \end{bmatrix} = \begin{bmatrix} x_{Th_1} & y_{Th_1} \\ x_{Th_2} & y_{Th_2} \\ x_{Th_3} & y_{Th_3} \\ x_{Th_4} & y_{Th_4} \\ x_{Th_5} & y_{Th_5} \\ x_{Th_6} & y_{Th_6} \\ x_{Th_7} & y_{Th_7} \\ x_{Th_8} & y_{Th_8} \end{bmatrix} \cdot \begin{bmatrix} 0 & -\frac{1}{(\sum_{i=1:n} y_{Th_i})^2} \\ \frac{1}{(\sum_{i=1:n} x_{Th_i})^2} & 0 \end{bmatrix} \cdot \begin{bmatrix} M_{cmd_x} \\ M_{cmd_y} \end{bmatrix} \quad (2.24)$$

2.5.4 Unification of thruster and reaction wheel models

The last operation that is performed by the model implemented in the simulator is to define vector \overline{U}_{Act} , which in turn contains vector \overline{F}_{Act} and vector \overline{M}_{Act} , respectively containing the forces and moments applied by the actuators to the spacecraft structure. This is done having in mind that, since all the actuators are oriented in concordance with the z Body axis, forces F_{act_x} and F_{act_y} will be null. These are in fact generated by the lander acting on its attitude, causing the roll and pitch angles to vary. Thrust contribution F_{Act_z} along z Body axis is defined as the sum of all contributions given by the twelve thrusters. In contrast, moments M_{act_x} and M_{act_y} are defined by multiplying all the components of vector \overline{T} , containing values in Newtons of all the thrusts generated by the thrusters, by the relative distance from the axis with respect to which the generated moment is being calculated. These steps are mathematically summarized and compressed into vector multiplication reported at Equation 2.25.

$$\overline{U}_{Act} = \begin{bmatrix} \overline{F}_{Act_x} \\ \overline{F}_{Act_y} \\ \overline{F}_{Act_z} \\ \overline{M}_{Act_x} \\ \overline{M}_{Act_y} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 1 & 1 & 1 & \dots & 1 \\ y_{Th_1} & y_{Th_2} & y_{Th_3} & \dots & y_{Th_8} \\ -x_{Th_1} & -x_{Th_2} & -x_{Th_3} & \dots & -x_{Th_8} \end{bmatrix} \cdot \begin{bmatrix} T_{Th_1} \\ T_{Th_2} \\ T_{Th_3} \\ \vdots \\ T_{Th_{12}} \end{bmatrix} \quad (2.25)$$

2.6 Fuel consumption model

Fuel consumption turns out to be a very important variable regarding these types of missions because the mass variations involved are large. These, in fact, can be detrimental to the operation of an unrobust control system that could see its performance diminished or going into instability. Therefore, implementing the first order differential equation 2.26 known as the Tsitsolkovsky rocket equation in the simulator was decided in order to make it possible to determine the variations of spacecraft's mass due to the fuel consumption required in order to implement the forces contained in vector \overline{T} , output of the propulsion system model:

$$\dot{M}_{SC} = \sum_{i=1:12} |T_{Th_i}| \cdot \frac{-1}{g_0 \cdot I_{sp}} \quad (2.26)$$

Where g_0 is the calculated gravitational acceleration on the surface, which for Mars equals $3.71m/s^2$, whereas I_{sp} is the specific impulse, which turns out to be related to both the type of thrusters installed and the fuel used. Usually, for missions of a similar kind to the one to be reproduced in this paper, specific impulse values of about 200 seconds are mentioned.

2.7 Mars disturbance

This subchapter will discuss the disturbances implemented in the simulator with the task of testing the robustness of the GNC system. The disturbances present on planet Mars are manifold and can act on both the positional and attitude dynamics of the spacecraft. Among these, it was decided to simulate those correlated to the gravitational field, taking into consideration the variations with respect to the ideal model reported in Equation 2.3 given by the J_2 effect and the moments generated on the spacecraft due to an uneven distribution of weight. Moreover, disturbances correlated to the aerodynamics of the lander have been taken into consideration as well by simulating the presence of aerodynamic friction due to the difference in speed between the atmosphere in which the spacecraft is immersed and the latter. The remaining ones, related to magnetic fields and solar radiation, have therefore been ignored, as they are of minor magnitude and difficult to implement.

2.7.1 J_2 effect

As already mentioned in subchapter 2.1, the shape of planet Mars is far from being a sphere with constant density throughout its volume. This causes discrepancies in relation to the gravitational field predicted by Newton's universal gravitational law, which predicts the assumptions and approximations stated in Section 2.3.1. The J_2 effect compensates for the approximation of perfect sphericity previously made by adding a contribution in terms of acceleration to the gravitational field that takes into account the fact that Mars is actually ellipsoidal in shape. The result is that there will be a gravitational force that no longer perfectly points at the planet's centre of mass, but will be slightly more inclined towards the equatorial plane instead. A consequence of this is the fact that, at the poles, the force of gravity will be greater in modulus than that measured at the equator. In fact, looking at Figure 2.6 [25], it can be seen that the modulus of the latter reaches a minimum of 3.69m/s^2 in the equatorial zones near the Tharsis Montes and Olympus Mons and a maximum of 3.74m/s^2 near the poles.

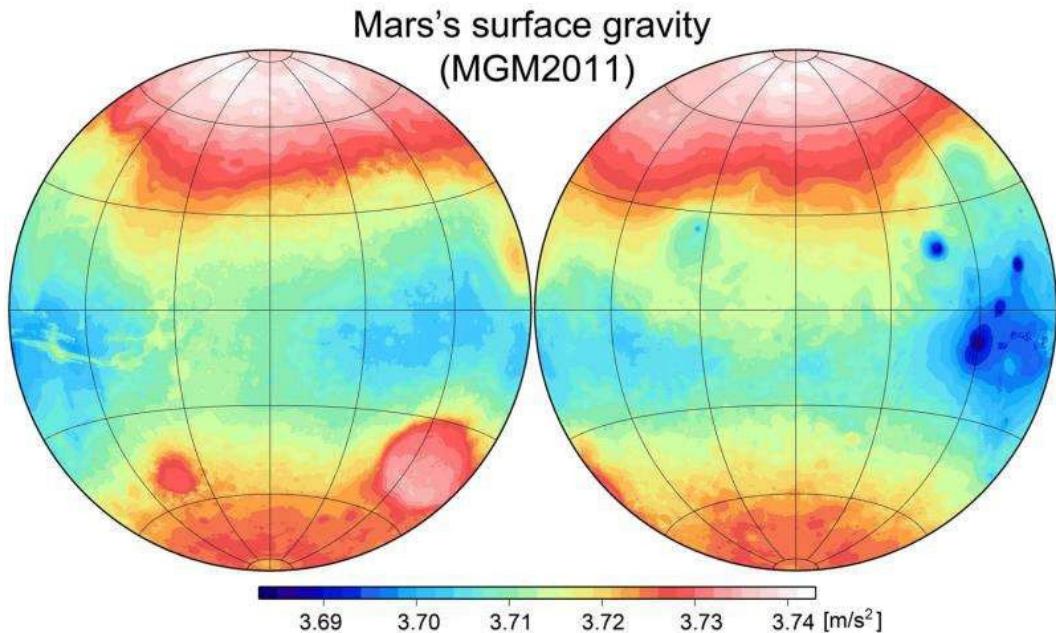


Figure 2.6: Gravity field of Mars surface.

The model taken as reference in order to implement the phenomenon discussed in this section in the simulator also takes into account the J_3 effect and the J_4 effect. These harmonic coefficients make adding definition to the model possible, which thus comes closer and closer to reality.

Parameters	Units	Mars	Earth
J_2	[--]	$1.9555 \cdot 10^{-3}$	$1.0826 \cdot 10^{-3}$
J_3	[--]	$3.1450 \cdot 10^{-5}$	$-2.5327 \cdot 10^{-6}$
J_4	[--]	$-1.5377 \cdot 10^{-5}$	$-1.6196 \cdot 10^{-6}$

Table 2.1: Table comparing harmonic coefficients for Earth and Mars.

Since they are, as can be seen in Table 2.1, two orders of magnitude smaller than J_2 , and in favour of greater computational lightness, it was decided not to take them into account by placing them at zero. Equation 2.27 of the implemented model is shown below and is written taking as reference an inertial system with the same characteristics possessed by reference system M present in the simulator:

$$\bar{g}(\overline{R_{SC}}) = \overline{g_c} + \overline{g_{J2}} = -\frac{\mu}{R_{SC}^3} \cdot \overline{R_{SC}} + \left[\begin{array}{l} \frac{3\cdot\mu\cdot J_2\cdot R_M}{2\cdot R_{SC}^5} \cdot \left(\frac{5\cdot Z_{SC}^2}{R_{SC}^2} - 1 \right) \cdot X_{SC} \\ \frac{3\cdot\mu\cdot J_2\cdot R_M}{2\cdot R_{SC}^5} \cdot \left(\frac{5\cdot Z_{SC}^2}{R_{SC}^2} - 1 \right) \cdot Y_{SC} \\ \frac{3\cdot\mu\cdot J_2\cdot R_M}{2\cdot R_{SC}^5} \cdot \left(\frac{5\cdot Z_{SC}^2}{R_{SC}^2} - 3 \right) \cdot Z_{SC} \end{array} \right] \quad (2.27)$$

Where R_M is the radius measured in the equatorial plane of Mars, whereas R_{SC} is the vector joining the centre of mass of the planet to the one of the S/C and whose components are X_{SC} , Y_{SC} and Z_{SC} , expressed with reference to reference system M.

2.7.2 Wind, atmospheric and aerodynamic drag models

As will be stated in the following subsections, wind, atmospheric density and aerodynamic friction force are closely related to each other; therefore, a model for each of these must be implemented.

Wind model

Winds are major players in the Martian environment, so much so that rovers such as the US Curiosity have among their tasks to measure these phenomena. Studies have revealed that winds on Mars can gust up to 100 km/h during dust storms. In order to be able to reproduce as faithfully as possible the disturbances that the GNC system must be able to counteract, it is essential to implement a model of this phenomenon in the simulator.

Figures 2.7, 2.8 and 2.9 propose graphs showing the average values of the three components of wind at an altitude of 500m as a function of longitude and latitude and for a situation of ordinary solar activity [26].

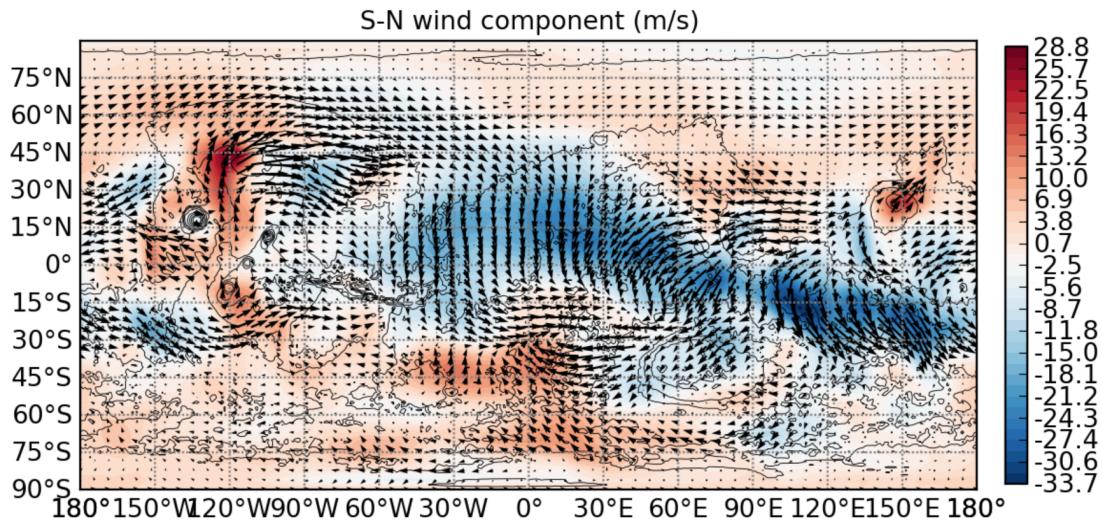


Figure 2.7: Average wind in South/North direction on Mars at 1000m altitude. [26]

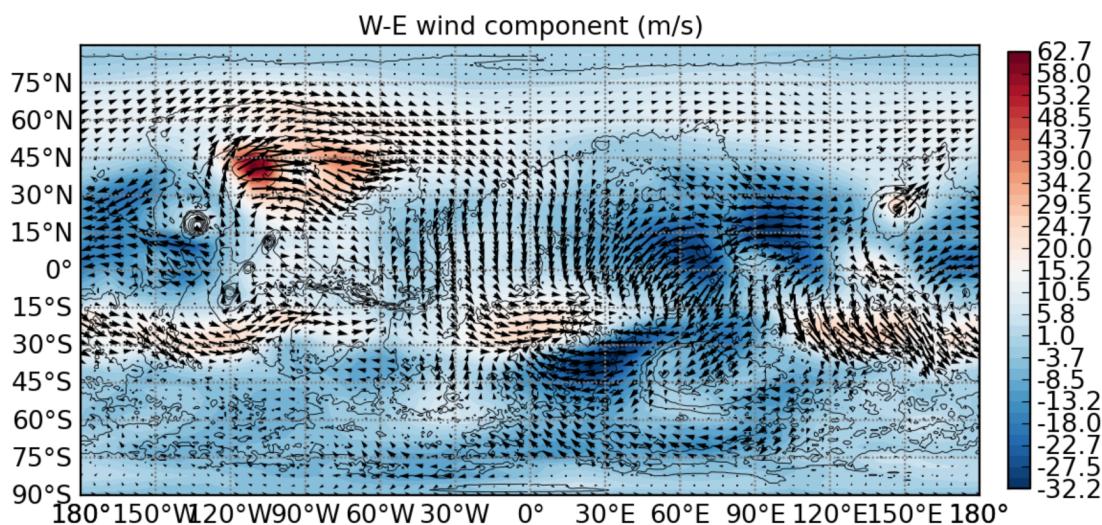


Figure 2.8: Average wind in West/East direction on Mars at 1000m altitude. [26]

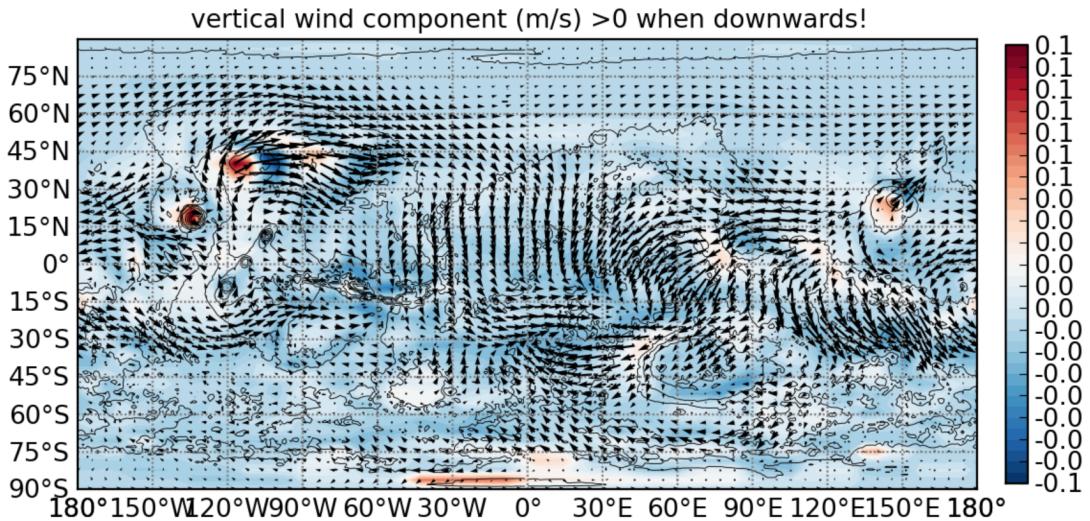


Figure 2.9: Average wind in vertical direction on Mars at 1000m altitude. [26]

In order to best reproduce the conditions in terms of wind intensity and direction, reference was made to a 2020 study [28] in which various wind gusts were analysed in frequency by applying the Fourier transform. This study is actually based on measurements made on Earth and not on planet Mars, but assuming that the dynamics behind these two phenomena are almost the same is an acceptable approximation. An attempt was therefore made to construct a model that would produce outputs, in terms of oscillation amplitude, at least similar to those in Figure 2.10.

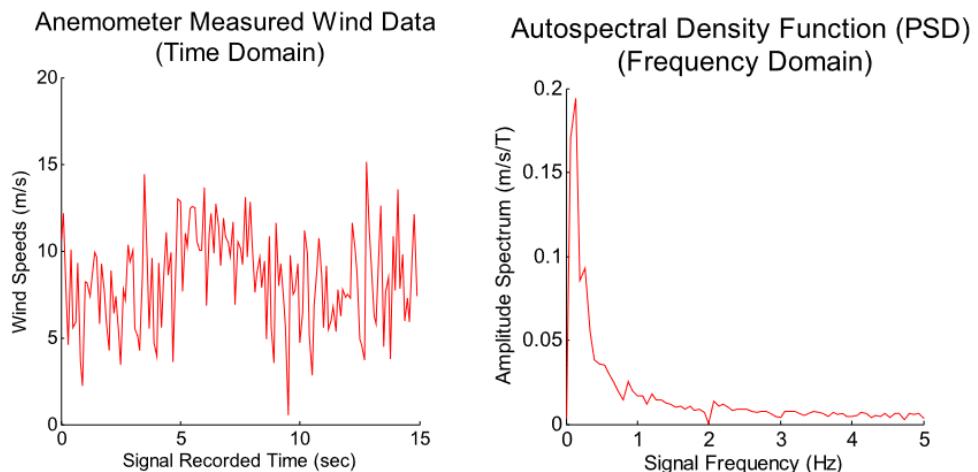


Figure 2.10: Anemometer data measured in a 50 s long record and spectral density analysis in frequency domain. [28]

In order to reproduce the unpredictable and rapid changes in wind intensity on the Red Planet, it was decided to define the components of the vector representing wind speed as formed by the sum of two stochastic contributions. The first has been randomly chosen in a range of $\pm 10\%$ around an average value taken from the matrix $\overline{wind_{avr}}$. The second simulates white noise and has been again randomly chosen, but still respecting the Gaussian distribution. The frequencies of the two contribution have been chosen taking into account spectral density analysis and trying to reproduce the velocity variations shown in Figure 2.10. Therefore, a frequency of $1 Hz$ has been chosen for the first contribution, whereas a much higher frequency of $40 Hz$ has been chosen for the second. Instead, the average values on which the first contribution was based were retrieved from a NASA database [26] by choosing as reference point the landing site of the Phoenix Lander, which is located at the coordinates Latitude $68.22 North$ Longitude $234.25 East$. Equation 2.28 shows the aforementioned values referring to an altitude of 500m above the Martian surface. Below in Figure 2.11, the value that the three components of the wind speed vector $\overline{V_{wind}}$ assume during a test simulation is shown.

$$\overline{wind_{avr}} = \begin{bmatrix} -3.24625 \\ 4.67611 \\ 0.000597911 \end{bmatrix} [m/s] \quad (2.28)$$

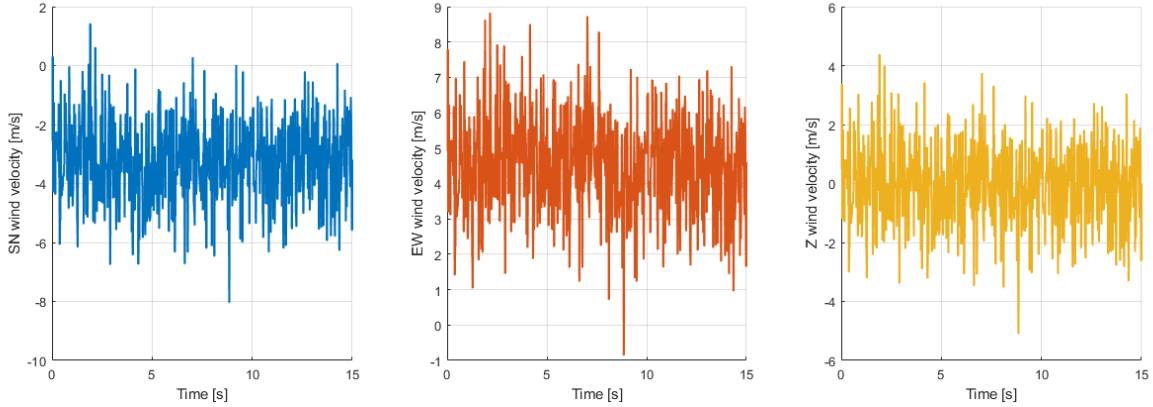


Figure 2.11: Value components of vector $\overline{V_{wind}}$ as seen from reference system T.

Analyzing outputs regarding the three components of vector $\overline{V_{wind}}$ that have been produced by the simulator, frequencies appear to be much higher than those shown in the model taken as reference. In order to tackle this issue, the frequency of white noise could have been lowered; however, it was decided not to act this way and to overestimate the impetus of the disturbance instead in order to test the efficiency of the control system.

Atmospheric model

In order to be able to define the friction force, knowing the characteristics of the fluid in which the spacecraft is moving as a function of the altitude at which it is located is necessary. On a chemical level, the atmosphere of planet Mars is extremely different from that of Earth. It is largely composed of carbon dioxide; therefore, searching for a model specifically created for the Martian atmosphere was necessary. The model proposed by NASA and based on the measurements made in 1996 by the Mars Global Surveyor probe [29] was chosen. The temperature equation 2.29, the pressure equation 2.30 and the density equation 2.31 that together make up the atmospheric model as implemented in the simulator are shown below:

$$T [C^\circ] = \begin{cases} -31 - 0.000998 \cdot h & if h \leq 7000m \\ -23.4 - 0.00222 \cdot h & if h > 7000m \end{cases} \quad (2.29)$$

$$p [kPa] = 0.699 \cdot e^{-0.00009 \cdot h} \quad (2.30)$$

$$\rho \left[\frac{kg}{m^3} \right] = k \cdot \left(\frac{p}{0.1921(T + 273.1)} \right) \quad (2.31)$$

Where h is the altitude at which the spacecraft is located, measured in meters, whereas in equation 2.31 constant k is shown, which has been differentiated in order to fit the model depending on the area of Mars in which it is applied.

Aerodynamic drag

In order to simulate the atmospheric friction force, Drag equation was implemented in the formulation typically used in fluid dynamics, given below as equation 2.32.

$$F_{Drag} = \frac{1}{2} \cdot \rho \cdot V_{S/C_{wind}}^2 \cdot c_D \cdot S \quad (2.32)$$

Where $V_{S/C_{wind}}$ is the velocity the spacecraft possesses with respect to the flow in which it is moving, calculated taking the wind into account. Term c_D is the dimensionless drag coefficient, that depends on the shape of the body that is immersed in the flow. Finally, S is the reference surface. Landers in the powered descent phase are in the vast majority of cases not fairing; therefore, it is difficult to define reference surfaces and drag coefficients for their structures. In order to overcome this, as can be seen in Figure 2.12, the shape of the spacecraft has been approximated to that of a flattened cylinder; this way, determining the three reference surfaces and estimating three different drag coefficients, one for all three axes, was possible. In particular, the areas of the sections relative to the planes created by the body axes were taken as reference surfaces. By doing so, applying the drag equation by decomposing and treating the components of vector $V_{S/C_{wind}}$ individually was then possible.

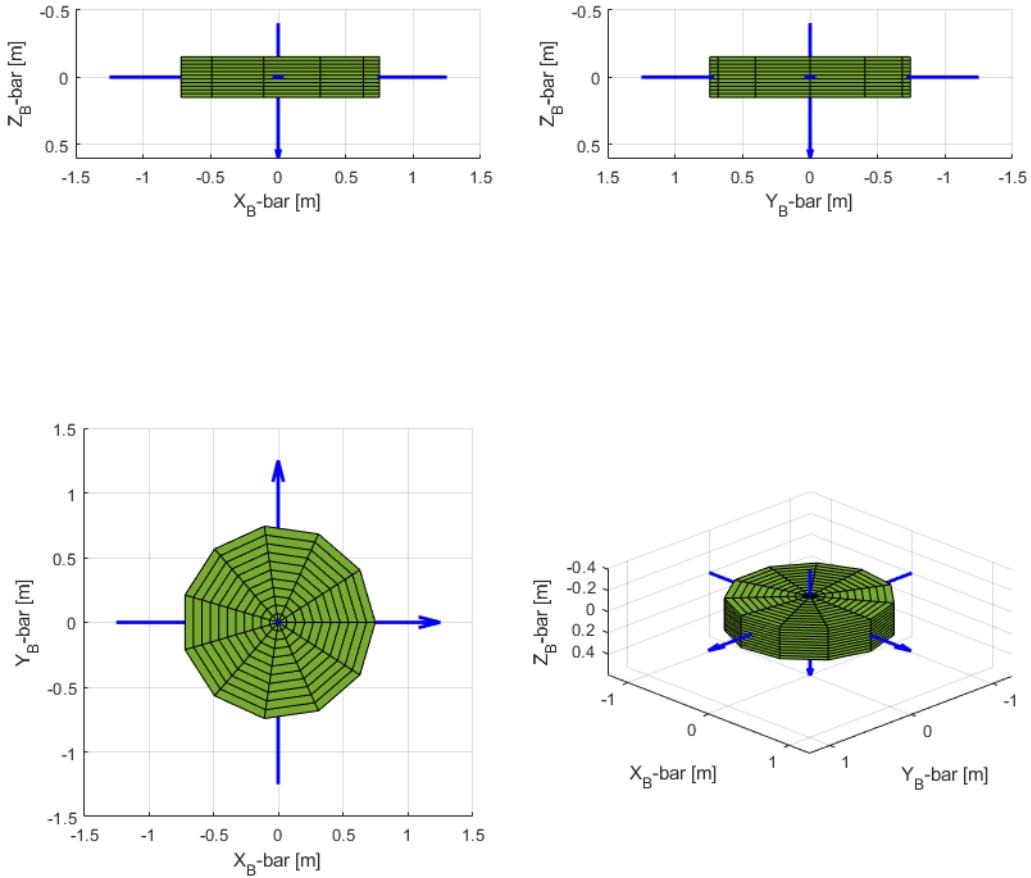


Figure 2.12: 3D model of the spacecraft body implemented in the simulator.

Thus, the Drag equation can be reformulated in vector form as given in Equation 2.33:

$$\overline{F_{Drag}} = \frac{1}{2} \cdot \rho \cdot \overline{V_{S/C_{wind}}}^2 \cdot \overline{c_D} \cdot \overline{S} \quad (2.33)$$

Where $\overline{c_D}$ is the vector containing the aerodynamic drag coefficients associated with each axis of reference system B. Similarly, vector \overline{S} contains the areas of the spacecraft sections, making sure that the first component contains the area of the section defined by the plane defined by the y and z axes of the Body system, the second component contains the area of the section defined by the plane defined by the x and z axes of the Body reference system, and finally, the third shall contain the area of the section defined by the plane defined by the x and y axes of the body reference system.

2.7.3 Gravity gradient effect

Unlike the previous ones, the Gravity gradient effect is a disturbance acting on the attitude of the spacecraft. This, in fact, represents the action that the force of gravity has on bodies that have an inertia matrix with all three polar inertias of equal magnitude on the diagonal. In fact, as can be seen in Figure 2.13, if the spacecraft is found to have a polar inertia lower than the other two along one of its main axes of inertia, a momentum related to gravitational attraction will arise. This will tend to align the spacecraft with respect to the local vertical. This effect is also often exploited in order to obtain a pointing of the satellites called Nadir, i.e. pointing towards the centre of mass of the planet around which it is orbiting [30]. This effect was then simulated by simply implementing Equation 2.34, which describes it:

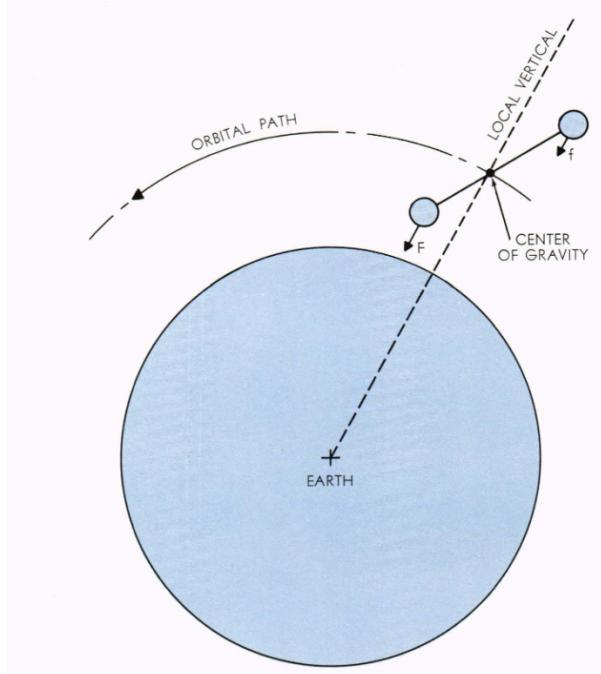


Figure 2.13: Gravitational forces acting on a dumbbell-shaped satellite.

$$\overline{M_D} = 3 \cdot \mu \cdot \frac{\overline{R_{SC}}}{|\overline{R_{SC}}|^5} \wedge \overline{\bar{J}} \cdot \overline{R_{SC}} \quad (2.34)$$

Where $\overline{R_{SC}}$ is the spacecraft's position vector with respect to reference system M expressed, however, with components referring to the Body reference system. The standard gravitational parameter μ , as already mentioned in Section 2.3.1, carries information about how intense the gravity field developed by planet Mars is. The inertia tensor $\overline{\bar{J}}$ carries with it information about how the spacecraft mass is distributed around its centre of mass.

Chapter 3

Guidance algorithms

The algorithm chosen to be implemented for spacecraft control in the missions that will be simulated in Chapter 4 is Optimal Sliding Guidance (OSG). However, having implemented, as explicated in Section 2.5.3, a propulsion system control logic not able to generate any acceleration along the x-axis and y-axis of reference system B, acting on attitude dynamics has been necessary in order to overcome this issue. Moreover, implementing an algorithm for attitude control has turned out to be mandatory as well. Super-Twisting Sliding Mode control has therefore been chosen in order to fill the latter role.

3.1 Position dynamics control

Optimal Sliding Mode (OGS) is a driving algorithm based on the hybridization of the Zero-Effort-Miss/Zero-Effort-Velocity driving strategy and the Sliding Mode Control method [31]. The result of this fusion is a driving algorithm that can provide as output an acceleration calculated solely on the basis of an estimate of the position and speed at the end of the maneuver. In order to better explain the structure and functioning of the OSG law, the two control strategies from which it derives have first been described in general terms and then the focus has been shifted to how they have been melded together.

3.1.1 Zero-Effort-Miss/Zero-Effort-Velocity

The Zero-Effort-Miss/Zero-Effort-Velocity (ZEM/ZEV) feedback guidance law, which originated in the missile field, has been designed in order to successfully carry out missions in which the interception of a moving target is involved and which require excellent robustness and flexibility. In recent years, the space sector has looked with interest at this guidance law, as highlighted by numerous examples of studies conducted in order to demonstrate the adaptability of this technology to missions involving rendezvous [32] [33] rather than landings [34] [16].

The ZEM/ZEV algorithm is based on the definition of two terms known as ZEM and ZEV, which represent the deviation between the coordinates in terms of final position and velocity with respect to those desired for the spacecraft at final time t_f which, in the case discussed in this paper, is the time at which the landing is desired to happen. These two terms are constantly calculated during the manoeuvre and are based, as can be seen from Equations 3.1 and 3.2 that define them, on the propagation in time of the spacecraft's state variables.

This has been done by applying the hypothesis that, from that moment onwards, only the acceleration that is due to the force of gravity acts on the spacecraft, thus excluding all the others, including those generated by the actuation system:

$$\overline{ZEM}(t) = \overline{R_{SC_f}} - \left[\overline{R_{SC}}(t) + t_{go} \cdot \overline{V_{SC}}(t) + \int_t^{t_f} (t_f - \tau) \cdot \bar{g}(\tau) d\tau \right] \quad (3.1)$$

$$\overline{ZEV}(t) = \overline{V_{SC_f}} - \left[\overline{V_{SC}}(t) + \int_t^{t_f} \bar{g}(\tau) d\tau \right] \quad (3.2)$$

Where $\overline{R_{SC_f}}$ and $\overline{V_{SC_f}}$ are respectively the desired final position and velocity, whereas term t_{go} is called Time To Go and is defined in Equation 3.3 as the time remaining to complete the maneuver:

$$t_{go} = t_f - t \quad (3.3)$$

In order to follow the policy applied so far, which imposes a certain computational lightness, and in order to allow an eventual on-board computer to calculate these contributions in real time, considering vector $\bar{g}(\tau)$ as a constant was decided, thus allowing the solution of the integrals to be made explicit and relations to be rewritten, obtaining Equations 3.4 and 3.5:

$$\overline{ZEM}(t) = \overline{R_{SC_f}} - \left[\overline{R_{SC}}(t) + t_{go} \cdot \overline{V_{SC}}(t) + \frac{1}{2} \cdot t_{go}^2 \cdot \bar{g}(t) \right] \quad (3.4)$$

$$\overline{ZEV}(t) = \overline{V_{SC_f}} - [\overline{V_{SC}}(t) + t_{go} \cdot \bar{g}(t)] \quad (3.5)$$

Once the two terms \overline{ZEM} and \overline{ZEV} have been calculated, acceleration $\overline{u_{ZEM/ZEV}}$ is then defined as given in Equation 3.6. In fact, the acceleration $\overline{u_{ZEM/ZEV}}$ thus defined is able to act on the spacecraft's state variables with the aim of cancelling the two terms mentioned above. The nullifying of the latter occurs when the conditions predicted by the propagation in time of the state variables define conditions in terms of final landing position and velocity exactly equal to those desired and set as a reference:

$$\overline{u_{ZEM/ZEV}}(t) = \frac{k_r}{t_{go}^2} \cdot \overline{ZEM} + \frac{k_v}{t_{go}} \cdot \overline{ZEV} \quad (3.6)$$

Where k_r and k_v are two gains that act respectively on position error, through the multiplication with the ZEM term, and velocity error, through the multiplication with the ZEV term.

3.1.2 Sliding Mode Control

The Sliding Mode Control technique is a control law that has among its best qualities that of being able to counter external disturbances very effectively. This property allows to achieve accuracy levels that are prohibitive for many other control laws. A further quality is brought by the possibility of being able to formally demonstrate and guarantee, through the use of Lyapunov theory, that the developed control system is globally finite-time stable, provided that perturbations are bounded between pre-defined values [35].

The logic behind Sliding Mode control laws involves modelling the dynamics of the phenomenon to be controlled by treating perturbations as real discrepancies between the real model and the ideal model. In fact, the model that is defined and on which the control law is then constructed includes perturbations even though they are unknown. An example is given in Equation 3.7, in which perturbations are represented by term $f(x_1, x_2, t)$, on which the subsequent definitions will be based:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = u(x_1, x_2) + f(x_1, x_2, t) \end{cases} \quad (3.7)$$

Subsequently, in Equation 3.8, Slip Variable σ is defined as a linear composition of the state variables defining the model mentioned before. A further concept that is defined is that of Sliding Surface, whose equation is defined by imposing the equation of the sliding variable as null, as done in Equation 3.9:

$$\sigma(t) = x_2 + c \cdot x_1 \quad \text{with } c > 0 \quad (3.8)$$

$$\sigma(t) = x_2 + c \cdot x_1 = 0 \quad (3.9)$$

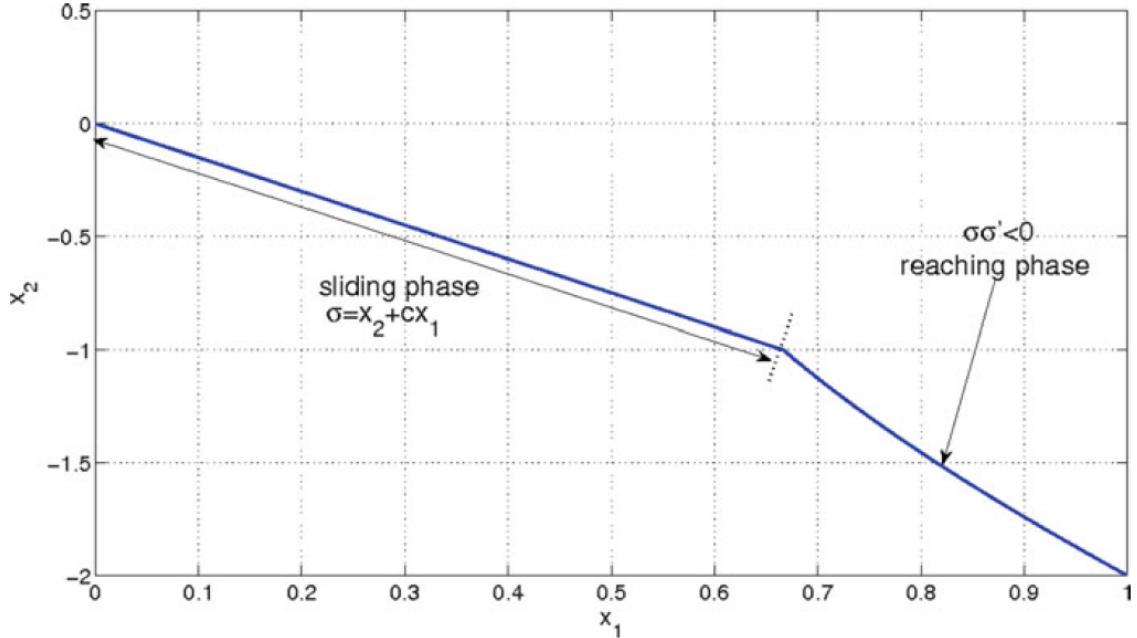


Figure 3.1: State-space representation in which reaching phase and sliding phase are clearly distinguished. [35]

The last step is to define control Equation 3.10, known as Sliding Mode Controller. This defines the desired acceleration that the actuating system must generate in order for the sliding variable, first of all, to reach the Sliding Surface, which is equivalent to saying that $\sigma(t)$ must be nullified. Then, it subsequently remains confined there by *sliding* along it until it reaches the equilibrium condition, which is defined as the condition in which the contributions forming the sliding variable, that in the example are x_1 and x_2 , are all nullified:

$$u(t) = -c \cdot x_2 - \Phi \cdot \text{sign}(\sigma) \quad (3.10)$$

In Figure 3.1, a graphic representation of state space of the system in which the first phase of reaching the sliding surface and the second phase of sliding are clearly distinguished can be seen. In the Sliding Mode Controller, named $u(x_1, x_2)$ in this example, a contribution multiplied by the sign function of the sliding variable is inserted. This term makes the control law discontinuous and is responsible for the chattering effect that occurs in the sliding phase visible in Figure 3.2. This effect is a consequence of the fact that, as a matter of facts, the model on which the control law is based does not take into account perturbations contained in term $f(x_1, x_2, t)$ since they are unknown and therefore, once the sliding surface is reached, the contribution given by the sign function acts by going against this discrepancy. If there were the possibility of perfectly knowing the time course of the perturbations, implementing them in the model and then developing the sliding mode controller; the chattering effect would not appear. Gain Φ , by which function $\text{sign}(\sigma)$ is multiplied, defines how influential this contribution may be in defining the acceleration to be implemented. Another aspect to take into consideration when choosing gain Φ is that the greater in modulus the latter is, the more nervous the control system will be.

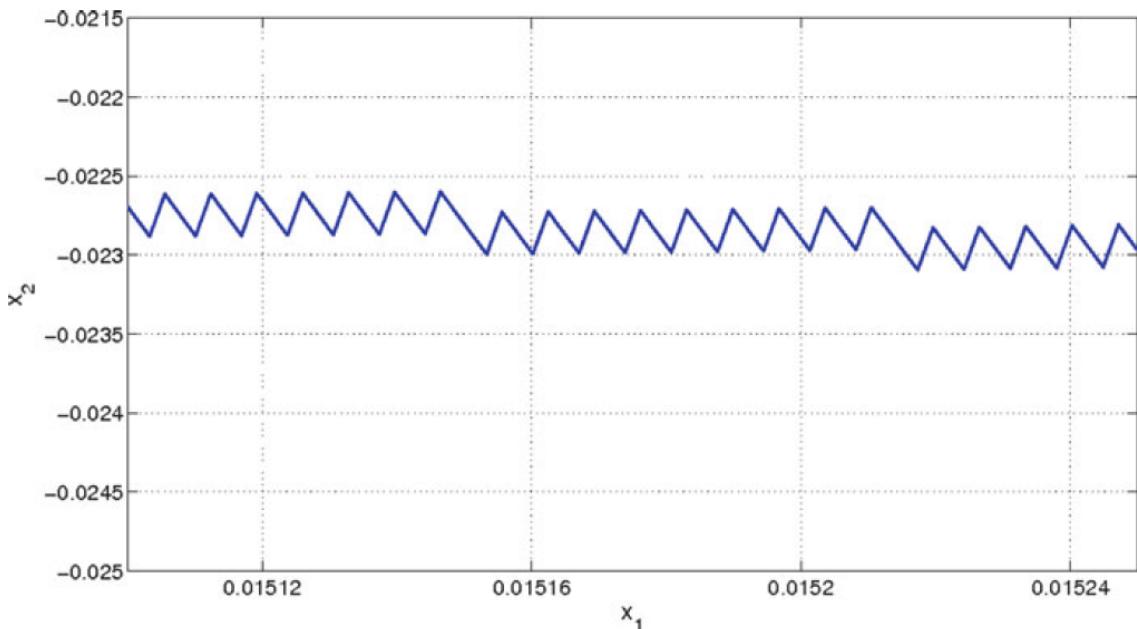


Figure 3.2: Zoom of Figure 3.1 in which chattering effect is noticeable.

The OSG control algorithm, as mentioned earlier, is a hybridisation of ZEM/ZEV and Sliding Mode; therefore, defining a sliding variable and a sliding mode controller designed in order to work with the two terms \overline{ZEM} and \overline{ZEV} was necessary [31]. Sliding variable σ has then been defined, as shown in Equation 3.11:

$$\sigma(t) = \overline{ZEM} + \lambda \cdot \overline{ZEV} \quad (3.11)$$

Whereas the sliding mode controller, named u_{SM} , was defined as only consisting of the sign function related to σ multiplied by the gain called Φ , as represented in Equation 3.12:

$$u_{SM}(t) = -\Phi \cdot \text{sign}(\sigma) \quad (3.12)$$

The term λ is the gain which in the previous example, and precisely in Equation 3.8, was called c , and on which the slope of the sliding surface drawn in the state-space graph depends. The definition of the latter is given in Equation 3.13 and has been chosen following the reference "Optimal sliding guidance algorithm for Mars powered descent phase" by Professor Furfaro Roberto [31], which derives the value of constant λ during the demonstration of global finite time stability in which it is imposed that the equation used for the definition of the sliding surface is a Lyapunov function. This procedure is very important, as it allows to prove the global stability of the algorithm, which is equivalent to having the certainty that it will lead the system to converge to the equilibrium point in a finite time, provided that the problem has been globally defined and that perturbations are bounded.

$$\lambda = -\frac{2}{t_{go}} \quad (3.13)$$

Finally, substituting Equation 3.13 with Equation 3.14 gives the final definition of the sliding variable in equation 3.14:

$$\sigma(t) = \overline{ZEM} - \frac{2}{t_{go}} \cdot \overline{ZEV} \quad (3.14)$$

3.1.3 Union of the two control logics.

Having defined the two control logics in sections 3.1.1 and 3.1.2, these have been combined in order to form the Optimal Sliding Guidance algorithm. This union, given in Equation 3.15, simply consists of summing $\overline{u_{ZEM/ZEV}}$ and $\overline{u_{SM}}$, thus defining accelerations $\overline{A_{cmd}}$ to be implemented by the propulsion system control logic.

$$\overline{A_{cmd}} = \overline{u_{ZEM/ZEV}}(t) + \overline{u_{SM}}(t) = \frac{k_r}{t_{go}^2} \cdot \overline{ZEM} + \frac{k_v}{t_{go}} \cdot \overline{ZEV} - \Phi \cdot \text{sign}(\sigma) \quad (3.15)$$

The OSG algorithm is therefore able to determine the accelerations that must be generated in order to complete the mission by simply processing the feedback received from sensors in terms of velocity and position relative to the landing point, ergo to reference system T. Moreover, among the several advantages of this type of control logic is the fact that it is totally autonomous, robust and extremely well suited for the implementation of possible additions such as collision avoidance logic [36].

3.2 Attitude dynamics control

The simulator that has been implemented belongs to the category of six degrees of freedom simulators, which means that not only is it capable of simulating the dynamics related to the position of the spacecraft, but also to determine the evolution of the spacecraft's attitude over time. This was necessary because of the fact that the actuation system, as is conceived in Subchapter 2.5, bases its operation on the thrusters' orienting in the direction imposed by the OSG guidance algorithm. It was therefore necessary to add a control system that could be able, upon receiving vector \bar{A}_{cmd} as input, to determine the moments required in order to orient the spacecraft so that vector \bar{T} was as closely aligned with the requirements of the OSG algorithm as possible.

Again, a controller from the sliding mode family has been chosen for this attitude control role, but this time a second-order controller called Super-Twisting Controller has been opted for [35]. This is called second-order because not only is it able to drive sliding variable σ to zero, but also to nullify its derivative $\dot{\sigma}$ and, therefore, to avoid the problem of chattering that first-order controllers suffer from. In keeping up with the policy adopted in Section 2.3.2, the implementation of a Super-Twisting attitude control system adapted to work with quaternions and angular velocities as input has been chosen [37].

3.2.1 Calculation of the reference quaternion

The calculation of the reference quaternion that defines the desired attitude that the Super-Twisting Attitude control will have to pursue has been implemented on the basis of the definition of quaternion given in Equation 2.10. Using the latter, defining a quaternion by knowing the orientation of two vectors in space is possible. The two vectors with respect to which the reference quaternion is to be calculated are the two versors $\widehat{z_B}$ and $\widehat{z_{cmd}}$. As can be seen by looking at Equations 3.16 and 3.17, verson $\widehat{z_B}$ defines the orientation of the z-axis of reference system B expressed in coordinates referring to the latter, whereas verson $\widehat{z_{cmd}}$ represents the orientation that the spacecraft must assume in order to orient thrusters in the direction commanded by the OSG algorithm:

$$\widehat{z_B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.16)$$

$$\widehat{z_{cmd}} = \frac{1}{\|\bar{A}_{cmd}\|} \cdot \begin{bmatrix} -A_{cmd_x} \\ -A_{cmd_y} \\ |A_{cmd_z}| \end{bmatrix} \quad (3.17)$$

Two peculiarities can be noticed in the definition of $\widehat{z_{cmd}}$ verson. The first is that the first two components of vector \bar{A}_{cmd} are taken as opposites, which is necessary because, for instance, at a request for acceleration along the x-axis, the spacecraft will have to change its attitude by tilting to the opposite side. A two-dimensional representation of this concept is shown in Figure 3.3. The second is that an absolute value is applied to the last component of vector \bar{A}_{cmd} ; this is necessary in order to avoid the need to command attitudes that would lead to inverted flight.

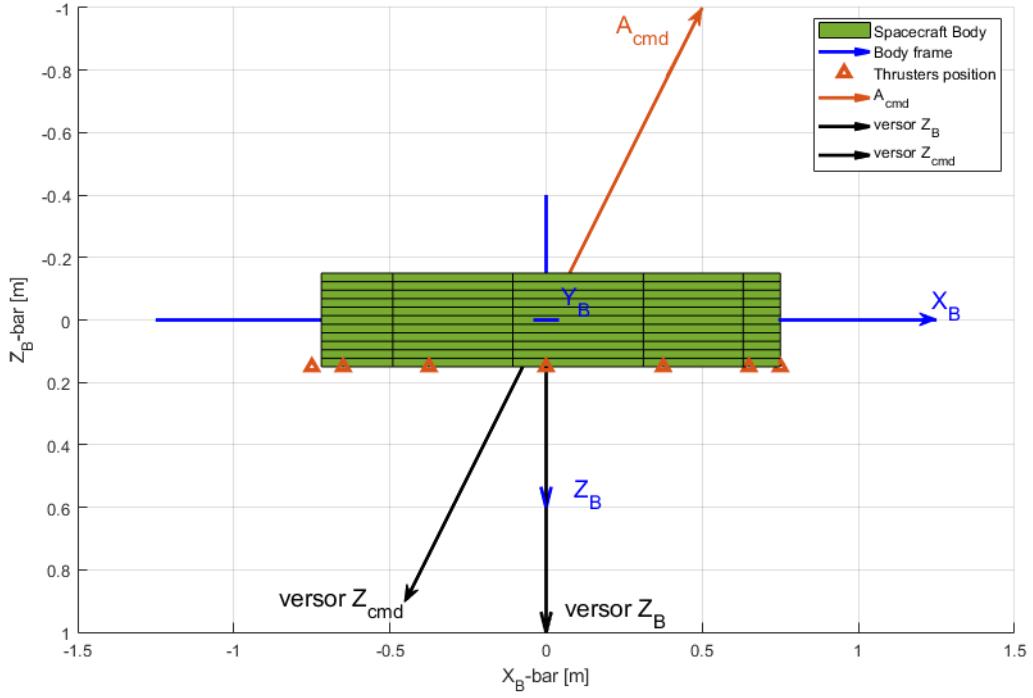


Figure 3.3: Two-dimensional representation of how \widehat{z}_B versor and \widehat{z}_{cmd} versor act.

Once the two versors have been defined, it is necessary to determine angle α between them and Euler axis \hat{e} perpendicular to them. These are necessary elements for the calculation, through the application of Equation 2.10, of the quaternion that defines the rotation necessary to have the orientation of \widehat{z}_B and \widehat{z}_{cmd} coincide. This is done by applying analytical geometry equations 3.18 and 3.19:

$$\alpha_{SMA} = \text{atan}(\widehat{z}_B \cdot \widehat{z}_{cmd}) \quad (3.18)$$

$$\widehat{e}_{SMA} = \frac{\widehat{z}_B \wedge \widehat{z}_{cmd}}{\|\widehat{z}_B \wedge \widehat{z}_{cmd}\|} \quad (3.19)$$

3.2.2 Super-Twisting Sliding Mode Controller

In recent years, there have been numerous examples of the application of this algorithm for attitude control in space environment [37] [38] [39] and many of these are based on the nullification of quaternion error $\overline{q_{err}}$ and angular velocities $\overline{\omega_{SC}}$. This quaternion error is defined by Equation 3.20 and represents the discrepancy between the desired attitude expressed by quaternion $\overline{q_{des}}$, which is received by the OSG algorithm, and the attitude of the spacecraft, represented by quaternion $\overline{q_{att}}$:

$$\overline{q_{err}} = \begin{bmatrix} q_{err_0} \\ q_{err_1} \\ q_{err_2} \\ q_{err_3} \end{bmatrix} = \overline{q_{des}}^{-1} \otimes \overline{q_{att}} \quad (3.20)$$

Where \otimes represents the quaternion multiplication. As far as angular velocities are concerned, there is no need to calculate any error since, at equilibrium, these are desired to be null. In fact, the desired angular velocity vector that would be subtracted would only consist of null components, rendering the operation useless.

Having calculated quaternion error $\overline{q_{err}}$ and having angular velocities $\overline{\omega_{SC}}$, expressed with respect to reference system B, sliding variable $\overline{\sigma_{SMA}}$ is defined in equation 3.21 as a linear composition of the two terms:

$$\overline{\sigma_{SMA}} = \begin{bmatrix} q_{err_1} \\ q_{err_2} \\ q_{err_3} \end{bmatrix} + \begin{bmatrix} \omega_{SC_x} \\ \omega_{SC_y} \\ \omega_{SC_z} \end{bmatrix} \quad (3.21)$$

Having defined the sliding variable, the sliding mode controller is defined in Equation 3.22, in which a differential equation-related contribution also appears as part that makes the control second-order:

$$\begin{cases} \overline{u_{SMA}} = -\lambda_{SMA} \cdot \sqrt{|\overline{\sigma_{SMA}}|} \cdot sign(\overline{\sigma_{SMA}}) - \overline{u_1} \\ \dot{\overline{u_1}} = \beta_{SMA} \cdot sign(\overline{\sigma_{SMA}}) \end{cases} \quad (3.22)$$

The sliding mode controller $\overline{u_{SMA}}$ thus defined directly contains the moments that must be acted upon by the actuation system in order to control the attitude. It is therefore legitimate to enunciate Equation 3.23:

$$\overline{M_{cmd}} = \overline{u_{SMA}} \quad (3.23)$$

Chapter 4

Simulations

This chapter will describe all the elements involved in mission simulations starting from the choice of the spacecraft and the mission taken as reference for the establishment of the initial conditions, going through the decision of the settings and gains related to the implemented algorithms and finally arriving at the analysis of the results obtained from the code implemented in the Matlab/Simulink environment.

4.1 Spacecraft and parameters selected

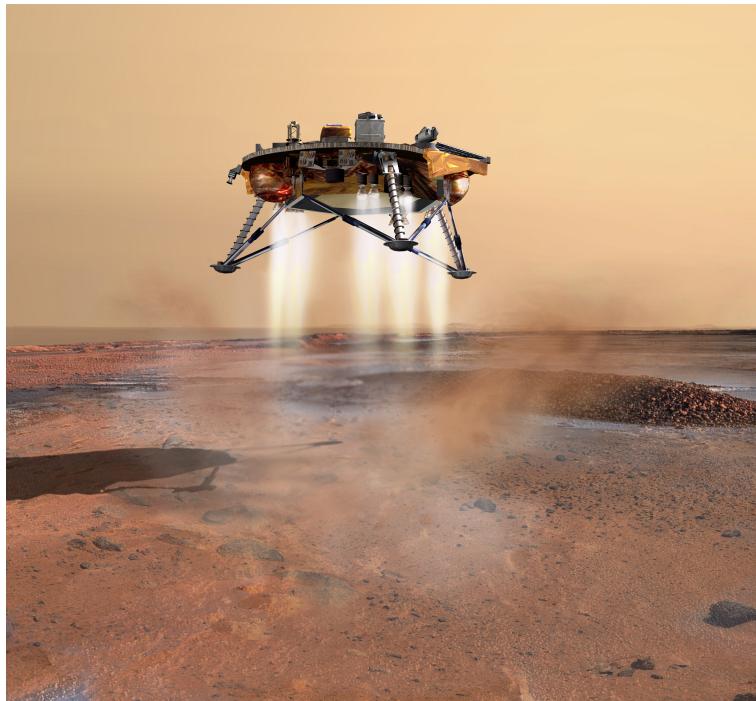


Figure 4.1: Concept depicts NASA's Phoenix Mars Lander during touchdown on the arctic plains of Mars. [22]

The choice of spacecraft was made taking various aspects into consideration. Consistency with the objective of this study, which is to test the behaviour of the implemented algorithms in a Mars landing, certainly played a key role. For this reason, several missions among those mentioned in subchapter 1.3 and including an EDL phase were analyzed. Among many of the missions considered; the Curiosity mission, the Perseverance mission and also the InSight mission can be mentioned. Secondly, the amount of information that could be retrieved was also decisive. This

aspect guided the choice by decreeing as mission to be taken as a reference the one conducted by NASA in 2008 and called Phoenix, whose final approach and landing phases were taken as an example and previously explained in subchapter 1.4.

The Phoenix Mars Lander developed by NASA also has a structure and actuation system that are comparable to those modelled in the simulator and respectively described in subchapters 2.7.2 and 2.5. In fact, it has a deck diameter of $1.5m$ for a height, estimated from the drawings due to a lack of reference, of approximately $0.4m$ [23]. Surrounding this structure, as can be seen in Figure 4.1, are twelve thrusters, each one capable of generating up to $293N$ of thrust with a specific impulse of $212.5s$. The model of the thrusters implemented in the simulator has been conceived with analogue modulation; in other words, they can vary the thrust generated in a range from 1% to 95% of the maximum thrust that can be generated.

In the documents found about the Phoenix Mars Lander, only thrusters were mentioned as actuating systems; therefore, a control of the moments around the yaw axis is missing. By doing so, however, some of the drives required by the implemented algorithm for attitude control would be lost. In order to overcome this drawback, it was therefore decided to also implement a reaction wheel capable of generating a maximum moment of $12N \cdot m$ acting on the z-axis of the B system, which would otherwise remain uncontrolled. In order to make the actuator model more realistic, $\tau_{TH} = 0.1$ and $\tau_{RW} = 0.05$ were chosen as time constants to be inserted in the first-order filters; this operation was previously explained in subchapter 2.5. As far as the weight of the lander is concerned, this is $410kg$, a parameter that will decrease each time thrusters are used during the mission simulation due to fuel consumption. As far as the inertia matrix is concerned, this was taken to be the same as the real one of NASA's spacecraft [40]. The components of the latter are, for reasons of space, directly shown in summary table 4.5.

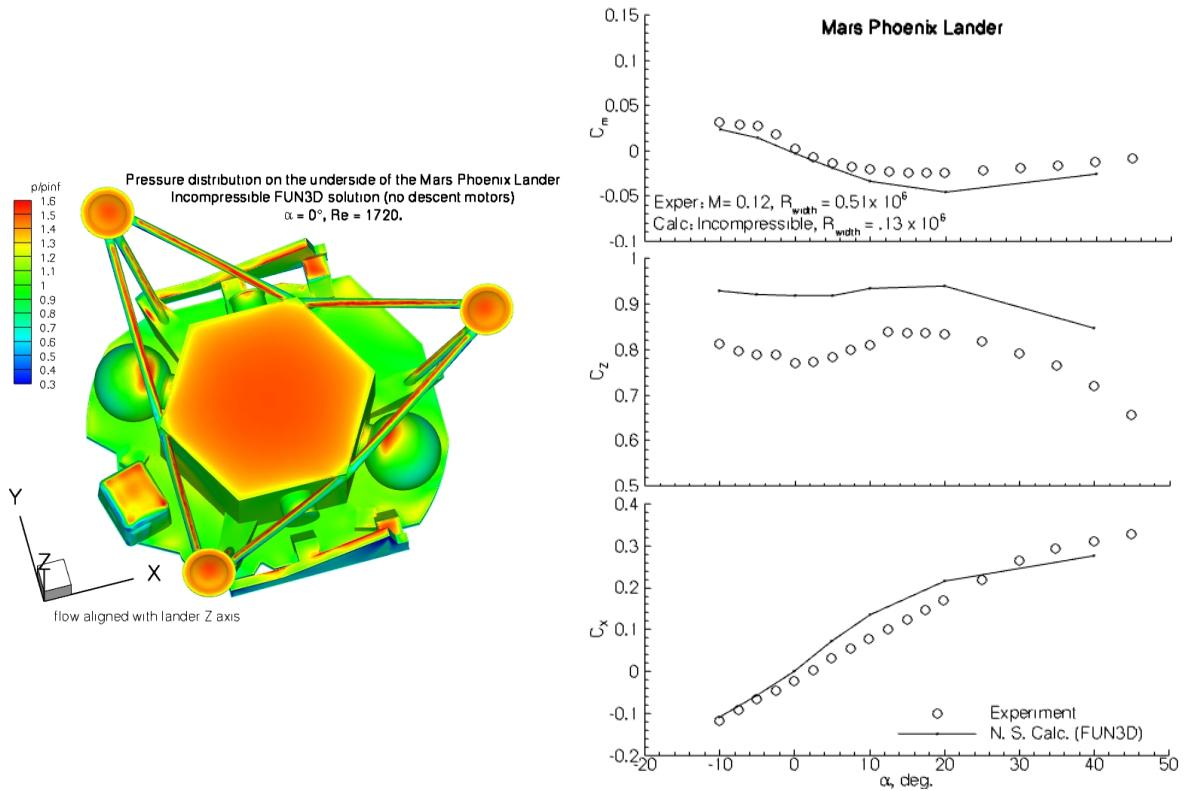


Figure 4.2: Dimensionless drag coefficients derived from both CFD simulations and wind tunnel experiments.[41]

The last parameters to be entered into the simulator are those related to the model simulating the aerodynamic friction force introduced in subchapter 2.7.2. These include the surfaces to be entered in vector \bar{S} , which are obtained by approximating the structure of the lander to a cylindrical disc, exactly as represented in Figure 2.12. Instead, dimensionless drag coefficients are set the same as those shown in Figure 4.2, which are derived from both CFD simulations and wind tunnel experiments [41]. Analyzing the graphs proposed for the choice of the coefficients, it was decided to take $c_{D_x} = c_{D_y} = 0.3$ as the values for the x and y axes, whereas the value $c_{D_z} = 0.9$ was chosen for the z-axis.

S/C Parameters	Value	Units	Description
M_{SC_0}	410	kg	Initial mass of the spacecraft
$Dim(1)$	1.5	m	Spacecraft dimension along x-axis
$Dim(2)$	1.5	m	Spacecraft dimension along y-axis
$Dim(3)$	0.4	m	Spacecraft dimension along z-axis
$Sup(1)$	0.45	m^2	Area section plane y-z axes
$Sup(2)$	0.45	m^2	Area section plane x-z axes
$Sup(3)$	1.7671	m^2	Area section plane x-y axes
J_{xx}	169.2	$kg \cdot m^2$	Inertia moment around x
J_{yy}	145.81	$kg \cdot m^2$	Inertia moment around y
J_{zz}	231.25	$kg \cdot m^2$	Inertia moment around z
J_{xy}	0.333	$kg \cdot m^2$	Cross inertia moment x-y
J_{xz}	-4.204	$kg \cdot m^2$	Cross inertia moment x-z
J_{yz}	0.283	$kg \cdot m^2$	Cross inertia moment y-z
$c_D(1)$	0.3	--	Drag coefficient direction x-axis
$c_D(2)$	0.3	--	Drag coefficient direction y-axis
$c_D(3)$	0.9	--	Drag coefficient direction z-axis
I_{sp}	212.5	s	Specific impulse Monopropellant hydrazine thruster
T_{Th_m}	293	N	Maximum thrust produced by a single thruster
$uplim$	90	%	Maximum usable thrust percentage
$lowlim$	5	%	Minimum usable thrust percentage
n_{Th}	12	--	Number of thrusters implemented
τ_{Th}	0.1	s	Time constant of thrusters first order filter
τ_{RW}	0.05	s	Time constant of reaction wheel first order filter
$U_{max_{RW}}$	12	$N \cdot m$	Reaction wheel nominal moment generated

Table 4.1: Table summarizing the characteristics of the Phoenix-like spacecraft implemented in the simulator.

4.2 Missions and landing constraints definition

In this subchapter, NASA's Phoenix mission, described in more detail in subchapter 1.4, is analyzed focusing on its final phase, called Terminal Descent. This has been done with the aim of being able to define initial conditions for the simulated missions which are as realistic as possible and also in order to determinate which requirements and limits must be met during touchdown so as to state whether it was successful or not.

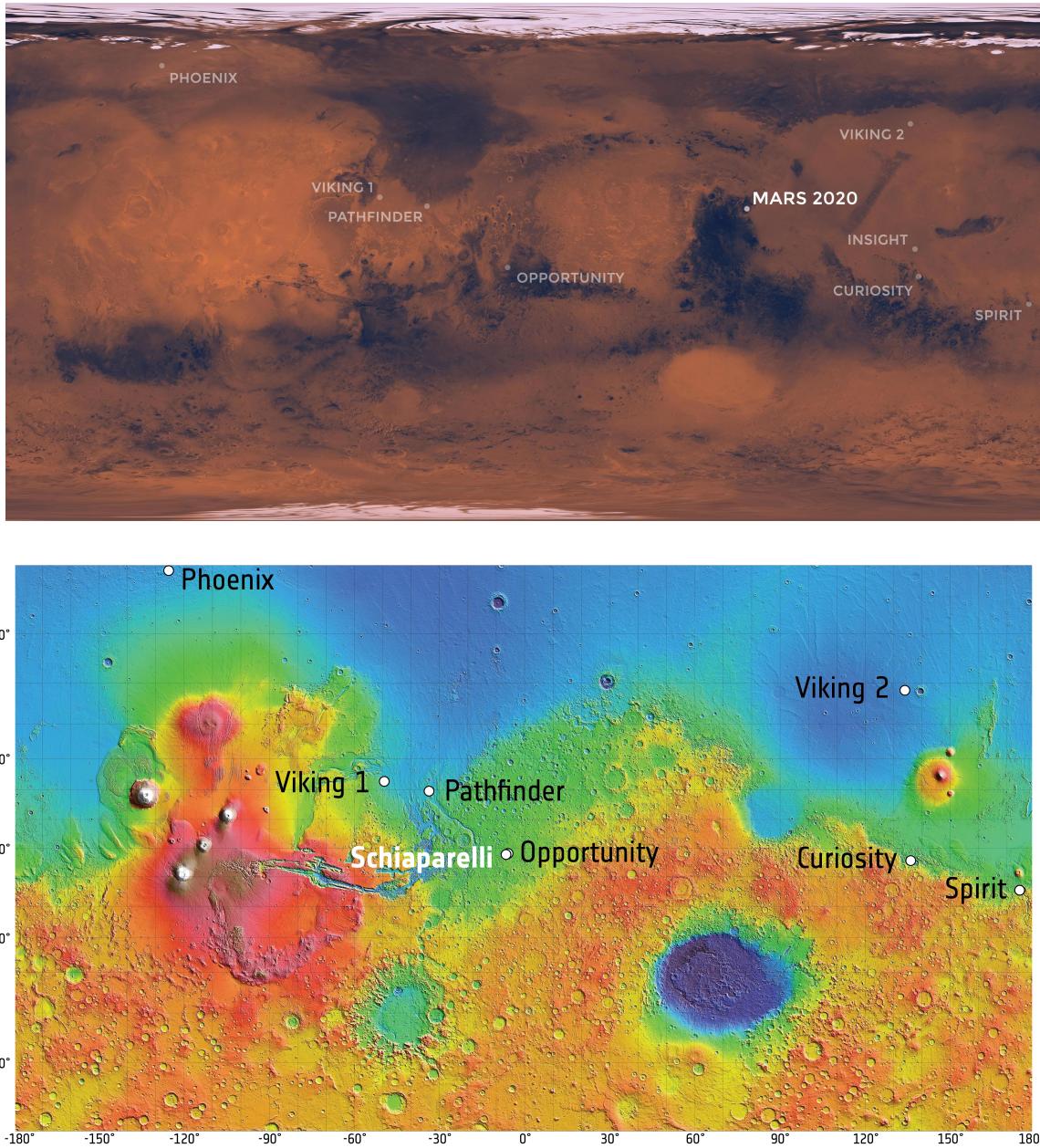


Figure 4.3: Map of Mars in which the landing sites of rovers and landers that have reached the surface of Mars are indicated. [42]

First, the coordinates of the point on the Martian surface where the Phoenix lander landed in May 2008 have been taken as a reference; these are 68.2deg North Latitude and 233.4deg East Longitude. They have then been used and set into the simulator in order to define the position of reference system T. Looking at the coordinates of the point in which the NASA lander landed, it is evident how close this was to the pole. One consequence of this detail is that the spacecraft touched down at a point where the Martian surface was below the MOLA areoid of 4.1km . Having made in subchapter 2.1 the approximation of a perfectly spherical surface with radius equal to the equatorial radius, there will be a discrepancy with the actual mission. Nevertheless, this does not affect the evaluations of the effectiveness of the implemented algorithms.

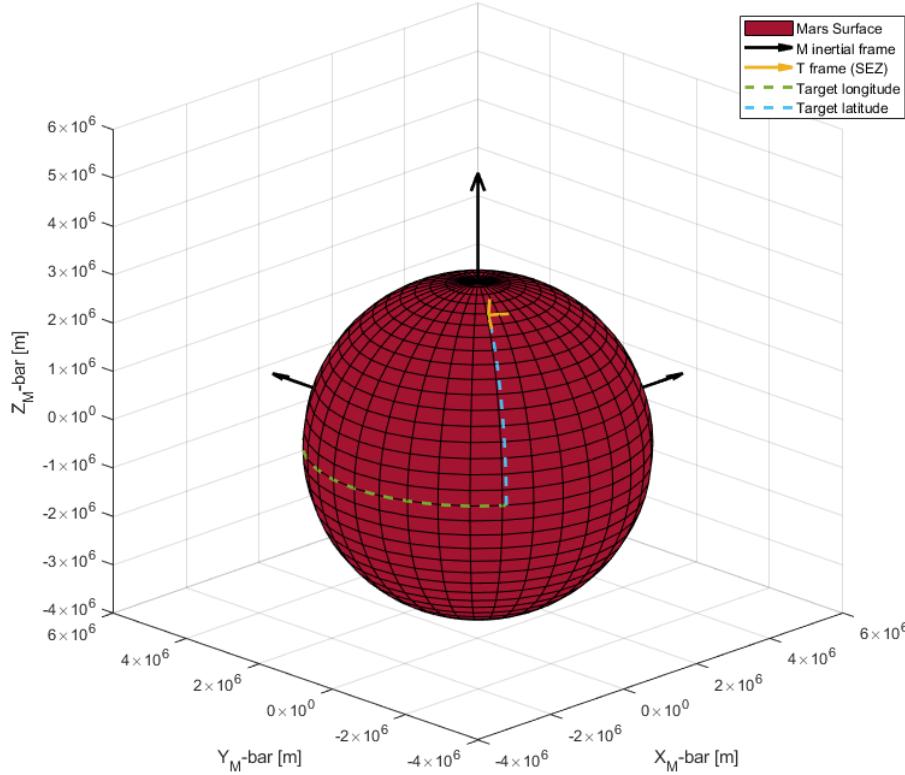


Figure 4.4: Position of reference system T in the Mars model implemented in the simulator.

However, not only is it sufficient for the spacecraft to reach the predetermined landing point in order to decree the success of the mission; indeed, it is also necessary that certain constraints be met. One of the most important requirements is related to the residual kinetic energy still possessed by the lander at the time of touchdown, which is what decrees how violent the impact with the ground is. The requirement that therefore was asked of the Phoenix lander was expressed in terms of residual velocity at touchdown. Indeed, it was required that the residual vertical velocity of the lander with respect to the ground be between 1.4m/s and 3.4m/s [40]. Similarly, as for the residual velocity projected on the horizontal plane, it had to be of less than 1.4m/s in modulus.

Another constraint to take in consideration is related to the attitude of the spacecraft at the time of landing. In fact it is important that the three legs mounted on the bottom of the structure are optimally oriented so as to absorb the aforementioned residual kinetic energy. On the other hand, no requirements in terms of attitude and angular velocities were found in the analyzed papers; therefore, a limit of $\pm 30\text{deg}$ was chosen for attitude in order to apply roll and pitch angles, whereas a requirement of $\pm 5\text{deg}/\text{s}$ was imposed for angular velocities.

At last, there would be limits in terms of consumable fuel as the NASA-developed Phoenix lander was equipped with a tank capable of holding about 70 kg of fuel. However, these have been partially ignored as they have been taken into account, but have not affected the success of the simulated missions. This is because they are intended to test the implemented algorithms and, therefore, will require significantly more energy-intensive maneuvers than the one for which the US lander was designed and optimized.

All requirements have therefore been summarized in Table 4.2 below.

Terminal Descent Phase Requirements	Detail
Final vertical velocity	$1.4\text{m}/\text{s} < V_{f_v} < 3.4\text{m}/\text{s}$
Final horizontal velocity	$V_{f_h} < 1.4\text{m}/\text{s}$
Final roll angle	$-30\text{deg} < \phi_f < 30\text{deg}$
Final pitch angle	$-30\text{deg} < \theta_f < 30\text{deg}$
Final roll rate	$-5\text{deg}/\text{s} < \omega_{x_f} < 5\text{deg}/\text{s}$
Final pitch rate	$-5\text{deg}/\text{s} < \omega_{y_f} < 5\text{deg}/\text{s}$
Final yaw rate	$-5\text{deg}/\text{s} < \omega_{z_f} < 5\text{deg}/\text{s}$

Table 4.2: Table summarizing requirements imposed for the simulated missions.

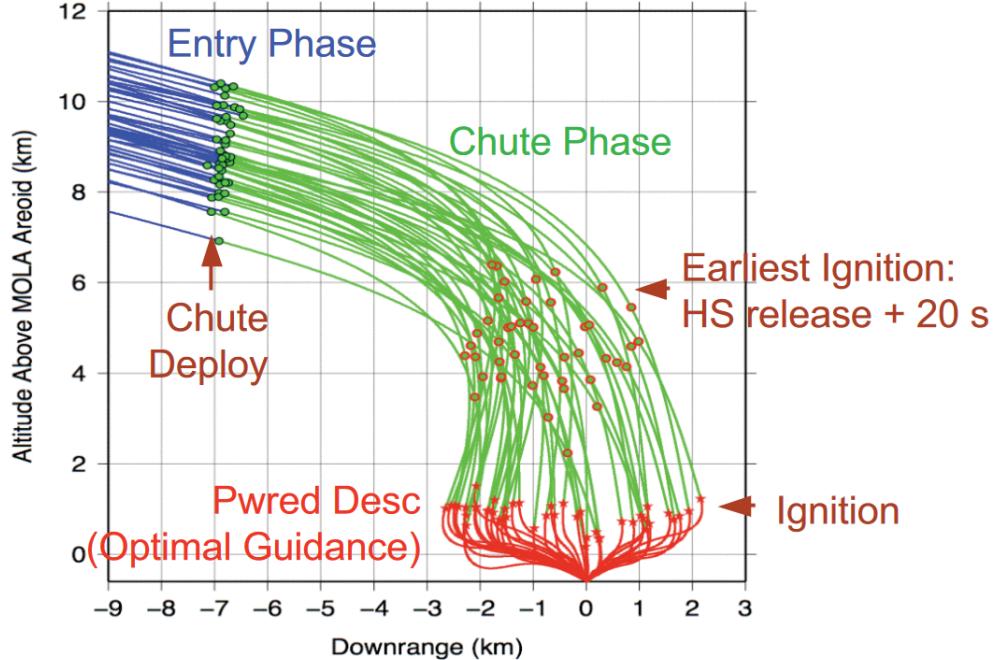


Figure 4.5: A 2D representation of some Monte Carlo runs carried out by Jet Propulsion Laboratory. [43]

Subsequently, initial starting conditions were chosen, which turn out to be particularly important since, depending on these, the maneuver may be more or less challenging and difficult to successfully be completed. In order to test the implemented algorithms, it was decided to foreshorten from how the terminal descent phase was performed by the Phoenix lander. In fact, it was decided to set up a mission in which the objective is to reach reference system T, which is located in the planned landing point, starting from an altitude of 1800 m and a distance of 2.5 km. These parameters were chosen referring to typical accuracy values that could be obtained in 2008 at the end of a Parachute phase preceded by a Ballistic Hypersonic Entry [44].

The following subsections will show the results obtained from the simulations carried out on the basis of the concepts expressed in previous chapters. First, the results of a single nominal mission in which the initial conditions have been chosen with the purpose of verifying the correct operation of the implemented architecture are shown and analyzed in detail. Next, a mission with more hostile initial conditions will be proposed so as to verify the robustness of the implemented algorithms.

4.3 Nominal Mission

This subchapter presents the results obtained from a single simulation by analyzing in detail the evolution of all state variables of interest in time. The purpose of this mission is mainly to understand how guidance algorithms handle the maneuver and if the settings of the latter were correctly set allowing the maneuver to successfully take place respecting constraints mentioned in previous subchapters.

4.3.1 Initial conditions and mission overview

In this first simulation, a mission is set up structured as follows. At the beginning of the maneuver, the spacecraft is at $1800m$ of altitude, having recently deployed its parachutes and still possessing a speed of about $200km/h$ mainly directed downwards. The landing maneuver has been divided into two submaneuvers. In the first part of the mission, the main objectives are to cancel the downrange and crossrange errors accumulated during previous phases by reaching a waypoint located $175m$ above the planned landing point and to dispose of the relative velocity with respect to the surface, i.e. reference system T. Once the waypoint is reached, the vertical descent maneuver begins. In this the spacecraft, guided by guidance algorithms, has the goal of reaching the predicted landing point while respecting constraints mentioned in subchapter 4.2. This structure was given to the mission in order to achieve a trajectory that would allow the S/C to avoid as many natural obstacles as possible, as there might be some in the proximity of the landing site. Table 4.3 shows the initial conditions of the maneuver in more detail.

Parameters	Value	Units	Description
Lat_{T_0}	68.22	deg	Latitude of reference system T
Lng_{T_0}	234.25	deg	Longitude of reference system T
Lat_{SC_0}	68.22	deg	Latitude of reference system T
Lng_{SC_0}	234.365	deg	Longitude of reference system T
X_{SC_0}	-2.35	m	Initial position viewed from reference system T
Y_{SC_0}	2530.6	m	Initial position viewed from reference system T
Z_{SC_0}	1800	m	Initial altitude viewed from reference system T
\dot{X}_{SC_0}	0	m/s	Initial velocity viewed from reference system T
\dot{Y}_{SC_0}	0	m/s	Initial velocity viewed from reference system T
\dot{Z}_{SC_0}	-56	m/s	Initial velocity viewed from reference system T
ϕ_0	0	deg	Initial roll angle
θ_0	0	deg	Initial pitch angle
ψ_0	0	deg	Initial yaw angle
Ψ_0	-90	deg	Initial heading angle (between x-axis and local North)
ω_{x_0}	0	deg/s	Initial roll rate
ω_{y_0}	0	deg/s	Initial pitch rate
ω_{z_0}	0	deg/s	Initial yaw rate

Table 4.3: Table summarizing the initial conditions of the nominal mission.

4.3.2 Guidance algorithm settings

Of paramount importance is the setting of the parameters of guidance algorithms, for it is on these that the success or failure of the mission depends. These were in some cases chosen by referring to references external to this paper, an example being the k_r and k_v gains related to the ZEM/ZEV type part of code within the OSG algorithm [45]. On the other hand, others were optimized as a result of numerous test simulations during the implementation of the same algorithms in the simulator. Having divided the maneuver into two submaneuvers, settings regarding the OSG algorithm will be duplicated and will be distinguished thanks to a subscript that will be equal to 1WP if the parameter refers to the end of the first part, i.e. reaching the first waypoint. If, on the other hand, the parameter is intended to refer to the last part of the maneuver, this will have as subscript the abbreviation LND, that recalls the fact that, at the end of this, there is the landing. All parameters are reported in Table 4.4.

Parameters	Value	Units	Description
OSG Guidance Algorithm settings			
k_r	6	--	ZEM/ZEV position gain
k_v	-2	--	ZEM/ZEV velocity gain
Φ	0.005	--	Sliding variable sign gain
t_f	80	s	Total maneuver time
$t_{f_{1WP}}$	66	s	Maneuver time reaching waypoint
$r_{x_{1WP}}$	0	m	Position viewed from ref. sys. T required at first waypoint
$r_{y_{1WP}}$	0	m	Position viewed from ref. sys. T required at first waypoint
$r_{z_{1WP}}$	175	m	Position viewed from ref. sys. T required at first waypoint
$v_{x_{1WP}}$	0	m/s	Velocity viewed from ref. sys. T required at first waypoint
$v_{y_{1WP}}$	0	m/s	Velocity viewed from ref. sys. T required at first waypoint
$v_{z_{1WP}}$	-6.25	m/s	Velocity viewed from ref. sys. T required at first waypoint
$t_{f_{LND}}$	14	s	Maneuver time reaching landing site from waypoint
$r_{x_{LND}}$	0	m	Position viewed from ref. sys. T required at touchdown
$r_{y_{LND}}$	0	m	Position viewed from ref. sys. T required at touchdown
$r_{z_{LND}}$	0	m	Position viewed from ref. sys. T required at touchdown
$v_{x_{LND}}$	0	m/s	Velocity viewed from ref. sys. T required at touchdown
$v_{y_{LND}}$	0	m/s	Velocity viewed from ref. sys. T required at touchdown
$v_{z_{LND}}$	-1.5	m/s	Velocity viewed from ref. sys. T required at touchdown
Super-Twisting Sliding Mode settings			
λ_{SMA}	77.735	--	Sliding variable gain
β_{SMA}	0.0005	--	Sliding variable derivate gain

Table 4.4: Summary table of parameters related to the OSG and Super-Twisting Sliding Mode algorithms implemented in the simulator.

4.3.3 Results and comments

The results obtained from the nominal mission simulation are given below. A 3D representation of the trajectory that the spacecraft followed during the mission is shown in Figure 4.6. From this representation, the two phases into which the maneuver was divided can easily be distinguished, the first one bringing the spacecraft close to the waypoint, represented in blue, and the second one leading to touchdown. In this mission, initial attitude conditions were set by having the lander have the x-axis of reference system B in the direction of the landing point. This was done in order to make reading the graphs depicting Euler angles and developed angular velocities easier, as doing so will almost uniquely require attitude changes affecting the pitch angle to be accomplished.

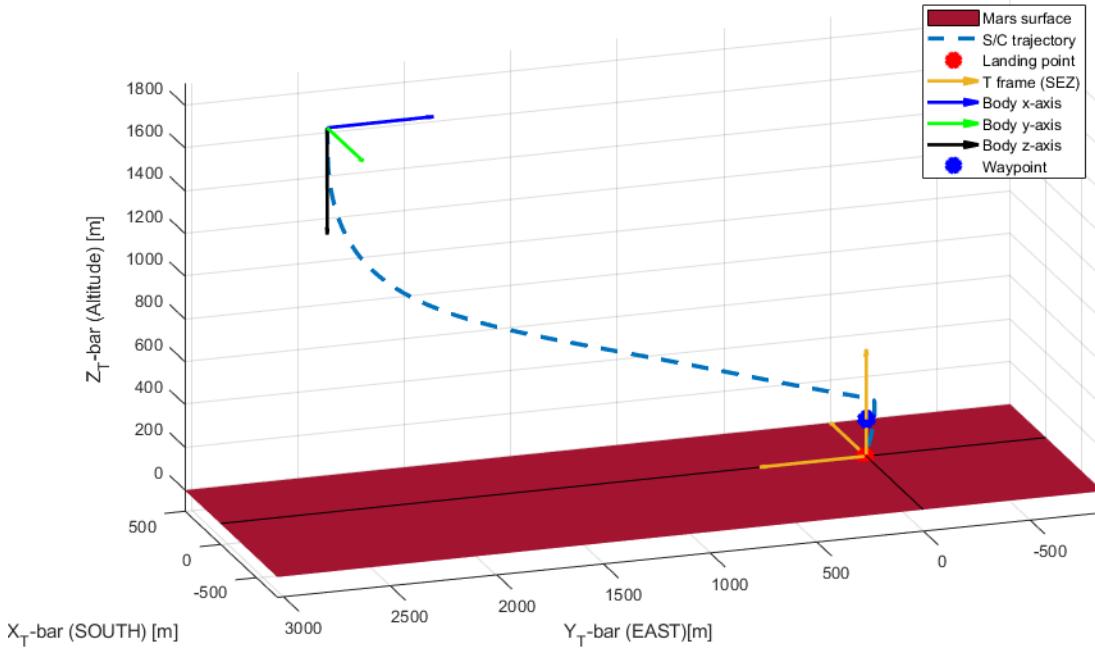
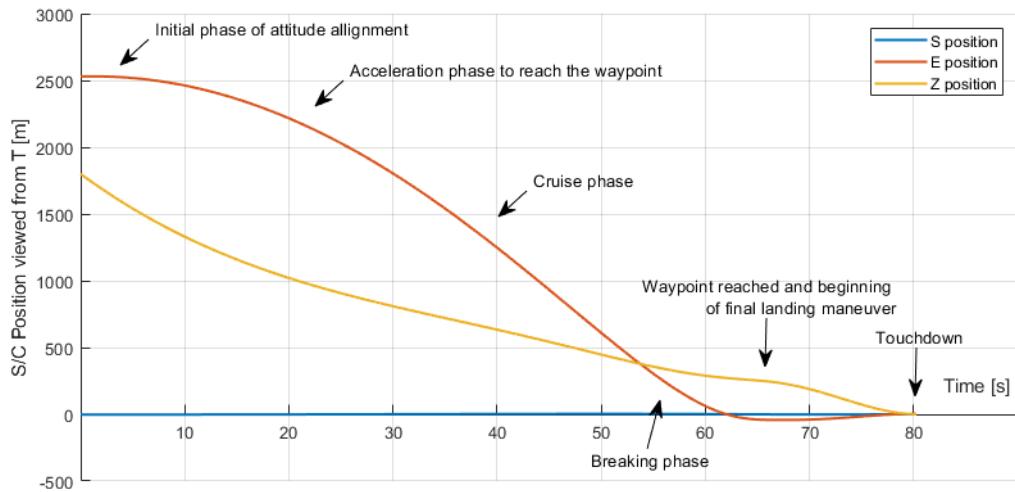


Figure 4.6: 3D representation of the trajectory followed by the lander during the mission.

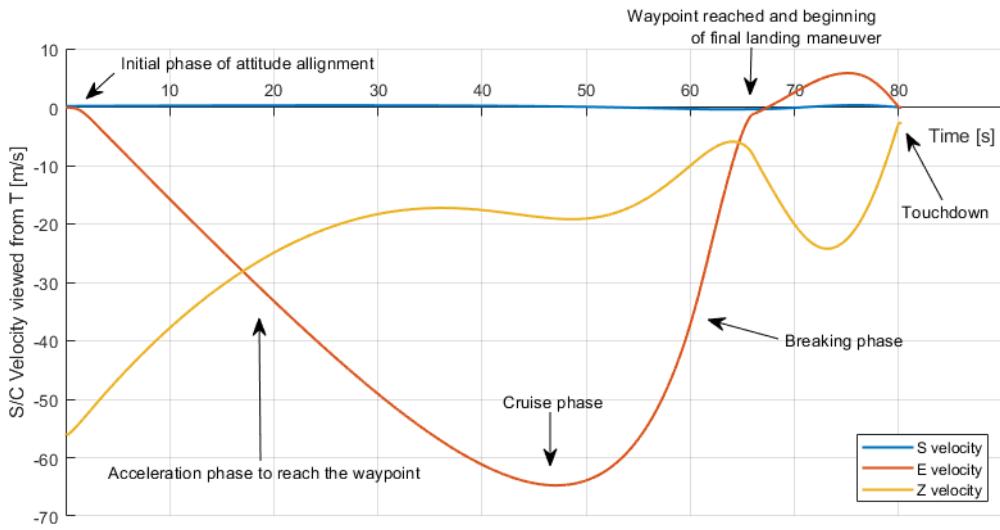
By analyzing more in detail the trajectory drawn by the spacecraft and the relative velocities developed during the mission, shown in Figure 4.7, the distinction between the two phases of the maneuver appear even clearer. In fact, it is possible to notice that, for the first 67 seconds of the maneuver, the lander is mainly concerned with reducing its vertical velocity and reaching the waypoint, located 175 meters above the landing site. Then, the final phase begins, in which the lander straightens its attitude and concludes the maneuver with touchdown while trying to meet the constraints imposed on final velocity.

In performing the initial maneuver, the spacecraft, after having released its parachute, tries to change its attitude by orienting thrusters and making sure that the thrust generated by them creates an acceleration in the direction of the waypoint. As can be seen from Figure 4.8a, the control of the pitch angle is of paramount importance in order to make sure that the aforementioned acceleration arises. Indeed, a negative pitch angle which is equivalent to a dive is commanded, which for the way the spacecraft has been oriented corresponds to a tilt towards the landing site. This change in attitude angles occurs because of the moments generated by thrusters, which are shown in Figure 4.9b. At the end of the first maneuver, when the waypoint is being approached, the spacecraft performs a pullback and reaches a pitch angle of 60deg. This happens as thrusters must be oriented making sure that, once turned on, these decelerate

the vehicle by dropping the previously accumulated speed. Finally, before touchdown, the GNC system commands large moments that straighten the spacecraft, causing it to make a final push that allows the fulfilment of the landing requirements previously mentioned in subchapter 4.2. Forces developed by thrusters during the simulation, reported in Figure 4.9a and related to vector \vec{F}_{act} , are expressed referring to reference system B. Therefore, forces along the x and y axes are null, as thrusters are just oriented in the direction concordant to the z-axis, which also explains why positive accelerations cannot be developed by the actuation system.



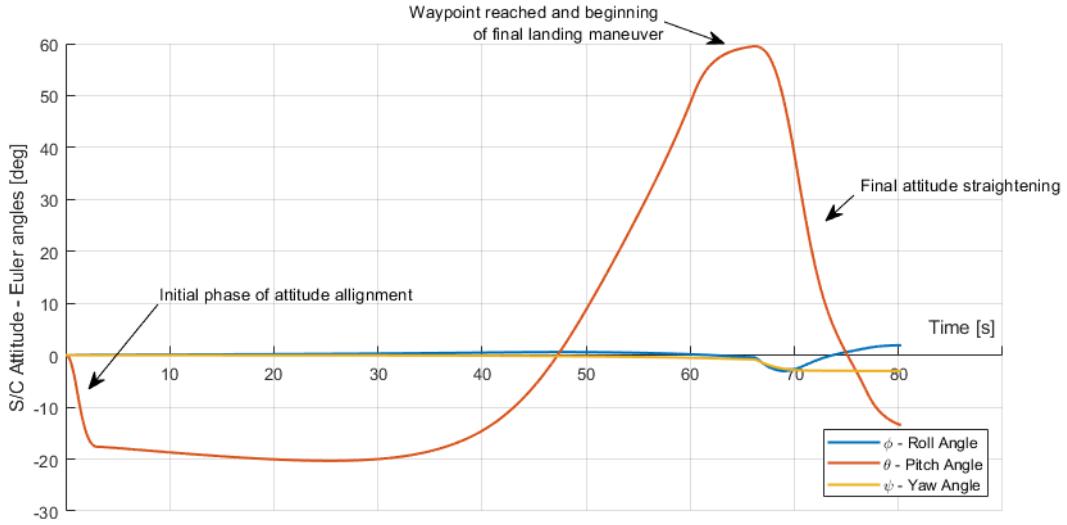
(a) Spacecraft position Vs Time.



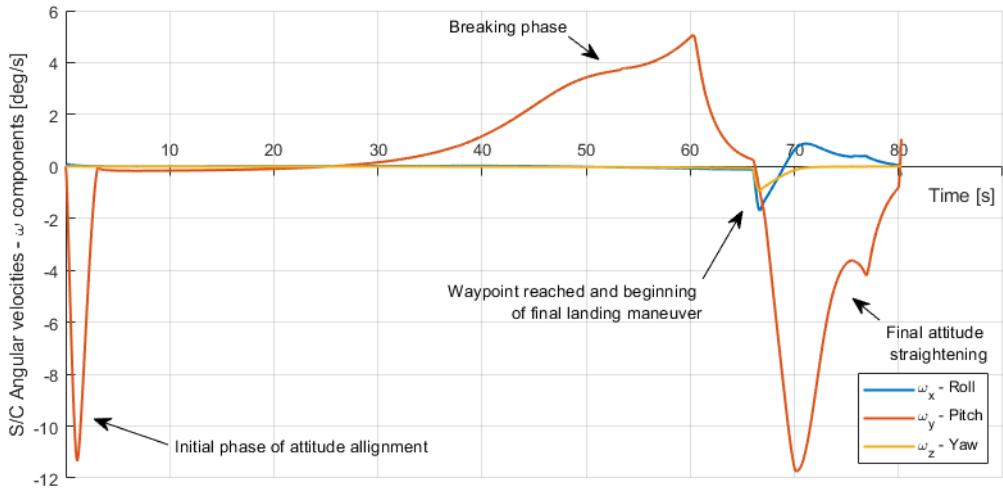
(b) Spacecraft velocity Vs Time.

Figure 4.7: Evolution in time of the components of the spacecraft position and velocity vectors viewed from reference system T.

In addition to the details mentioned in Figure 4.9a, it is also possible to see that, at the beginning of the maneuver, a large amount of thrust is required. This is due to the fact that, at that time, the spacecraft is oriented with the thrusters downwards and uses them in order to brake its free fall by contrasting, as previously mentioned, the vertical velocity left over from the previous parachute phase. After that, there is a phase in which the thrust generated by thrusters is modulated in relation to attitude angles in order to guide the spacecraft towards the waypoint. The absolute minimum value of thrust generated by thrusters during this displacement phase occurs at the moment when the spacecraft performs the flare maneuver in order to slow down. This happens because, at that moment, thrusters are directed towards the ground and an excessive thrust would risk causing the spacecraft to regain altitude, which would be counterproductive. Once the waypoint is reached, the modulus of thrust generated by thrusters drops, reaching the minimum limit. This is done in order to allow the vehicle to perform the righting maneuver before the final push that will eventually lead to touchdown.



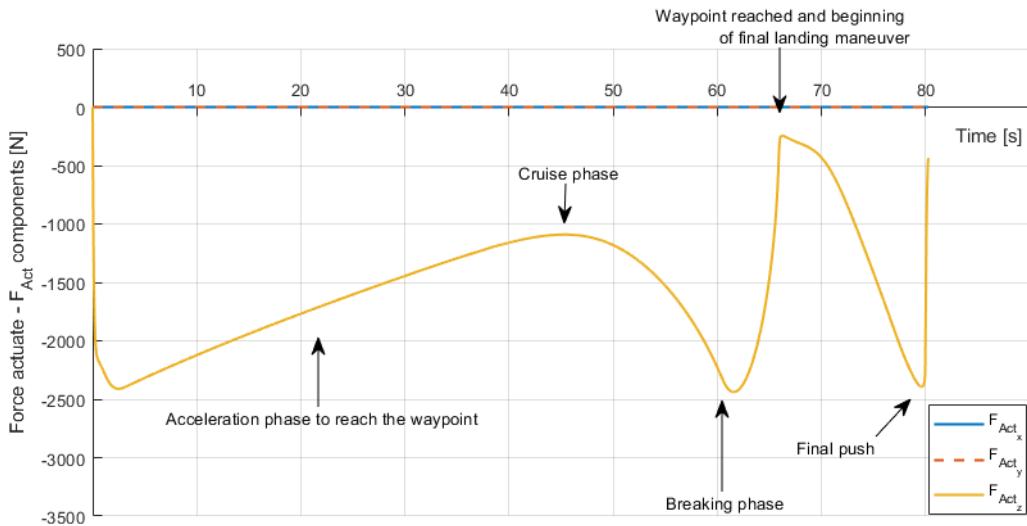
(a) Spacecraft attitude angles Vs Time.



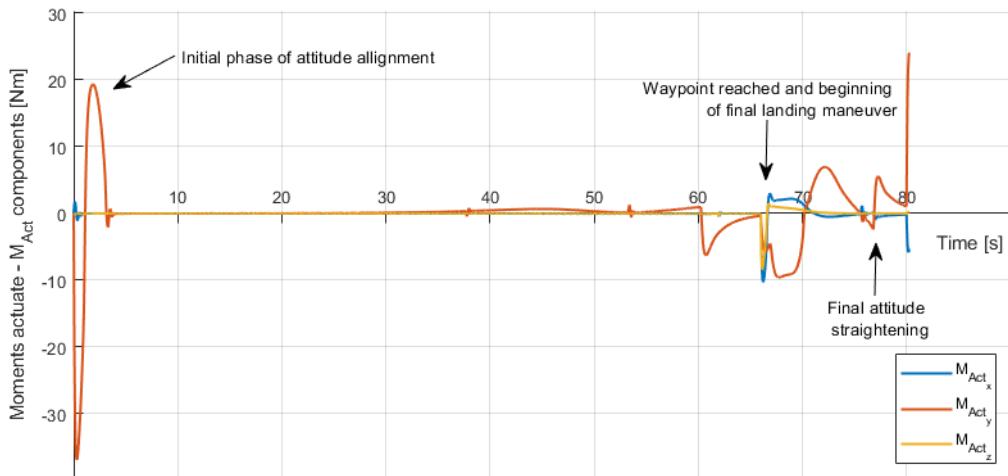
(b) Spacecraft attitude rates Vs Time.

Figure 4.8: Evolution in time of the components of the Euler angles and attitude rates that describe the spacecraft attitude.

Looking at Figure 4.8b, representing time trends of the angular velocities to which the spacecraft is subjected during the mission, a peak of almost 17deg/s can be noticed, which could be a prohibitive speed for the lander structure. As a consequence, research was then conducted based on the Phoenix lander and, in accordance with it, angular velocities of this amplitude have been considered acceptable for maneuvers of this type. In order to confirm this, as can be seen from Figure 4.10, NASA carried out some simulations in order to validate the powered descent phase that reports peaks of up to 20deg/s .



(a) Forces actuated Vs Time.



(b) Moments actuated Vs Time.

Figure 4.9: Evolution in time of the forces and moments actuated viewed from reference system B.

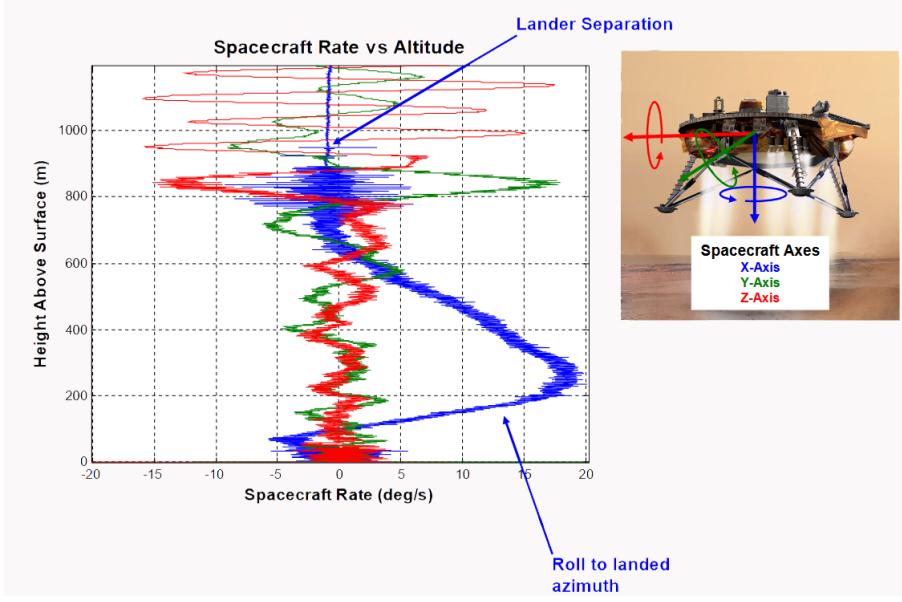


Figure 4.10: Attitude rates registered during a simulation of a powered descent phase of the Phoenix lander.[46]

Finally, after analyzing the mission in its entirety, touchdown performance was analyzed with special attention. For simplicity, results of the latter were summarized in Table 4.5. It can be seen that, despite the considerable distance between the maneuver starting point and the touchdown site, the implemented algorithms were able to bring the spacecraft to its destination committing an error of only about six meters and meeting the constraints imposed on velocity, final attitude angles and residual angular velocities at touchdown. The amount of fuel used corresponded to about 160 kg, which is more than double the capacity of the Phoenix's fuel tanks. However, it should be considered that the US lander was optimized for a much less energy-intensive mission than the one implemented in this simulator.

Parameters	Value	Units	Description
$X_{SC_{final}}$	0.9569	m	Position viewed from reference system T at touchdown.
$Y_{SC_{final}}$	5.7535	m	Position viewed from reference system T at touchdown.
$Z_{SC_{final}}$	0	m	Altitude viewed from reference system T at touchdown.
$\dot{X}_{SC_{final}}$	0.33444	m/s	Velocity viewed from reference system T at touchdown.
$\dot{Y}_{SC_{final}}$	0.9496	m/s	Velocity viewed from reference system T at touchdown.
$\dot{Z}_{SC_{final}}$	-3.2157	m/s	Velocity viewed from reference system T at touchdown.
ϕ_{final}	3.1688	deg	Roll angle at touchdown.
θ_{final}	-16.142	deg	Pitch angle at touchdown.
ψ_{final}	-4.2088	deg	Yaw angle at touchdown.
$\omega_{x_{final}}$	0.26328	deg/s	Roll rate at touchdown.
$\omega_{y_{final}}$	-0.82092	deg/s	Pitch rate at touchdown.
$\omega_{z_{final}}$	-0.0039696	deg/s	Yaw rate at touchdown.
$Fuel_{used}$	159.17	kg	Amount of fuel used to complete the mission.

Table 4.5: Table summarizing the initial conditions of the nominal mission.

4.4 Perturbed Mission

This subchapter reports and analyzes the results of a mission similar to the one described in subchapter 4.3, but with the addition of some perturbations to the initial conditions.

4.4.1 Initial conditions and mission overview

This second simulation aims at analyzing how perturbations of the initial conditions, specifically concerning velocity, attitude and angular velocities, can affect the performance of the implemented algorithms. In fact, it was decided to maintain the initial altitude of 1800m by, however, reducing the difference in latitude and longitude between the spacecraft at the maneuver starting point and the landing site. This way, not changing the setting of the algorithms affecting the timing of the maneuver was possible; therefore, touchdown is still expected to occur at 80 seconds from the start of the mission. As for the initial velocity of the spacecraft, this has been slightly modified with the addition of contributions to the two components that in the nominal mission of chapter 4.3 were null. Contributions regarding attitude angles and initial angular velocities possessed by the spacecraft have also been added in order to simulate a parachute release that did not perfectly occur. Table 4.6 shows the initial conditions of the maneuver in more detail.

Parameters	Value	Units	Description
Lat_{T_0}	68.22	deg	Latitude of reference system T
Lng_{T_0}	234.25	deg	Longitude of reference system T
Lat_{SC_0}	68.226	deg	Latitude of reference system T
Lng_{SC_0}	234.33	deg	Longitude of reference system T
X_{SC_0}	-356.98	m	Initial position viewed from reference system T
Y_{SC_0}	1760	m	Initial position viewed from reference system T
Z_{SC_0}	1800	m	Initial altitude viewed from reference system T
\dot{X}_{SC_0}	-7	m/s	Initial velocity viewed from reference system T
\dot{Y}_{SC_0}	-5	m/s	Initial velocity viewed from reference system T
\dot{Z}_{SC_0}	-56	m/s	Initial velocity viewed from reference system T
ϕ_0	10	deg	Initial roll angle
θ_0	8	deg	Initial pitch angle
ψ_0	0	deg	Initial yaw angle
Ψ_0	-90	deg	Initial heading angle (between x-axis and local North)
ω_{x_0}	12	deg/s	Initial roll rate
ω_{y_0}	8	deg/s	Initial pitch rate
ω_{z_0}	10	deg/s	Initial yaw rate

Table 4.6: Table summarizing the initial conditions of the perturbed mission.

4.4.2 Guidance algorithm settings

Again, driving algorithm settings play a key role in achieving touchdown requirements. These were kept the same as those chosen for the nominal mission with the aim of demonstrating that, despite initial conditions having been perturbed, they are nevertheless able to conduct the spacecraft to its destination. All parameters are reported in Table 4.4.

4.4.3 Results and comments

Results obtained from the simulation of the perturbed mission are shown below. A 3D representation of the trajectory that the spacecraft followed during the mission is shown in Figure 4.11. From this representation, it is clear how, despite the different initial conditions, the spacecraft draws a trajectory very similar to that of the nominal mission. This happens as once perturbations on the initial conditions are counteracted, the two maneuvers converge to the landing site in the same manner.

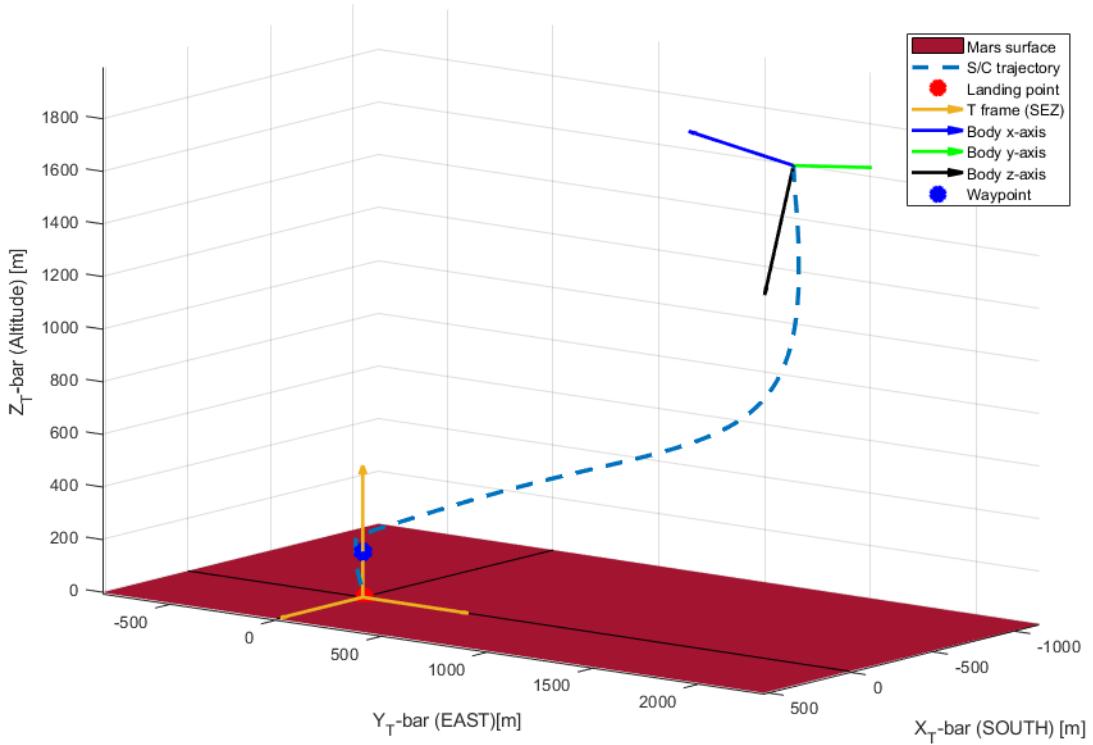
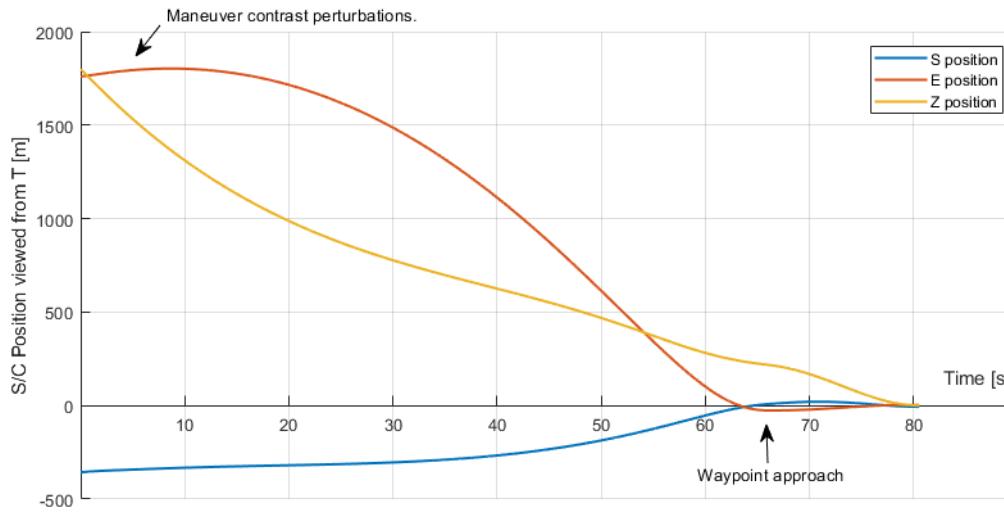
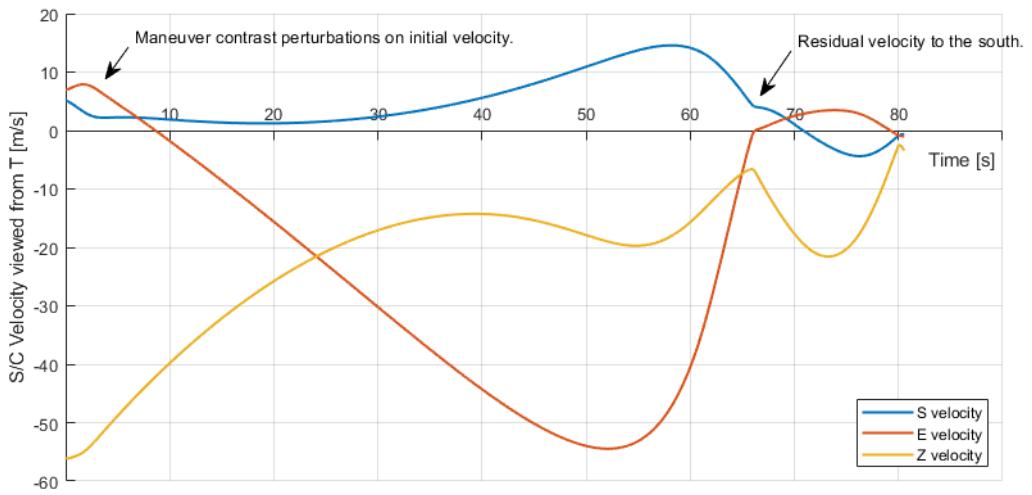


Figure 4.11: 3D representation of the trajectory followed by the lander during the mission.

By analyzing in more detail the trajectory drawn by the spacecraft and the relative velocities developed during the mission, shown in Figure 4.7, it can be seen that in the first 10 seconds of the simulations the GNC system is engaged in counteracting the component referring to the East direction of the velocity vector, that tends to carry the spacecraft away from the landing point. The time taken in order to perform this maneuver puts a strain on the control algorithm, which, in fact, reaches the waypoint with a southwards velocity relative to the target that is not null. This aspect could be solved by increasing the expected time $t_{f_{1WP}}$ for the first part of the maneuver. However, the second phase, i.e. the landing phase, compensates the aforementioned drawback by bringing the spacecraft close to the landing point within the imposed constraints.



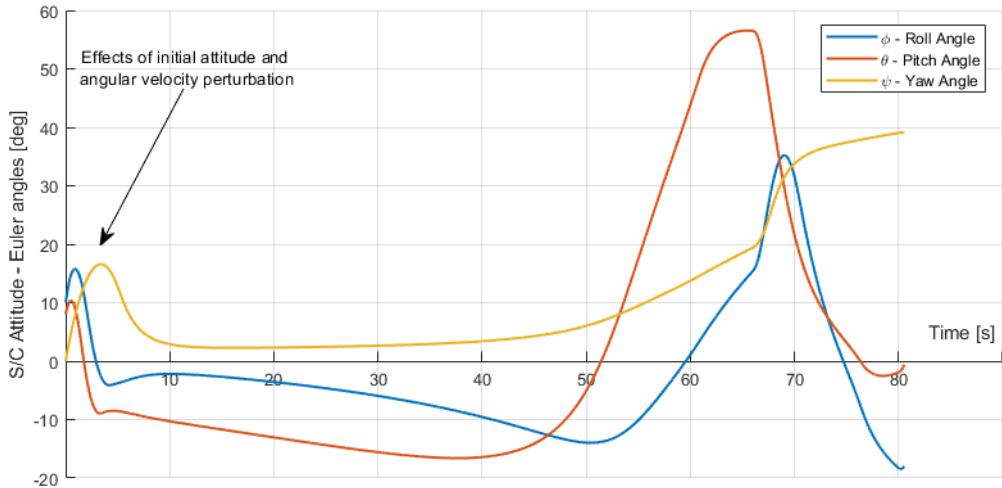
(a) Spacecraft position Vs Time.



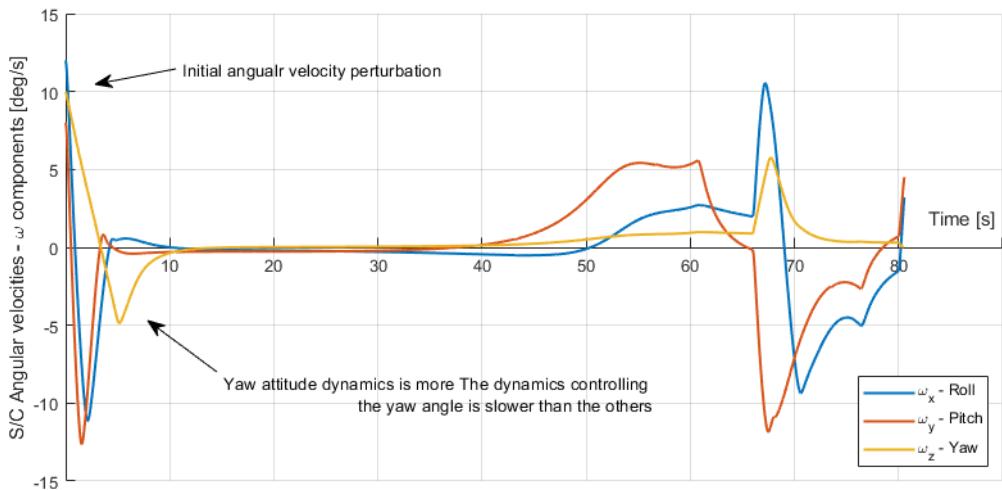
(b) Spacecraft velocity Vs Time.

Figure 4.12: Evolution in time of the components of the spacecraft position and velocity vectors viewed from reference system T.

However, analyzing Figures 4.13a and 4.13b, respectively representing the time trends of Euler angles and angular velocities, it is possible to see the effect that perturbations on the initial conditions had on these parameters. It can first be noticed how these values no longer start from a null value and it can also be noticed that, in the first few seconds of simulation, the algorithms work in order to have the values of these parameters coincide with those needed in order to conduct the maneuver.



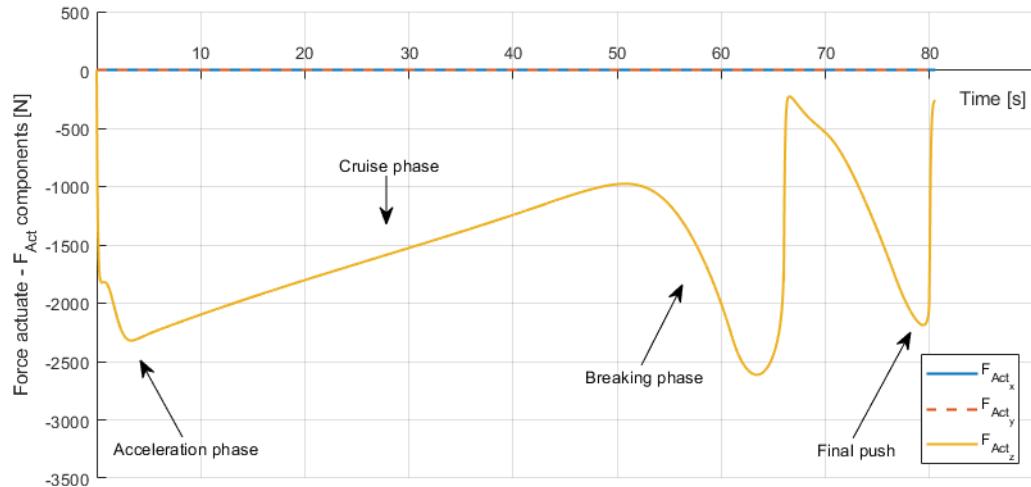
(a) Spacecraft attitude angles Vs Time.



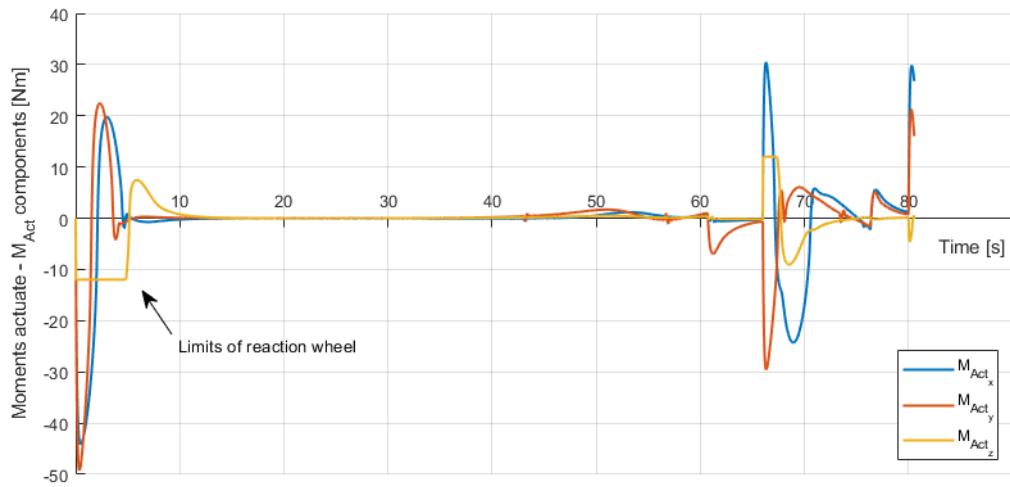
(b) Spacecraft attitude rates Vs Time.

Figure 4.13: Evolution in time of the components of the Euler angles and attitude rates that describe the spacecraft attitude.

Then, by analyzing the trend of the forces implemented by thrusters during the mission, shown in Figure 4.14a, it can be seen that this does not significantly differ from that obtained in the nominal mission. In fact, the acceleration, cruise and braking phases related to the first part of the maneuver and also the final push related to the landing maneuver can be noticed just as could be done for the nominal mission.



(a) Forces actuated Vs Time.



(b) Moments actuated Vs Time.

Figure 4.14: Evolution in time of the forces and moments actuated viewed from reference system B.

Another feature of the attitude control dynamics can be noticed in the first moments of simulation, as the yaw angle control turns out to be at least twice as slow as that of the other two Euler angles. This is because, as can be seen analyzing Figure 4.14b, the moments implemented by thrusters acting on the roll and pitch axes turn out to be much more effective as they are free to reach very high values of generated moments; on the other hand, the reaction wheel that controls the yaw angle dynamics turns out to have much stricter limits as it is unable to provide a moment higher than $12N \cdot m$, thus having the phenomenon of saturation. As far as time course of the angular velocities is concerned, this appears not to have peaks of excessive amplitude, according to the considerations made for the nominal mission as well.

Lastly, parameter values obtained at the time of touchdown are analyzed and reported in Table 4.7. A degradation in the accuracy of the final landing position can be noticed, as the spacecraft ends up landing almost seven meters away from the predicted landing point. The same can be noticed for velocities recorded at the moment of the impact with the ground, as the vertical velocity turns out to be at the limit of the constraints imposed for the success of the maneuver. Attitude angles ϕ and θ turn out to comply with the imposed constraints, just as the three angular velocities do, since just the pitch angular velocity approaches the predetermined limit. On the other hand, as far as fuel consumption is concerned, the same considerations previously made for the nominal mission can be made. Therefore, the landing can be decreed as successful.

Parameters	Value	Units	Description
$X_{SC_{final}}$	0.21263	m	Position viewed from reference system T at touchdown.
$Y_{SC_{final}}$	3.073	m	Position viewed from reference system T at touchdown.
$Z_{SC_{final}}$	0	m	Altitude viewed from reference system T at touchdown.
$\dot{X}_{SC_{final}}$	-0.042684	m/s	Velocity viewed from reference system T at touchdown.
$\dot{Y}_{SC_{final}}$	-0.17673	m/s	Velocity viewed from reference system T at touchdown.
$\dot{Z}_{SC_{final}}$	-2.8684	m/s	Velocity viewed from reference system T at touchdown.
ϕ_{final}	1.9007	deg	Roll angle at touchdown.
θ_{final}	-13.313	deg	Pitch angle at touchdown.
ψ_{final}	-3.0167	deg	Yaw angle at touchdown.
$\omega_{x_{final}}$	-0.36545	deg/s	Roll rate at touchdown.
$\omega_{y_{final}}$	1.0675	deg/s	Pitch rate at touchdown.
$\omega_{z_{final}}$	0.0014148	deg/s	Yaw rate at touchdown.
$Fuel_{used}$	159.59	kg	Amount of fuel used to complete the mission.

Table 4.7: Table summarizing parameters involved in touchdown.

Chapter 5

Conclusions, future works and possible development

5.1 Conclusions

In conclusion, based on the simulations performed, it can be said that the implemented guidance algorithms seem to respond well to perturbations, managing to demonstrate robustness with respect to variations in initial conditions from nominal ones. The simulated missions were successfully completed as they meet the limits for touchdown conditions despite the fact that the maneuver time settings of the OSG algorithm were not optimized, , i.e. they have not been varied between the two missions. The choice to carry out the simulations by implementing a spacecraft similar to Phoenix turned out to partially be an unfavorable condition, as the US lander was designed and optimized for its mission and not for the maneuvers that have been performed in this work, thus decrying suboptimal conditions for the implemented algorithms that could therefore not express their full potential. This is highlighted by the fact that the fuel consumption obtained from the simulations exceeded the maximum capacity of the tanks installed in the US lander by more than their double. This, however, is the price to be paid for using a real vehicle in order to have the assurance that the parameters related to it are consistent. However, it can be said that guidance algorithms that exploit sliding mode logic and the ZEM/ZEV technique possess enormous potential for this type of application, proving to be extremely malleable and capable of being adapted to applications, even if not strictly designed in order to work with this precise type of algorithms. In addition, the enormous advantage brought by the great ease with which it is possible to add features such as the possibility of dividing the maneuver into several phases divided by waypoints as to obtain a trajectory that allows avoiding obstacles or collisions with the terrain can be mentioned. In short, the path taken by the scientific community, which has chosen to invest in these technologies, will surely lead to results and goals that years ago seemed unattainable.

5.2 Future works and possible development

By furtherly developing this work and especially the implemented simulator, simulations closer and closer to reality could be obtained. This would allow more informed and accurate evaluation of the performance of the algorithms. Obviously, in order to ascertain the actual robustness of the Optimal Sliding Guidance algorithm and the Super-Twisting Sliding Mode controller applied to this type of mission, much more accurate and complex models than those developed for this paper would be required. In addition, many more simulations applying the Monte Carlo method and testing even more prohibitive initial conditions than those used in this paper would be required. Other improvements that could be made to the simulator are diversifying the update rate of the dynamic equations from those of the algorithms and the propulsion system, so that the GNC system installed on the on-board computer would be limited by the update rate of the latter. The implementation of a spacecraft other than the Phoenix lander, which was chosen for this work, could also prove to be an excellent way to be able to evaluate how much the type of spacecraft and its actuators affect the performance of the implemented guidance algorithms. The natural continuation of this work could also be to identify a larger spacecraft, perhaps designed for possible manned missions. A further assessment that could be made concerns optimizing the algorithm settings to achieve lower fuel consumption. This would allow, for the same overall spacecraft weight, a larger payload to be embarked.

Bibliography

- [1] WILLIAMS, David R. Mars fact sheet.
<http://nssdc.gsfc.nasa.gov/planetary/factsheet/marsfact.html>, 2004.
- [2] The European Space Agency Webpage. Beautiful new images from Rosettas approach to Mars: OSIRIS UPDATE.
https://www.esa.int/Science_Exploration/Space_Science/Rosetta
- [3] The European Space Agency Webpage. Missions to Mars.
https://www.esa.int/ESA_Multimedia/Images/2019/05/Missions_to_Mars, 2019
- [4] Wikipedia contributors. Mars — Wikipedia, The Free Encyclopedia.
<https://en.wikipedia.org/w/index.php?title=Mars&oldid=1094089512>, 2022.
- [5] ULRICH, Steve. Guidance and Adaptive Control for Precision Landing on Mars. Library and Archives Canada= Bibliothèque et Archives Canada, Ottawa, 2008.
- [6] DESAI, Prasun N., et al. Entry, descent, and landing performance of the mars phoenix lander. *Journal of Spacecraft and Rockets*, 2011, 48.5: 798-808.
- [7] NASA Science, Mars exploration program. Mars Phoenix
<https://mars.nasa.gov/mars-exploration/missions/phoenix/>
- [8] GROVER III, Myron R.; CICHY, Benjamin D.; DESAI, Prasun N. Overview of the Phoenix entry, descent, and landing system architecture. *Journal of Spacecraft and Rockets*, 2011, 48.5: 706-712.
- [9] PORTOCK, Brian, et al. Navigation challenges of the Mars Phoenix Lander mission. In: AIAA/AAS Astrodynamics Specialist Conference and Exhibit. 2008. p. 7214.
- [10] QUEEN, Eric; DESAI, Prasun; PRINCE, Jill. Multi-Body Modeling and Simulation for the Mars Phoenix Lander Entry, Descent and Landing. In: AIAA/AAS Astrodynamics Specialist Conference and Exhibit. 2009. p. 7347.
- [11] GOLDSTEIN, Barry; SHOTWELL, Robert. Phoenix-the first Mars scout mission (a mid-term report). In: 2006 IEEE Aerospace Conference. IEEE, 2006. p. 18 pp.
- [12] AMARAL, Tiago. Guidance Optimization for Mars Pinpoint Landing with Optimal Trigger and Re-optimization. 2019.
- [13] SpaceX. Mars And Beyond The road to making Humanity Multiplanetary.
<https://www.spacex.com/human-spaceflight/mars/index.html>, 2022.
- [14] DAVIS, Jody, et al. Guidance and control algorithms for the Mars entry, descent and landing systems analysis. In: AIAA/AAS Astrodynamics Specialist Conference. 2010. p. 7972.
- [15] FURFARO, Roberto, et al. Adaptive generalized ZEM-ZEV feedback guidance for planetary landing via a deep reinforcement learning approach. *Acta Astronautica*, 2020, 171: 156-171.

- [16] Yao Zhang, Yanning Guo, Guangfu Ma, Tianyi Zeng (2017) Collision avoidance ZEM/ZEV optimal feedback guidance for powered descent phase of landing on Mars
- [17] KARLGAARD, Christopher D.; TYNIS, Jake A. Mars Phoenix EDL Trajectory and Atmosphere Reconstruction Using NewSTEP. 2019.
- [18] GROVER III, Myron R.; CICHY, Benjamin D.; DESAI, Prasun N. Overview of the Phoenix entry, descent, and landing system architecture. *Journal of Spacecraft and Rockets*, 2011, 48.5: 706-712.
- [19] PRINCE, Jill L., et al. Mars phoenix entry, descent, and landing simulation design and modeling analysis. *Journal of Spacecraft and Rockets*, 2011, 48.5: 756-764.
- [20] BATE, Roger R., et al. *Fundamentals of astrodynamics*. Courier Dover Publications, 2020.
- [21] YANG, Yaguang. Spacecraft attitude determination and control: Quaternion based method. *Annual Reviews in Control*, 2012, 36.2: 198-219.
- [22] NASA. Image Gallery.
<https://www.jpl.nasa.gov/images>.
- [23] NASA. Press Kit/MAY 2008 Phoenix Mission.
<https://www.nasa.gov/news/media/presskits/index.html>
- [24] Arizona university. Mars Education - Wind.
<https://marsed.asu.edu/mep/wind>
- [25] HIRT, Christian, et al. Kilometer-resolution gravity field of Mars: MGM2011. *Planetary and Space Science*, 2012, 67.1: 147-154.
- [26] NASA. Mars Climate Database.
https://www-mars.lmd.jussieu.fr/mcd_python/index.html
- [27] TAMPPARI, Leslie K., et al. Expected atmospheric environment for the Phoenix landing season and location. *Journal of Geophysical Research: Planets*, 2008, 113.E3.
- [28] SHIH, Yusi. Wind science and engineering: Solutions and new Certificate of Approvals proposed for offshore wind turbine installation. *Energy Reports*, 2020, 6: 1574-1584.
- [29] Nasa. Mars Atmosphere Model.
<https://www.grc.nasa.gov/www/k-12/airplane/atmosmrm.html>.
- [30] KERSHNER, Richard B.; FISCHELL, Robert E. Gravity-Gradient Stabilization of Earth Satellites. In: *Peaceful Uses of Automation in Outer Space*. Springer, Boston, MA, 1966. p. 249-267.
- [31] WIBBEN, Daniel R.; FURFARO, Roberto. Optimal sliding guidance algorithm for Mars powered descent phase. *Advances in space research*, 2016, 57.4: 948-961.
- [32] WANG, Pengyu; GUO, Yanning; WIE, Bong. Orbital rendezvous performance comparison of differential geometric and ZEM/ZEV feedback guidance algorithms. *Astroynamics*, 2019, 3.1: 79-92.
- [33] FURFARO, Roberto, et al. Analysis and Performance Evaluation of the ZEM/ZEV Guidance And Its Sliding Robustification For Autonomous Rendezvous in Relative Motion. 2016.
- [34] FURFARO, Roberto, et al. Adaptive generalized ZEM-ZEV feedback guidance for planetary landing via a deep reinforcement learning approach. *Acta Astronautica*, 2020, 171: 156-171.

- [35] SHTESSEL, Yuri, et al. Sliding mode control and observation. New York: Springer New York, 2014.
- [36] ZHANG, Yao, et al. Collision avoidance ZEM/ZEV optimal feedback guidance for powered descent phase of landing on Mars. *Advances in space research*, 2017, 59.6: 1514-1525.
- [37] LI, Yang-Rui; PENG, Chao-Chung. Super-Twisting Sliding Mode Control Law Design for Attitude Tracking Task of a Spacecraft via Reaction Wheels. *Mathematical Problems in Engineering*, 2021, 2021.
- [38] NADAFI, Reza, et al. Super-twisting sliding mode control design based on Lyapunov criteria for attitude tracking control and vibration suppression of a flexible spacecraft. *Measurement and Control*, 2019, 52.7-8: 814-831.
- [39] LU, Kunfeng; XIA, Yuanqing. Finitetime attitude control for rigid spacecraftbased on adaptive supertwisting algorithm. *IET Control Theory & Applications*, 2014, 8.15: 1465-1477.
- [40] PRINCE, Jill L., et al. Mars phoenix entry, descent, and landing simulation design and modeling analysis. *Journal of Spacecraft and Rockets*, 2011, 48.5: 756-764.
- [41] NASA. FUN3D:Application #33. Mars Phoenix Lander.
<https://fun3d.larc.nasa.gov/example-33.html>
- [42] ESA. Landing Sites on Mars.
https://www.esa.int/ESA_Multimedia/Images/2016/10/Landing_Sites_on_Mars
- [43] WOLF, Aron A. Pinpoint landing technology assessment. 2008.
- [44] WOLF, Aron A., et al. Systems for pinpoint landing at Mars. 2004.
- [45] AHN, Jaemyung; GUO, Yanning; WIE, Bong. Precision ZEM/ZEV feedback guidance algorithm utilizing Vintis analytic solution of perturbed Kepler problem. In: *Proceedings of the AAS/AIAA Space Flight Mechanics Meeting*. 2016.
- [46] CICHY, Benjamin D., et al. The Phoenix Mars Landing: An Initial Look.

