



UNIVERSITÀ
DI PISA

DEPARTMENT OF CIVIL AND INDUSTRIAL ENGINEERING
Master of Science in Space Engineering

Space System - Final Term Report

HERMES Halo EaRth Moon rElay Satellites

Pre Phase-A Mission Report

Authors - Group I:

BIANCHINI Elia
DESHMUKH Lalit
FURFARI Mauro
MAESTRIPIERI Michela
MARRADI Francesco
NOCILLI Mario
PODESTÀ Lucrezia

Supervisor:

MARCUCCIO Salvo
BASSETTO Marco

Accademic Year 2023/2024

Pisa, 20 Maggio 2024



Abstract

Several missions to the Moon are planned for the coming decades, ranging from NASA's Lunar Gateway research station to smaller missions as China's Chang'e 7. The heightened interest in going to the Moon, shows that there could be a market in providing satellite communications beyond Earth, especially to the farside of the Moon that has not line of sight towards the Earth.

This report presents a proposal of a commercial mission to offer support to future human activities on the Moon, providing continuous communication both with Earth and Moon, and a Lunar Navigation Satellite System service, ensuring precise positioning of vehicles on the surface.

An entire Pre-Phase A study was conducted to identify detailed objectives to be achieved, assess the feasibility of such tasks, define a set of detailed requirements and develop a preliminary mission proposal. Four architectures were considered, compared, analyzed, and evaluated to identify the most promising one in terms of performance, risks, and costs.

The selected architecture consists of a constellation of four twin satellites, that will be placed in a Halo orbit around EML2. Each satellite will have 9 main antennas mounted on the front face of the spacecraft, plus another 8 antennas distributed on the other faces for redundancy. The expected data rate ranges from 5 Gbps to 1 Gbps depending on the lunar application.

The constellation will be launched on a Falcon 9 rocket in five years and will last at least 7 years, with 50% lifetime expansion expected. The mission shows a great potential of scalability with the possibility to launch other four or more satellites in a northern halo orbit enlarging the surface coverage and data transmission capability.

Contents

1	Introduction	1
1.1	Lunar Far-Side Features	2
1.2	Earth-Moon Lagrange Point 2	5
2	Mission Objective	7
2.1	Lunar Human Activities	7
2.2	Support Mission Objectives	12
2.3	Mission Requirements	13
3	Mission Architecture	16
3.1	Architecture A	17
3.2	Architecture B	18
3.3	Architecture C	19
3.4	Architecture D	20
3.5	Decisional Matrix	22
4	Mission Profile	23
4.1	Working orbit	23
4.2	Transfer trajectory	25
4.3	Results Obtained from Keplerian approaches	41
5	Satellites and Subsystems Design	42
5.1	Telecommunication Systems	43
5.1.1	System Architecture	43
5.1.2	Ground Section	48
5.1.3	Lunar Navigation Satellite System (LNSS)	50
5.1.4	Selected COTS Components	51
5.2	Propulsion Module	54
5.2.1	Selection Rationale	54
5.2.2	Orbit Injection	54
5.2.3	Station Keeping Activities	55
5.2.4	Technology Comparison	56
5.2.5	Propellant and tank	59
5.3	Attitude Determination and Control System	63
5.3.1	Requirements	64
5.3.2	Perturbation Torques	67
5.3.3	Actuators	72
5.3.4	Operations	73
5.3.5	Sensors	75
5.4	Power Generation and Supply	76
5.4.1	Solar Panels	76
5.4.2	Power Storage	77

5.4.3 Stationkeeping power requirements: load shift	79
5.5 Thermal Control	81
5.5.1 Spacecraft Thermal balance	81
5.5.2 External surfaces thermal balance	83
5.5.3 Final results	85
6 Mission Budgets	87
6.1 Link Budget	87
6.2 Mass Budget	87
6.3 Power Budget	90
6.4 Launcher Selection	92
6.5 Risk Analysis	94
6.6 Cost Estimation	97
7 Phase-A Mission Expansion	99
7.1 Work Breakdown Structure	99
7.2 Gantt Chart	101
8 Conclusions	103
A Nomenclature	106
B Appendix	108
B.1 Mission Profile	108
B.1.1 First Order Method	108
B.1.2 Introduction to the "2nd order" approaches	109
B.1.3 Second Order Approaches	112
B.1.4 Restricted Second Order Methods	114
B.2 3 Body Dynamics	120
B.3 Propulsion Modules	125
B.3.1 Impulsive Approximation	125
B.3.2 Technology Comparison: Ep calculations	125
B.3.3 Thruster Selection	126
C Calculations	128
C.1 Restricted Second Order Approaches	128
C.2 Propulsion Calculations	167
C.3 Propellant Evaluation	182
C.4 Link Budget	192
C.5 Inertia Matrix	209
C.6 Attitude Perturbations	210
C.7 Camera Orbit (used in Attitude Plot and Calculation)	213
C.8 Attitude Plot and Calculation	215
C.9 Power Generation battery load shift evaluation	236
C.10 Temperature cold face	238
C.11 Temperature hot face	240
C.12 Thermal equilibrium temperature	242



HERMES
SUPREMÆ DIGNITATIS
ANNO 1343

UNIVERSITÀ
DI PISA

References

245



1. Introduction

It is common knowledge that the Moon rotates at the same rate as its orbital motion around the earth with synchronous rotation, thus keeping the same face towards our planet.

This involve that the majority of the past earth' satellite missions were directed to the near side, both for simplicity of the transfer and a more reliable and direct connection with the ground center in almost every phase of the mission. The "dark side" is remote and harder to reach and that is the main reason why most of information about lunar features regards the earth's side and more commercial and exploration appeal was concentrated there.

However, interest and curiosity moved humans towards the unknown and starting from 1959 with the Soviet Union's *Luna 3* mission that captured and transmitted the first images of the far side (Fig. 1.1) a new space mission window opened.

This was early followed by Soviet Unions' *Luna 10* in 1966, USA *Lunar Orbiter Program* in 1966-1967 and NASA's missions *Explorer 35* in 1967 and *Apollo 8* in 1968, which allowed more detailed images of the surface and information on the Moon's gravitational field, even if their primal mission objective was not specifically to study the far side.

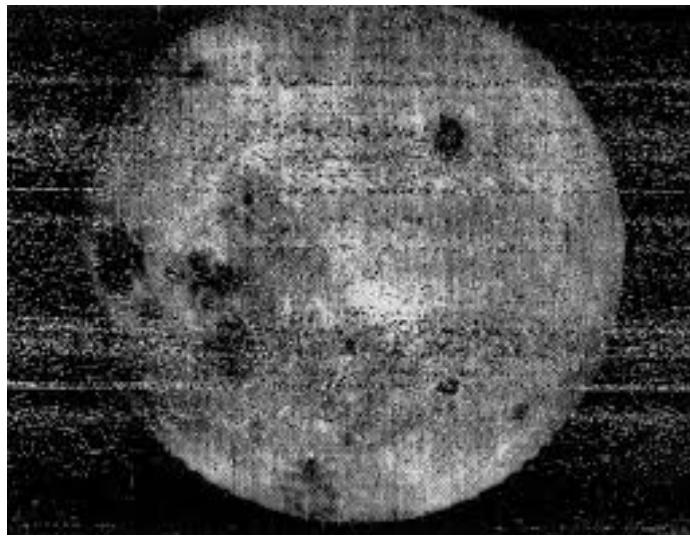


Figure 1.1: *Luna 3* moon far side picture, 1959.

Anyway, most of the information about moon hidden face's features comes from later missions, such as:

- **Lunar Prospector (NASA, 1998):** designed to study the Moon's geology, geo-physics, and environment, providing valuable data about the Moon's composition,



magnetic field, and surface features;

- **Lunar Reconnaissance Orbiter (LRO) (NASA, 2009):** extensively mapped the entire lunar surface with high-resolution images, scouting potential landing sites for future crewed missions and understanding the Moon's topography while identifying potential resources;
- **Chang'e 4 (China, 2019):** the first mission to successfully land on the "dark" hemisphere, which intended to study the geology, composition, and environment of the *Von Kármán crater* in the *South Pole-Aitken Basin* region, as well as testing technologies for future lunar exploration;
- **Queqiao (China, 2018):** a relay satellite stationed at the Earth-Moon Lagrange Point 2 to facilitate communication between Earth and the hidden side of the Moon, which goal was to support *Chang'e 4* and other future missions.

1.1. Lunar Far-Side Features

The moon has a small dense core with a mass of 1% to 4% of the entire satellite's and the crust is composed by high density materials called *mascons* that are usually associated with impact basins, which are mostly detected in the near side as for example *Mare Imbrium* or *Mare Sere*. [1]

As we can see (Fig. 1.2 - 1.3), the far side comprises mainly rocky terrain, with limited flat landing areas with the floors of *Crater Tsiolkovsky* and the *Mare Moskve* being the only obvious flat areas. [1] [2]

Moreover, 16 % of the moon surface is composed of lunar maria, made mainly of iron, 92.9% of which are in the nearside, while 37.9 % of basins in the far side are concentrated in the *South-Pole-Aitken*. [3]



Figure 1.2: Near side photographed by the Galileo spacecraft on its way to Jupiter (left), and far side photographed by the Apollo 16 spacecraft (right). [4]



1. INTRODUCTION

That shows that the hidden lunar hemisphere has a thicker crust and present many more small craters and irregularities on the surface, probably caused by meteoroids and micrometeoroids impacts. [5]

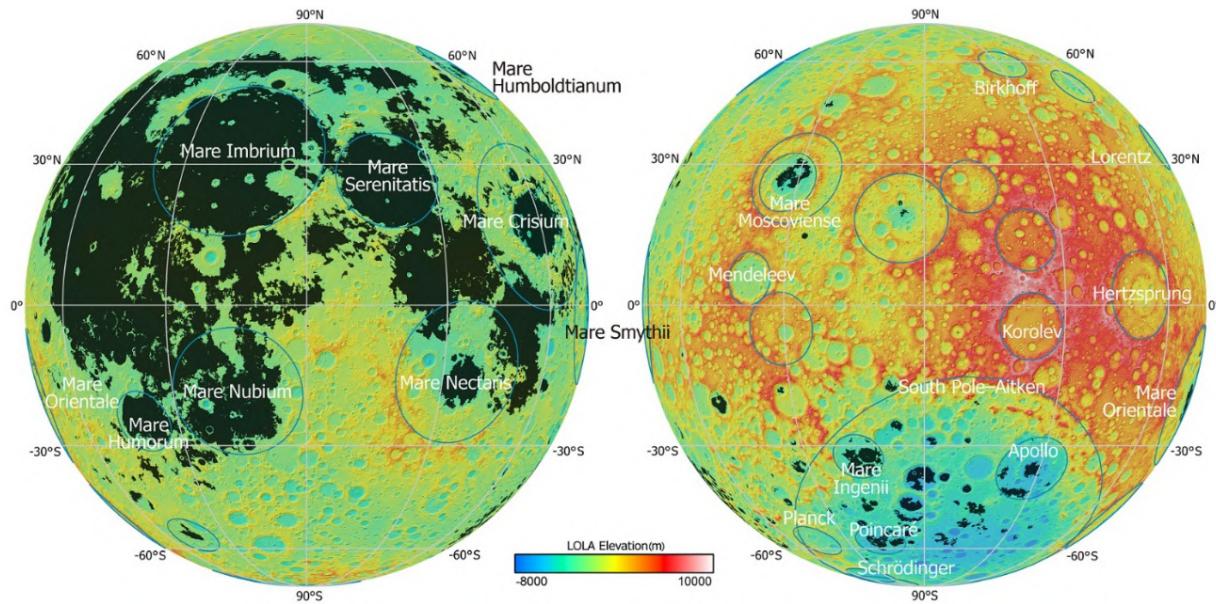


Figure 1.3: Maps of the lunar nearside (left) and farside (right) topography, showing the spatial distribution of mare basalts (dark grey areas). [6]

Moon does not have a global magnetic field but it shows localized fields, particularly strong fields in the regions antipodal (on the exact opposite side of the Moon) to the large *Mare Imbrium* and *Mare Serentatis* basins (both in the earth's side), which are so strong they can deflect solar wind particles and form their own small magnetospheric system. [1] [2] [7]

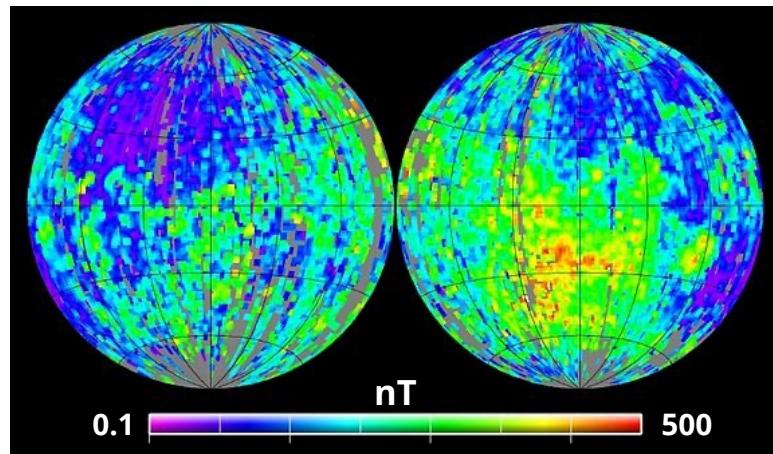


Figure 1.4: Total magnetic field strength at the surface of the Moon as derived from the **Lunar Prospector** electron reflectometer experiment. [1]



The NASA missions *Lunar Reconnaissance Orbiter* (2009) and *Lunar Atmosphere and Dust Environment Explorer* (2013), have shown that there is a thin atmosphere with a number density of atmospheric molecules no more than 10^6 cm^3 , so a sort of "exosphere" with no ionosphere. [2] [8]

The surface is almost fully covered in lunar regolith layer (Fig. 1.5) which interior structures knowledge is essential for understanding surface modification caused by processes such as impact cratering, as shown for example by the *Lunar Penetrating Radar* installed onboard of *Chang'E-5* lander (Cina, 2020). [9]

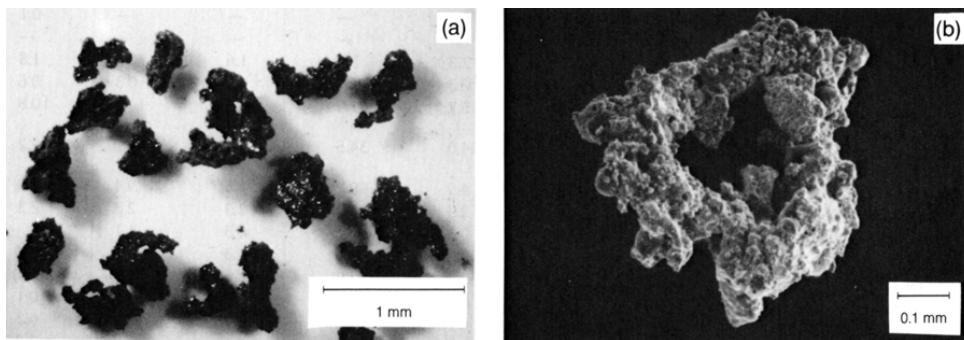


Figure 1.5: Typical lunar soil agglutinates. **(a)** Optical microscope photograph of a number of agglutinates separated from Apollo 11 soil sample. **(b)** Scanning electron photomicrograph of a doughnut-shaped agglutinate. [9]

For what concern the materials distribution on the moon surface, three main provinces can be :

- **Mare basalts**, rich in Fe and Ti;
- **Highland areas**, poor in Fe and Ti;
- **Floor of the South Pole-Aitken basin and rims of the circular maria on the nearside**, rich in KREEP materials. [2] [1]

At moon's North and South Poles, the Sun is never more than 1.5° above or below the horizon; therefore sunlight can never reach the floors of some deep craters. These are some of the coldest spots in the solar system, and so they trap volatile chemicals.

Thank to a Mini-RF an low energy neutron detector on the *Lunar Reconnaissance Orbiter* mission and the *Moon Mineralogy Mapper* (M3), a NASA's instrument carried on the indian mission *Chandrayaan-1* (2009), presence of water-ice in the permanent shadow areas of the lunar poles was proven and water was detected in the interior structure of some lunar materials. [8] [10] [11]

1.2. Earth-Moon Lagrange Point 2

EML2 is the Lagrange 2 Point of the Earth-Moon system located about 64515 kilometers beyond the Moon, a position in space where the gravitational pull of the principal body (Earth) and the secondary (Moon), precisely equals the centripetal force required for a small object to move with them, (Fig. 1.7).

Even if this point is unstable it is an excellent location for satellites, as orbit corrections and fuel requirements needed to maintain the desired configuration, are kept at a minimum.

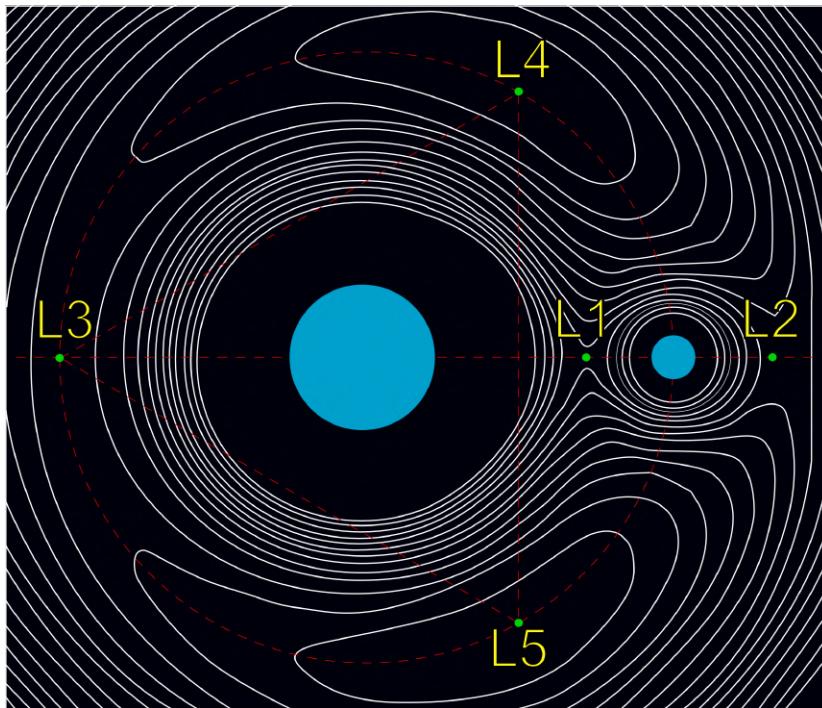


Figure 1.6: NASA illustration of Lagrange points maps in the 3-body problem. [12]

There are many available libration orbits around this point, but **Halo orbits** are the most used since they allow satellites to always be in sight of both the far side of the moon and almost all earth surface, since it is really far (around 448914 km). [13]

Two recent mission have used satellites orbiting into Halo orbits in EML2 to stay in constant connection with the far side:

- **Queqiao - Chang'e 4 satellite**, as talked in 1;
- **EQUULEUS**: a 6U CubeSat from Jaxa & NASA launched in 2022 to demonstrate astrodynamics techniques for CubeSat missions to reach Earth-Moon libration orbits and detecting high-velocity meteoroid impacts and evaluating their fluxes in the cis-lunar region; [14]

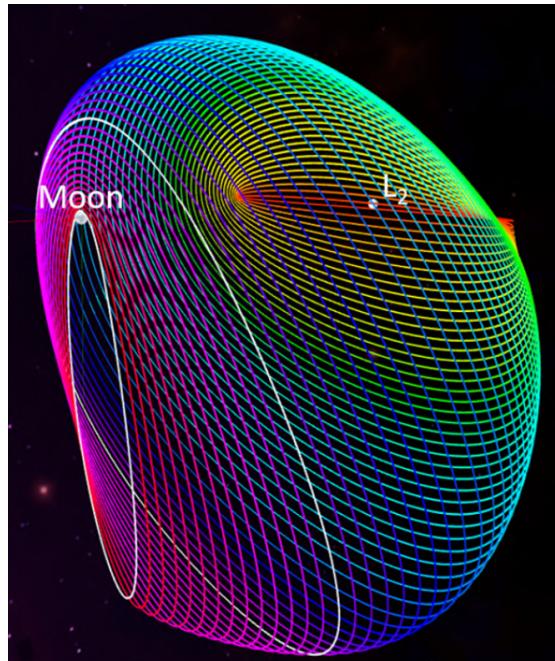


Figure 1.7: Zoomed-in view of the L2 halo family delineating the bounds of the Near-Rectilinear-Halo-Orbits (NRHOs) on white. [13]

and another mission is planned for the very next future:

- **LUMIO**: an ESA CubeSat set for 2026, that will observe, quantify, and characterize the meteoroid impacts by detecting their flashes thanks to the optimal illumination of the far side; [15] [16]



2. Mission Objective

The project refers to a commercial mission developed starting from the stakeholder prerequisite stated as: ***Support for human activities on the moon from EML2.*** So the first challenge was to find which are the possible lunar human activities the mission could support and how, based on the recent space operations focused on the far side of the moon and the ones planned in the very next future, considering that the spacecraft should be ready to launch in 5 years.

2.1. Lunar Human Activities

Initially four main classes of human activities were identified based on their goals, always concerning tasks on the "hidden" face lunar surface.

Surface Exploration

Recent missions and planned ventures are driven by the tantalizing prospect of discovering and analyzing water and ice deposits nestled within the Moon's polar craters, as specified in Section 1.1.

These reservoirs, shielded from the Sun's scorching rays, could hold valuable resources for sustaining future lunar settlements or fueling further space exploration endeavors.

Moreover, the study of the lunar regolith promises insights into its composition, formation processes, and possibly even clues about the early history of our solar system.

Ultimately, the needing of new landing spots on the far side, not discovered yet, will move many agencies and companies to exploring the surface; a challenging task since the presence of many more small craters and irregularities on the surface.

In the following list, some past, recent(**light blue**) and future (**yellow**) lunar missions focused in the hidden face of the moon exploration:

- ***Chang'e 4*** (CNSA - 2018)

Conducting in-depth geological and scientific investigations utilizing a combination of a lander, rover, and a communication relay satellite to analyze lunar soil composition, radiation levels, and surface temperatures, Fig. 2.1. [17]

- ***SLIM*** (JAXA - 2023)

The "Smart Lander for Investigating Moon" was designed for the deployment robotic lander to conduct detailed seismic measurements for better understanding Moon's interior structure and tectonic activity, investigate the distribution and characteristics of lunar regolith. [18]



- **Chang'e 6** (CNSA - 2024)

Collection of the first far side lunar samples (*South Pole–Aitken basin*) and investigation of potential lunar resources, such as helium-3 and water ice, while testing advanced landing and sample return technologies. The mission consist in four modules: a lander, an ascender, an orbiter and a return capsule. [19]

- **VIPER** (NASA - 2024)

Landing of "Volatile Investigating Polar Exploration Rover" to characterize and map the distribution of water ice and other volatile resources, and to conduct detailed geological and chemical analyses of the lunar regolith. [20]

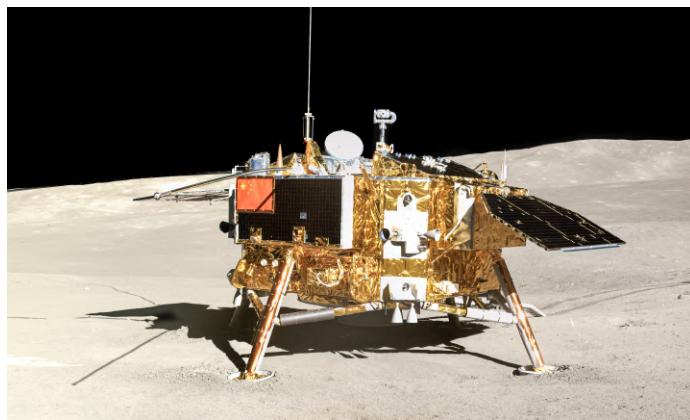


Figure 2.1: The Chang'e-4 lander imaged by the Yutu-2 rover on the lunar far side.
Credit: CLEP/CNSA.

Resource Mining

Following the work and discoveries of many explorations mission, space agencies are planning on expanding the resource mining field with several project all concentrated in the far side of the moon.

Those missions aim to prospect and extract valuable materials such as water ice, rare metals, and helium-3, all vastly present in that region (Section 1.1), thus holding a significant promise for sustaining long-term human presence beyond Earth.

Some of them are summarized in the following list:

- **Chandrayaan 3** (ISRO - 2023)

Mapping of lunar water resources and conducting detailed surface analysis to support future manned missions. Investigating the presence of Helium-3, a potential fuel source for future fusion reactors, further enhancing its significance in the realm of space exploration, Fig. 2.2. [21]

- **Chandrayaan 4** (ISRO - 2026/2028)

Sample return mission continuing the work of the ISRO's Chandrayaan program conducting high-resolution mapping of the lunar surface and analyzing mineral composition to study lunar resources.

- **Lunar Outpost** (NASA)

Anticipating a future base-camp, resource extraction will be necessary since water could be used at the outpost, or more importantly decomposed into oxygen and hydrogen paving the way for refueling reusable lunar landers and eventually delivering fuel to the Gateway for refueling spacecraft. [22]



Figure 2.2: Chandrayaan-3 Integrated Module in clean-room. Credit: ISRO.

Moon Astronomy

Observational astronomy is one of the main scientific fields that will benefit from renewed operations on the lunar surface, particularly in the far side.

One of the principal benefits would be for low-frequency radio astronomy from the radio-shielded far side: radio waves with wavelengths longer than about 20m cannot

penetrate the Earth's ionosphere, and so must be observed in space. [?]

The possibility for passive cooling of IR instruments in permanently shadowed ultra-cold lunar craters near the lunar south pole, with crater rims in perpetual sunlight and the provision of a solid substrate on which to mount optical/IR/farIR instruments, will enable the usage of telescopes up to 100 m diameter.

Moreover, this open up the possibility of installing architecture such as low-frequency radio interferometers located on the lunar surface. [23]

The absence of atmosphere and the presence of many deep craters on the far-side, point many possible site where to install such instruments and infrastructures.

In the following lists future concept missions are presented, many of which already have a started development:

- **Lunar Crater Radio Telescopes** (NASA)

LCRT, a ultra-long wavelength telescope design proposal to deploy a 1km-diameter wire-mesh using robots in a 3-5km-diameter crater on the far-side to form a spherical-cap reflector. It will be the largest filled-aperture radio telescope in the Solar System, Fig. 2.3. [24] [25]

- **Laser Interferometer On Moon** (NASA)

A laser interferometer gravitational wave detector on the moon, would be sensitive to frequencies from sub Hz to kHz. This will help to understand how supermassive black holes got their colossal masses on the cosmological landscape, and it can observe in detail intermediate-mass ratio inspirals at distances as large as at least 100 Gpc.[26]

- **LuSEE-Night** (NASA - 2025/2026)

The Lunar Surface Electromagnetics Explorer is a low frequency radio astronomy experiment that consists of a 4 channel, 50 MHz Nyquist baseband receiver system and 2 orthogonal 6m tip-to-tip electric dipole antenna. It will make unique measurements of the global radio sky below 50 MHz a first step in a measurement strategy to access the high redshift 21 cm signature associated with the so-called 'Dark Ages'. [27] [28]

Moon Bases

The absence of large lunar maria on the far side reduces the risk of volcanic activity and seismic disturbances, providing a more stable foundation for infrastructure development. [6]

2. MISSION OBJECTIVE

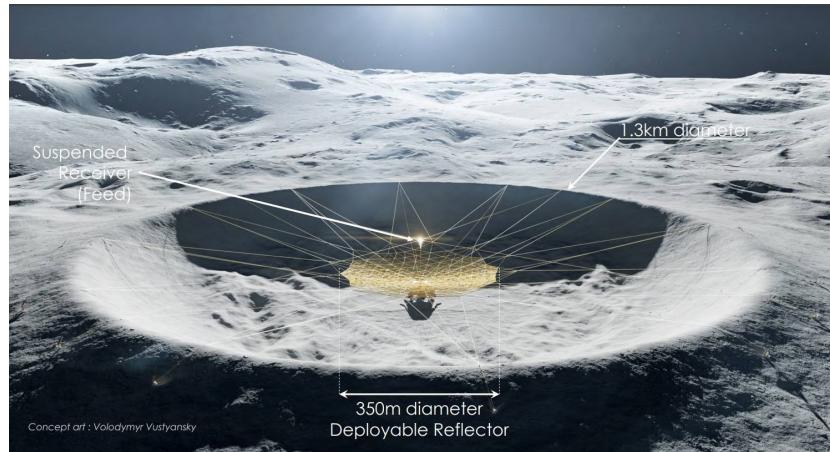


Figure 2.3: Lunar Crater Radio Telescope concept. [25]

This stability enhances the safety and longevity of lunar bases, reducing operational risks associated with geological hazards.

Moreover, many small-scale remanent magnetic fields are scattered over the lunar surface, mainly in the far side, and they are believed to divert the incident solar wind and shield the local lunar surface beneath, thus producing unique local surface environment that is critical to activities of human beings/facilities. [2] [7]

Many agencies have already speculated lunar outpost and moon bases concepts, as the *Artemis Base Camp* [29], and such missions could become more and more concrete in the very next future.

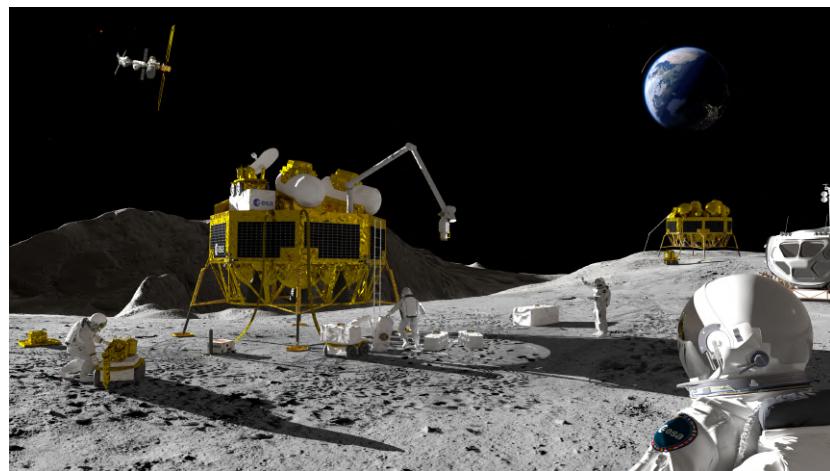


Figure 2.4: ESA Moon base concept. Credit: ESA-ATG

2.2. Support Mission Objectives

The preceding section delineates a comprehensive array of prospective human activities on the lunar far side, predominantly focused on the forthcoming years, thereby describing the future trend of moon missions.

This portrayal shows the increasing effort aimed at leveraging the unique opportunities presented by the "dark" side.

To ensure optimal operational performance, safety, and effective communication with Earth-based stations, all aforementioned activities necessitate an assist mission.

Surface Exploration mission necessitate of a constant communication link both during the landing phases and deployment of rovers since the presence of many small craters and irregularities in the far side surface.

Moreover, they could benefit of a tracking system during the wide exploration phases and a relay link with the earth eliminating data storage limitations.

Resource Mining operations have similar requirements together with the possible necessity of a orbiting docking system for the collection of extracted lunar samples.

With regard to Moon Astronomy activities, high data link transfers are mandatory given the enormous amount of information to be transmitted, beside with the long operative time of such applications.

In conclusion, Moon Bases' requirements sum up all the other activities' ones along with large coverage areas for the relay link to reach, therefore raising the complexity and precision of the support mission.

Since all of those moon operations are concentrated in the hidden lunar hemisphere, the *Earth-Moon Lagrange Point 2* is the optimal location for an assist space operation: a spacecraft in orbit around EML2 is far enough from its gravitational bodies to be able to simultaneously see both earth and moon while always facing the far side.

Based on those information, the support mission objectives can be extrapolated in order to give assistance to the greatest extent possible of human activities on the moon:

- Earth-Moon relay system;
- High data rate communications;
- Constant and simultaneous link for all the assisted moon activities;



2. MISSION OBJECTIVE

- Precise position tracking supply system during surface navigation and landing phases;
- Reliable and long time operative support.

The aforementioned information and results can be concentrated into the following mission statement:

"HERMES mission aim to support all kinds of lunar human activities as a relay system positioned in EML2.

*Our spacecrafts will be able to perform **constant high data rate transfer** with the ground stations, ensuring a safe and reliable link.*

Moreover, the mission will offer an innovative tracking system capable of tracing the precise position of lunar users."

2.3. Mission Requirements

In order to extrapolate the set of requirements for the aforementioned support operations, a first study about the human activity requisite in terms of:

- Data rate;
- Operative and communication intervals;
- Mission lifetime;
- Number of satellites needed;
- Surface coverage, it directly impact on the selection of the orbit;

was conducted, gathering information from similar past operations' report.

In particular, the *Chang'e 4* support satellite *Quequiao* served as a reference because of the similarities with the mission in question: both for the position in EML2 and tasks to be carried out. [30]

The Table 2.1 presents the required data rate and link intervals for *Chang'e 4* (2018), *Chandrayaan 3* (2023) and *Chang'e 7* (2026) missions, while moon astronomy applications information are shown in Table 2.2, taking as refernece the *LION* interferometer and *SKA* telescope that has similar requirements in terms of data rate.



		Chang'e 4			Chandrayaan 3	Chang'e 7 ¹
		S/C - Earth	S/C - Rover	S/C - Lander	Lander - Earth	S/C - Earth/Rover
Data Rate	DL UL	1÷10 Mbps -	125 kbps 0.7÷285 kbps	125 kbps 1.4÷555 kbps	1 Mbps 4 kbps	1 Gbps
Link	Constant			Intermittent	-	
Lifetime	5 years			<1 year	8 years	
Orbiter	1			1 (Backup)	1	

Table 2.1: Surface exploration missions communication requirements. [30] [21] [31]

	LION ^{1 2}	SKA ^{1 2 3}
Data Rate	640 Mbps÷10 Gbps	≈ 9.5 Gbps
Communications interval	Intermittent	≈ Constant
Lifetime	>5 years	>5 years

Table 2.2: Moon astronomy applications communication requirements. [26] [32]

Ultimately, from those data and information the full set of requirements has been derived taking in consideration the necessity to satisfy all type of lunar human activities requisites.

The results are shown in Table 2.3.

As will be discussed in Section 3, those requirements directly affect not only the selection of the type of orbit, but also the type of communication links to establish and the number of satellites to use.

¹Estimated data, not publicly available.

²Just preliminary concept designs.

³Application on earth.



2. MISSION OBJECTIVE

Mission Requirements		Rover/Lander Relay	Position Tracking System	Moon bases and Astronomy Activities
Achievable Data Rate	Earth Data	600 Mbps	-	1 Gbps
	Moon Data	800 Mbps	-	5 Gbps
	TT&C	150 kbps	-	150 kbps
Communication Intervals		≈Constant	Constant	Sporadic ¹
Mission Lifetime		≈5 years	≈5 years	>7 years
Number of Orbiting Satellites		1	>3	>1 ²
Surface Coverage		20° ÷ -90°	±45°	±20°

Table 2.3: Principal Mission Requirements.

Based on those assumption, the mission objectives can be presented as:

- High data transmission: ≤ 5 Gbps;
- Low data transmission: ≤ 1 Gbps;
- Lunar navigation and position tracking;
- Constant and simultaneous links with earth-moon;
- Central-Southern coverage of the far side;
- At least 7 years lifetime + 50% expansion expected.

¹Once every orbit, in the nearest point to the surface.

²One single satellite cannot supply such data rate.



3. Mission Architecture

With the aim of providing high-level communication services to a location where it has never been done before, current and upcoming missions with similar purposes were studied. This analysis enabled us to examine the diverse architectures employed by these missions to accomplish their objectives, thereby facilitating the development of our own.

Starting from the missions requirements, four mission architecture were derived and a comparative analysis between them was conducted. These preliminary studies dictated the selection of the optimal configuration from the four presented, which was subsequently further developed.

The different architecture studied are all shown in the scheme below (Fig. 3.1), they are classified by the number of satellites required and also by the typology of the selected working orbit. The chosen architecture is highlighted with a blue solid line.

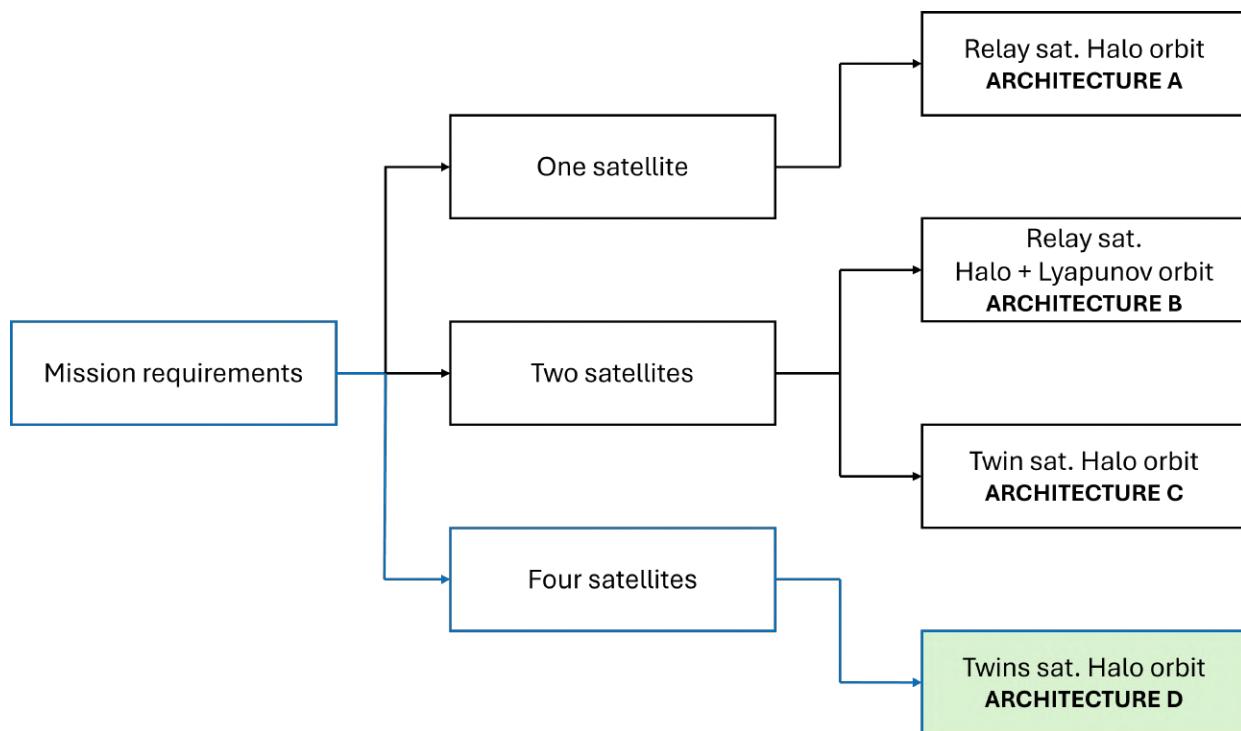


Figure 3.1: Mission architecture selection flow

3.1. Architecture A

This first configuration rely on a single communication satellite placed on a high z-amplitude halo orbit around EML2. This kind of orbit, thanks to its high stability, allow for a low propellant consumption requirement for the station-keeping activities. The significant benefit of the halo orbit lies in its capability to provide constant visibility of both the far side of the Moon and Earth. This feature is given by the particular geometry of this kind of orbits.

The advantages of a single satellite configurations includes a high flight heritage¹ and the cost-effectiveness of the mission, since only one satellite is launched, making it the most economical among all other configurations.

However, there are evident drawbacks, there is a increased risk of losing direct communication with the Moon due to reliance on a single satellite and also the spacecraft will require very high power and antennas gains to meet the high data rate requirements.

Subsystem	Mass Budget	Power Budget
Telecomm	34%	11%
Propulsion	6%	67%
Power Gen	17%	5%
Structure	34%	4%
ADCS	5%	6%
Thermal	4%	7%
Total	215-301Kg	620-848W

Table 3.1: Architecture A preliminary Mass and Power budget

¹This configuration shares the same characteristics as the Queqiao mission

3.2. Architecture B

The concept behind this configuration involves splitting the higher data rate service and the lower data rate service into two satellites: one will be placed on a Halo orbit and the other on a Lyapunov orbit.

The first satellite it's similar to the previous one, but somewhat smaller, given the opportunity to split the telecommunication load over two different spacecraft.

For the second satellite, a Lyapunov orbit was chosen to achieve a lower altitude above the lunar surface, thereby reducing the power needed for the lunar customer to send data to the satellite. This configuration will employ a store-and-forward kind of communication strategy and so, will necessitates a large storage memory, as well as a larger amount of batteries due to more frequent and longer eclipses from the Sun.

In this configuration, both satellites are essential to achieve maximum performance. In the event of the loss of one, there would be a severe reduction in communication capabilities from the far side of the moon.

Subsystem	Halo Sat		Lyapunov Sat	
	Mass Budget	Power Budget	Mass Budget	Power Budget
Telecomm	17%	7%	16%	22%
Propulsion	7%	68%	8%	53%
Power Gen	16%	5%	22%	3%
Structure	45%	5%	40%	3%
ADCS	8%	8%	7%	11%
Thermal	7%	7%	7%	8%
Total	205-245Kg	570-680W	190-230Kg	510-670W

Table 3.2: Architecture B preliminary Mass and Power budget

3.3. Architecture C

This configuration comprises two identical satellites positioned on a halo orbit. They are situated at opposite ends of the orbit, facilitating coverage of almost the entire far side simultaneously. This setup enhances reliability due to inherent redundancy and enables communication with a larger number of landers and rovers at the same time compared to a single-satellite version.

The twin spacecraft cannot directly communicate with each other. Also, in the event of receiving data from the lunar surface, the data can be split between them. This arrangement effectively doubles the data rate incoming to Earth.

Both satellites provide very high data rates and are lightweight compared to other architectures, thanks to the use of K-band antennas. However, this also comes with its drawbacks: the power required to sustain the telecommunication is very high compared to the other architectures.

Subsystem	Mass Budget	Power Budget
Telecomm	15%	25%
Propulsion	9%	61%
Power Gen	22%	4%
Structure	41%	4%
ADCS	6%	3%
Thermal	7%	3%
Total	215-265Kg	902-1030W

Table 3.3: Architecture C preliminary Mass and Power budget for a single satellite



3.4. Architecture D

This last architecture was conceived to provide the navigation service to our customer on the Moon, later referred as **LNSS (Lunar Navigation Space System)**.

A constellation design with *4 satellites* was analyzed, this is the minimum number of satellites required to provide triangulation and elevation of a receiver on the moon surface or low altitude.

Furthermore, the choice of using an *halo orbit* was made because the positioning accuracy not only depends on the instruments on board the satellites but also on the geometrical and spatial distribution between the satellites and the receiver. This effect on the position error is called **geometric dilution of precision (GDOP)**. So, in order to maximise the ground coverage, maximise the accuracy of the positioning system and avoiding obstruction of the moon, we have excluded other working orbits like Lyapunov and Lissajous that had very good ground coverage but they also had very poor GDOP. The Halo orbit chosen has a higher z-amplitude with respect of the other configuration and also has less heritage.

At last a choice of using *Southern* halo orbit was made in order to service the south pole with a better navigational signal. - The south pole region has more interest to be explored and exploited by human activities as discussed previously.

Subsystem	Mass Budget	Power Budget
Telecomm	32%	11%
Propulsion	8%	67%
Power Gen	15%	5%
Structure	36%	5%
ADCS	4%	6%
Thermal	5%	6%
Total	200-290Kg	600-812W

Table 3.4: Architecture D preliminary Mass and Power budget for a single satellite

3. MISSION ARCHITECTURE

Here a graphical comparison of the total launch mass is shown. The minimum value of the total dry mass has a dark blue color, while the difference with the maximum value has a light blue color. Finally only the maximum value of the propellant mass for orbit insertion and station keeping were considered.

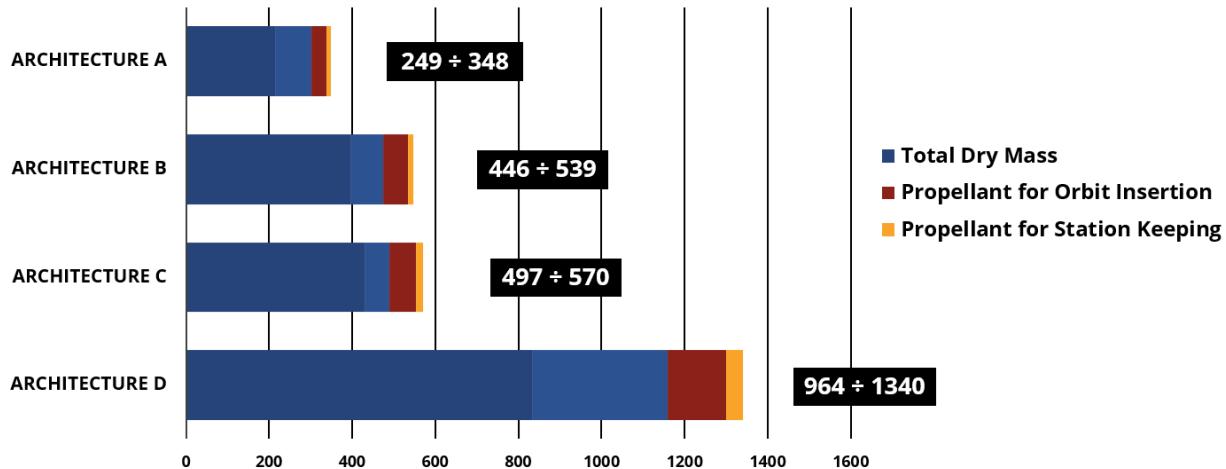


Figure 3.2: Architectures total launch mass.



3.5. Decisional Matrix

After analyzing all the architectures details and conducting a trade-off between their masses and costs, a decision matrix was drawn.

Within this framework, the different architectures are being compared, based on the parameters of performance, cost, and risk, with values ranging from one to five.

The first architecture is the cheaper but has less capabilities, while the difference in risk value between architecture B and C was attributed to the fact that architecture B has less heritage.

A value of 5 for the performance for architecture D was chosen because it offers better coverage of the lunar surface and has the potential for increasing data rates (e.g., by increasing antenna sizes and powers on the surface). Additionally, it can provide Navigation service.

After adding all the values, our choice fell on Architecture D.

More details of the mission profile of this architecture on the next chapter.

Architectures	A	B	C	D
Performance	1	3	3	5
Cost	5	3	3	2
Risk	2	3	4	5
Total	8	9	10	12

Table 3.5: Decisional Matrix



4. Mission Profile

In this chapter, the various factors that contributed to the selection of the spacecraft trajectory for this mission will be outlined. Additionally, the rationale that led to the choice of the working orbit will be shown.

4.1. Working orbit

The requirements used to select the operational orbit in which the satellite will be positioned are defined as follows:

- Constant communication of each satellite over the far side
- Continuous communication with the South Pole
- Optimal GPS coverage of the Moon far side and south pole

Orbit selection to meet LNSS requirements

In principle, the halo orbits, Lissajous orbits, quasi-halo orbits and the combinations of the horizontal and vertical Lyapunov orbits can be used as orbits of the LNSS constellations.

As discussed in the previous chapter, a decision was made to utilize Halo orbits due to their capability to guarantee continuous communication with earth without moon occlusions.

Specifically, a Southern Halo orbit was selected because this family of orbits will extend further into the southern hemisphere and thus a south pole coverage could be achieved.

- Halo orbits with small z-amplitude¹ ($< 10000\text{km}$) are not suitable to be used as the orbit of the LNSS: these orbits always bring relative high percentage of ground coverage, but the satellites are almost in the same plane (X-O-Y plane), which will cause an alignment between two satellites and a receiver and the position can not be calculated. *To have a working LNSS, the minimum number of satellite required increases.*
- With bigger halo orbits ($> 45000\text{km}$), the situation is opposite: the quality of the signal is good because the satellite are never in the same plane, but the ground coverage is poor because they are very distant from each other. *To have a good ground coverage, the minimum number of satellite required increases.*

¹An Halo orbit can be identified among an halo orbit family in different ways. Amplitude-z is a parameter common used that is equal to the maximum value of the z coordinate of the halo orbit, where the xy-plane is the orbital plane of the moon around earth.



A trade-off must be made between them and for those reason a southern halo orbit with an amplitude of 25000 km was chosen. This amplitude is optimal for this mission because it can give good positioning performance with only 4 satellites but can be also improved by adding more satellite in a southern or northern orbit with future launches.

Parameters	Values
z-Amplitude	24'986.12 Km
Jacobi Constant	3.134848
Period	14.987 days
Stability index	444.32
Max Distance to Earth	463'830 Km
Max Distance to Moon	75'870 Km

Table 4.1: Selected Halo Orbit parameters

4.2. Transfer trajectory

The requirements behind the transfer trajectory selection are that the trajectory needs to intersect the Halo orbit with minimal fuel consumption and the maneuver must be executed within a suitable time frame.

Problem Assumptions

- All computer-based calculations are based on a simplified model where the three-body problem (Spacecraft - Moon - Earth) are reduced to two separate Keplerian problems¹.
- All analyzed trajectories lie on the Moon's orbital plane.
- The departure circular parking orbit altitude set to 300 km
- The trans-lunar injection maneuver is performed by the launcher, while subsequent maneuvers depend on the spacecraft's propulsion.
- To enter the halo orbit trajectory, the spacecraft needs to match its velocity by at the intersection point.

From the intersection of the Halo orbit with the Earth-Moon orbital plane, two intersection points (P1; P2) are obtained Fig. 4.1 These are the points that we're aiming to reach.

A mathematical model of the problem was constructed through which the complete transfer geometry can be derived, given the following initial conditions:

- The Earth's parking orbit altitude r_0
- The trans-lunar injection velocity from Earth v_0
- The departure flight path angle from Earth ϕ_0
- The entry angle into the Lunar soi λ_0

By leveraging the previous mathematical model, three distinct Earth-Moon transfer approaches were analyzed. For each approach, the precise trajectories covered by the spacecraft and the corresponding individual ΔV values needed to accomplish the mission were determined (section C.1)

The objective of this analysis was to identify the transfer method that minimizes the spacecraft propellant consumption to the greatest extent possible. To do so, an algorithm was developed to identify the optimal trajectories (section C.1)

¹by means of the patched conic approximation

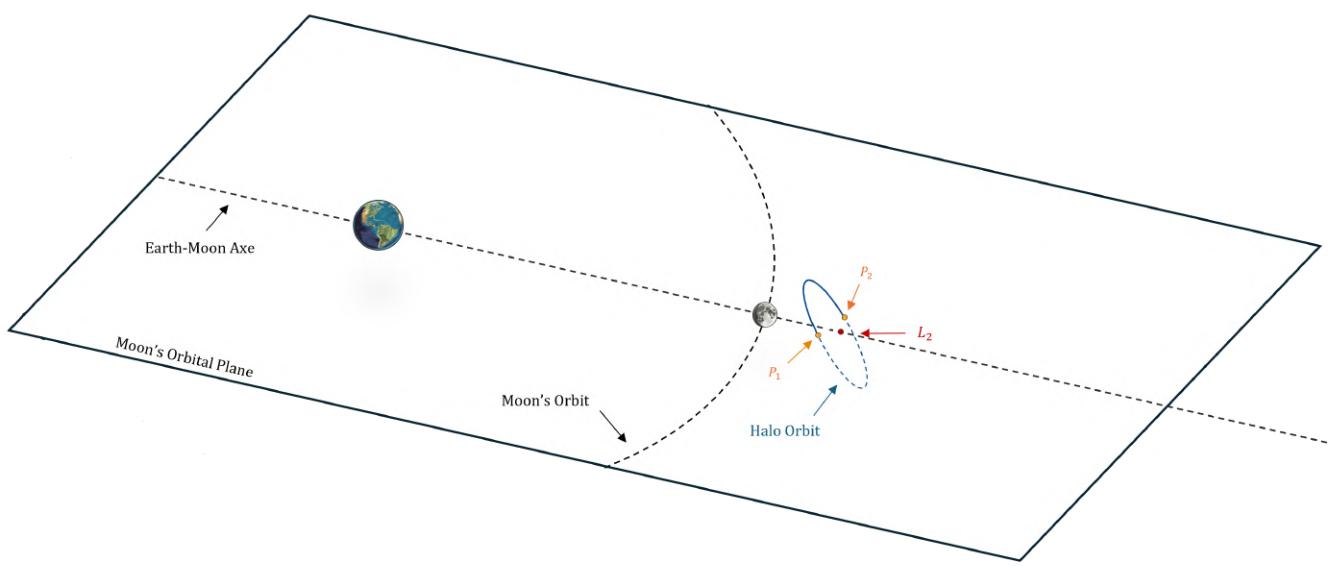


Figure 4.1: Working trajectory illustration

Direct Approach

The trajectory analyzed in this first method is defined as follows: from the 300km parking orbit around Earth the spacecraft is placed into the transfer orbit via trans-lunar injection manouver performed by the transfer stage, after which we have the spacecraft release.

After the separation, the satellite follows a trajectory aimed at intersecting the halo interception points. Upon reaching these points, it will utilize its own propulsion system to execute the insertion maneuver.

The approach required identifying the trans-lunar injection trajectory, which generates a selenocentric hyperbolic trajectory intersecting one of the two designated points.

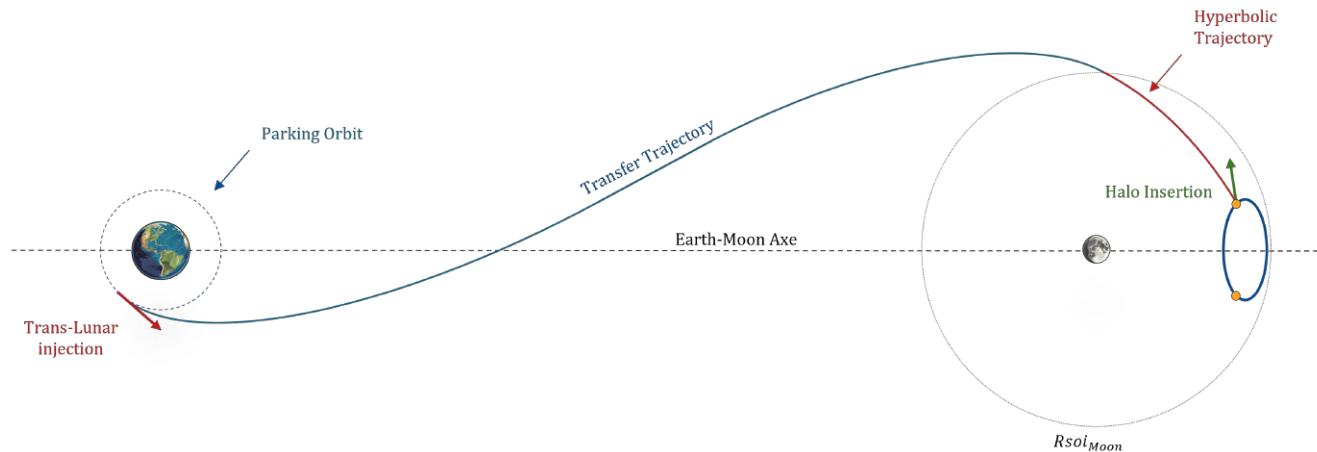


Figure 4.2: "Direct" Approach trajectory illustration. This trajectory draws inspiration from the "free return trajectory" utilized by the Apollo missions to reach the Moon.

Given the stringent requirements concerning the spacecraft propulsion system, the data from the five orbits that exhibited the lowest propellant consumption, were used to compare the results obtained through this method with the subsequent ones Tab. 4.2¹.

In figures 4.3 - 4.4 - 4.5, the tabulated trajectories are visually represented.

¹The "Spacecraft ΔV " and "Halo orbit insertion ΔV " values are identical. This is attributed to the fact that the propulsion system will only execute that maneuver.



Trajectory	v_0 [km/s]	λ_0 [degree]	S/C Δ_v [km/s]	Halo insertion Δ_v [km/s]	Translunar Injection Δ_v [km/s]
Light blue	10.825	51.23	0.791	0.791	3.100
Red	10.826	46.63	0.802	0.802	3.101
Orange	10.827	44.10	0.814	0.814	3.102
Green	10.828	42.25	0.827	0.827	3.102
Purple	10.829	40.75	0.839	0.839	3.103

Table 4.2: "Direct" Approach Δ_v trajectories values

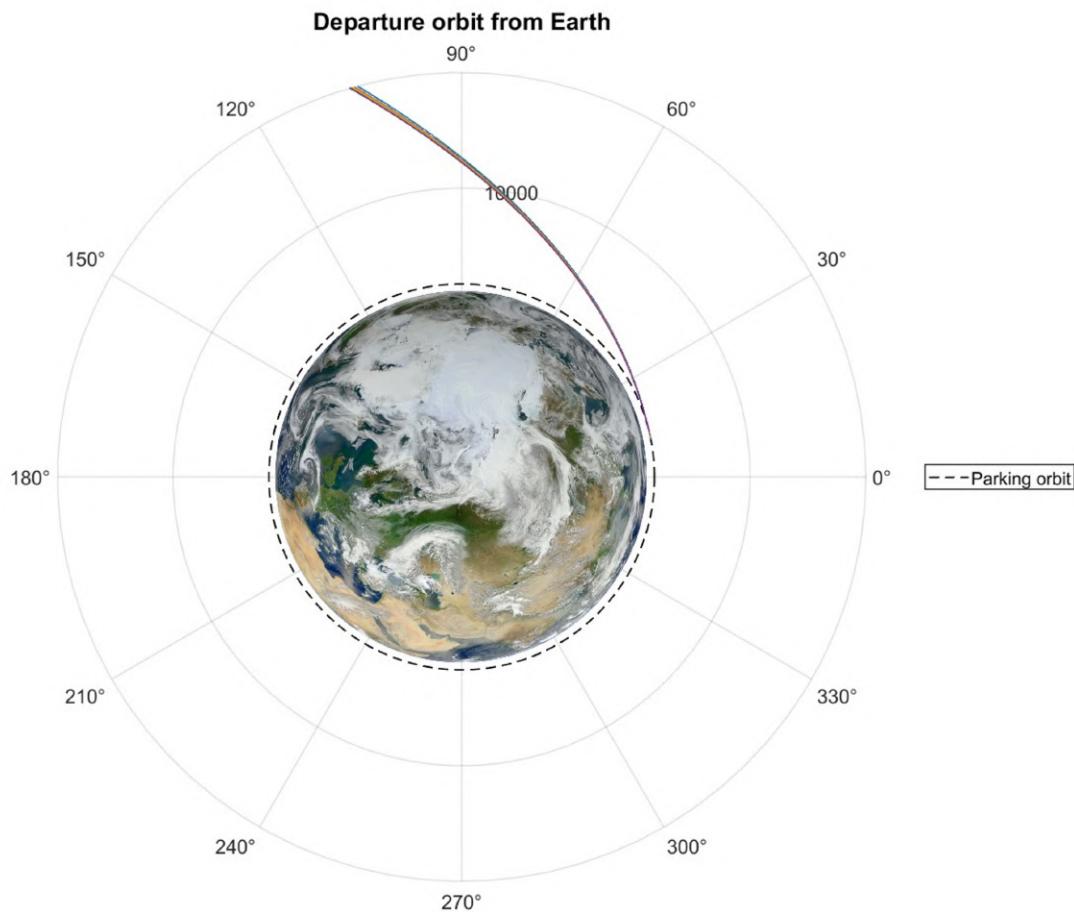


Figure 4.3: "Direct" Departure Orbits from Earth

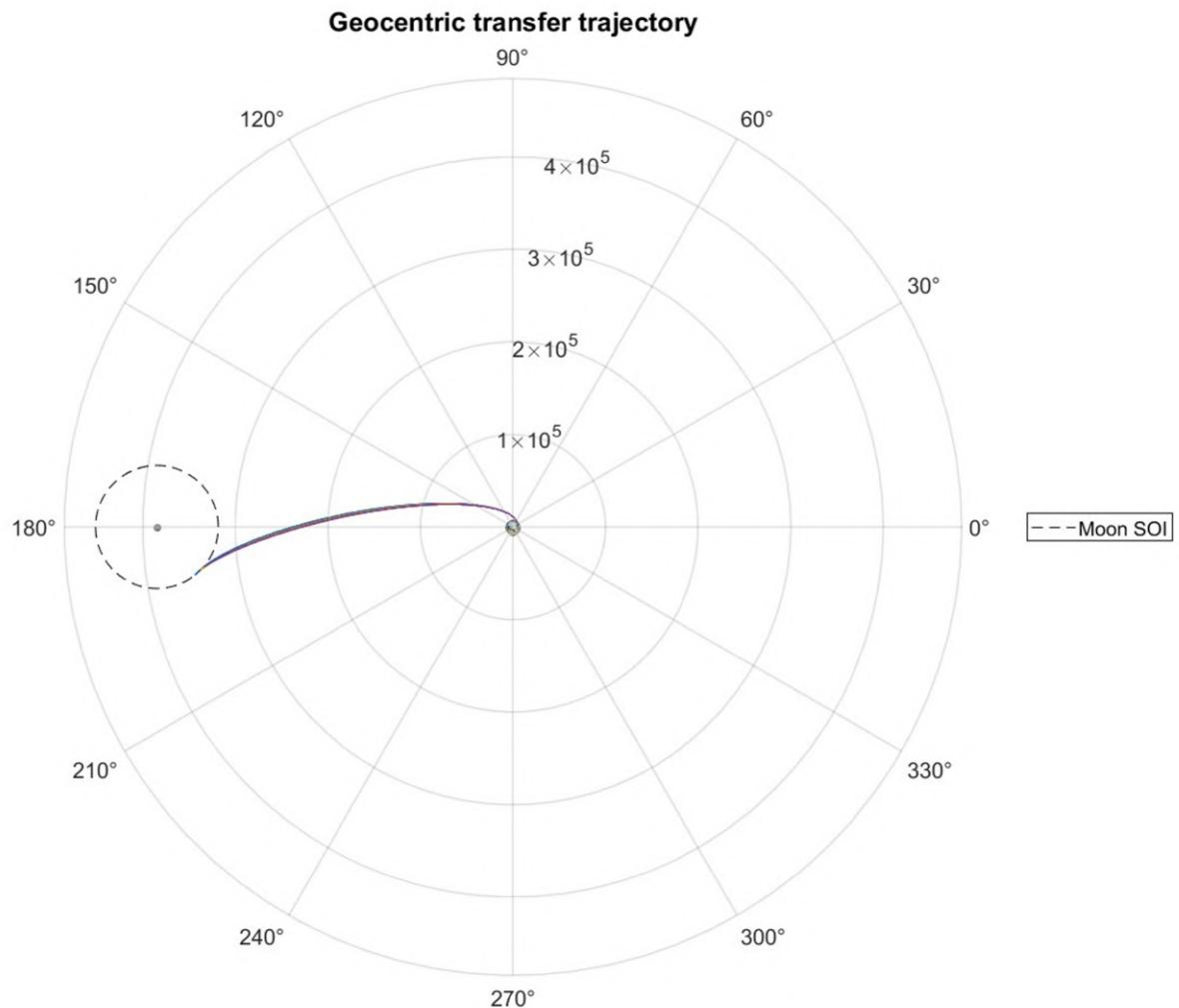


Figure 4.4: "Direct" Geocentric Transfer Trajectories

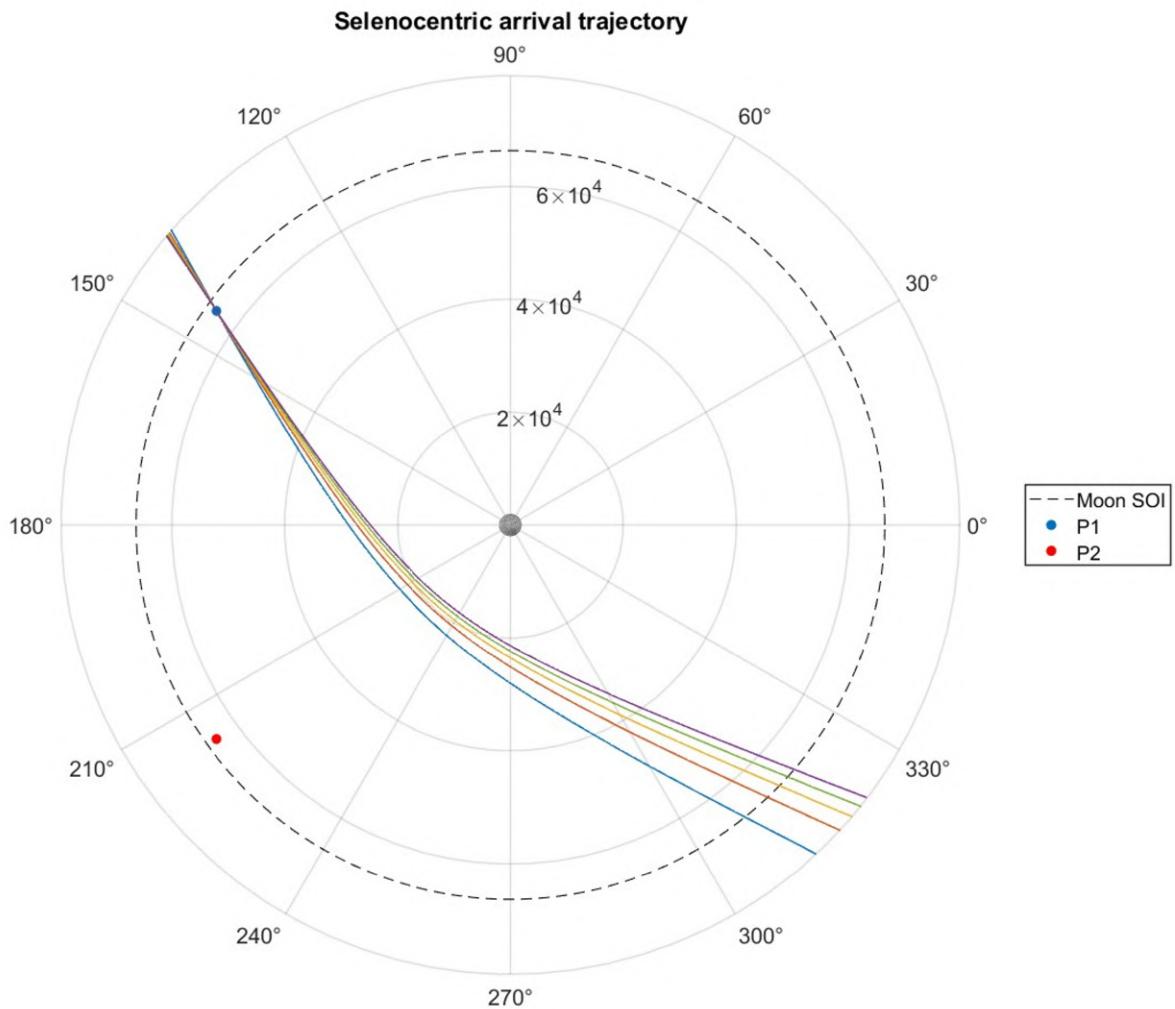


Figure 4.5: "Direct" Arrival Trajectories

Selenocentric Bi-Impulse" Approach

The trajectory presented in this second method shares the same transfer phase as the previous one. However, unlike before, it involves a more complex selenocentric phase. The concept behind this second approach involves an attempt to reduce the Δv required by the spacecraft's propulsion system by assuming it executes multiple, more efficient maneuvers rather than a single one.

The lunar insertion occurs at the periapsis of the selenocentric hyperbola, maximizing the efficiency of the maneuver. The resulting elliptical trajectory has an apoapsis equal to the distance between points P1 and P2 from the Moon. Subsequently, the circularization maneuver takes place at the apoapsis of the selenocentric ellipse, transitioning the spacecraft into a circular orbit.

This circular orbit intersects both points P1 and P2 (given their equal distance from the moon). Thus, the spacecraft executes the insertion maneuver into the halo orbit at the most advantageous intersection point, minimizing propellant consumption.

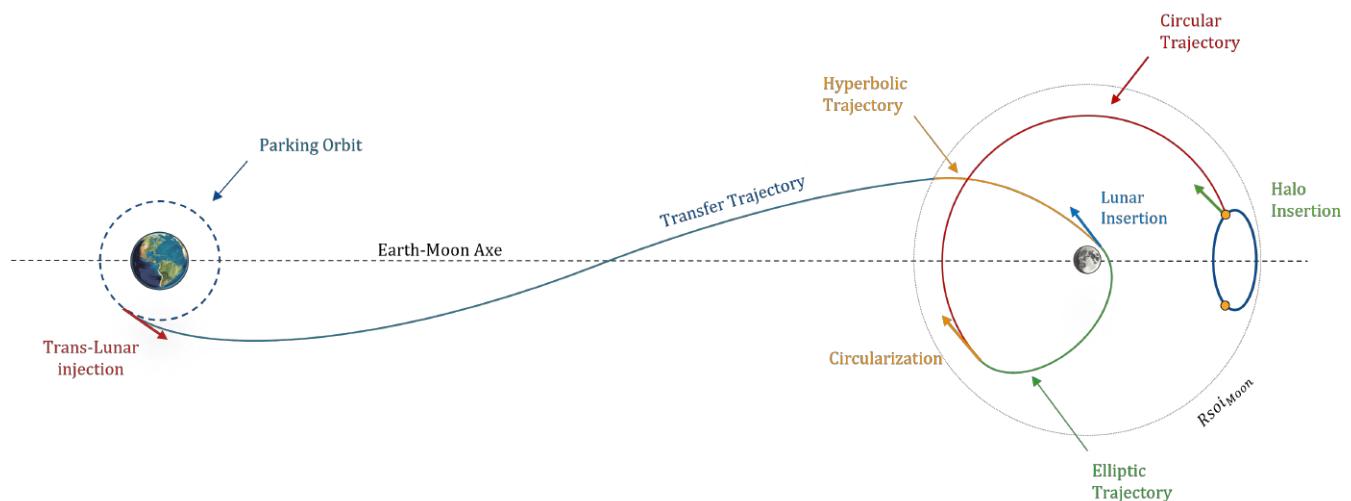


Figure 4.6: "Selenocentric Bi-Impulse" Approach trajectory illustration. This trajectory is a Keplerian model inspired by the trajectory employed by the Chinese Queqiao mission directed towards a halo orbit around L2.

This method holds the same initial assumptions as the previous one, only the selenocentric part has been modified.

Unlike the preceding method, this approach yields a broader spectrum of solutions. This is attributed to the fact that the selenocentric trajectory doesn't need to intersect points P1 and P2 directly; rather, any transfer orbit capable of reaching the moon's



sphere of influence will result in insertion into the halo orbit.

As expected, the closer the selenocentric hyperbolic orbit approaches the Moon, the lower the ΔV required by the spacecraft's propulsion system become, while the ΔV required by the launcher increases.

In this table, the ΔV results obtained from the five orbits that exhibited the lowest propellant consumption are shown Tab. 4.3, where the "Spacecraft ΔV " values is the sum of the two maneuvers done by the spacecraft propulsion system.

In figures Fig. 4.7 - 4.8 - 4.9, the tabulated trajectories are visually represented.

Trajectory	v_0 [km/s]	λ_0 [degree]	S/C ΔV [km/s]	Halo insertion ΔV [km/s]	Translunar Injection ΔV [km/s]
Light blue	11.096	13.1	0.663	0.231	3.370
Red	11.092	13.2	0.665	0.230	3.336
Orange	11.098	13.0	0.666	0.232	3.372
Green	11.094	13.1	0.667	0.231	3.369
Purple	11.087	13.3	0.669	0.230	3.361

Table 4.3: "Selenocetric Bi-Impulse" Approach ΔV trajectories values

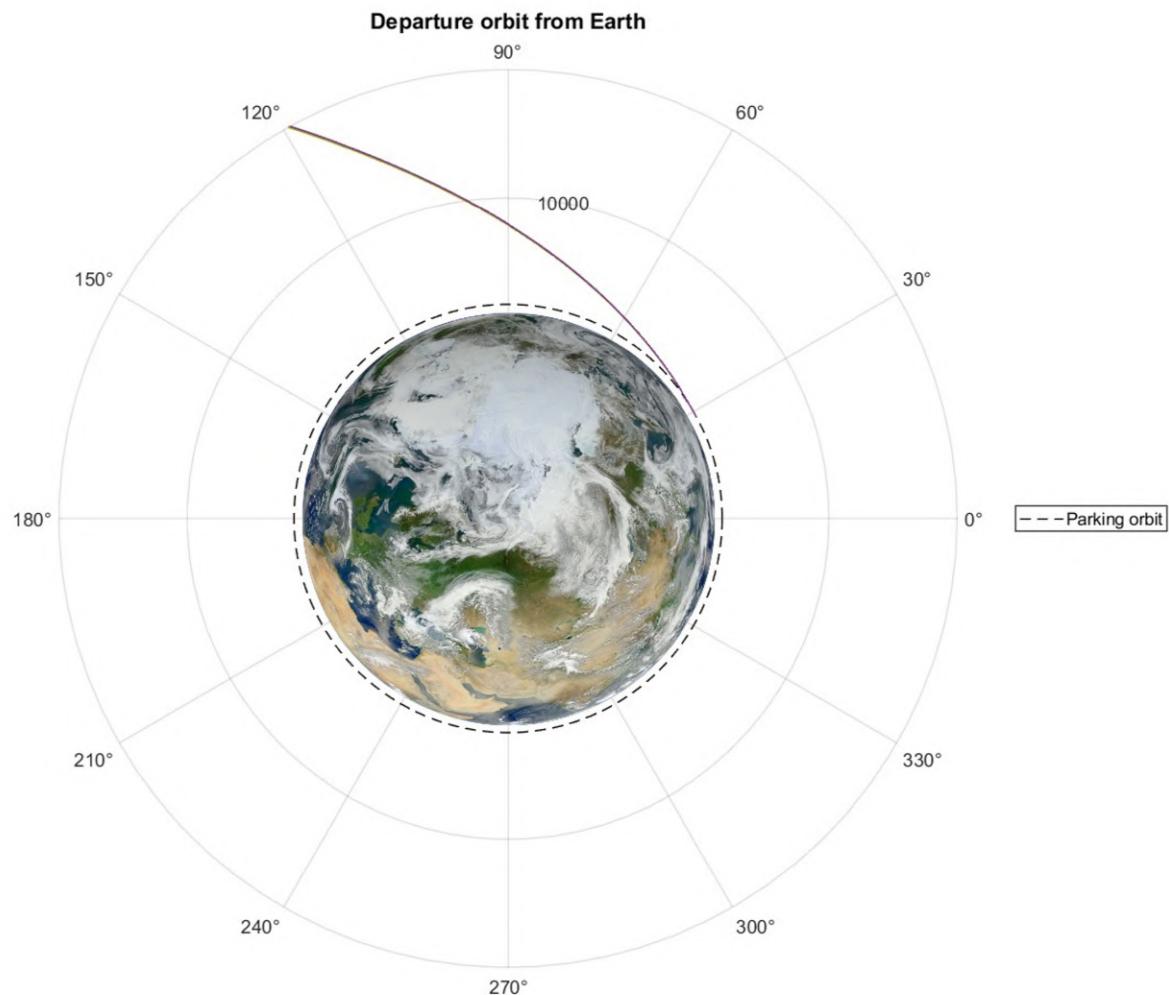


Figure 4.7: "Selenocentric Bi-Impulse" Departure Orbits from Earth

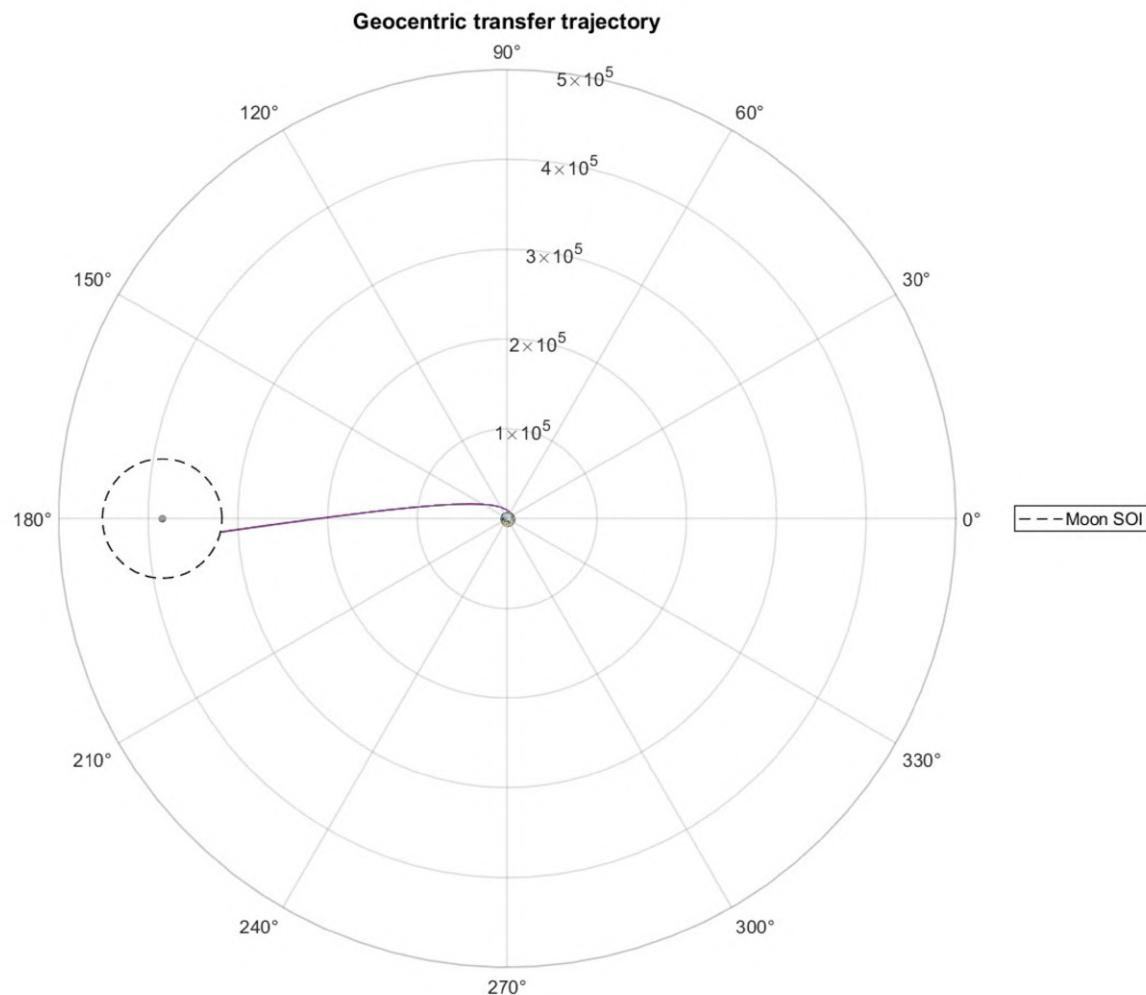


Figure 4.8: "Selenocentric Bi-Impulse" Geocentric Transfer Trajectories

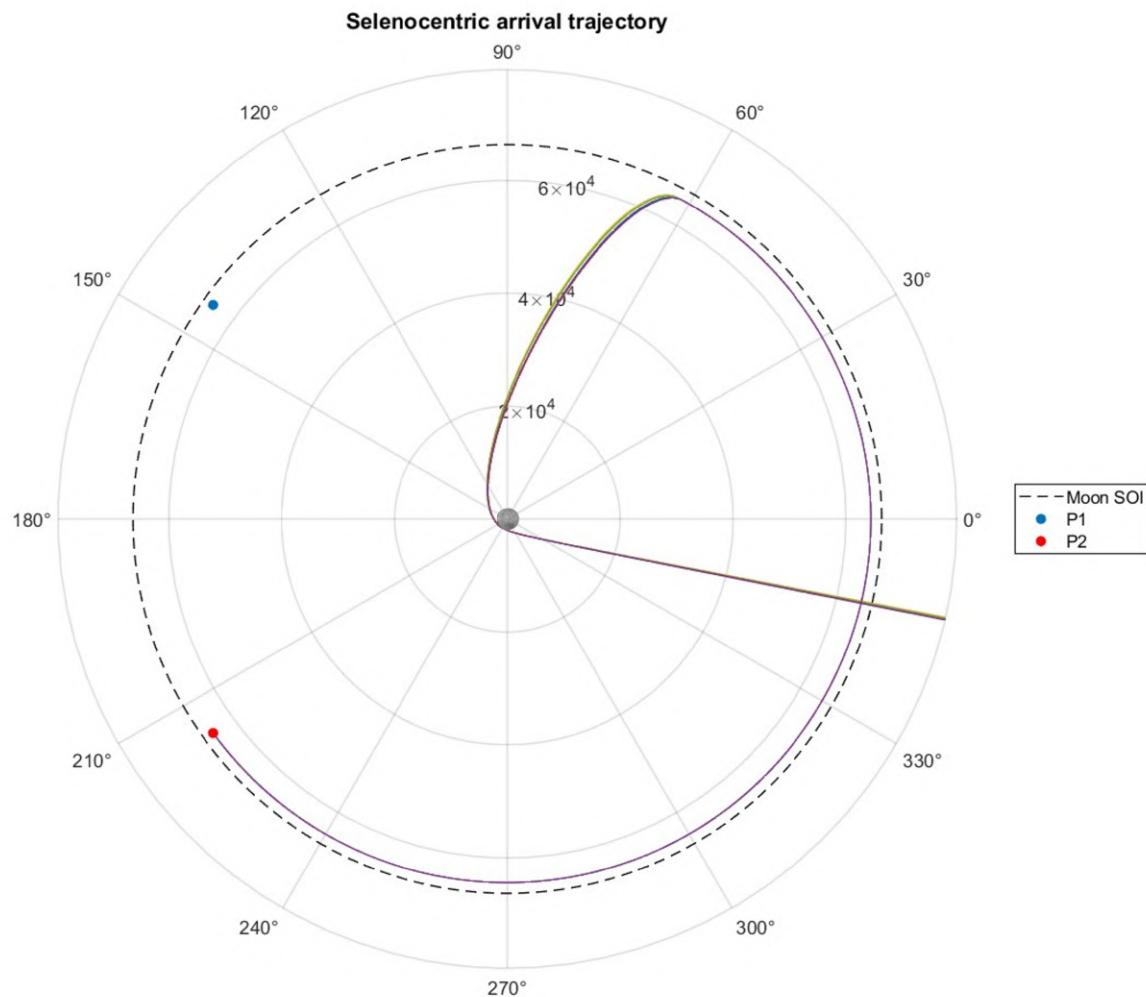


Figure 4.9: "Selenocentric Bi-Impulse" Arrival Trajectories



"Geocentric Bi-Elliptic" Approach

The concept behind this latest method is to attempt to reduce the ΔV required for halo orbit insertion. As observed in the previous methods, the spacecraft's entry into the lunar sphere of influence occurs solely "in front of the moon". This is due to the spacecraft's velocity decreasing along the transfer orbit as it moves away from Earth.

As the spacecraft approaches the lunar orbit, its tangential velocity becomes lower than the one of the Moon. In a sense, the moon "moves away" from the approaching spacecraft. Consequently, it is necessary for the geocentric transfer trajectory to be more energetic than the lunar one in order to allow the spacecraft to enter the lunar soi "from behind".

To minimize the ΔV required for insertion into the halo orbit, it is crucial for the spacecraft's velocity vector to be as closely aligned as possible with the one of the halo orbit at points P1 or P2. As observed, the halo orbit "revolves" clockwise (when viewed from L2 toward Earth). Therefore, this alignment can only be achieved if the spacecraft enters the lunar sphere of influence "from behind".

For this reasons, a bi-elliptical transfer was considered to achieve a suitably energetic transfer orbit.

Maintaining the initial assumptions unchanged from the previous two methods, the spacecraft will now follow the subsequent trajectory:

The spacecraft will initially be placed into a first elliptical orbit having an apoapsis equal to the Earth's sphere of influence radius by the launcher.

Upon reaching the apoapsis, the spacecraft will execute its first maneuver which will cause it to track a second geocentric ellipse intersecting the Moon sphere of influence. The tracked selenocentric hyperbola will pass through one of the two intersection points, where once reached, the spacecraft will then perform the insertion maneuver into the halo orbit.

In contrast to previous methods, this approach yields only a few specific solutions rather than a continuous range of viable angle-velocity combinations.

This limitation arises because the second geocentric transfer ellipse must be precisely configured to bring the spacecraft to a particular point on the lunar sphere of influence edge with a specific velocity. This ensures that the spacecraft is directed toward points P1 or P2 as a result of lunar gravitational attraction.

In figures Fig. 4.11 - 4.12 - 4.13, the tabulated trajectories are visually represented.

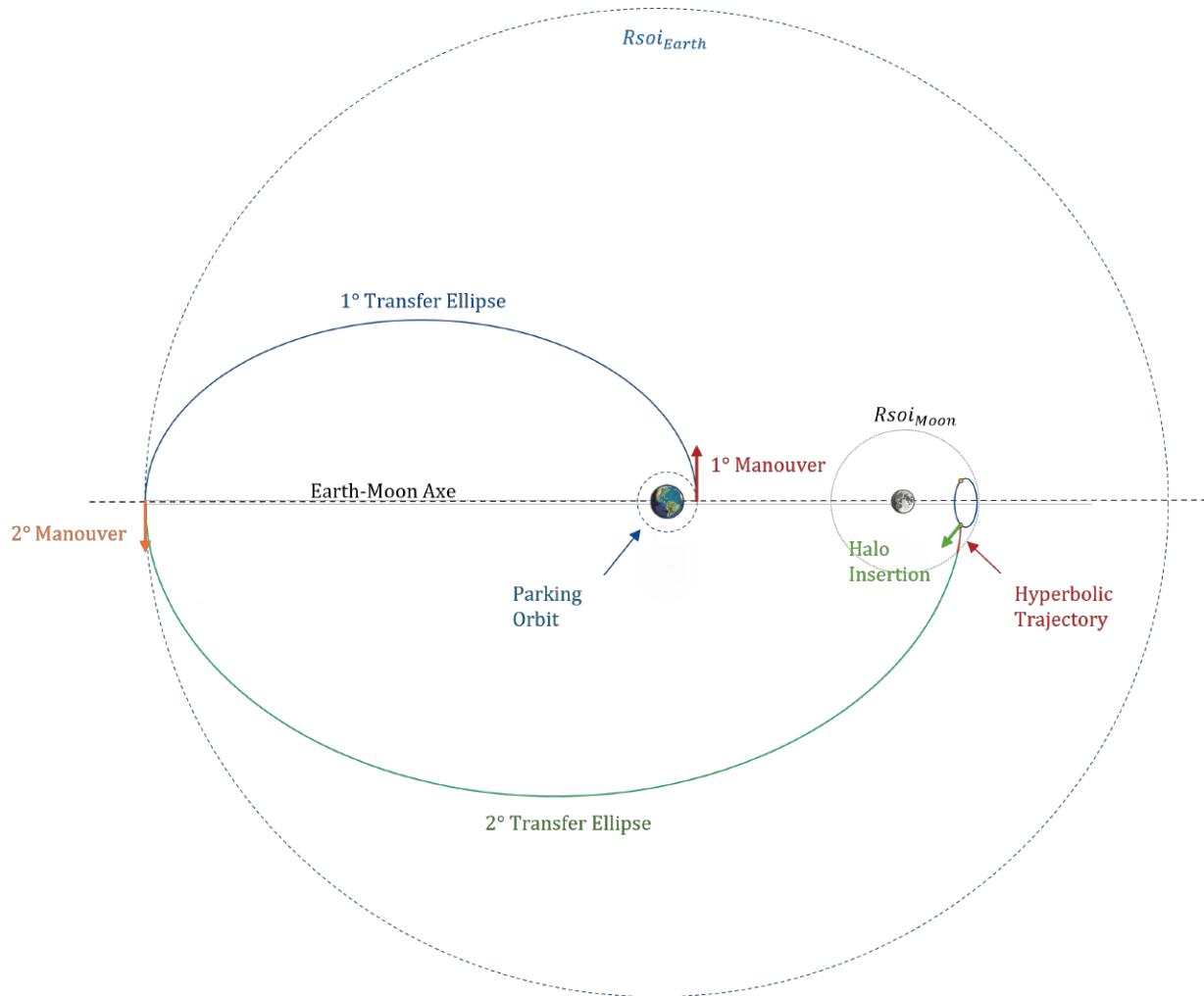


Figure 4.10: "Geocentric Bi-Elliptic" Approach trajectory illustration. This trajectory is a Keplerian model inspired by the trajectory tracked by the Japanese Hiten mission directed towards the Moon.

Trajectory	v_0 [km/s]	λ_0 [degree]	S/C Δv [km/s]	Halo insertion Δv [km/s]	Translunar Injection Δv [km/s]
Light blue	0.522	144.011	0.659	0.215	3.161
Red	0.527	144.010	0.663	0.214	3.161
Orange	0.520	144.011	0.665	0.224	3.161
Green	0.517	144.011	0.667	0.234	3.160
Purple	0.506	144.011	0.723	0.295	3.160

Table 4.4: "Geocentric Bi-Elliptic" Approach Δv trajectories values

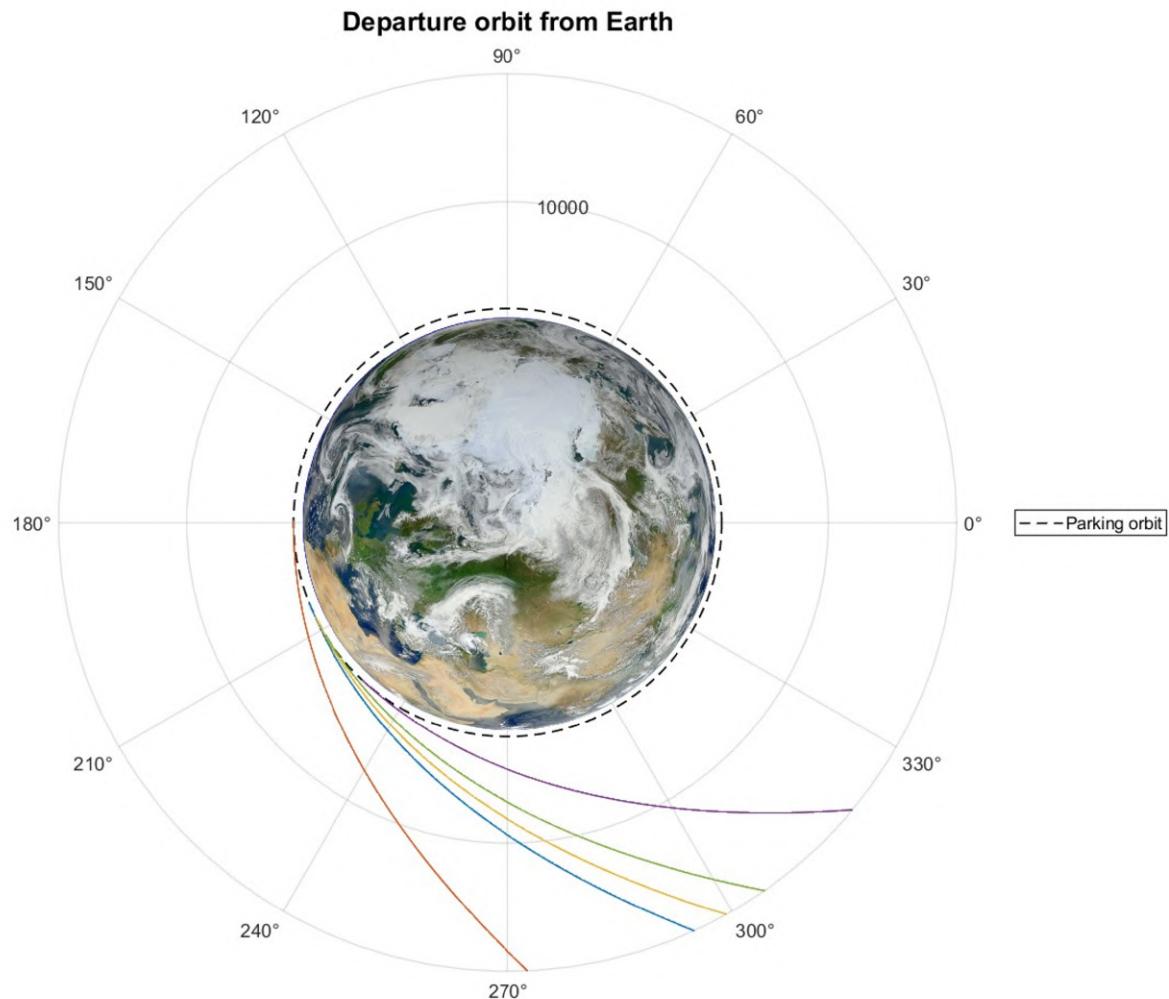


Figure 4.11: "Geocentric Bi-Elliptic" Departure Orbits from Earth

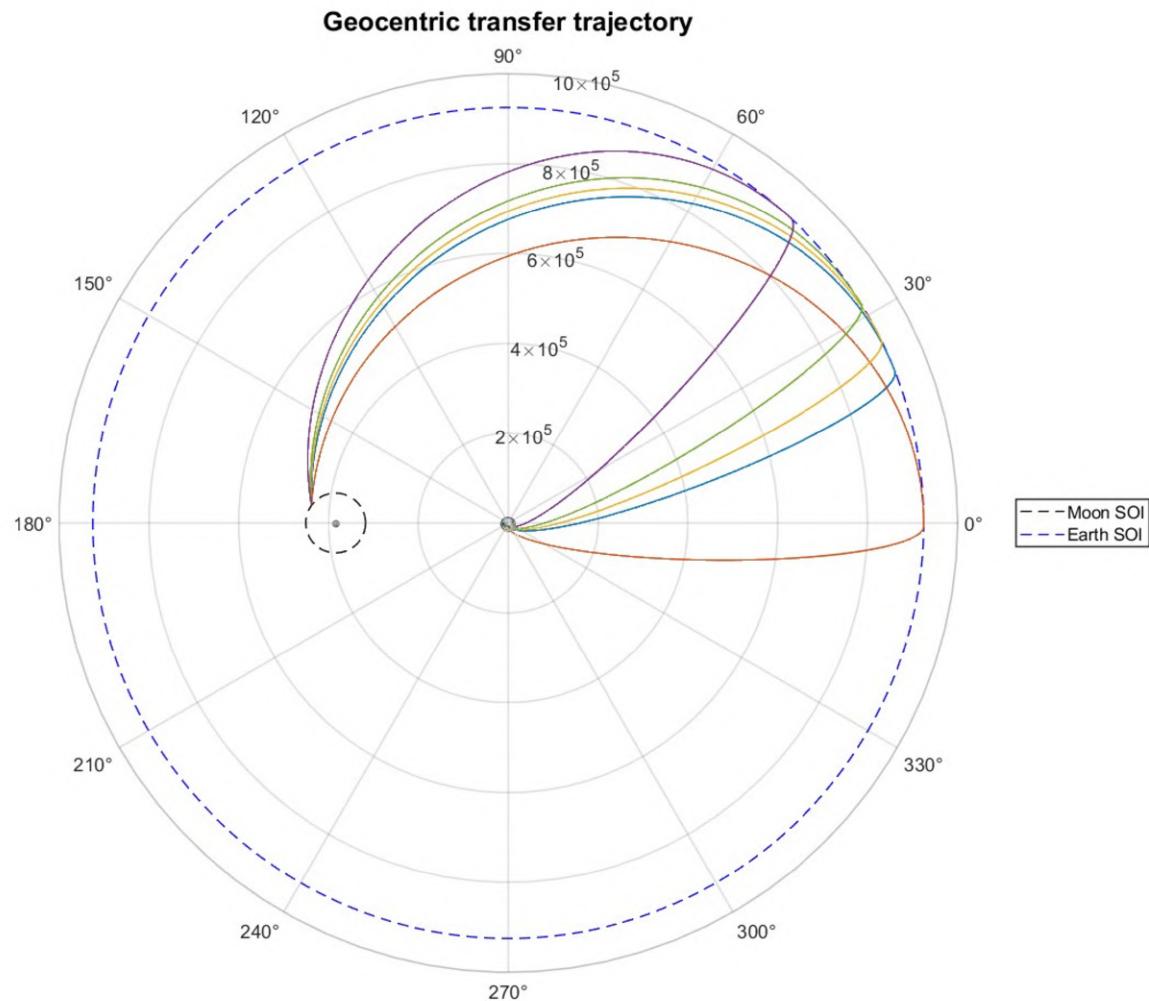


Figure 4.12: "Geocentric Bi-Elliptic" Geocentric Transfer Trajectories

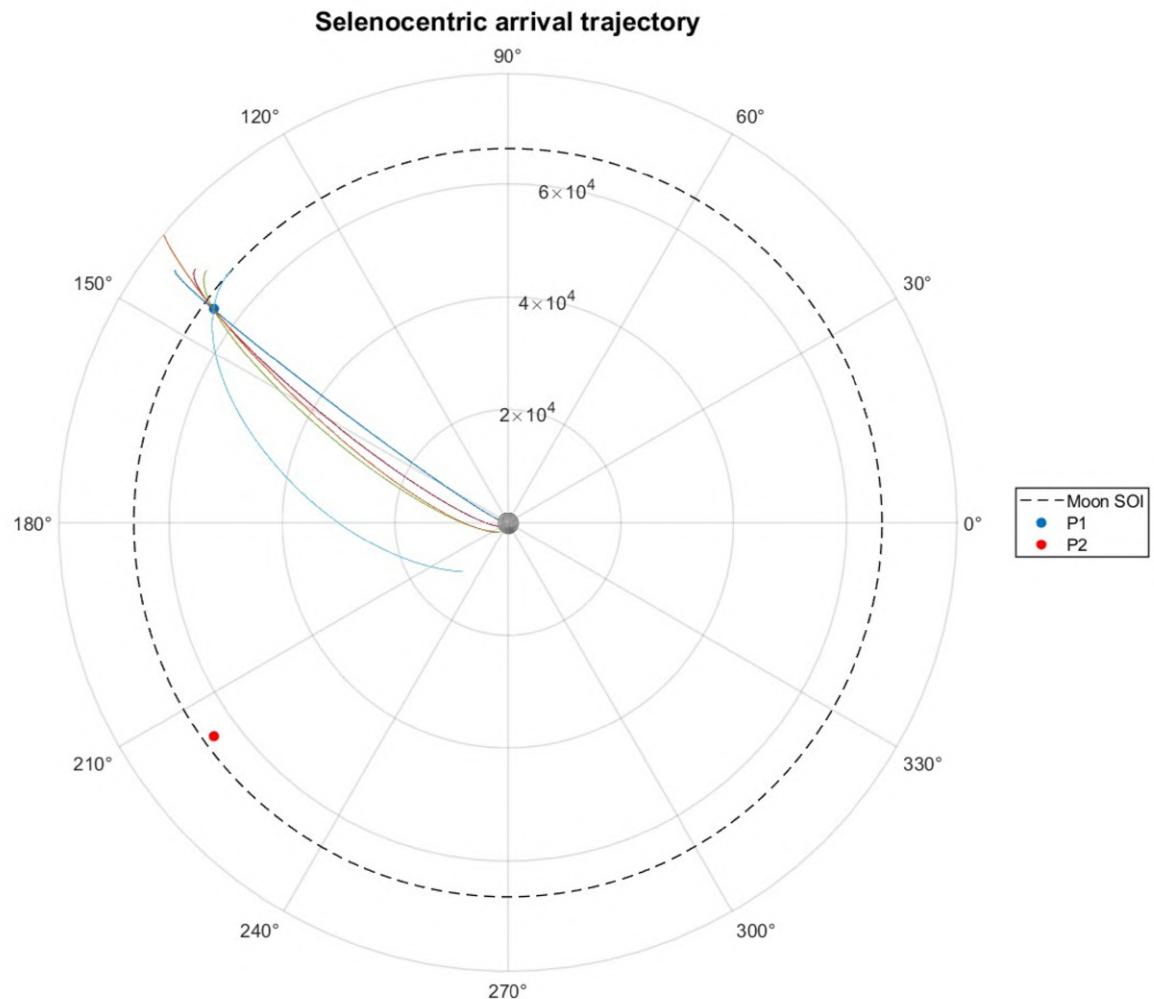


Figure 4.13: "Geocentric Bi-Elliptic" Arrival Trajectories

4.3. Results Obtained from Keplerian approaches

To select the optimal trajectory, the top 5 results obtained from each analyzed approach were compared.

Approach Used	Spacecraft Δ_v [km/s]	Launcher Δ_v [km/s]	Flight Time days
"Direct"	$\sim 0.791 - 0.839$	~ 3.09	$\sim 3 - 4$
"Bi-Impulse"	$\sim 0.663 - 0.669$	~ 3.36	$\sim 3 - 4$
"Bi-Elliptic"	$\sim 0.659 - 0.723$	~ 3.16	$\sim 40 - 45$

Table 4.5: Comparison of the different approach studied

As observed from the table, the direct method emerges as the most propellant-intensive option for the spacecraft, leading us to discard it. Comparing the other two methods, the "Bi-elliptical" approach stands out for minimizing both the onboard propellant and the trans-lunar injection propellant ("Launcher Δ_v ") requirement. However, this comes at the cost of a total flight time exceeding 10 times that of the Bi-Impulsive approach.

Considering that the Bi-Impulsive method also provided "greater solutions flexibility" due to its numerous working angle-velocity combinations, we believe that this trajectory offers the best choice for our case.

Parameters	Values
Lunar SOI entry angle	13.1°
Translunar Injection Δ_v	3.36 km/s
Lunar Orbital Insertion Δ_v	0.161 km/s
Lunar Circular orbit insertion Δ_v	0.271 km/s
Halo insertion Δ_v	0.231 km/s

Table 4.6: "Bi-Impulse" Trajectory parameters

This analysis was based only on the study of simple Keplerian transfer trajectories lying on the lunar orbital plane. An optimization of these is certainly achievable by considering the possibility of varying the inclination of the transfer orbit, thus being able to enter the halo orbit in more suitable points, minimizing the Δ_v required by the spacecraft propulsion system.

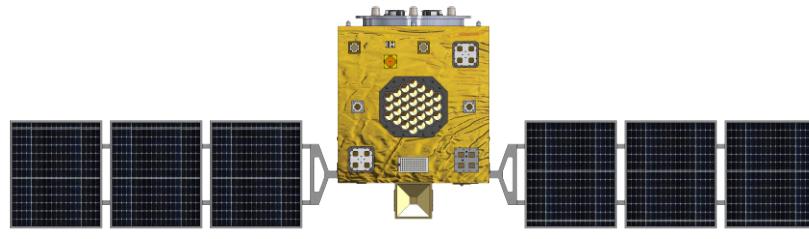


5. Satellites and Subsystems Design

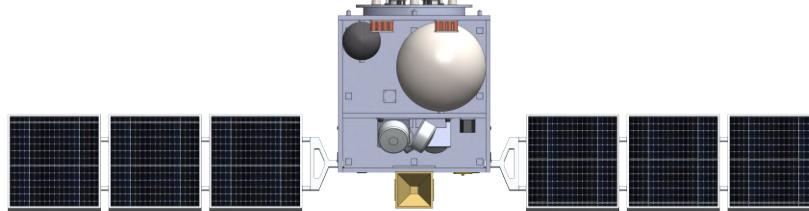
As already said, the mission is composed of four identical satellites which design is shown in Fig. 5.1b.

The front side of the spacecraft, always facing earth and moon, is mainly covered in antennas for the data communications and by the Laser Retroreflector Array used to determine the satellite position, Fig. 5.1a. Other antennas are positioned in the remaining faces of the satellite, mainly for backup/redundancies, as explained in Sec. 5.1.

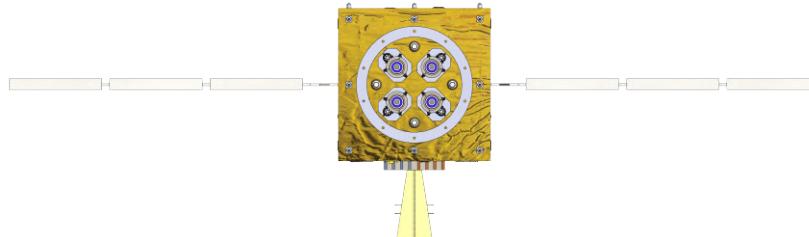
In the bottom face, the electric and chemical thrusters are located inside of the launch adapter plate, Fig. 5.1c, which architecture and selection rationale are better illustrated in Sec. 5.2.



(a) HERMES Frontal View



(b) HERMES inside components view



(c) HERMES Propulsion view

5.1. Telecommunication Systems

The Telecommunication Subsystem is one the major unit of the satellite, since the design concept is intended to serve as a relay system for both smaller architectures as well as larger applications and it has to provide a constant reliable link between Earth stations and the many activities on the lunar surface, Section 2.

Based on the simple but challenging requirements shown in Tab.2.3, the telecommunication system architecture was derived, Sec.5.1.1, and from that the selection of the off-the-shelf components to be mounted on board, Sec.5.1.4, which ultimately led to the Link Budget analysis, see Sec.C.4.

5.1.1. System Architecture

From the requirements' data, considering to subdivide each link within the four satellites in orbit, the system architecture was derived in order to exploit each link with the S/C.

Moreover, DL transfers were subdivided from UL's, both for data transmission and TT&C, in order to assure a constant communication.

Earth-Satellite Link

Just one data UL was considered from the Earth, sending together the information for both smaller surface exploration facilities and greater human infrastructures.

Those may include scientific data, commercial communications and others useful information for moon human activities.

A similar thing was done for the downlink:

- The possibility to constantly transmit medium/lower data rates towards many accessible ground stations on the Earth was supposed: those data are mainly coming from scientific smaller surface exploration facilities;
- Sporadically very high data transmissions is ensured with specific and selected ground stations: those information combine Moon astronomy scientific data, base-camp information and surface exploration data.

For redundancy and safety reasons, a separated link for telemetry and command link with the Earth was chosen.

The frequency ranges for telemetry and commands were also selected in order to not have any interference between the UL and the DL and TT&C, since they all work in S-band.



Moon-Satellite Link

The same approach was adopted for the Moon trunk-link, ensuring a constant communication for smaller lunar users, while a simultaneous higher data rate link is established with greater human infrastructures once every orbit.

First of all, the frequency ranges to be used in each segment has been selected driven by the standards of nowadays space communication[33] considering the velocity constrain on data transmission.

As explained in Sec. C.4, TT&C is operated in S-band with low data rate, alongside with Earth data uplink. Data downlink is instead transmitted in X-band for the Earth link and in Ka-band for the Moon link at greater transmission velocities, while the Moon uplink is carried in the lower X-band. Last, the low data LNSS downlink with the lunar users in L-band.

For what regards the Ground Stations, the antennas to use were selected as explained in Sec. 5.1.2, while in Sec. 5.1.3 the architecture for the position tracking system (**LNSS**) is described.

Next, based on past missions' telecommunications architecture, an average value for the TX/RX Gain and for transmitted power was derived, see Fig. 5.3.

Taken those values, a Matlab code was created, C.4, to extrapolate the S/C minimum values of transmitted power and RX/TX Gain considering to maintain a sufficient value of the Signal to Noise Ratio of at least:

- $\frac{S}{N} = 10 \text{ dB}$ for Earth-Links;
- $\frac{S}{N} = 6 \text{ dB}$ for Moon-Links.

In particular, from the *Shannon–Hartley Theorem*[34], Eq. 5.1, Channel Capacity (C) was taken equal to the wanted achievable data rate, extrapolating the needed Bandwidth (B) having fixed the Signal to Noise Ratio ($\frac{S}{N}$).

Then, entering in the so called *Telecommunication Equation*[34], Eq. 5.2, with B and $\frac{S}{N}$, selecting the Rx (Tx) Gain G_R (G_T) and Tx Power P_T , the Rx Power P_R - negligible - and the Tx Gain G_T (Rx respectively - G_R) was obtained, fixing the antenna receiver temperature T_{sys} in the various scenarios.

$$C = B \log_2 \left(1 + \frac{S}{N} \right) \quad (5.1)$$

$$\frac{S}{N} = \frac{1}{k} P_T G_T \frac{1}{L_S} \frac{1}{L_A} \frac{G_R}{T_{sys}} \frac{1}{B} \quad (5.2)$$

where:

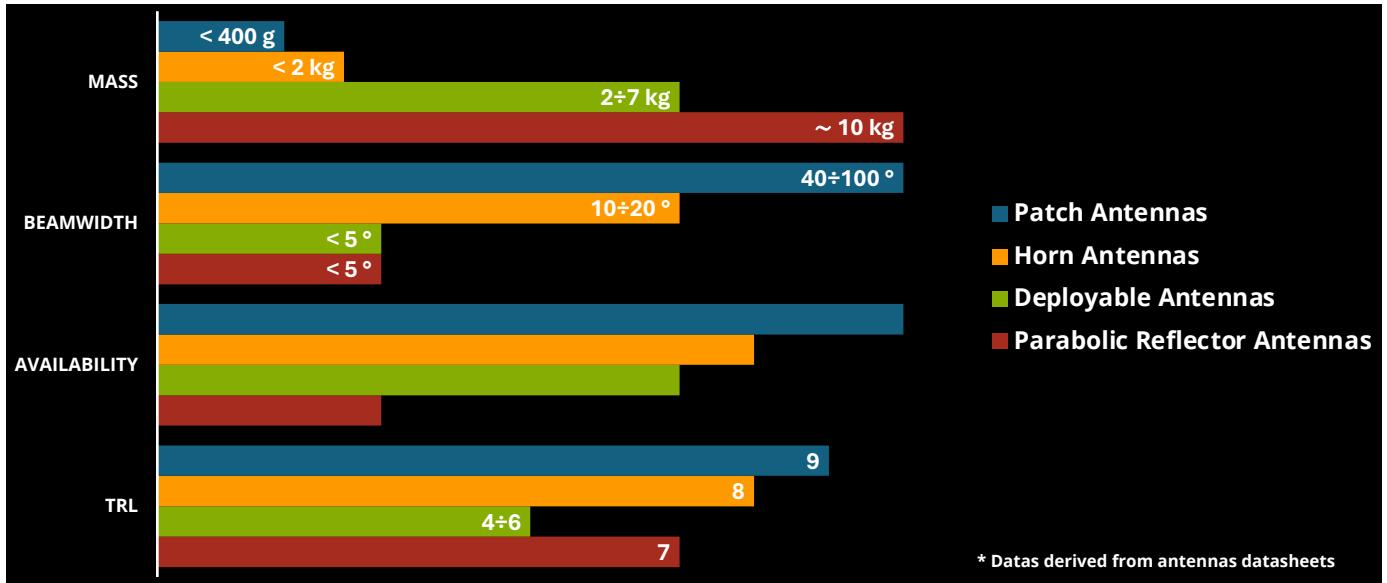


Figure 5.2: Antennas' type selection trade off.

$L_S = \frac{4\pi r^2}{\lambda}$ is the Free-Space loss;

$L_A = 10^{(\frac{1}{100r})}$ is the Atmospheric Attenuation ($L_A=1$ for Moon Link).

In order to simplify the selection, a first trade off between COTS antennas' type was conducted, Fig. 5.2, which led to the choice of *Patch-Patch Array Antennas*, for almost every trunk-link, since they have an overall higher TRL and availability.

Most importantly, they show greater beamwidth that eliminate the problem of a massive usage of antennas' pointing system, and a lower mass, reducing drastically the overall weight of the satellite.

As shown in Fig. 5.4, in order to achieve the requested data rate, the lower are the TX-RX gain and transmitted power of the Earth-Moon facilities communication systems, the higher are the constrains on the satellites' one, in terms of TX-RX gain.

From the obtained information of Earth-Moon section antennas, a first value of the minimum satellites communication performances, letting the code cycle to obtain more detailed data.

After the first selection of the COTS antennas, the calculations were iterated until almost every trunk-link was capable of reaching the data rates imposed in the requirements, changing the antennas if needed.

Finally, the calculations led to the selection of the antennas, which are illustrated in Sec. 5.1.4.

The results of the maximum achievable data rate are shown in Fig. 5.3, where it's clear that a compromise has to be reached with regard to the requirements for the communication with smaller lunar activities' systems.

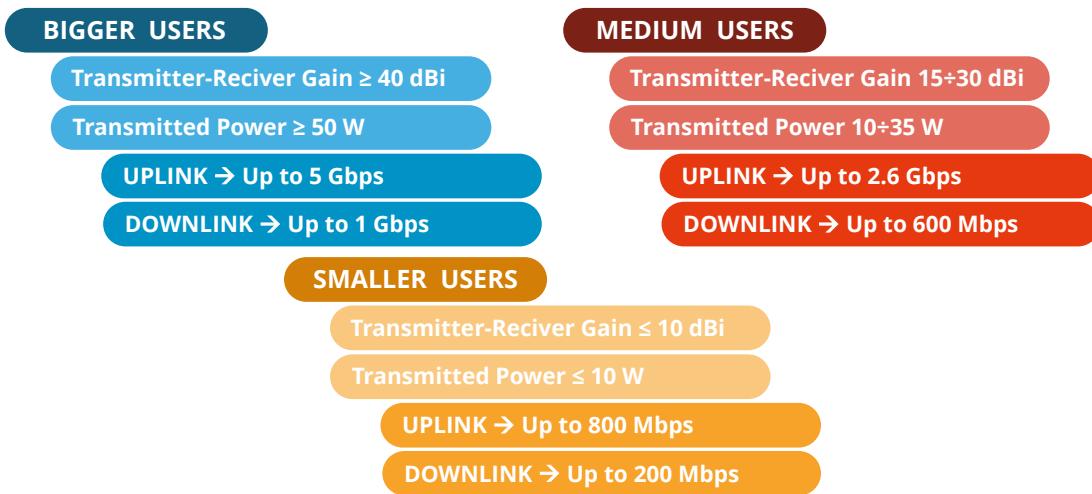


Figure 5.3: Final results of the achievable data rate for each macro-application.

The work flow for the antennas selection is illustrated in Fig. 5.5.

5. SATELLITES AND SUBSYSTEMS DESIGN



SUPREMÆ
DIGNITATIS
1343

UNIVERSITÀ
DI PISA

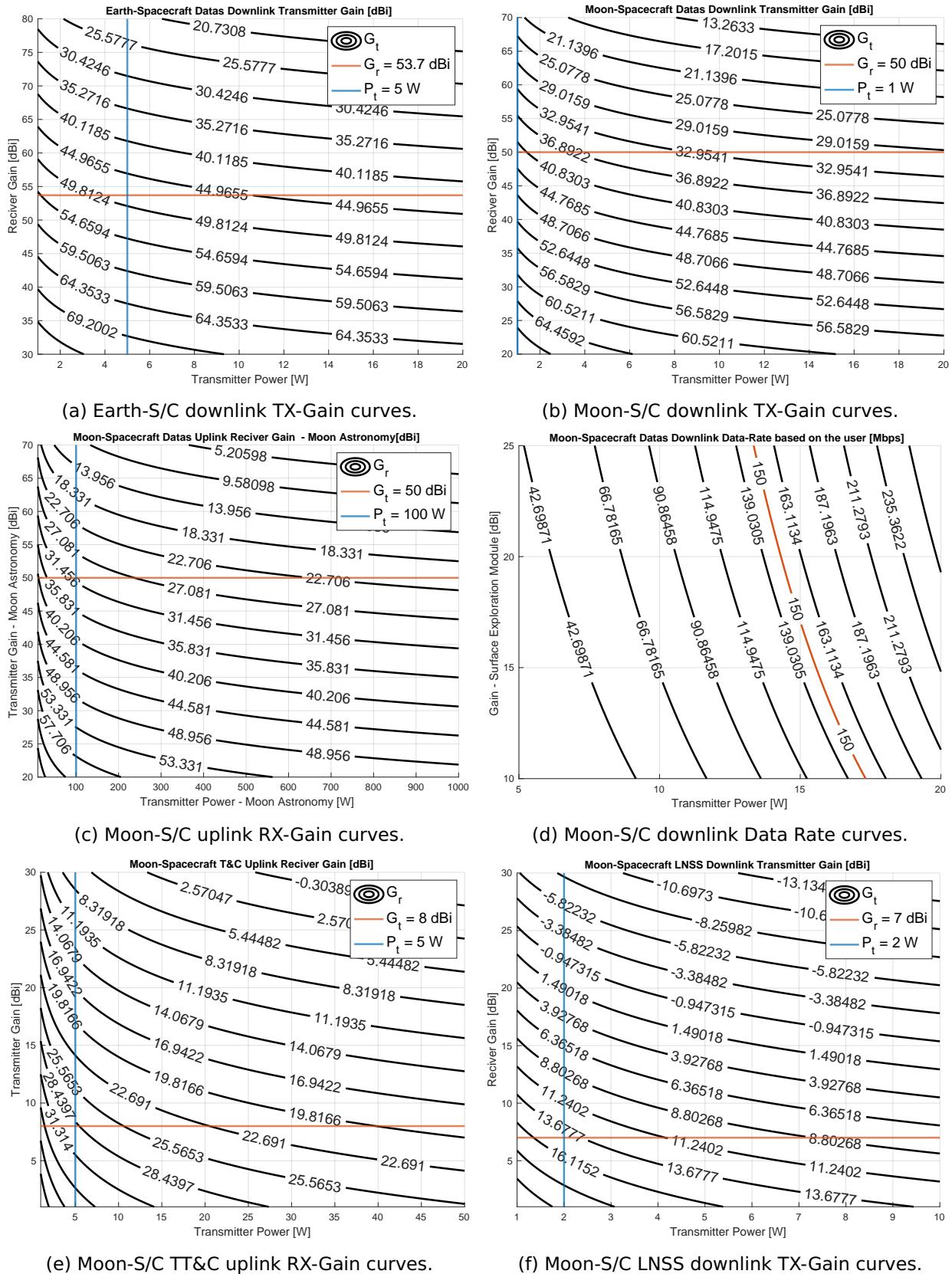


Figure 5.4: Gain and Data rates curves expressed as earth-moon communications system facilities' performances.

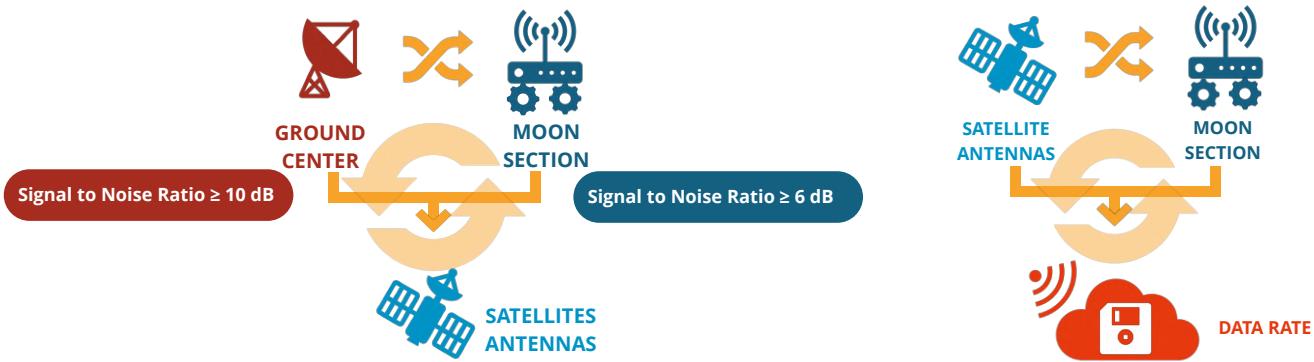


Figure 5.5: Work flow used for the selection of the antennas.

5.1.2. Ground Section

The possibility of using existing structures provided by various companies around the world has been checked, since it would have been the most efficient solution. Two possible providers for each of the two major human activities groups aforementioned has been found.

For what concern the support to smaller surface exploration facilities, the companies Swedish Space Corporation (SSC) and Atlas-Space Operations have been chosen: they can provide the access to multiple ground stations around the world, as shown in Fig.5.6.

This is a primary need for the mission, in order to guarantee a constant communication between Earth and Moon; in fact, for this reason, at least four stations (considering the possible reduced mobility or visibility) that will operate in shifts are needed.

They can provide single ground stations with multiple antennas or also multiple ground stations, located in the neighborhood of the same parallels, with a single antenna each: one antenna for each transmission link is needed for the mission.

Their operative characteristics are:

- SSC: [35]
 - Transmission:
 - * X-band: 7190-7235 MHz with EIRP = 75-85 dBW;
 - * S-band: 2025-2120 MHz with EIRP = 68-72 dBW;
 - Receiving:
 - * X-band: 8450-8500 MHz with G/T = 22-30 dB/K;
 - * S-band: 2200-2300 MHz with G/T = 22-24 dB/K.
- Atlas: [36]

- Trasmission:

 - * S-band: 2025-2120 MHz with EIRP = 48-65 dBW;

- Receiving:

 - * X-band: 7750-8500 MHz (variable among the different sites) with G/T = 25-32 dB/K;
 - * S-band: 2200-2300 MHz with G/T = 11.3-21.5 dB/K.



(a) SSC Antenna Network Map



(b) ATLAS Antenna Network Map

Figure 5.6: Ground stations for smaller surface exploration facilities

For what concern the support to greater moon surface human infrastructures, the requirement of a constant communication is not needed, but they have to be more effective in the receiving of the data.

For these reasons, the chosen antennas for the reception of the data in X-band are the GHY-3 provided by Goonhilly ($G/T = 53.7 \text{ dB/K}$) [37] and the Sardinia Deep Space Antenna, managed by INAF ($G = 72.4 \text{ dBi}$), [38]. They are shown in Fig. 5.7

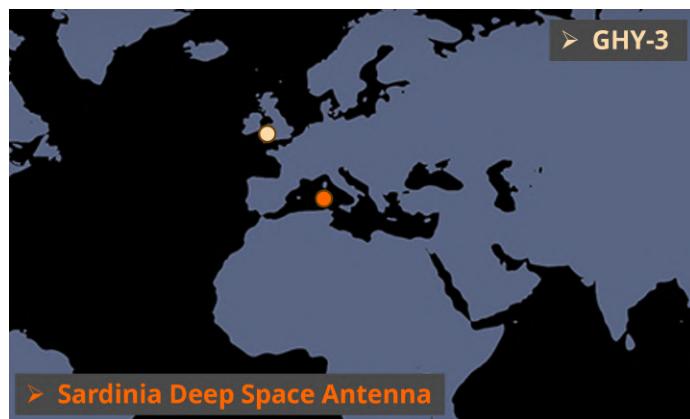


Figure 5.7: Ground stations for greater moon surface human infrastructures



5.1.3. Lunar Navigation Satellite System (LNSS)

To provide the Lunar Navigation Satellite System service, the operational methods and internal components of GALILEO, the European Global Navigation Satellite System, have been considered.

To accurately determine a user's position on the moon, the precise locations of the HERMES satellites must be known. This is achieved by measuring the distance of each satellite from Earth using a method that involves sending laser beams toward Laser Retroreflector Arrays mounted on the satellites and then measuring the travel time of these beams.



Figure 5.8: Example of a Laser Retroreflector Array mounted on the Lunar Pathfinder mission.

With the knowledge of the position of the HERMES satellites, the internal atomic clock can be updated and kept in sync.

GALILEO satellites use 2 RAFS (Rubidium Atomic Frequency Standard) by Airbus (Fig:5.9) and 2 PHM¹ (Passive Hydrogen Maser) by Leonardo (Fig:5.10). The first are more accurate in short periods of time but require frequent synchronizations, while the second ones are more robust and thus are more accurate in the long run. There are 2 of both for redundancy reasons.

GALILEO employ different pieces of hardware to convert the reference signal generated by the clocks into the final navigation signal transmitted by the antenna to the ground. A brief recap of those needed components follows:

- Clock Monitoring and Control Unit: responsible for generating the onboard reference frequency at 10 MHz.
- Navigation Signal Generation Unit: encode the navigation data onto the reference signal.
- Frequency Generation and Up-conversion Unit: to modulate the signal into a L-band signal at 1.5 kHz.

¹while GALILEO uses 2 PHM, the choice was ultimately made to use 2 Mini PHM, also by Leonardo.



Figure 5.9: RAFS Clock



Figure 5.10: Mini PHM.

5.1.4. Selected COTS Components

Here a brief overview of the antennas and their Low Noise/High Power Amplifier selected.

The LNA and HPA allow the communication links to reach the performance requirements found with calculations, and are provided by the company ERZIA (Tab. 5.2 and Tab. 5.3) [39].

There are nine primary antennas (Tab. 5.1): eighth for the uplink, downlink and UL/DL TT&C with Moon and Earth and the ninth for the LNSS service; they are all located on the same spacecraft face, in order to realize the communication link as desired (Sec. 5.1.1).

In addition, there are a couple of redundancy antennas (Tab. 5.4) located on all the remain faces, with the exception of the one where the engines are located, because they would not be able to work there, due to the plume interference.

They will serve as back-up antennas (one for uplink and one for downlink) in case of loss of the orientation of the spacecraft, in order to guarantee a minimum connection and data transmission.

They will work in S-band (UL: 2025-2110 MHz/ DL: 2200-2290 MHz) and will have the following performance:

- Uplink Capacity from Earth: 1100 kbps;



- Downlink Capacity to Earth is: 450 kbps;
- Uplink Capacity from moon is: 650 kbps;
- Downlink Capacity to moon is: 2 kbps.

Link	Company	Gain	Beamwidth
Data Uplink from Earth [40]	IQ Spacecom	6 ÷ 11 dBi	± 40°
Data Downlink to Earth [41]	Spaceteq	23 dBic	10° ÷ 12°
Data Uplink from Moon [42]	IQ Spacecom	10 dBi	40°
Data Downlink to Moon [43]	Printech	24-25 dBic	4° (X-Cut) / 8° (Y-Cut)
LNSS [44]	Printech	5 dBic	92° ÷ 102°
TT&C Uplink from Earth [45]	Planewave	4 ÷ 5 dBic	Omni
TT&C Uplink from Moon [45]			
TT&C Downlink from Earth [46]	Planewave	10 ÷ 12 dBic	± 30°
TT&C Downlink from Moon [46]			

Table 5.1: Selected COTS Antennas

Link	Gain [dB]	Power [W]
Data Downlink to Earth [47]	19	2
Data Downlink to Moon [48]	40	3
LNSS [49]	12	2
TT&C Downlink from Earth [50]	19	1.2
TT&C Downlink from Moon [50]		

Table 5.2: Selected COTS High Power Amplifier

Link	Gain [dB]	Power [W]
Data Uplink from Earth [51]	41	3.5
Data Uplink from Moon [52]	27	0.3
TT&C Uplink from Earth [45]		
TT&C Uplink from Moon [45]	26 ÷ 32	0.3

Table 5.3: Selected COTS Low Noise Amplifier

Link and Components	Company	Gain	Power [W]
Uplink Omni-Antenna [53]	Planewave	4 ÷ 5 dBic	-
LNA [51]	ERZIA	41 dB	3.5
Downlink Omni-Antenna [54]	Planewave	4 ÷ 5 dBic	-
HPA [55]	ERZIA	36 dB	7

Table 5.4: Selected COTS Redundancy Components

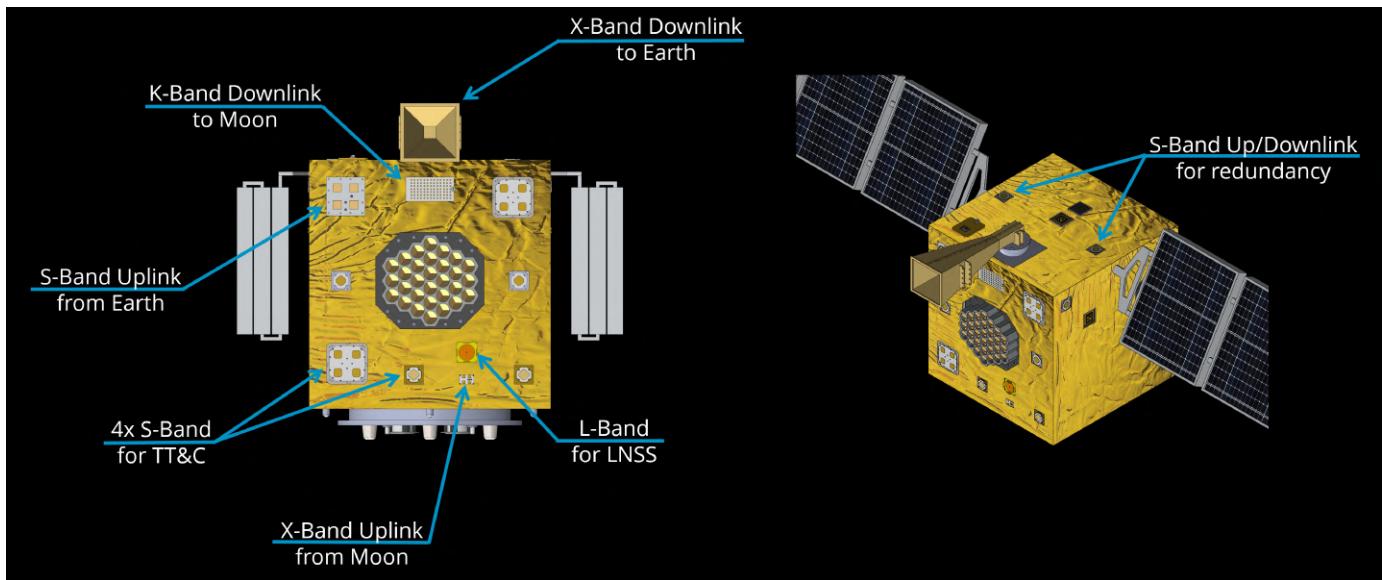


Figure 5.11: Antennas Arrangement



5.2. Propulsion Module

In this chapter, the various analyses that led to the selection of the spacecraft propulsion system will be presented.

5.2.1. Selection Rationale

The study conducted examined both propulsion technologies currently available on the market: electric propulsion and chemical propulsion. Both technologies were examined for the in-flight maneuvers, halo orbit insertion and station keeping phases. Given the commercial nature of our mission, the main objective of this analysis was to determine whether mass savings could be achieved by choosing the appropriate propulsion technology.

5.2.2. Orbit Injection

The decision to use electric propulsion for the halo orbit insertion maneuver was quickly dismissed due to the high ΔV required to be developed within a tight time frame. Also, the entire trajectory analysis was based on the assumption of using impulsive maneuvers 4.2¹, making the use of electric propulsion for this phase, impractical. Therefore, the propulsion technology chosen was chemical.

The chemical thruster selected for insertion maneuver shares very similar specifications as those used by Queqiao Tab. 5.5

Parameters	Values
Thrust	20 N
N° of thrusters	4
Specific Impulse	222 s
Thruster Mass	0.65 kg

Table 5.5: "20N Hydrazine Thrusters" made by "Ariane Group"

This thruster specification were used to calculate the propellant mass required for the insertion maneuver. By means of Tsiolkovsky equation, knowing the satellite mass, the

¹Through the use of impulsive approximate maneuvers, the piece of orbit where the maneuver is performed becomes negligible compared to the entire orbit, so we can safely consider the spacecraft motion to be keplerian



propellant mass can be easily computed.

With the mass of propellant known, the weight of the required hydrazine tanks was estimated (Appendix B.3.2).

5.2.3. Station Keeping Activities

The objective of this comparison was to determine the mass of "equipment"¹ required to be able to maintain the halo orbit for a duration of 10 years. The calculations were reiterated using various values of an "inert mass"².

In order to comprehend how the mass of "equipment" required on board varies between the two technologies, it is essential to examine how it changes as the "inert mass" value to stationkeep changes.

In order to compare the two technologies regarding stationkeeping activities, several simplifying assumptions were made:

- Electric stationkeeping maneuvers were approximated to impulsive maneuvers
- The power required by electric thrusters during stationkeeping maneuvers is entirely supplied by the solar panels
- Only one thruster was considered for the calculation, disregarding the redundancy aspect entirely³.

In addition, a search for studies that could provide us with data on the average annual ΔV required for stationkeeping in a halo orbit for both chemical and electric propulsion was conducted.

According to the paper [56], Queqiao achieves an annual consumption of approximately ~ 22 m/s using hydrazine propellant thrusters, while according to [57] and [58] it is evident that an electric propulsion system can achieve an annual consumption of approximately 30-40 m/s.⁴.

Impulsive Approximation

To simplify the analysis, an "act and wait" control strategy for the stationkeeping maneuvers was chosen. To approximate these maneuvers as impulsive, a very small fraction of the orbital period was allocated for the stationkeeping activities. This value was set at 1% of the orbital period. Consequently, for the 99% of time, the spacecraft follows a Keplerian trajectory, while during the remaining part, its motion is "perturbed" by the propulsive action of the engines.

¹The weight of this "equipment" includes both the mass of the engines and the propellant tanks. In the case of electric propulsion, it also includes the weight of additional solar panels required to sustain the propulsion needs.

²Representing an object with no propulsive capabilities

³In general, this assumption results in a minor variation in the total mass of the satellite

⁴These values were defined under the assumption of continuous propulsion in the case of [57] while in the case of [58] station-keeping activities are carried out at regular intervals (act and wait strategy)

These adjustment operations are performed once a day, reducing the total execution time of each individual maneuver to 0.06%¹ of the Halo orbital period. This approach allows us to achieve even more precise control over the orbit tracked by the spacecraft.

Having known the orbital period of the chosen orbit, the total time spent on station-keeping maneuvers was calculated and then the time spent on the single maneuver was computed.

- Orbital period: 15 days
- N° of maneuvers to be done for each orbit: 15^2
- Time allocated for each maneuver: 14,4 minutes ³

With this information in mind, and considering that electric stationkeeping consumes approximately 40 m/s per year⁴, the average velocity change required to be produced with each maneuver was computed ⁵

The subsequent step involved calculating the thrust required by the electric engines to execute the maneuver within the specified time frame⁶.

5.2.4. Technology Comparison

With that said, let's proceed to define the characteristics of the representative engines for the respective technologies used for comparison.

For the chemical propulsion stationkeeping calculations, the same engines as Queqiao were used, which possess the following characteristics:

- Thrust: 4N
- Specific Impulse (Isp): 211s ⁷ ⁸
- Mass: 0.4kg
- N° of engines onboard = 12

For the electric propulsion stationkeeping, the initial comparison between the two technologies was conducted assuming that Hall thrusters would be used as representatives

¹ Comes from 0.01/15 i.e. total maneuver time during an orbit divided by the halo orbital period [days]

² One each day

³ Comes from: $15\text{days} \cdot 1\% = 3.6\text{hours}$, that the total maneuver time allocated with each orbit, so dividing it by the number of maneuvers to be done each orbit gives 14,4 minutes as result

⁴ Using the "act and wait" control strategy [58]

⁵ $40\text{m/s} / 365 = 0.11\text{m/s}$

⁶ Required by the Impulsive approximation hypothesis

⁷ from the paper [56] knowing the Δ_V required by Queqiao for the insertion maneuver and the consumed propellant, we were able to compute the Isp of the Hydrazine engines via Tsiolkovsky equation. This calculation was done in the hypothesis that Queqiao uses a scaled version of the insertion thruster for the stationkeeping maneuvers

⁸ This derived Isp value aligns perfectly with the specifications of some engines currently available on the market.



5. SATELLITES AND SUBSYSTEMS DESIGN

for this technology. This decision was made because, according to the impulsive approximation hypothesis, electric engines must be capable of delivering a minimum thrust value.¹

Given that for Hall Thrusters technology, thrust, isp and power consumption are not linearly dependent, an "ideal Hall thruster" was modelled by deriving "average performance parameters" from engines currently available on the market.

The four thrusters selected² possess a power consumption of less than 1000W. This limit was chosen as an initial assumption; generally, engines with higher powers are not very widespread commercially, but are dedicated to special cases.

In order to estimate the possible weight and consumption of the engines needed for our purposes, given the "average performance parameters", "performance factors per unit weight" were derived ³, so that we could be able to calculate the weights of the electric propulsion system once the thrust required for the "impulse approximation" was known.

These calculations were repeated for various values of the inert mass, as initially discussed, ranging from 100kg to 450kg with deviations of 50kg.(Appendix B.3.2)

From the results, it was observed that generally, the weight of the equipment required to support electric propulsion is higher than the chemical propulsion's one. However, this is true until we factor in the weight of the consumed propellant.

As depicted in the graphs Fig. 5.12, opting for electric propulsion for the stationkeeping activities enables a reduction in the total mass of the satellite. This gap tends to amplify as the total mass of the satellite to be stationkept increases.

Given the commercial nature of our mission, electric propulsion technology was chosen for the stationkeeping maneuvers. This decision was made due to the overall mass savings compared to its chemical counterpart.

¹ Gridded Ions provide very high Isp but generate poor thrust relative to their weight, while MPDs are still a developing technology. Given these premises, the choice fell on the Hall Thrusters technology

² "Bht-100", "Bht-200", "Bht-300" and "Bht-600" are manufactured by Busek, and their specifications can be found on the website

³ The parameters computed were "average Thrust per kg", "average Power Consumption per kg" and "average Isp"

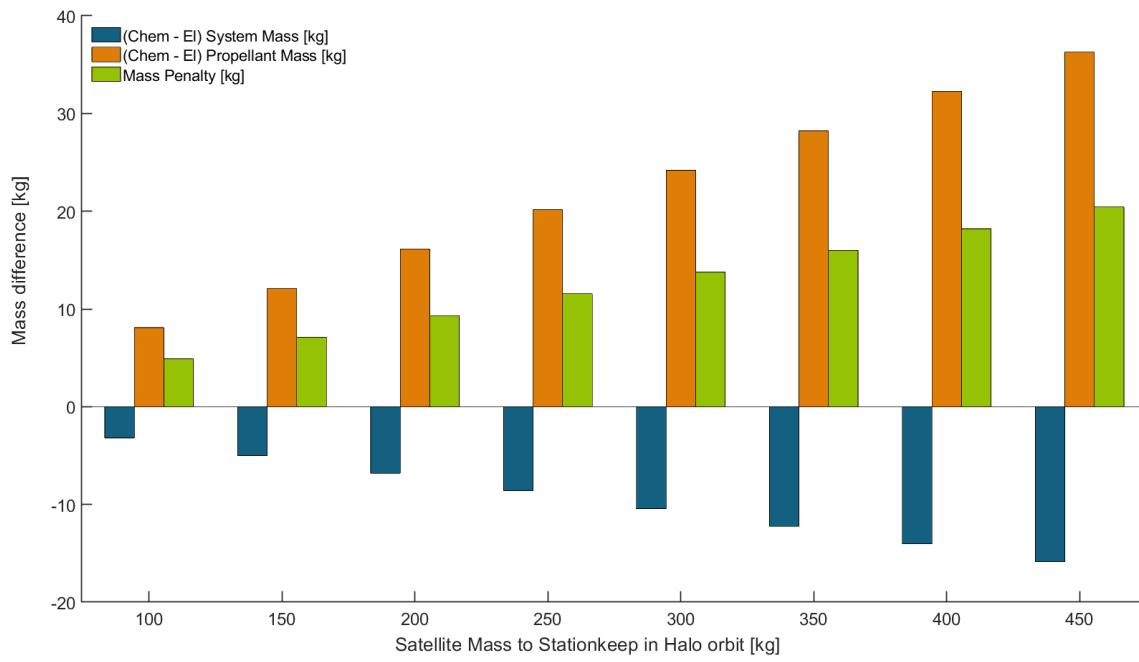


Figure 5.12: The blue bar represents the difference between the Chemical system mass and the Electric one. As you can see its negative because Electric propulsion requires "more equipment" on board to be functional. The orange bar represents the difference between the propellant mass required by the two propulsion systems while the green bar shows the mass penalty the satellite will have if we choose to use a chemical propulsion system for the stationkeeping activities

Thruster Selection

Considering the results obtained from the technology trade-off, a more in-depth analysis was conducted.

The objective was to determine the specific thruster model capable of minimizing the total mass of the satellite. To achieve this, the mass range from 100 to 200 kg was analyzed¹ with intervals of 20 kg.

This study was conducted using an updated version of the algorithm used here 5.2.4, in order to achieve higher accuracy in the results.

Sevne commercially available thrusters were selected as reference², and with each of them we tried to stationkeep the inert mass values taken. For each "engine-inert mass value" combination, we calculated the mass of the necessary "equipment" to have on board³ and at last we computed the final mass that the satellite would have (Appendix B.3.3), identifying the best thruster to choose for each mass value Tab. 5.6

¹Based on our preliminary mass estimations, the satellite weighed around 150-200 kg, therefore we decided to investigate that mass range

²This engines were both Hall thrusters (BHT-100 BHT-200 BHT-350 BHT-600) and Gridded ion (BIT-3RF RIT-10EVOv1 RIT-10EVOv2)

³following the same approach as before



Inert Mass values	BHT-100	BHT-200	BHT-350	BHT-600	BIT-3RF	RIT-10EVO v1	RIT-10EVO v2
100kg	155	155	159	167	-	161	159
120kg	189	184	189	196	-	190	188
140kg	219	214	218	226	-	226	217
160kg	248	243	248	255	-	256	246
180kg	278	273	277	284	-	285	275
200kg	308	309	307	314	-	314	305

Table 5.6: Total satellite mass[kg] computed with the stationkeeping performed by each thruster. The BIT-3RF didnt produce acceptable results (Appendix B.3.3)

From the table, it is evident which thruster will ensure the lowest total satellite mass. Therefore, we could choose the appropriate engine once the correct total dry mass of the satellite was determined.

5.2.5. Propellant and tank

In order to optimize the satellite propulsion system, a trade off between propellant was performed, that ultimately led to the selection of Krypton for the station keeping manoeuvres.

The analysis was conducted between the selected thruster's compatible propellant: Iodine, Krypton and Xenon, comparing:

- Mass of the propellant needed;
- Cost;
- Storage conditions;
- Satellite compatibility.

To asses the problem correctly, a first evaluation on the propellant tank was conducted thanks to a matlab code, C.3, where different material and shape was analyzed for xenon and krypton in supercritical storage condition. [59]

In particular titanium and composite carbon fiber reservoir¹ was considered for the trade off, with a spherical or cylindrical shape.

As shown in Fig. 5.13, as the propellant mass increases, the difference in terms of mass between titanium and composites tank increases exponentially, with a maximum difference of more than 50% for Krypton reservoirs.



Moreover, considering the eccentricity factor γ^2 , the more the tank is cylindrical, higher γ , the higher the mass, Fig. 5.14.

Since iodine can be stored at its solid state at room temperature, such aspects are not relevant, but material selection depends only on the material compatibility because of the high reactivity and corrosive property of the propellant.

So, for the aforementioned consideration, spherical composite carbon fiber tank were selected for xenon and krypton. A inconel 625 alloy³ material is instead preferable for iodine.

The calculations results show that xenon tank are lighter, requires lower storage volume⁴, Fig. 5.15, and the overall propellant mass calculated is less than the others, even if there is no much difference.

The cost comparison shows that xenon is, of course, the most expensive one whit an order of magnitude difference with the others, with iodine the cheapest one⁵.

For the storage condition, even if iodine can be stored solid it requires more power and control through the feeding line to maintain the sublimated state until the injection valves. The other gases instead doesn't require particular storage condition, with krypton that has a slightly lower storage density.

Last, while krypton and xenon don't give much problems in terms of satellite compatibility, iodine plume could interfere with the antennas scattered all over the satellites' surfaces. Since it's really reactive it could affect their proper functioning, compromising the entire mission.

For all the aforementioned krypton has been chosen has the propellant for station keeping thrusters, and the overall required mass is of 6 kg.

¹Titanium has a yield strength of about $1.4 \times 10^9 N/m^2$ and a density of $4850 \text{ kg}/m^3$.

Carbon fiber composite material has instead a yield strength of $0.701 \times 10^9 N/m^2$ while its density is around $1550 \text{ kg}/m^3$.

²The eccentricity factor γ is defined as $\gamma = \frac{(r+l)}{r}$, where r is the cylinder radius and l is the cylinder length. [59]

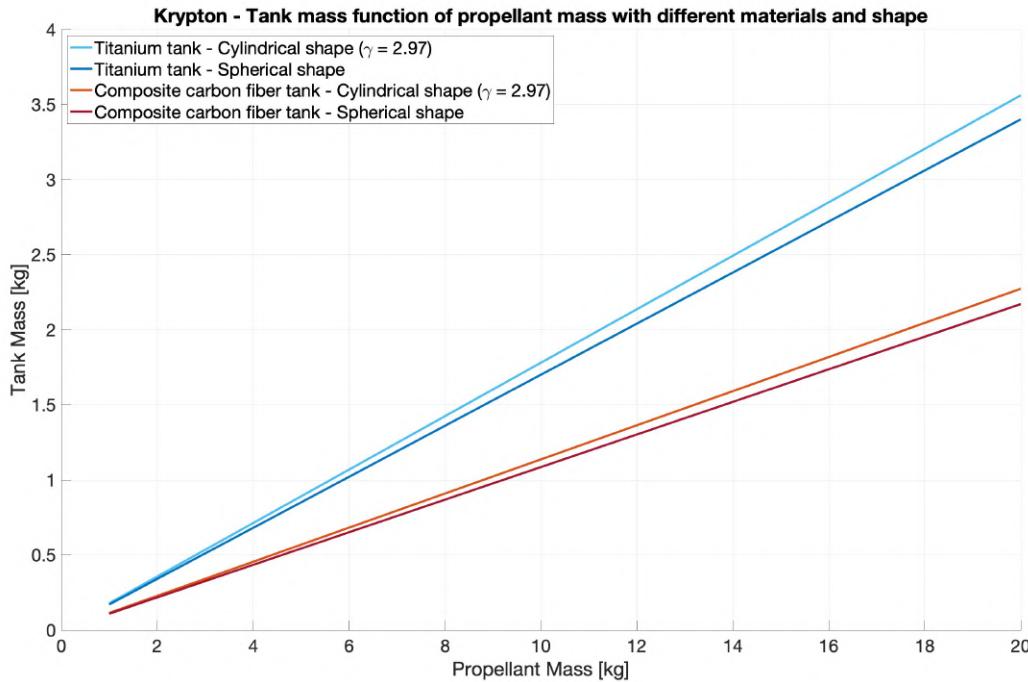
³Inconel is a nickel-chromium-based superalloy often utilized in extreme environments where components are subjected to high temperature, pressure or mechanical loads. The 625 alloy id particularly resistant to corrosive fluids.

⁴A 20% ullage volume was considered for iodine.

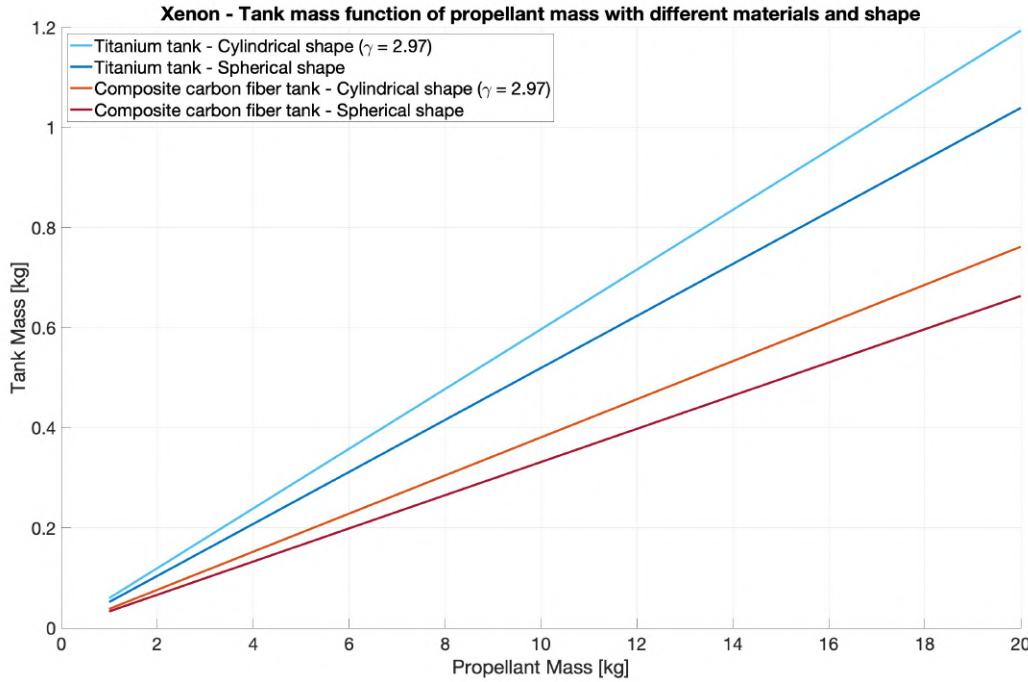
⁵Xenon > 15000 €/kg.

Krypon \approx 1500 €/kg.

Iodine = 975 €/kg.



(a) Krypton tank mass.



(b) Xenon tank mass.

Figure 5.13: Tank Mass trend based on propellant mass for different tank's shape and materials.

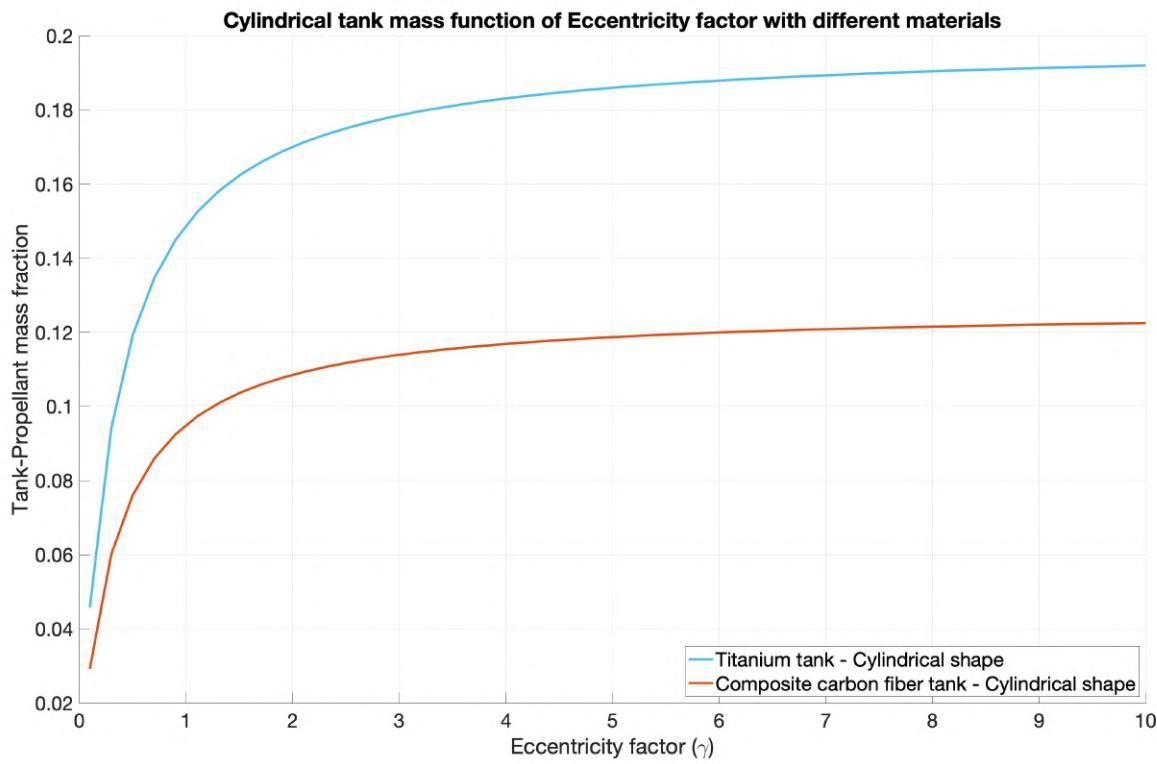


Figure 5.14: Tank mass trend at varying eccentricity factor values.

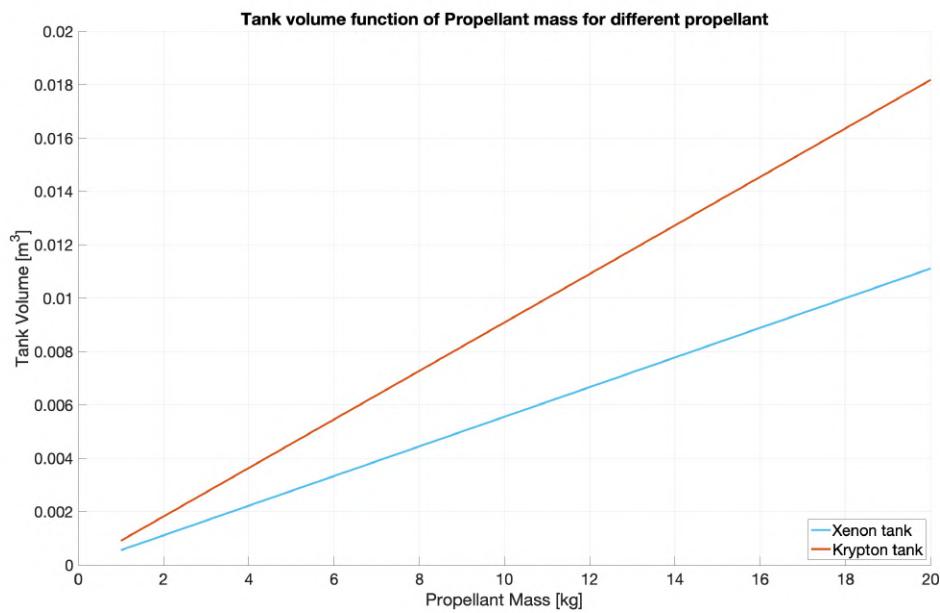


Figure 5.15: Tankage volume as a function of propellant mass, for krypton and xenon.

5.3. Attitude Determination and Control System

In the perspective of a mission orbiting in the vicinity of the EML2, the attitude determination and control subsystem plays a crucial role by allowing the satellite to reorient itself in space during its operational phases. The following procedure outlines the adopted approach for the preliminary sizing of this subsystem, acknowledging that a more accurate sizing would require a more in-depth analysis, which will be addressed in a subsequent phase of the mission study.

The major difficulties in sizing this subsystem are dictated by the innovative use of halo orbits. In particular, orbit perturbations are not easily modeled for this non-keplarian orbit and some approximations have to be done.

- **Adopted Reference Frames**

An inertial reference frame τ_I was selected at the center of the Moon, with the x-axis pointing from the Moon toward EML2. The z-axis points upward toward the Moon's north pole and is orthogonal to the Earth-Moon orbital plane, finally the y-axis complete the left-handed trio.

Regarding the body reference frame τ_B , the z-axis is directed along the antenna's pointing direction (that will be discussed later), the x-axis is orthogonal to the z-axis and pointing in the direction of the velocity vector, while the y-axis complete the left-handed trio, as shown in the figure 5.21 made with the matlab code C.8 that uses orbit parameter taken from [60].

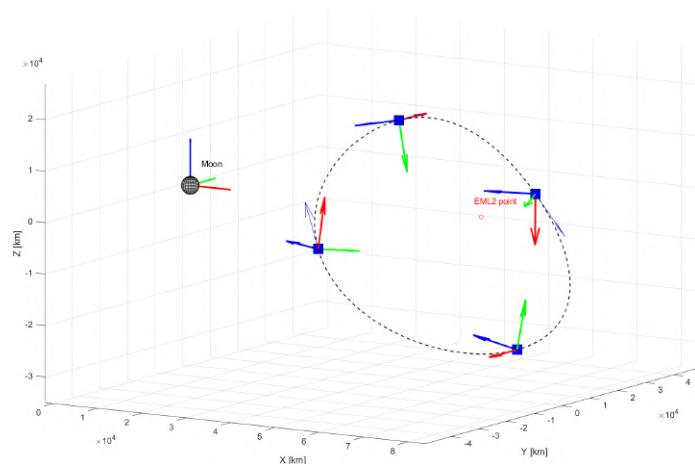


Figure 5.16: Reference systems used: Inertial reference system (Left) and Body reference system for each satellites (Right).

x, y, z shown in red, green, blue; velocity vector of the spacecraft in light blue

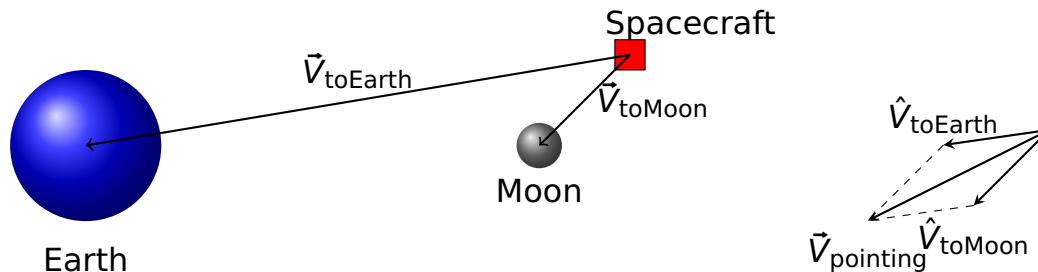


5.3.1. Requirements

The main requirements for the ADCS are dictated by the pointing accuracy of the antennas, necessary to have continuous communication with both Earth and Moon.

- **Pointing of the antennas**

All directional antennas are mounted on one front side of the spacecraft and so, to fulfill the requirement of constant and uninterrupted communication with Earth and Moon, the satellite will point in a direction that is always in between the two as shown in the figure below.



The direction where the antennas should point is defined as:

$$\hat{V}_{\text{Pointing}} = \frac{\hat{V}_{\text{toMoon}}}{\|\hat{V}_{\text{toMoon}}\|} + \frac{\hat{V}_{\text{toEarth}}}{\|\hat{V}_{\text{toEarth}}\|} \quad (5.3)$$

All the directional antennas are chosen to have a beamwidth so that the radiofrequency beam can reach both Earth and Moon.

- **Accuracy, range and slew rate**

We have analyzed the Halo orbit in the XY-Plane and in the XZ-Plane.

Among all the antennas, the smallest beamwidth is of about 40° . On the XY-plane, the greatest divergence between Earth and Moon is of about 30° . So we need an **accuracy** of at least $\pm 5^\circ$.

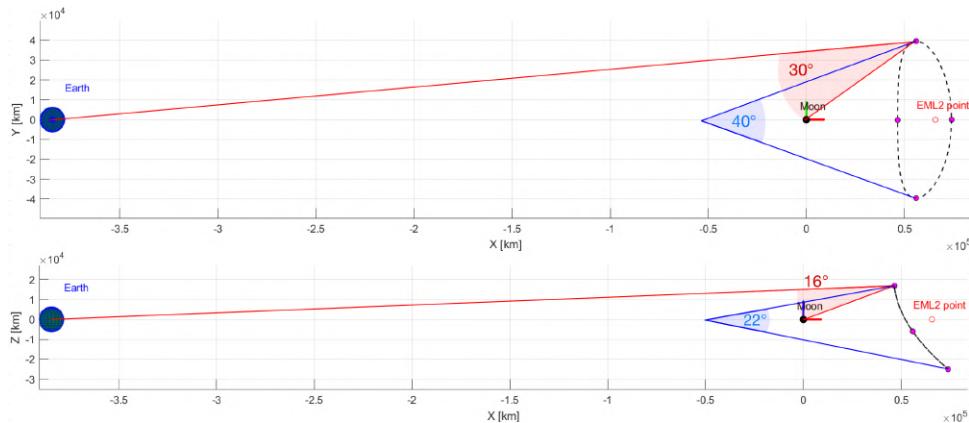


Figure 5.17: In red: angles between Earth and Moon
In blue: range of Pointing vector

During one orbit, in the xy-plane the pointing vector of the antennas goes from 160° to 200° , giving a **range** of about 40° ; whereas on the XZ-plane it goes from 169° to 191° , giving a **range** of 22° , as shown in blue in the figure 5.17.

In this context, the angular velocity of the body frame τ_B corresponds to the one the satellite should have to correctly pointing the antennas, that is the slew rate. The angular velocities of the spacecraft are shown in the figure 5.19 and the most demanding **slew rate** turns out to be $1.03 \times 10^{-3} \text{ deg/s}$.

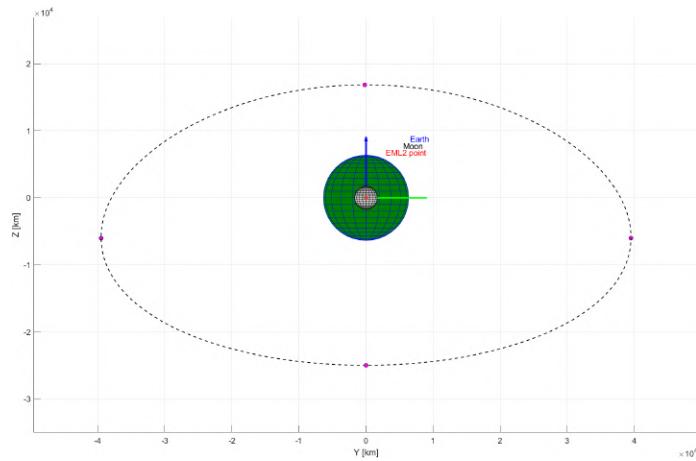


Figure 5.18: View of the points of maximum angular velocity and acceleration

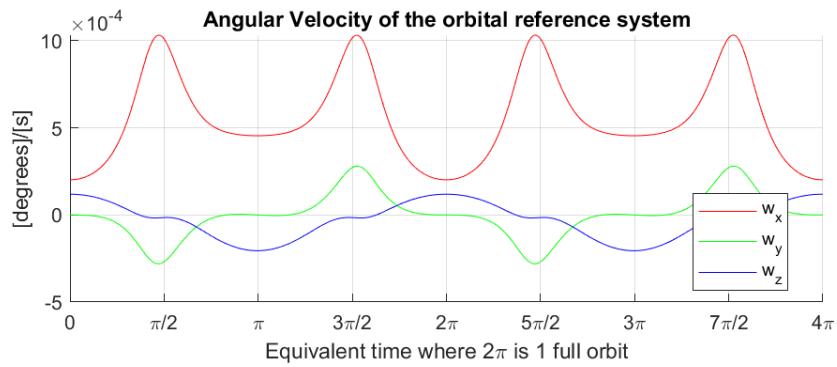


Figure 5.19: Angular Velocity of the orbital reference system

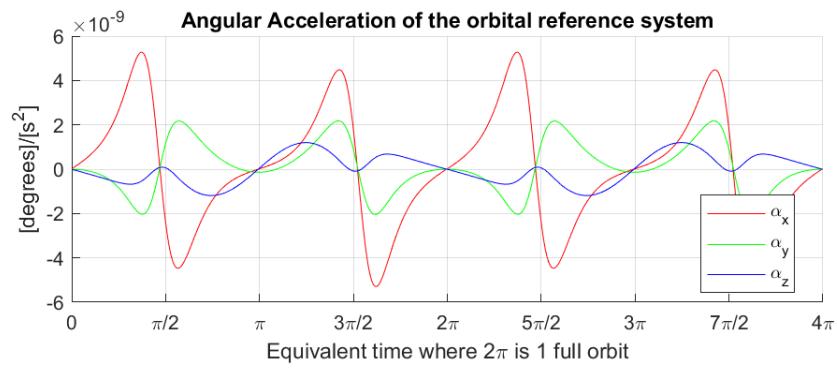


Figure 5.20: Angular Acceleration of the orbital reference system

5.3.2. Perturbation Torques

The main perturbative source acting on our spacecraft are represented by the gravitational pull of both Earth and Moon, and by the solar radiation pressure. Magnetic torques and aerodynamic torques are instead negligible.

- **Gravity Gradient Torque**

As a first approximation, the satellite was modeled as an homogeneous cube with a mass equal to the calculated mass and two side panel with the mass of the two solar panel, calculations were made with this script C.5. The resulting inertial matrix, calculated with reference to τ_B , is:

$$I = \begin{bmatrix} 107,6094 & 0 & 0 \\ 0 & 118,75 & 0 \\ 0 & 0 & 126,3594 \end{bmatrix} [kg \cdot m^2] \quad (5.4)$$

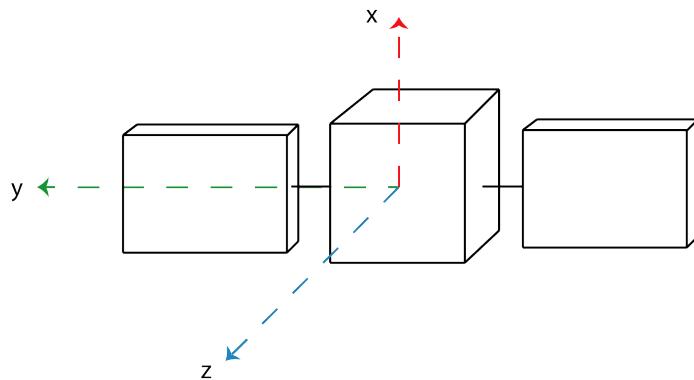


Figure 5.21: Simplified Model used to calculate the Moment of Inertia of the spacecraft

Given the inertial reference frame τ_I and the body reference frame τ_B previously defined, the gravity gradient torque has been computed considering the worst-case scenario, where the spacecraft is located at r_{sc}^{min} , at its closest point to the Moon:



$$M_{gravity} = \max \left(\frac{3\mu}{r^3} \begin{bmatrix} (I_z - I_y) \sin(\phi) \cos(\phi) \cos(\theta)^2 \\ (I_z - I_x) \cos(\phi) \sin(\theta) \cos(\theta) \\ (I_x - I_y) \sin(\phi) \sin(\theta) \cos(\theta) \end{bmatrix} \right) = \frac{3\mu}{2r^3} \begin{bmatrix} I_z - I_y \\ I_z - I_x \\ I_x - I_y \end{bmatrix} \quad (5.5)$$

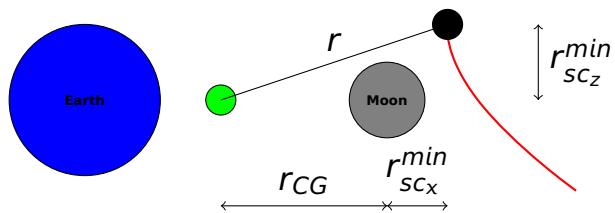
where ϕ , θ are the Euler angles and I_x , I_y and I_z are the inertia momentums of the spacecraft with respect to the selected body frame, while:

$$\mu = G(m_{Moon} + m_{Earth}) \quad (5.6)$$

$$\vec{r}_{CG} = \frac{m_{Moon}\vec{r}_{Moon} + m_{Earth}\vec{r}_{Earth}}{m_{Moon} + m_{Earth}} \quad (5.7)$$

$$r = \sqrt{(r_{CG} + r_{sc_x}^{min})^2 + (r_{sc_z}^{min})^2} \quad (5.8)$$

To calculate the momentum stored in one orbit this maximum torque has been multiplied by the orbital period, considering thus an overestimation.



• Solar Radiation Pressure Torque

The satellite has a symmetrical design in mind to decrease to the maximum the effect of the solar radiation pressure. It was chosen to artificially add an offset of the center of pressure from the center of gravity to account for any asymmetry in the final design. An educated guess of the offset was chosen as 1/100 of the total length ($L = 4.5m$) of the spacecraft, as shown in the figure 5.22.

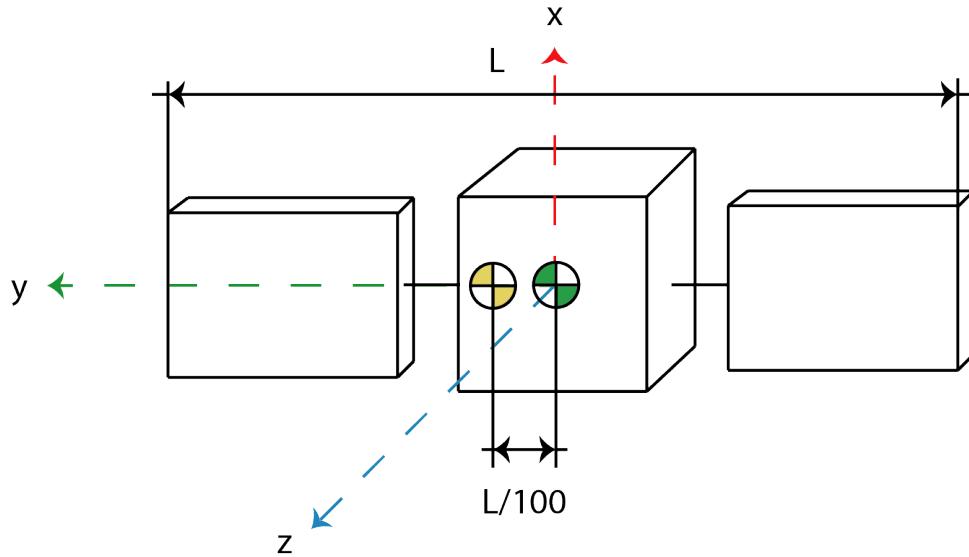


Figure 5.22: Simplified Aerodynamic Model used to calculate the solar radiation pressure torque acting on the spacecraft.

The solar radiation pressure torque result in:

$$\vec{M}_{solar} = \vec{r}_{cp} \times \vec{F}_{solar} \quad (5.9)$$

where \vec{F}_{solar} has been considered perpendicular to the solar panels (by putting $\alpha = 0$) to account for the worst-case scenario:

$$\vec{F}_{solar} = PA \left[(\rho_a + \rho_d) \hat{s} + (2\rho_s \cos(\alpha) + \frac{2}{3}\rho_d) \hat{n} \right] \quad (5.10)$$

with \hat{n} the normal to the exposure surface and \hat{s} the direction of the incident solar rays; ρ_a , ρ_d , ρ_s are respectively the absorptivity, diffusive reflectivity and specular reflectivity coefficients, chosen in such a way that $\rho_a + \rho_d + \rho_s = 1$. The area of exposure is $A = 3.5m^2$ that is the area of one side of the spacecraft plus the area of the two solar panels, while P is given by:

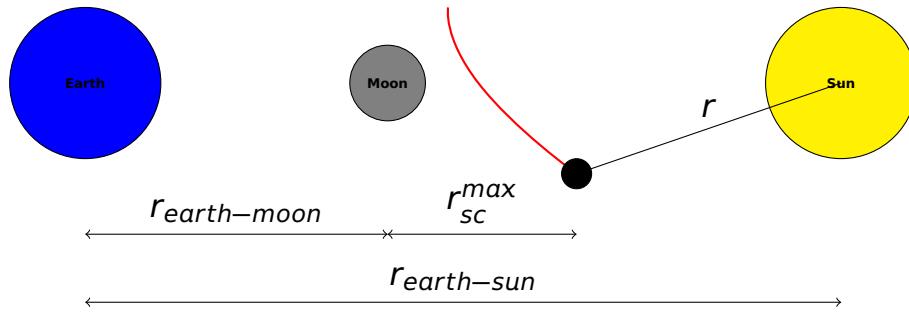
$$P = P_{earth} \left(\frac{r_{earth-sun}}{r} \right)^2 \quad (5.11)$$

with $P_{earth} = 4.5632 \times 10^{-6} Pa$.

To maximise the Solar radiation pressure, r is taken as the minimum distance from the spacecraft to the Sun, which occurs when the Moon is between Earth and Sun and the spacecraft is at its farthest point from the Moon:



$$r = r_{\text{earth-sun}} - r_{\text{earth-moon}} - r_{sc}^{\max} \quad (5.12)$$



• Results

The resulting torques and momentums to be stored are summarized in table 5.7 and table 5.8 respectively:

Disturbance Torques [Nm]			
	M_x	M_y	M_z
Gravity Gradient	5.917×10^{-11}	1.458×10^{-10}	-8.663×10^{-11}
Solar Radiation Pressure	9.039×10^{-7}	-	-
Total	9.039×10^{-7}	1.458×10^{-10}	-8.663×10^{-11}

Table 5.7: Resulting Disturbance Torques Acting On The Spacecraft.

Angular Momentums To Be Stored [Nms]			
	h_x	h_y	h_z
Gravity Gradient	7.662×10^{-5}	1.888×10^{-4}	-1.123×10^{-4}
Solar Radiation Pressure	1.171	-	-
Total	1.171	1.888×10^{-4}	-1.123×10^{-4}

Table 5.8: Resulting Momentums to be Stored In One Orbit (≈ 15 Days).

The dominant disturbance momentum is that one of the Sun as it can be seen.

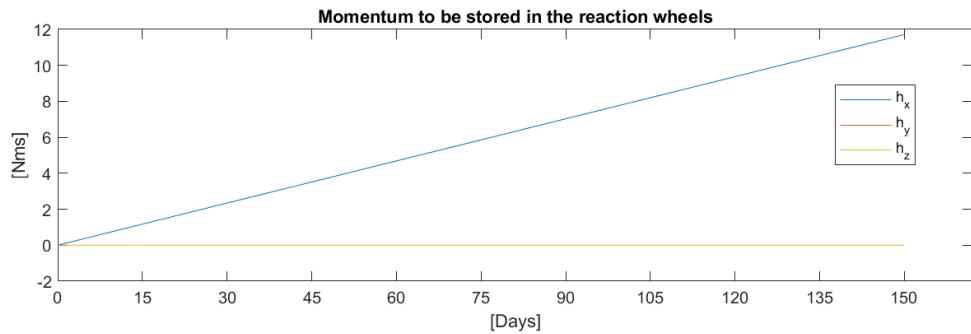


Figure 5.23: Momentum accumulated during 10 orbit or about 150 Days.



5.3.3. Actuators

Several stabilization strategies have been considered:

- **Passive Stabilization:** soon marked as impractical, this type of stabilization is typically useful in stable, well-known environment such as Earth orbits. This mission, however, operates in the vicinity of the Moon, and even if this is a well-studied environment, Halo orbits around Lagrangian point are inherently unstable. To stabilize the orbit, station keeping must be performed regularly and any passive stabilization technique will eventually be disrupted by the station keeping maneuvers. Furthermore, since EML2 is located over 60,000 km from the Moon, magnetic and aerodynamic passive stabilization are not feasible, and Gravity Gradient stabilization fails to meet the pointing accuracy requirements.
- **Spin Stabilization:** offers disturbance torque rejection through gyroscopic rigidity, but is only effective when few maneuvers are expected.
- **Double-spin-Stabilization:** utilized in geostationary telecommunication satellites with antennas on a fixed platform; unsuitable for this mission due to the non-planar, non-circular nature of Halo orbits.
- **3-Axis Stabilization:** although this method requires more energy and propellant, it provides complete control over the spacecraft's attitude, independent of the operational environment.

Based on these consideration 3-Axis Stabilization was selected.

Four reaction wheels are employed for the 3-Axis stabilization: one for each axis and one for redundancy. After reviewing previous calculations and available market options, the model **RWT250** by **ASTROFEIN** is recommended [61]. This model can store up to 4 Nms and deliver a maximum torque of 0.3 Nm.

ASTROFEIN Reaction Wheel Family	SmallSat TorqueLine		
Angular Momentum	RWT150	RWT250	µCMG
	1.0 Nms (@6000rpm)	4.0 Nms (@5000rpm)	8 Nms (@3820rpm)
Nominal Rotation Speed	6000 rpm	5000 rpm	3820 rpm
Maximum Rotation Speed	7080 rpm	5000 rpm	
Nominal Torque	0.1 Nm	0.3 Nm	14.0 Nm
Wheel Mass	<=1.5 kg	<=3.0 kg	<=8.0 kg
Moment of Inertia	$1.592 \cdot 10^{-3} \text{ kgm}^2$	$7.65 \cdot 10^{-3} \text{ kgm}^2$	
Dimensions	150mm*150mm*	200mm*200mm*	188mm*188mm*
	60mm	100mm	240mm

Figure 5.24: Possible choice of reaction wheels

The **RWT150** model was deemed insufficient due to its low angular momentum, leading to frequent momentum dumping. On the other hand, the highest performing model was excessively heavy and provided unnecessary torque. Thus, the **RWT250** model was chosen to provide adequate torque and so agility of the spacecraft, while containing weight.

5.3.4. Operations

The attitude control of the spacecraft allow us to perform a set of critical operations:

- **Tracking of the Antenna Pointing Vector**

In order to track $\vec{V}_{pointing}$ (eq: 5.3), the spacecraft must be able to increase or decrease its angular velocity and this can be done by increasing or decreasing the angular momentum to be stored or released by the reaction wheels. The angular momentum can be obtained by multiplying the maximum moment of inertia and the maximum angular velocity i.e. the slew rate; while the maximum torque to change the angular momentum is obtained with the angular acceleration.

$$h_{tracking} = I_{max}\omega_{max} = 0.00234 \text{ [Nms]} \quad (5.13)$$

$$M_{tracking} = I_{max}\alpha_{max} = 1.197 \times 10^{-8} \text{ [Nm]} \quad (5.14)$$

With the selected reaction wheels these requirements are largely met.

- **Momentum Dumping**

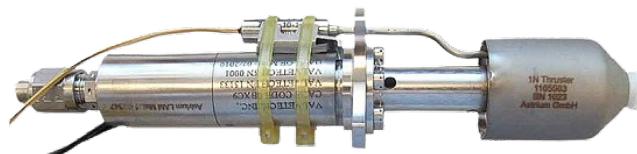
Analysis of prior calculations in tables 5.7 and 5.8 enables estimation of reaction wheel saturation frequency. Assuming the disturbance torque as previously described, an angular momentum of 4 Nms would require dumping every 3.42 orbits (approximately 51 days), as illustrated in figure 5.23. Considering a mission lifetime of ten years, a minimum of 71 momentum dumping maneuvers will be executed, cumulatively rejecting 284 Nms.

Since momentum dumping maneuvers results to be not very frequent, the choice of the propulsion system to be used to perform these maneuvers could fall on electric or chemical propulsion. We analyzed both solutions from the point of view of the total mass to be added on board, the time required to perform the maneuvers and the possibility of external contamination of the spacecraft due to the gases emitted by the thrusters (this eventuality has been considered for both technologies, and it has been thought to be solved by adopting appropriate engine placement on the surface of the spacecraft).

Regarding times and weights, as we have seen in fig: 5.12 electric propulsion tends to be heavier in terms of the “equipment” required on board as well as generating



(a) Reaction Wheel



(b) 1N Hydrazine Thruster

Figure 5.25: Attitude Control Hardware

less thrust. In fact, we saw how the times required for each desaturation maneuver increased by a factor of 10^2 compared to the use of chemical thrusters, and we also saw how the total mass of the electric momentum dumping system was about two times that of the chemical one.

Based on these considerations were selected 16 Hydrazine thrusters from the same family of the orbit injection thrusters selected in the chapter 5.2.2; each with 1 N thrust and positioned 0.5 m from the center of mass of the satellite. Turns out that approximately 250 g of Hydrazine will be consumed in individual maneuvers of 4 s totaling 284 seconds .

• In-Orbit Maneuvers

In case an orbit maneuvers is necessary for injection, insertion, or station keeping, it can be assumed that a rotation of up to 180° will be required to orient the hydrazine orbital thrusters or the station keeping electric thrusters appropriately. Assuming that the maneuver could take place along the axes of rotation with the highest inertia, the time required to have an angular acceleration up to 90° and a deceleration to stop at 180° can be calculated.

With a torque of 0.3 Nm and a moment of inertia of $130 \text{ kg} \cdot \text{m}^2$, the spacecraft has an angular acceleration:

$$\alpha_{sat} = \frac{M_{gyro}}{I_{max}} = 0.00231 \left[\frac{\text{rad}}{\text{s}^2} \right] \quad (5.15)$$

The time required to flip the spacecraft result to be: $t_{180^\circ} = 2t_{90^\circ} = 104 \text{ s}$, where:

$$t_{90^\circ} = \sqrt{\frac{2\pi}{\alpha_{sat}}} \quad (5.16)$$

5.3.5. Sensors

The detected minimum set of required sensors are Sun sensors, Earth sensor and an Inertial Measurement Unit (IMU).

- **Sun Sensors**

The sun sensors selected for the mission are or two types: Coarse Sun Sensor and Fine Sun Sensor, provided by Bradford. The Coarse Sun Sensor, with very low power consumption and a mass of only 200 g, provides a rough estimate of orientation relative to the sun, useful for initial attitude acquisition or fail-safe conditions. The Fine Sun Sensor, consuming only 0.25 W and weighing 375 g, offers more precise measurements, crucial for fine pointing operations and maintaining orientation during nominal operations.

- **Inertial Measurement Unit (IMU)**

The ASTRIX NS by Airbus, is a 3-axis Fiber-Optic Gyroscope, has a power consumption of 7 W and a mass of 1 kg and an error bias of 0.3°/1h and can provide accurate inertial data, essential for calculating the satellite's orientation and position in real-time.

- **Earth Sensor**

The Earth sensor's primary function is to determine the spacecraft's orientation relative to a celestial body. Since this satellite must point at all time toward Earth and Moon, this is a logical selection of attitude sensor for ensuring that the antennas are correctly oriented for communication and data gathering purposes.

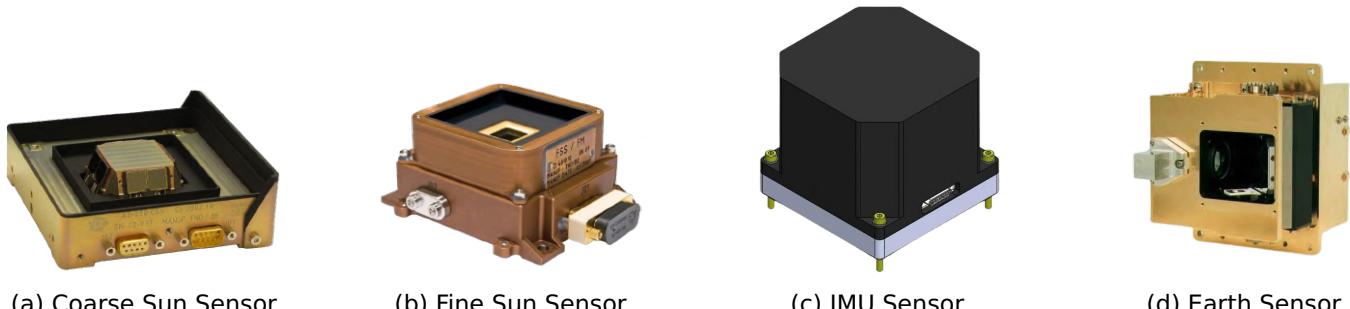


Figure 5.26: Attitude Determination Sensors



5.4. Power Generation and Supply

In this section, the study conducted to size of the power generation system on board the spacecraft will be presented.

5.4.1. Solar Panels

Once the components requiring power on board the spacecraft were identified, the performance required by the solar panels were computed.

Given that the solar constant varies negligibly between Earth and the Moon and considering the efficiency of current state-of-the-art solar cells is around 30%¹, the total area required by the solar panels onboard was computed to be approximately $1m^2$

$$\frac{\text{PowerRequired}}{\text{SolarConstant}} = \frac{418W}{1380[W/m^2]} = \sim 0.3 [m^2] \quad (5.17)$$

$$\frac{\text{SolarPanelArea}}{\text{CellsEfficiency}} = \frac{0.3m^2}{30\%} = \sim 1 [m^2] \quad (5.18)$$

This calculations need to take into account the EOL (end of life) performance of the solar cells, so considering an EOL efficiency of 25%, $0.3m^2/25\% = 1.2m^2$ of solar panels were obtained.

The previously calculated value didn't take into account the power required to sustain the stationkeeping activities. This value depends on the type of engine used, which in turn depends on the total mass of the satellite to be stationkept.

Based on current estimates, the satellite's weight for stationkeeping is projected to fall within the range of 160-190kg. Thus, referencing the data provided in the Tab. 5.6, the BHT 200 engine was selected for this computation.

In this scenario, a minimum of two engines are required to execute the stationkeeping maneuver, under the impulsive approximation hypothesis (section 5.2.3), resulting in total propulsion consumption of 400 watts. That's equal to adding another $\sim 1.2m^2$ of solar panels, for a total of $2.4m^2$. This value was then rounded to $2.5m^2$.

The total solar panels mass to be carried on board was estimated to be around 15kg using "SpaceTech GmbH" [62] solar panel technology.

¹values taken from triple junction gallium arsenide solar cells

5.4.2. Power Storage

For battery sizing, the primary factor to consider is the occurrence of eclipse times caused by the Moon and Earth. However, given the chosen halo orbit's position relative to both celestial bodies, the possibility of an eclipse event is very remote. Therefore, the batteries have been sized for emergency and power surge.

The design condition are ensuring the satellite's capability to communicate in a safe-mode for an extended period or guaranteeing the ability to perform at least one stationkeeping maneuver using its electric thrusters.

Since the last request is the most demanding one, an estimation was made regarding the battery capacity needed to fully satisfy the energy demand for the propulsive maneuver. This calculation were conducted considering the gradual degradation of the capacity of batteries due to numerous charge and discharge cycles.

In fact, to address this issue, a Depth of Discharge (DoD) of 70% and a battery capacity of 70% [63] of the total capacity were considered, assuming to work with batteries operating at end-of-life performance levels.

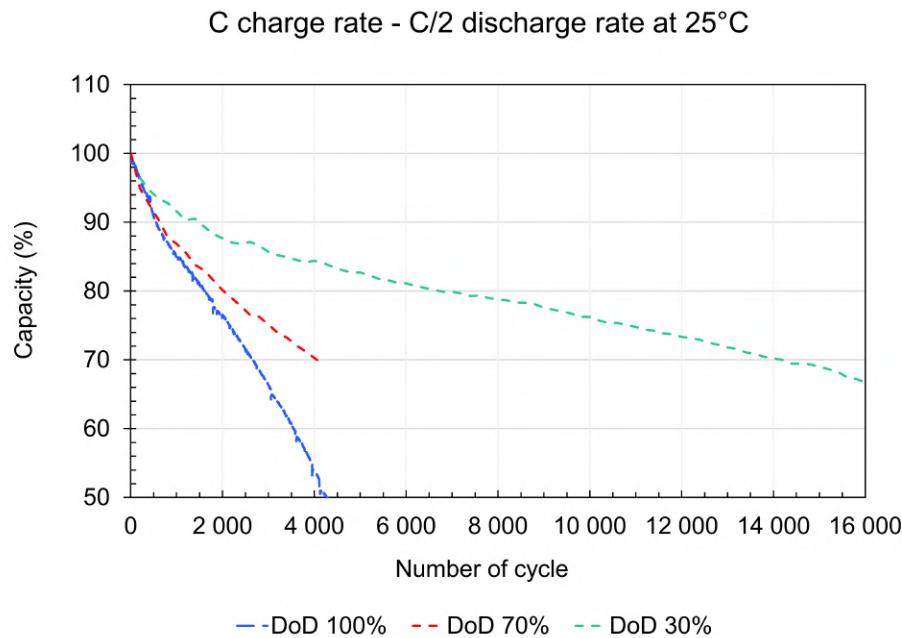


Figure 5.27: Saft battery DoD vs capacity values. The spacecraft will do 3650 stationkeeping maneuvers, which corresponds to 3650 charge-discharge cycles

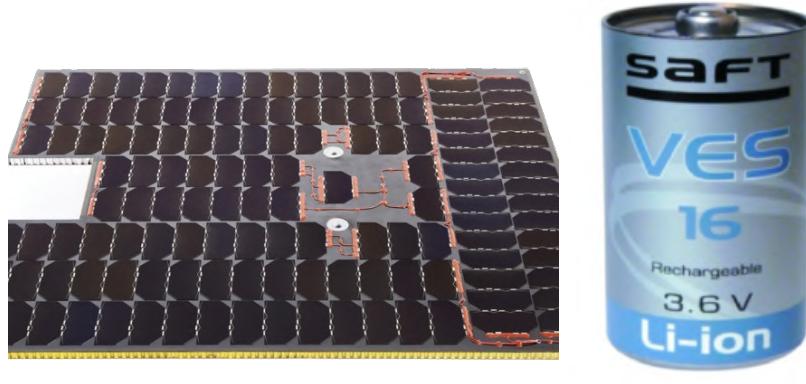
$$\frac{\text{BurnTime} \cdot \text{EngineCons} \cdot N_{\text{engines}}}{\text{EOLCellCapacityFactor}} = \frac{864s \cdot 200W \cdot 2}{70\%} = \sim 140Wh \quad (5.19)$$

From these calculations (section C.9), the required battery capacity of about 140Wh



were determined, rounded up to 200Wh considering a safety margin.

The power storage system of the spacecraft consists of two secondary batteries (more on this in the next section) that will be custom made starting from european made "SAFT VES16 cells" [64].



(a) SpaceTech solar cells

(b) SAFT VES-16 cell

Figure 5.28: Power Generation components

5.4.3. Stationkeeping power requirements: load shift

The objective of this study was to explore the possibility of redistributing some of the energy demand of the electric thrusters to the batteries, potentially allowing for the use of smaller solar panels, resulting in substantial mass savings¹.

To accomplish this, calculations were conducted (section C.9) by gradually shifting the "power load" required by the propulsion system from the solar panels to the batteries, calculating each time the sum of the two system masses.

Fig. 5.30 shows the results obtained by this analysis. As evident from the graph, opting for batteries instead of using solar panels to fully sustain the stationkeeping activities produce a mass saving of $\sim 4\text{kg}$. The results showed in the graph also include the mass for a redundant battery pack in the event of a failure.

Opting for batteries offers several significant advantages beyond mass saving:

- It permits to effortlessly incorporate redundancies into the system
- Decreases the total inertia of the satellite²
- Decreases the probability of a micrometeoroid impact on the solar panels surface is reduced, as the total impact area available is smaller.

With batteries fulfilling the station keeping propulsion requirements, the final power generation design was determined:

- **Solar panels:** rounded to 2m^2 ³ with a total mass of 12kg
- **Batteries:** total capacity rounded to 200Wh for each battery pack, with a total mass of $\sim 4.5\text{kg}$

¹Previously, the entire energy supply required for the stationkeeping activities was obtained through solar panels.

²thereby enhancing its maneuverability

³That's because when the satellite is pointed towards the moon, there is a shadow casted on the satellite solar panels, that diminish the power produced by them. To overcome this, the solar panels area was size with a 60% safety factor

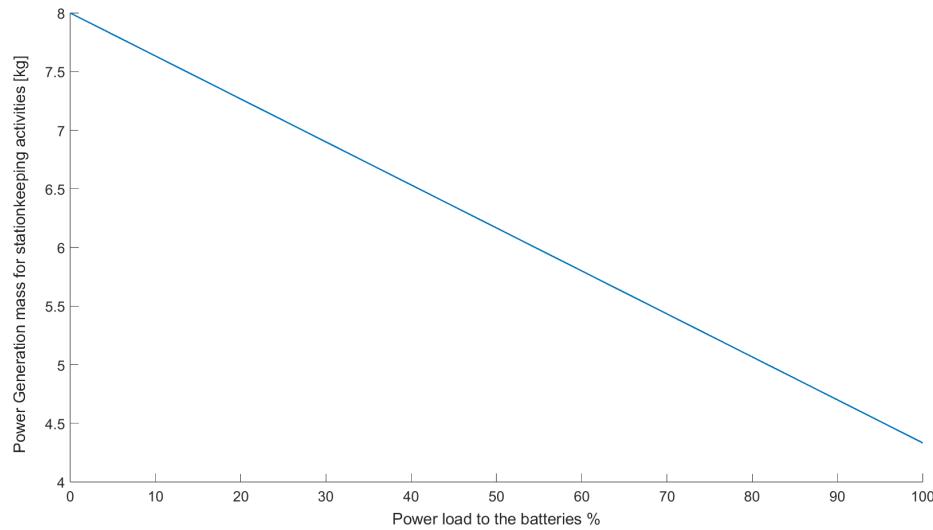


Figure 5.29: Power generation system mass values with different load distribution values.

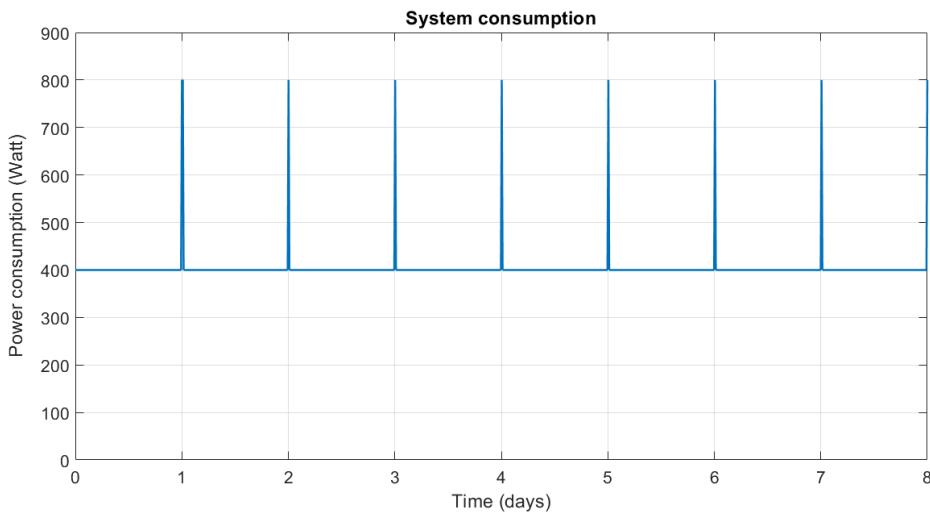


Figure 5.30: Spacecraft peak power consumption during 7 days. In this graph are clearly visible the stationkeeping activities, during which the spacecraft power consumption doubles

5.5. Thermal Control

The main purpose of this preliminary thermal analysis is to select a passive superficial thermal control technique in order to maintain the temperature of the spacecraft's components within their range.

These principal requirements are shown in Tab.5.9.

Sub-systems	Min. operative T	Max. operative T	Min. survival T	Max. survival T
Antennas	-30	60	-70	100
Batteries	10	30	-20	40
Momentum wheels	0	50	-20	70
Electronics	0	40	-20	70

Table 5.9: Temperature ranges of most thermally critical subsystems

Batteries, momentum wheels and electronics are reported because they are the components with the most critical ranges.

While the temperature range for the antennas is referred to the one with the most restrictive requirements, because one of the aim of the mission is the possibility to guarantee a constant connection between Earth and the far side of the Moon, so all the antennas must be able to work simultaneously.

5.5.1. Spacecraft Thermal balance

It has been chosen to investigate the temperature of the spacecraft, firstly as a whole, using the equation of thermal balance.

This choice has been made because, as preliminary analysis, the passive thermal control techniques, such as blankets and coatings, has been investigated, in order to satisfy the requirements with the less expensive techniques in terms of cost, mass and design.

This first step of the study has the primary goal of satisfying the requirements for internal components, while the patch antennas requirements must be satisfied looking at the equilibrium temperature of the external surface of the spacecraft, as it will be shown in the next section.

$$\dot{Q}_i + \dot{Q}_S + \dot{Q}_{MR} + \dot{Q}_{SM} = \dot{Q}_r \quad (5.20)$$

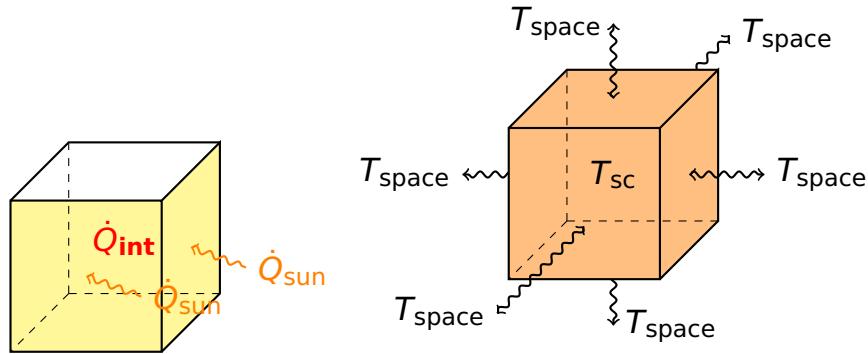
$$\dot{Q}_i + \alpha \cdot A_{\perp} \cdot I_{\text{sun}} + \alpha \cdot \alpha \cdot F_{S,SM} \cdot A_S \cdot I_{\text{sun}} + \sigma \cdot \epsilon \cdot F_{S,M} \cdot A_S \cdot (T_{sc}^4 - T_m^4) = \epsilon \cdot \sigma \cdot F_{S,S} \cdot A_S \cdot (T_{sc}^4 - T_{\text{space}}^4) \quad (5.21)$$



In this equation ², the contributions of albedo and irradiation with the Moon have been neglected since $F_{S,M}$ and $F_{S,SM}$ (view factor, spacecraft to Moon and spacecraft to sunlit Moon) are much smaller than $F_{S,S}$ (view factor, spacecraft to space), and since $F_{S,S} + F_{S,M} = 1$. For these reasons the approximation of $F_{S,S} = 1$ has been made.

The equation 5.21 becomes:

$$\dot{Q}_i + \alpha \cdot A_{\perp} \cdot I_{\text{sun}} = \epsilon \cdot \sigma \cdot A_s \cdot (T_{sc}^4 - T_{\text{space}}^4) \quad (5.22)$$



The calculation of T_{sc} has been made with the following values:

- \dot{Q}_i = about 60% of the power utilized from the spacecraft ($P_{SC} = 500 \text{ W}$)
- The absorptivity in the visible and the emissivity in the infrared of the spacecraft are chosen from the commercial datasheet of coatings and Multi-Layer-Insulation blankets (see Tab.5.10) [65].
- The area of one face of the spacecraft is 1 m^2 and the total surface area of the cube is 6 m^2 (the real cubic shape of the spacecraft has been considered).
- A medium value of the projected area perpendicular to the sunlight during the mission has been considered.

Material	α	ϵ	α/ϵ
Aluminum Paint	0.3	0.3	1
Aluminized Black Kapton	0.9	0.82	1.1
Aluminized Kapton	0.35	0.49	0.72

Table 5.10: Thermal characteristics of off-the-shelf coating and blankets of interest

The results are shown in the Fig.5.31

²See Appendix A for the nomenclature

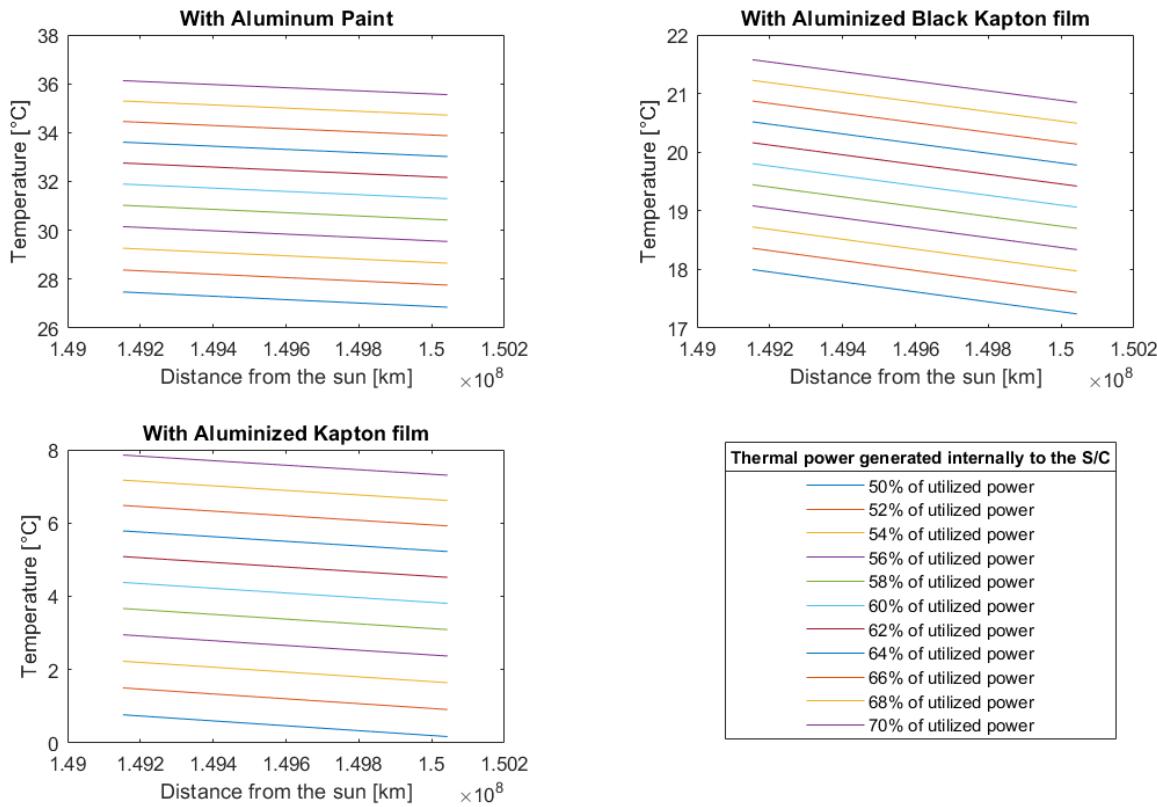


Figure 5.31: Spacecraft equilibrium temperatures

The graphs show the temperature of the spacecraft varying with the distance from the Sun and with different amounts of power generated internally to the spacecraft. This quantity of power has been varied in a range between 50% and 70% of the total power utilized within it.

The results show that the variation of distance from the Sun that occurs along with the rotation of the Moon around the Earth does not significantly affect the temperature.

Since that the spacecraft will experience a temperature gradient between the part exposed to the sunlight and the opposite one, a further investigation has been made.

5.5.2. External surfaces thermal balance

It has been chosen to calculate the temperature of a single face of the cubic spacecraft in order to find the thermal control solution that can satisfy also the temperature requirements of the antennas, in the scenarios of sunlight reaching and not reaching them.

In order to do this, the same equation has been used, but with two different assumptions:



- The fraction of \dot{Q}_i that will affect the thermal balance of a single face has been considered one sixth of the previous one, in the hypothesis of isotropic heat diffusion from the inside.
- The power transferred outside the surface has been divided into two components:
 - an irradiating power through the outside,
 - a heat transfer through the inside.

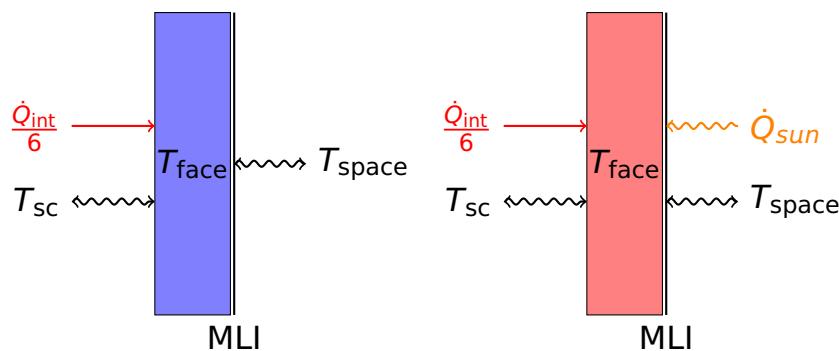
The last has been approximated with the criteria of worst-case scenario: since the heat transfer through the other part of the spacecraft will mainly be a mixture of irradiation and conduction, the first one has been chosen as the only one occurring.

This decision comes from the fact that a more detailed investigation of the heat conduction within the parts of the spacecraft would have not been significant at this level of the study, and also because the real maximum and minimum temperatures experienced by the surface of the spacecraft will be respectively lower and higher than the ones calculated here, and in this way the validity of this calculation can be guaranteed.

The equations for the "hot" scenario and "cold" scenario of the analyzed surface, respectively, become the following:

$$\frac{\dot{Q}_i}{6} + \alpha \cdot A_{\text{face}} \cdot I_{\text{sun}} = \epsilon \cdot \sigma \cdot A_{\text{face}} \cdot (T_{\text{face}}^4 - T_{\text{space}}^4) + \epsilon \cdot \sigma \cdot A_{\text{face}} \cdot (T_{\text{face}}^4 - T_{\text{SC}}^4) \quad (5.23)$$

$$\frac{\dot{Q}_i}{6} = \epsilon \cdot \sigma \cdot A_{\text{face}} \cdot (T_{\text{face}}^4 - T_{\text{space}}^4) + \epsilon \cdot \sigma \cdot A_{\text{face}} \cdot (T_{\text{face}}^4 - T_{\text{SC}}^4) \quad (5.24)$$



The results are shown in Fig. 5.32, and in Fig. 5.33

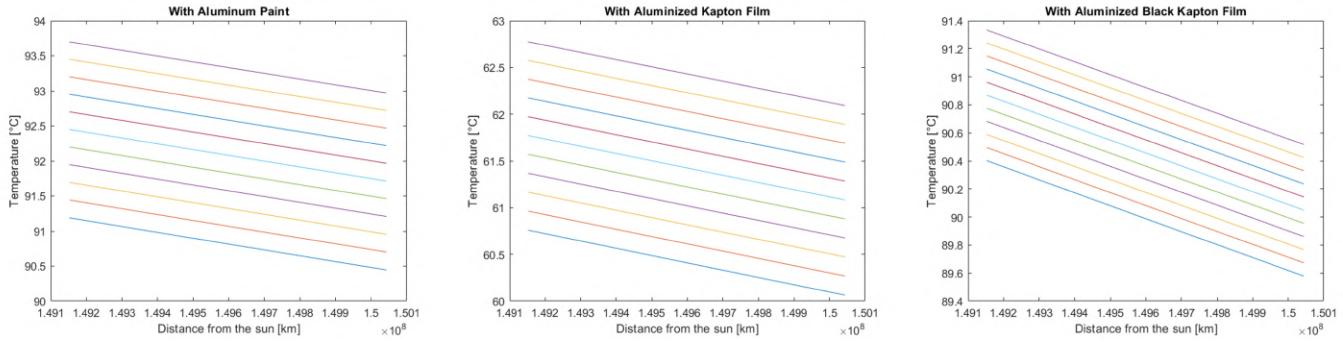


Figure 5.32: Temperatures of the spacecraft face in sunlight

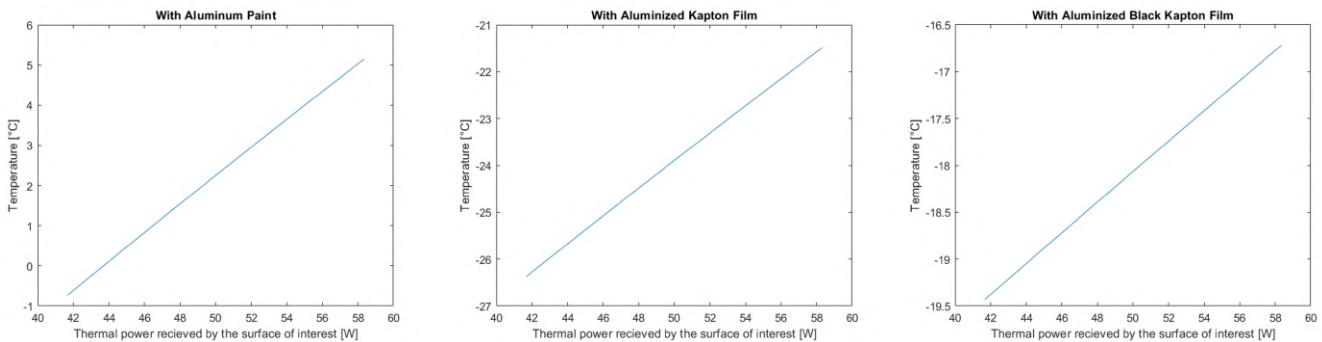


Figure 5.33: Temperatures of the spacecraft face in shadow

5.5.3. Final results

The results concerning the equilibrium temperature of the whole spacecraft and those of faces receiving and not receiving directly the sunlight are shown in Tab. 5.11.

Material	T_{sc} Range [°C]	Max. T_{face} Range [°C]	Min. T_{face} Range [°C]
Aluminum Paint	29÷36	90÷94	-1÷5
Aluminized Black Kapton	17÷22	89,5÷91,5	-19,5÷-16,5
Aluminized Kapton	0÷8	60÷63	-27÷-21

Table 5.11: Passive thermal control results and trade-off

Aluminum Paint and Aluminized Kapton film can satisfy the equilibrium temperature requirements for the internal components, with the exception of the batteries, while the Aluminized Black Kapton film can keep the equilibrium temperature within all that ranges.

However, the Aluminized Kapton film is the only one able to keep the temperature of the most critical surface (the one with the antennas) within the ranges.



From the previous observations, two options as been considered:

1. Using the black insulation, while adding heat pipes in order to reduce the temperature gradient within the spacecraft.
2. Using the Aluminized Kapton film, while adding heaters for keeping the batteries in the right range.

The second solution was selected, because it is less expensive and risky: the batteries already include their own heaters, reducing design costs.



6. Mission Budgets

6.1. Link Budget

Based on the work explained in Sec. 5.1, the Link Budget Analysis results are shown in this section.

In Fig. 6.1, all the frequency usage is shown alongside with the maximum data rate achievable for each trunk link. All earth-link has a signal to noise ratio of at least 10 dB, while all moon-link has a minimum signal to noise ratio equal to 6 dB.

For the data uplink from the moon, a maximum data rate of 5 Gbps can be achieved when the satellites communicate with larger moon users such as moon astronomy/outpost activities. The minimum is around 800 Mbps instead, for link with smaller lunar surface facilities.

The backup antennas' system installed for redundancy, is still capable to fulfill all primary mission objectives but resulting in much lower data rates link:

- **Uplink from Earth:** 1100 kbps;
- **Downlink to Earth:** 450 kbps;
- **Uplink from Moon:** 650 kbps;
- **Downlink to Moon:** 2 kbps.

6.2. Mass Budget

For each of the four configurations and mission profiles, the study was initiated with a link budget analysis to determine the appropriate antenna to use.

Subsequently, from the estimated consumption of the telecommunications package, the solar panels and power generation system were sized.

Looking at past missions, the weights and consumption of the attitude and thermal control systems were estimated for each configuration.

These estimated values served as the foundation for calculating an initial total dry mass, which was then used to further our investigations into the components tailored to our satellite class.

With this new and more detailed information, our initial dry mass estimation was refined. By using the updated total dry mass value, preliminary calculation for the required propellant to have on board was made.

These calculations involved applying the Tsiolkovsky equation for both the chemical

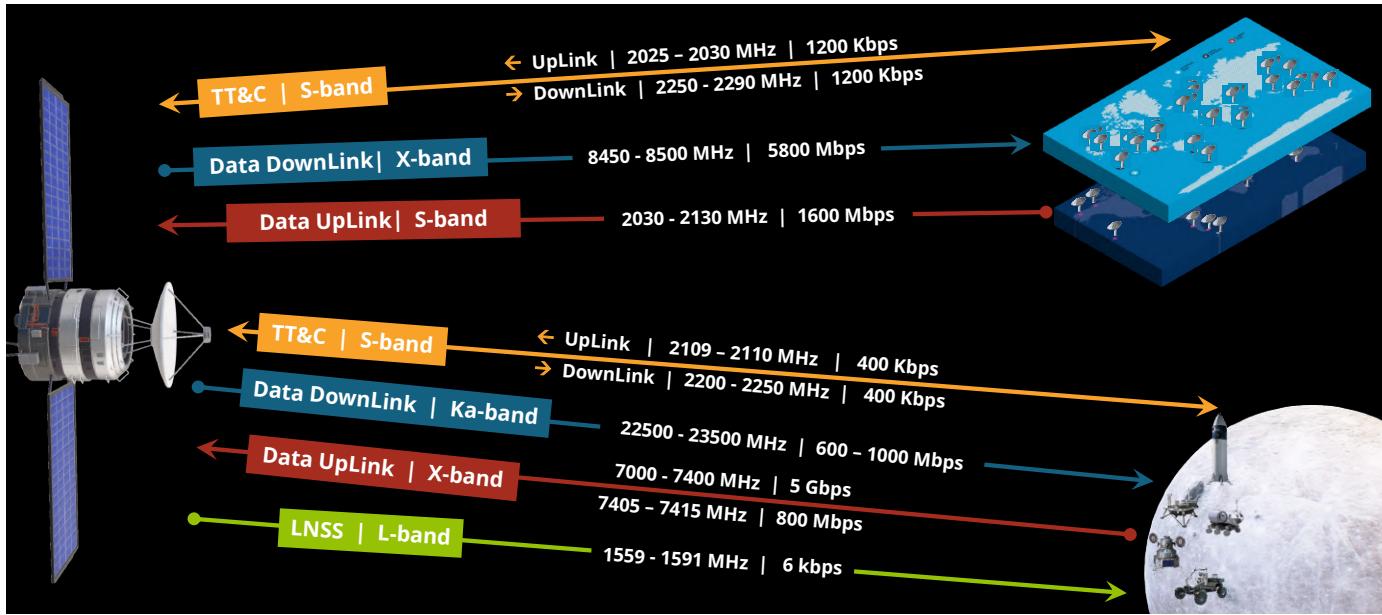


Figure 6.1: Link Budget results.

insertion burn and electrical stationkeeping activities (section 5.2)

Given the dry mass of the satellite, from 5.6 the appropriate engine was selected. Since it is US-made, an European engine with comparable performance was chosen. The selected thruster for the stationkeeping activities is the "PPSX00 Hall thruster" built by Safran.

By following this process, a preliminary mass budget was computed.

The result of the Mass Budget analysis is summarized in this table, which shows the estimated mass by each individual component or subsystem.

6. MISSION BUDGETS



UNIVERSITÀ
DI PISA

SUBSYSTEM	COMPONENTS	MASS [kg]
TELECOMMUNICATIONS	Antennas	4,1
	Amplifiers	0,5
	LNSS	25
	TOTAL	29,6
ATTITUDE & CONTROL	Actuators	12
	Reaction wheels desaturation	2,4
	Sensors	3,5
	TOTAL	17,9
STRUCTURE & MECHANISMS	Antenna pointing mechanisms	< 3
	Harness	14,3
	Frame	53,5
	TOTAL	67,8
THERMAL CONTROL	Passive	< 1
	Active	< 1
	TOTAL	< 2
POWER GENERATION & STORAGE	Solar panels	12
	Batteries	4,8
	Circuitry	5
	TOTAL	21,8
PROPELLION	Orbit injection	8,6
	Stationkeeping	14
	TOTAL	22,6
PROPELLANT	Insertion Maneuver	69,7
	Stationkeeping	6,1
	TOTAL	75,8

Figure 6.2: Architecture D Mass Budget estimation



6.3. Power Budget

A preliminary rough estimation of the architecture's power needs was obtained by multiplying a scaling factor with the power obtained via the link budget analysis. This factor was determined based on the technical data of existing satellites, particularly the Queqiao mission.

From this initial estimate, a very rough sizing of our solar arrays was performed. This was used to formulate a preliminary mass estimation.

Subsequently, the selection of all the required components and subsystems was updated, opting for components more suitable for our satellite class.

Finally, a more accurate evaluation of the power budget was obtained based technical data of commercial components.

The result of the Power Budget analysis is summarized in this table, which shows the estimated power required by each individual component or subsystem.



6. MISSION BUDGETS

SUBSYSTEM	COMPONENTS	POWER [W]
TELECOMMUNICATIONS	Antennas	47
	Amplifiers	14,6
	LNSS	121
TOTAL		182,6
ATTITUDE & CONTROL	Actuators	60
	Reaction wheels desaturation	0
	Sensors	11,3
TOTAL		71,3
STRUCTURE & MECHANISMS	Antenna pointing mechanisms	10
	Harness	0
	Frame	0
TOTAL		10
THERMAL CONTROL	Passive	0
	Active	40
	TOTAL	40
POWER GENERATION & STORAGE	Solar panels: Sada	100
	Batteries	0
	Circuitry	<30
TOTAL		130
PROPELLION	Orbit injection	0
	Stationkeeping	0
	TOTAL	0
PROPELLANT	Insertion Maneuver	0
	Stationkeeping	0
	TOTAL	0

Figure 6.3: Architecture D Power Budget estimation

6.4. Launcher Selection

There are some rockets capable of sending payloads into trans-lunar orbit:

- Ariane 64
- Long March 8
- Falcon 9
- Soyuz-2
- Vulcan Centaur VC6

Rocket Name	Payload up to GTO (Kg) ¹	Cost (Million Euro)	Remark
Ariane 64	11,500	77	
Long March 8	1500	70-80	Especially designed for SSO
Falcon 9	5,500	67	
Soyuz-2	3,250	70-80	Can't use due to sanctions
Vulcan Centaur VC6	11300	120	

Table 6.1:
Comparison Table [66]

After comparison, the most economical choice available is the Falcon 9 at present.

Parameters	values
Height	70m
Diameter	3.7m
Mass	549,054kg
Engine Name	Merlin

Table 6.2: Falcon 9 Overview

¹Since the total mass of the mission is almost 1000 kg and considering the addition of the cost of the inserting maneuver into the transfer orbit, this can be considered a rough estimate.

Parameters	values
Height	41.2m
Diameter	3.7m
Empty Mass	25,600 kg
Propellant Mass	395,700 kg
Propellant	LOX / RP-1
Thrust At Sea Level	8,227kN
Number of Engines	9
Thrust at sea level	7607kN
Thrust in vacuum	8227kN

Table 6.3: FIRST STAGE

Parameters	values
Height	13.8m
Diameter	3.7m
Empty Mass	3900 kg
Propellant Mass	92,670 kg
Propellant	LOX / RP-1
Number of Engines	1 Vacuum
Thrust	981kN
Specific Impulse (vacuum)	348 sec
Burn Time	397 Sec

Table 6.4: SECOND STAGE

Payload Fairing

Made of a carbon composite material, the fairing protects satellites on their way to orbit. The fairing is jettisoned approximately 3 minutes into flight, and SpaceX continues to recover fairing for reuse on future missions.



Parameters	values
Height	13.1m
Diameter	5.2m

Table 6.5: FALCON 9 Fairing

6.5. Risk Analysis

In order to perform the risk analysis, the major risk events were identified, considering only incidents that couldn't be eliminated during the design process and analyzing what effect they would have on the mission.

First of all, three scoring table were created to asses the risk level of a general event. The Table 6.6, in which severity of the event consequences on the mission was established, giving a score from A (minimum) to E (maximum) for each incident.

The Table 6.7, where the likelihood of occurrence of a certain event is assessed, assigning again a score from 1 (minimum) to 5 (maximum).

Finally, the Table 6.8, which is the union of the likelihood and severity tables. It establish how dangerous an event is, allocating a risk level from **most** (E5), so very likely to happen with catastrophic consequences, to **least** (A1), so improbable with negligible impacts.

Next, every possible event that could occur was identified, neglecting of course minimum risk event that has been easily eliminated during the design process.

All risks events, and corresponding level of severity and likelihood, were assigned considering that HERMES mission has an intrinsic triple redundancy, since only the failure of all four satellites would let to termination of overall mission.

The results of the Risk Analysis are presented in Fig. 6.4.

Score	Severity	Severity of Consequences
E	Catastrophic	Leads to termination of the mission
D	Critical	Satellites performances decrease > 75%
C	Major	Satellites performances decrease > 50%
B	Significant	Satellites performances decrease < 25%
A	Negligible	Minimal or no impact

Table 6.6: Severity of consequences of an event and the corresponding score.

SUPREMAE
DIGNITATIS

1343

UNIVERSITÀ
DI PISA

6. MISSION BUDGETS

Score	Likelihood	Likelihood of Occurrence
5	Maximum	Certain to occur
4	High	Will occur frequently, about 1 in 10 orbits
3	Medium	Will occur sometimes, about 1 in 100 orbits
2	Low	Will seldom occur, about 1 in 200 orbit
1	Minimum	Will almost never occur

Table 6.7: Probability of an event to occur and its relative score.

RISK LEVEL					
E	E1	E2	E3	E4	E5
D	D1	D2	D3	D4	D5
C	C1	C2	C3	C4	C5
B	B1	B2	B3	B4	B5
A	A1	A2	A3	A4	A5
	1	2	3	4	5

Table 6.8: Risk level matrix to identify the hazardousness of events.



Risks Event	Risk Level	Impact
Launch delay	A4	Delay in the orbit insertion
Single satellite overall failure	C2	Loosing of the LNSS feature - still operative
Propulsion system single failure	B2	The redundant thrusters module will be more sollicitated → will probably reduce its lifetime
Propulsion system double failure	C1	Will no longer be able to perform station keeping
Desaturation propulsion system failure	C3	Will no longer be able to perform desaturation of reaction wheel → loosing of the attitude control system
Principal telecommunications system failure	B3	Will reduce the maximum data rate - still operative
Micrometeoroids impact on the solar panels	B3	Will reduce the overall power generable → lower operation capacities
Batteries double failure	B2	Will reduce the station keeping propulsion performance
2-3 satellites overall failure	D1	Will drastically reduce the data transmission performances
All satellites overall failure	E1	Will led to termination of the entire mission

Figure 6.4: Risk analysis results, with individuated dangerous events with the assigned risk level and the corresponding impact on the mission.

6.6. Cost Estimation

This cost analysis has the only aim to evaluate the order of magnitude of the mission costs, since the study is at a very preliminary phase.

To do this evaluation, the comparison between the costs of similar missions would have been the most efficient method at this step of the study; however, the lack of analogue missions and of information about the few of them existing has influenced the possibility of using this approach.

For this reason, this estimation has been done considering the three main contributes to the costs:

- development and building of the satellites,
- launch,
- ground segment activities for all the last of the mission.

The cost of the satellites has been estimated around 100M€ per satellite.

This estimation has been made using two analogy methods: one with the cost of the NASA SMEX (Small-Explorer) satellites, and another one with the standard cost of GEO satellites.

Since the operative life of the SMEX missions is comparative to the one of HERMES and since their weight is also similar, a proportion has been made considering 50M€ every 150 kg [67], plus an additional 50% margin, in order to consider the complexity and the innovation of the mission.

On the other hand, the costs of GEO satellites, can vary between 100M€ and 400M€: they have a similar design and building time (3 to 5 years) and possible similar purposes (Communication and Navigation services), but they have a mass that can be in the order of few tons. [68]

The lower limit cost of a GEO satellite matches with the previous estimation and it has been considered suitable as an order of magnitude for the cost of one satellite.

The estimated cost of a launch with the Falcon 9 is 67M€ for a launch of 5500 kg up to GTO; since the total mass of the mission is almost 1000 kg and considering the addition of the cost of the inserting maneuver into the transfer orbit, this can be considered a rough estimate.[69]

About the ground segment cost, there is no a standard evaluation that can be made, since it depends a lot on the specific service that is desired.

The value reported here wants to be very generic. The considerations that have been made take into account the total amount of data that has to be handled, the number of ground stations needed, the great connection time and the complexity and innovation



of the mission.

Considering some examples, including Galileo [70] and an estimation made by the company LeafSpace [71], an annual cost of 50M€ has been considered for ground segment operations.

However, that could be widely variable depending on the actual contract that will be established with the selected provider; a more detailed estimation is left to the next phase-A study.

To summarize, **the expected cost of the mission, up to the launch, is around 500M€**, including the development and building of the satellites and the launch itself. Then, **another 500M€ is expected to be the cost of the ground segment operations for the lasting of ten years**, settling the **cost of the entire mission at 1B€**.



7. Phase-A Mission Expansion

To better understand how to manage the future work in the following mission stage, a detailed *Work Breakdown Structure* is presented, Fig. 7.1, commented and explained in Sec. 7.1, followed by the related *Gantt Chart*, Sec. ??.

7.1. Work Breakdown Structure

The suggested work to be arranged during Phase-A is presented as follow, considering the major mission sub-unit, in particular those which has still some design uncertainties, not removable during a Pre Phase-A study.

First of all, the **Mission Analysis** subunit where a numerical optimization analysis for the trajectories is mandatory before providing a detailed spacecraft maneuvers selection.

Because of time limitation of a Pre Phase-A, only a preliminary analysis based on some assumption has been made, reducing the three body problem into two separate Keplerian problems where only moon orbital plane trajectories has been considered.

Moreover an end of life disposal study for HERMES orbiting satellites is mandatory, particularly important considering the future of moon exploration which is tending more and more to the far-side.

Second, the **Ground Segment**, for which a mission control from earth has to be expanded in order to properly define contracts with possible vendor. This would allow a better understanding of the possible link to be installed with earth and so a precise refinement of the link budget that would led to an optimization of the telecommunications system.

The work would be followed by expansion of the mission control definition from the ground segment which is imperative for the better understanding of the full potential communications system achievable. Also contracts with potential vendors are pivotal to delineate expected outcomes and to ensure the earth support won't change.

Regarding the **BUS** subsystem, meticulous refinement of the link budget will be followed by optimization of the telecommunications system and its electronic components. This won't change much the telecommunications system since accurate design was performed during the prem phase-A.

The propulsion module within the BUS subunit necessitates attention, particularly in

the realm of modeling sloshing phenomena inside fuel tanks and developing system interfaces with spacecraft. At this stage of the mission only preliminary analysis for the power consumption and supply were carried out.

The Attitude Control System assumes paramount importance in maintaining antenna pointing accuracy, in particular considering the selected halo orbit around EML2. Consequently, refinement of perturbation models such as solar pressure, aerodynamic forces and more is essential, which were initially limited to gravitational forces only. Alongside with that, a precise inertial matrix of the satellites would help to develop an accurate desaturation model, which would lead to the precise selection and disposition of the small thrusters alongside with other possible momentum dumping systems.

The Power and Thermal subunits require meticulous design development, given the absence of crucial thermal and power flux information during this phase. This will aid in final power consumption estimations.

Enhanced structural design would optimize components arrangement within the satellites, mechanical solutions, a more accurate CAD design and, as said, a refined inertial matrix of the satellites.

Subsequent to these considerations, **Launcher** final selection warrants review, particularly in light of potential availability of alternative launchers, since the selected one is vastly exceeding current requirements having much more space available inside the fairing.

Furthermore, final **Cost Estimation** will be iterated by subsystem component selection and associated pricing information.

Conclusively, a comprehensive **Risk Assessment** is imperative, incorporating identification of new hazardous events and corresponding mitigation measures that could lead to new components and design.

In the next Section follows the detailed Gnatt Chart for the work flow expected during the Phase A of the mission.

7. PHASE-A MISSION EXPANSION

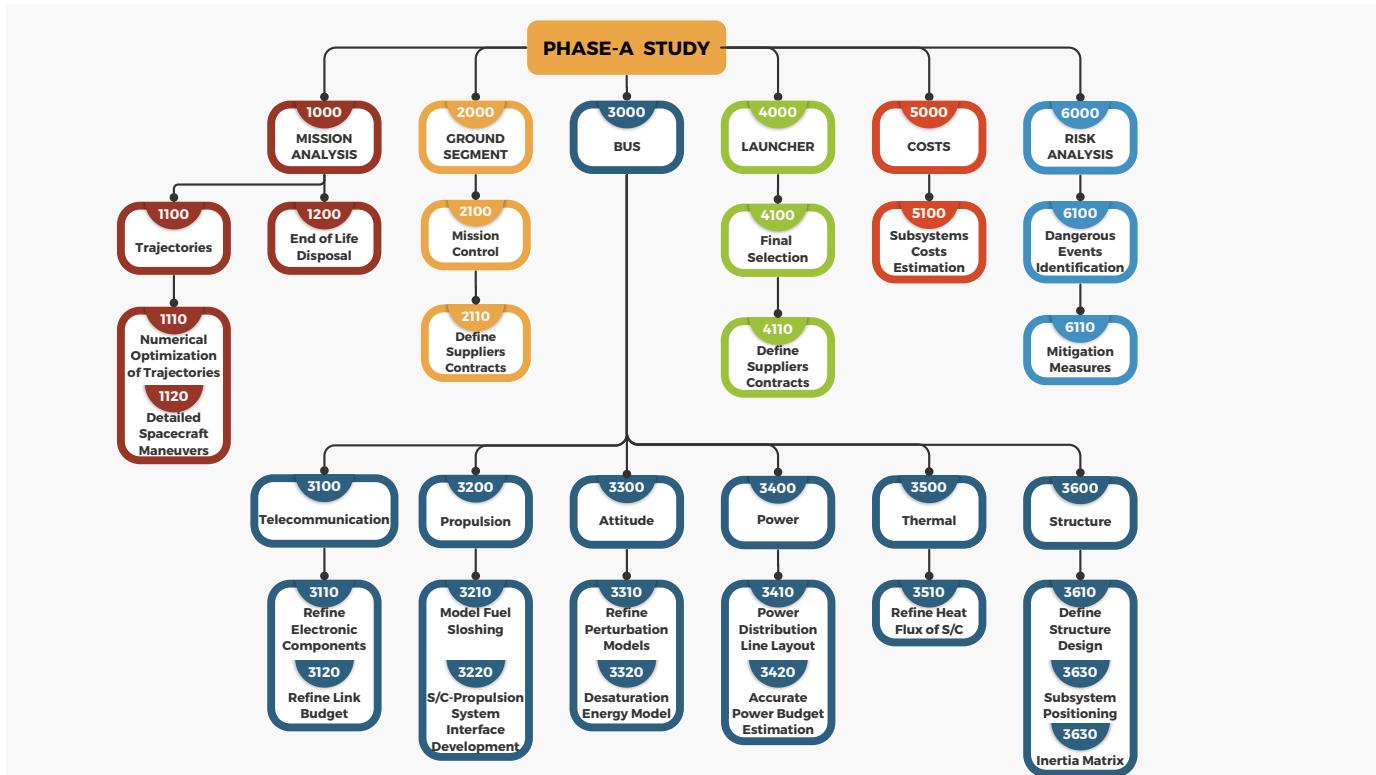
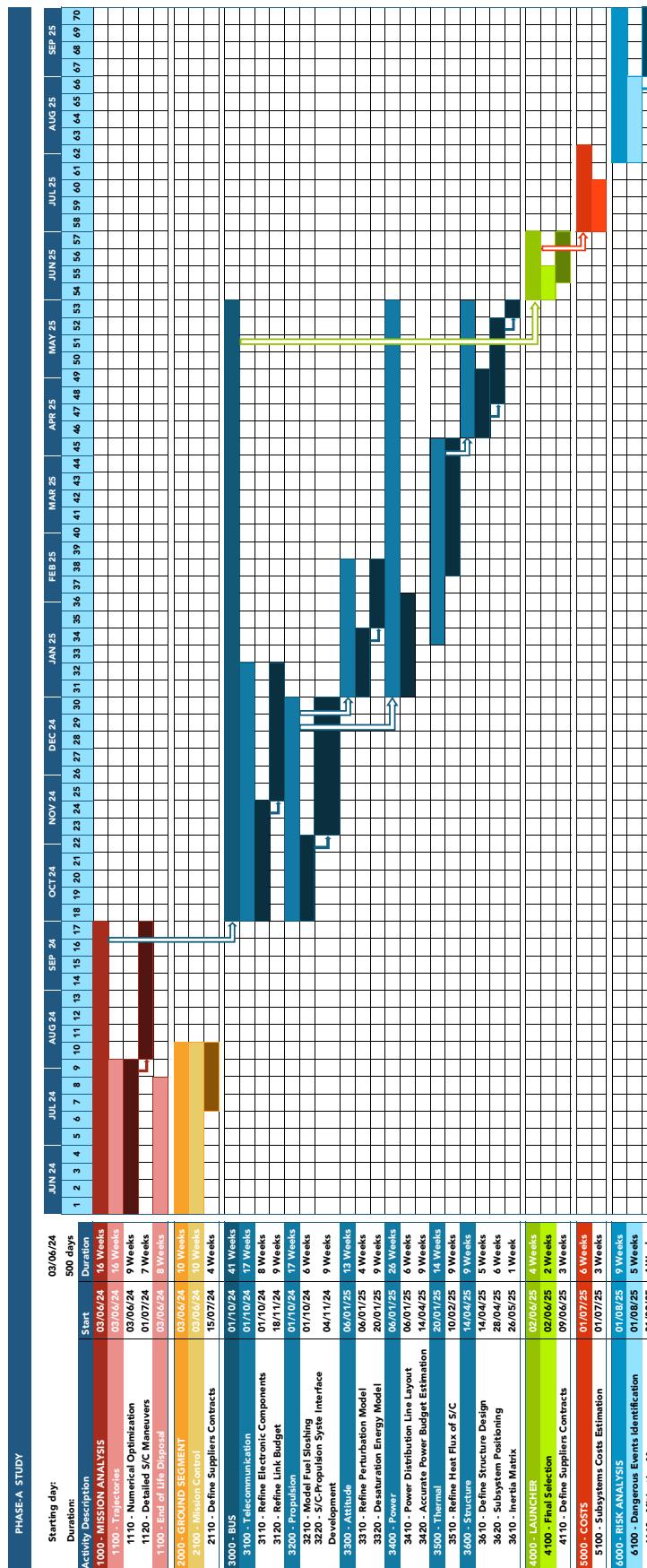


Figure 7.1: Overall Work Breakdown Structure for the Phase-A mission expansion.

7.2. Gantt Chart



HERMES - Pre Phase-A Mission Report

Group 1



8. Conclusions

During the Pre-Phase-A of the mission aimed at establishing communication infrastructure between Earth and the lunar surface, four satellites have been successfully designed to be deployed into an halo orbit around the EML2 Lagrange. These satellites serve as vital components in facilitating communication between various applications on the Moon, forming an essential part of lunar human infrastructure.

The study conducted highlights the significant scientific potential of the lunar far side. Its uncharted terrain presents promising opportunities for scientific research, resource extraction, and astronomical observation. Establishing communication capabilities on this side of the Moon is imperative for future endeavors, given its obscured visibility from Earth.

However, challenges have emerged. The selected halo orbit, while providing stability and a good trade off between coverage and distance, offers limited fast communication latitude range of the lunar surface since its low Z-Amplitude. Nevertheless, an higher Z-Amplitude orbit wouldn't be optimal to communicate with the central zones of the lunar surface.

Additionally, initial considerations of optical communication were abandoned due to complexity and insufficient information for link budget calculations at this stage of the mission, even though the technology could have provided faster and stronger communications link with the moon.

Communication issues persist, particularly regarding low data rate transmissions for smaller lunar applications with inadequate antenna size or gain. Addressing these challenges necessitates refining also attitude control system, to consider other complex perturbations and optimize antenna pointing and gain during the Phase-A study.

Furthermore, as said in Sec. 7, opportunities exist to refine propulsion strategies to minimize station-keeping requirements and a more accurate study for the link budget is essential, considering the evolving landscape of optical communication technology, which may become viable in future mission phases.

Anyway, this mission design demonstrates scalability, allowing for potential expansion of satellite deployment within the same orbit or into alternative orbits. This scalability enhances coverage across the lunar surface, aligning with emerging initiatives for lunar GPS systems, internet networks, and support infrastructure for future lunar bases.



In particular, the expansion of the mission with other four or more satellites in a northern halo orbit could enable, along with HERMES spacecraft, the possibility of a fast data transfer and tracking system of the south pole, particularly interesting for scientific research and already targeted by many space agencies and companies.

As shown in Sec. 2, there are already many missions planned for the very next future, focused on the lunar far side thus showing the increasing interest on this region, and many of them could take advantage of the design here proposed.

Since the greater requests and studies for greater infrastructures and radio-astronomy facilities that require huge amounts of data transmission, a concept like the proposed one will be more and more appealing, driving investors and space agencies into it, Fig. 8.1.

In conclusion, the preliminary mission represents a significant advancement in establishing robust communication infrastructure between Earth and the lunar surface. While challenges remain, addressing these issues will be crucial in realizing the full potential of lunar exploration and exploitation in the years ahead.

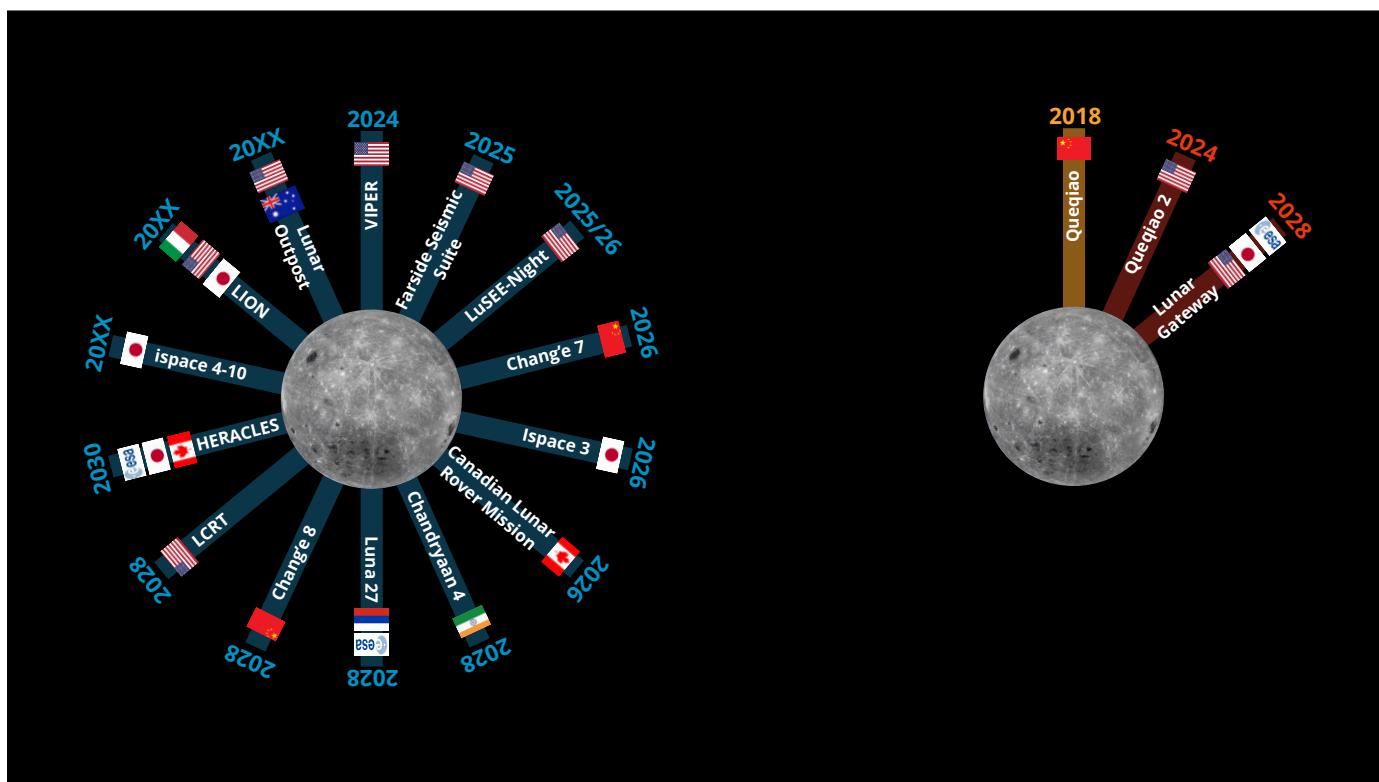


Figure 8.1: Many of the planned mission on the far side in the very next future (right), the only programs to support such missions (left).



A. Nomenclature

S/C	Spacecraft
ADCS	Attitude Determinetion and Control System
\dot{Q}_i	thermal power generated internally to the spacecraft
\dot{Q}_s	thermal power input from the Sun
\dot{Q}_{MR}	thermal power input from albedo radiation from Moon
\dot{Q}_{SM}	thermal power exchange between spacecraft and Moon
\dot{Q}_r	thermal power radiated to space
α	spacecraft absorptivity
A_{\perp}	projected area perpendicular to the sunlight
I_{sun}	solar constant
σ	Stefan-Boltzmann constant
ϵ	spacecraft emissivity
$F_{S,S}$	view factor, spacecraft to space
A_s	spacecraft surface area
T_{SC}	spacecraft temperature
T_{space}	space temperature (=3K)
$F_{S,M}$	view factor, spacecraft to Moon
T_M	Moon black body temperature
a	albedo
$F_{S,SM}$	view factor, spacecraft to sunlight on Moon
DL	Downlink
UL	Uplink
TT&C	Telemetry and Commands
COTS	Commercial Off The Shelf
λ	Signal wavelength
r	Spacecraft distance
r_E	Distance from earth
r_M	Distance from earth
B	Bandwidth
C	Channel Capacity
$\frac{S}{N}$	Signal to Noise Ratio
T_{sys}	Receiver Antenna Temperature



A. NOMENCLATURE

k	Boltzmann Constant ($k = 1.38064852 \times 10^{-23} m^2 kgs^{-2} K^{-1}$)
G_T	Transmitter Gain
G_R	Receiver Gain
P_T	Transmitter Power
P_R	Receiver Power

B. Appendix

B.1. Mission Profile

B.1.1. First Order Method

This initial approach relies on the strong assumption of approximating the Sphere of Influence (SOI) to a point in order to greatly simplify the transfer dynamics. This method provides us with initial comparative measurements with which we can compare more refined computational methods.

Additional Assumptions

- SOI approximated to a point
- Smooth trajectory angles

Problem Description

First Keplerian Problem: Orbital transfer with an Hohmann with the primary focus on Earth and perigee coinciding with the Moon (Fig. B.1).

Second Keplerian Problem: Parabolic orbit with focus on the Moon intersecting points P1 and P2 (Fig. B.2).

After being injected by the launcher into the Hohmann transfer trajectory, once the spacecraft reaches apogee, it is decelerated until it reaches, in magnitude, the same velocity as the Moon. At this point, it enters the Moon's Sphere of Influence, and in the Moon's reference frame, the satellite has a velocity of zero in magnitude. The trajectory angle must be set congruent to achieving a parabola that intersects P1 and P2. A second impulse Δv_2 allows us to transfer the spacecraft onto the Halo orbit.

Following this approach, we have derived a theoretical Δv requirement for the spacecraft of 0.525 km/s.

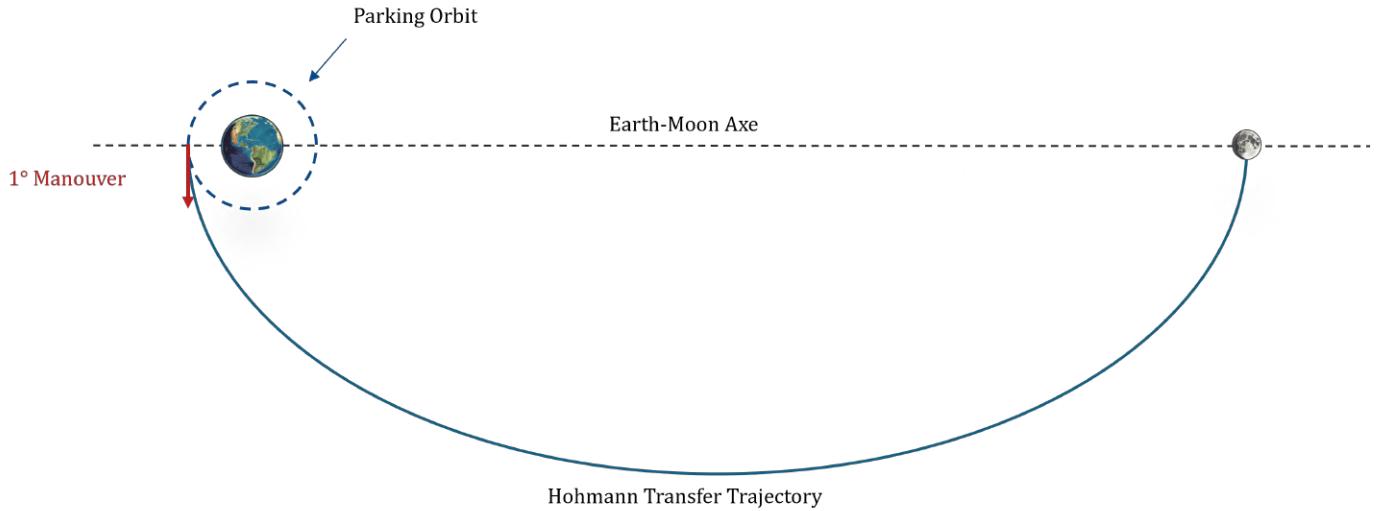


Figure B.1: First Keplerian problem

B.1.2. Introduction to the "2nd order" approaches

The analysis done with the previous model had strong simplifications, both physical and geometric. In fact, considering the lunar soi as a point coincident with the moon itself is a very crude approximation.

On the other hand, this kind of approximation is very fitting when we consider interplanetary transfers. in those cases, the dimensions of the soi of the individual planets are negligible compared to the distances between the primary attractor and the planet involved.

For example, let's consider the geometry of an Earth-Mars transfer: by means of the Laplace approximation we can calculate the dimensions of the spheres of influence of the two planets:

$$R_{soi} \sim a \cdot \frac{m^{\frac{2}{5}}}{M}$$

Where a is the semimajor axis of the smaller object's orbit around the larger body, m is the smaller body mass and M is the larger body mass.

In this case we have that:

$$R_{soi_{Earth}} \sim a_{Earth} \cdot \left(\frac{m_{Earth}}{M_{Sun}} \right)^{\frac{2}{5}} \sim 0.92 \cdot 10^6 \text{ km}$$

Similarly, we compute the radius of Mars's soi:

$$R_{soi_{Mars}} \sim 0.58 \cdot 10^6 \text{ km}$$

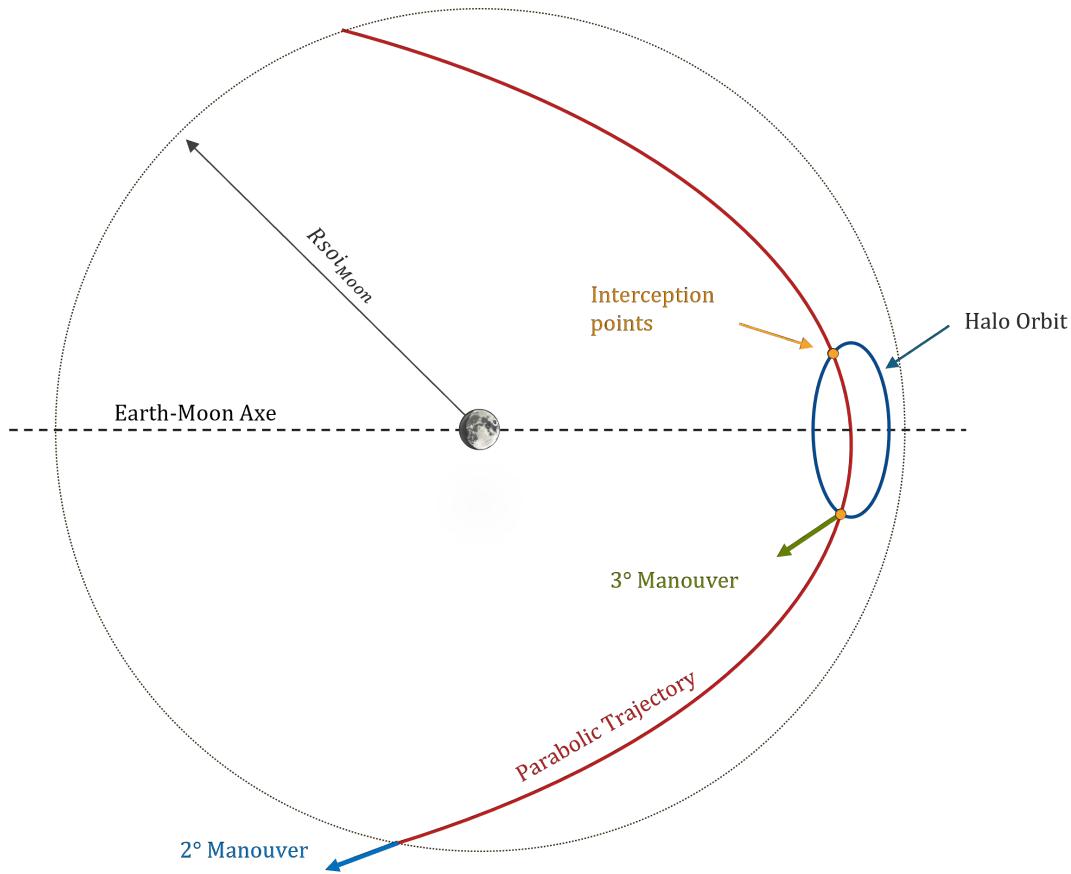


Figure B.2: Second Keplerian problem

Comparing these data with the average distances of individual planets from the sun we see that:

$$\frac{R_{SOI_{Earth}}}{a_{Earth}}, \frac{R_{SOI_{Mars}}}{a_{Mars}} \ll 0.01$$

From these considerations we can understand how the size individual spheres of influence are negligible compared to the distances of the planets from the primary attractor. "From the Sun" point of view, the soi of the individual planets are punctiform.

Repeating the same calculations for the Earth-Moon system we obtain that:

$$\frac{R_{SOI_{Moon}}}{a_{Moon}} \sim 0.2$$

Where $R_{SOI_{Moon}} \sim 0.66 \cdot 10^5 \text{ km}$ and $a_{Moon} \sim 3.84 \cdot 10^5 \text{ km}$.

As we can see, considering the lunar sphere of influence as point-like represents a strong approximation, which induces a major error in the calculations. In addition,



B. APPENDIX

crushing the lunar soi to a point causes the loss of the "entry angle" information within the moon's sphere of influence. Since our mission involves reaching a specific point in space, such information is of critical importance and will be used to discriminate the different trajectories studied.



B.1.3. Second Order Approaches

Analytic "Geocentric Bi-Elliptic" Approach

Starting from the same intersection point as before, a parabolic trajectory within the Moon system was computed, followed by the calculation of an elliptical trajectory in the Earth's reference frame necessary to establish connection with the parabolic path.

Additional Assumption

The velocity direction of the spacecraft (S/C) on the orbit intersecting point P1 is parallel to that of the Halo at point P1.

Problem Description

Based on this analytical insight, the orbits enabling the attainment of the specified velocity vector conditions were reconstructed (B.1.3). These orbits consist of a selenocentric hyperbolic trajectory extending up to the Sphere of Influence (SOI), and a geocentric bi-elliptical trajectory.(Fig.B.3)

It should be noted that, as illustrated in the graph, the two maneuvers are not executed on the Earth-Moon orbital plane but on another plane that can be calculated.

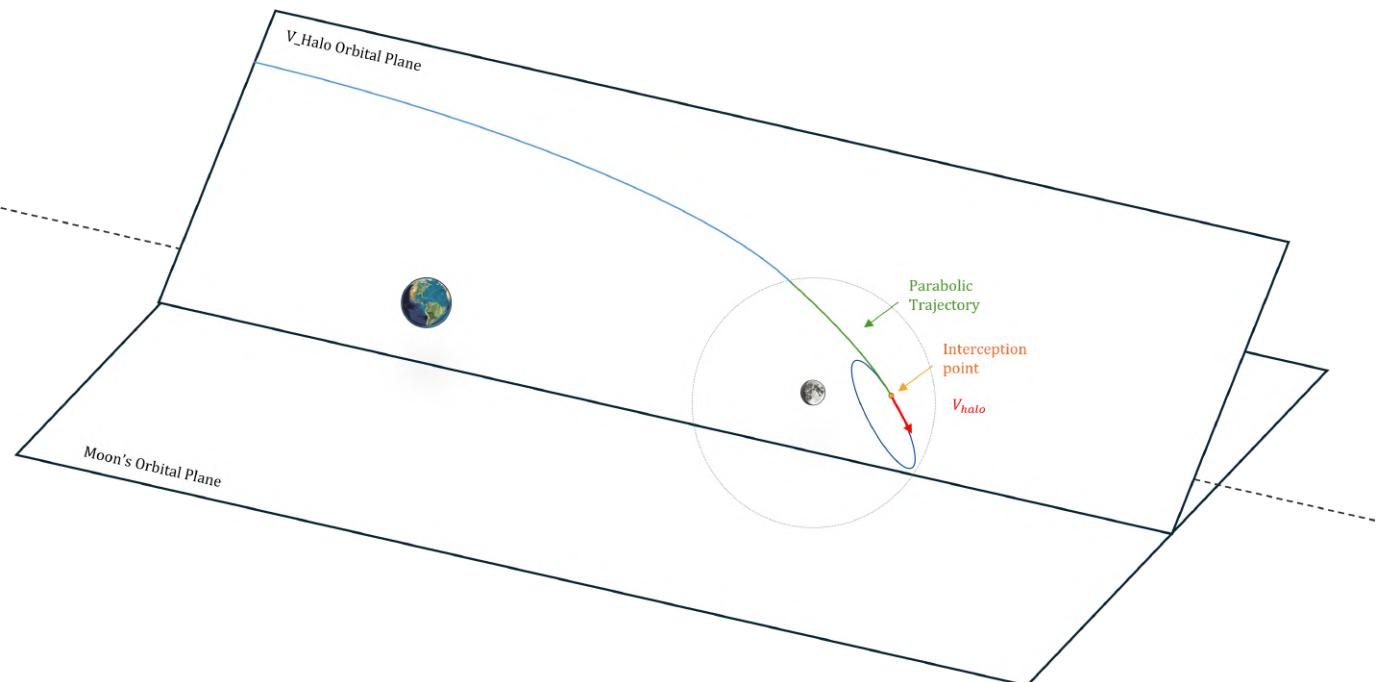


Figure B.3: Second order analytical approach

Limitation

From the calculations B.1.3 and the Fig B.4, it is evident that the apogee radius of the second transfer ellipse extends beyond the Sphere of Influence of the Earth. Thus, the fundamental assumption of using the two Keplerian problems for the calculation is no longer valid¹.

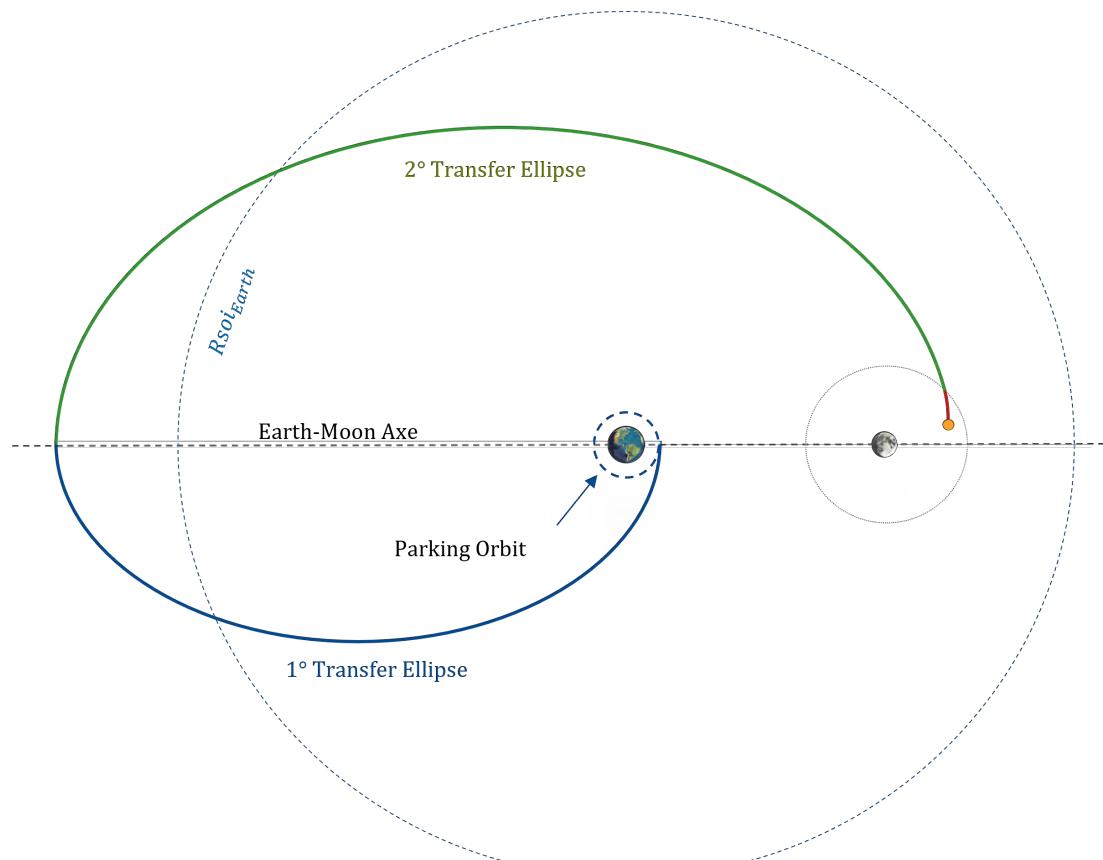


Figure B.4: Second order approach Limits

¹When exiting the Sphere of Influence of the Earth, it becomes necessary to also consider the segment of the orbit subjected to solar attraction



B.1.4. Restricted Second Order Methods

Given the difficulties associated with studying a 3-dimensional model, a decision was made to refine the results by exclusively focusing on trajectories confined to the Lunar orbital plane.

"Direct Approach"

The complete transfer geometry can be derived given the following initial conditions:

- The Earth's parking orbit altitude r_0
- The trans-lunar injection velocity from Earth v_0
- The departure flight path angle from Earth ϕ_0
- The entry angle into the Lunar soi λ_0

An algorithm was developed to identify the optimal trajectories. While maintaining constant r_0 ¹ and ϕ_0 ², v_0 and λ_0 were varied at each iteration. Specifically, a range of departure velocities from 10.8 km/s to 11.1 km/s was considered, with increments of 0.01 km/s. For each velocity value, attempts were made to enter the lunar sphere of influence with entry angles ranging from 0° to 180°, in increments of 0.01°.

For each angle-velocity combination tried, the algorithm varies the phasing angle of the translunar injection manouever looking for the optimal entry angle (in the lunar soi) in order to obtain that the hyperbolic selenocentric trajectory passes through one of the two intersection points of the halo orbit. Not all angle-velocity combinations achieve a correct transfer.

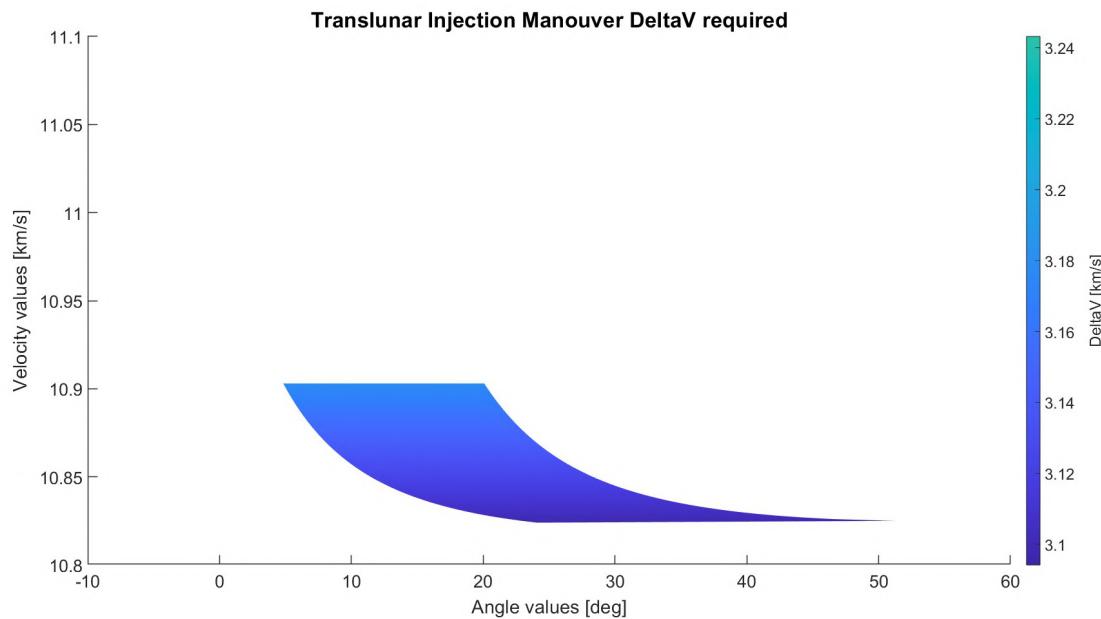
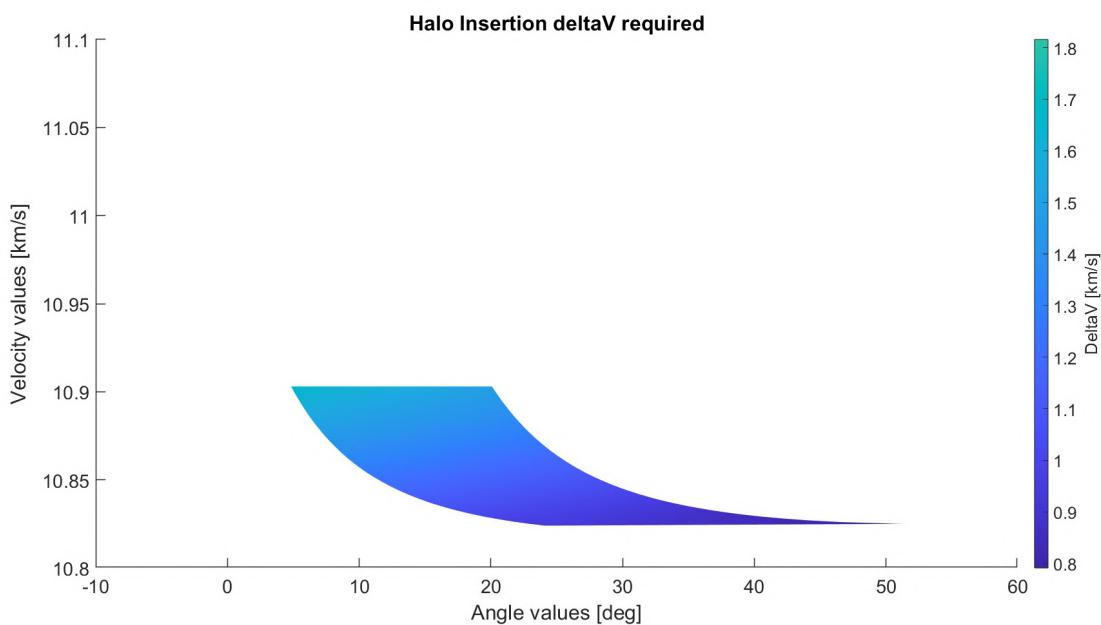
In fact, depending on the input values, the required λ_0 angle may require a more energetic orbit than the one available with the velocity v_0 considered.

All cases that were not physically possible were, therefore, discarded. The ΔV values were then calculated for each combination that produced a correct intersection with the halo orbit.

In the subsequent figures, the plotted data illustrate the angle-velocity combinations that resulted in trajectories intersecting the halo orbit (Fig. B.5 - B.6 - B.7). From these results, it emerges how with this method a velocity v_0 between ~ 10.82 - 10.9 km/s is required in order to produce a trajectory intersecting the desired points. As the velocity v_0 increases, the "slice" of entry angles (λ_0) that produce a proper intersection tends to decrease, moving from the initial 25°-50° to 7°-22°.

¹Calculations done by varying also r_0 showed that this variation only affects the deltaV required for the trans-lunar injection manouever. Consequently, we fixed its value at 300 km, considering it a reasonable choice.

²By assuming in the calculations that the trans-lunar injection maneuver is executed tangentially to the parking orbit, we fixed the value of ϕ_0 to 0°.

Figure B.5: "Direct" Translunar Injection ΔV Figure B.6: "Direct" Halo Orbit Insertion ΔV

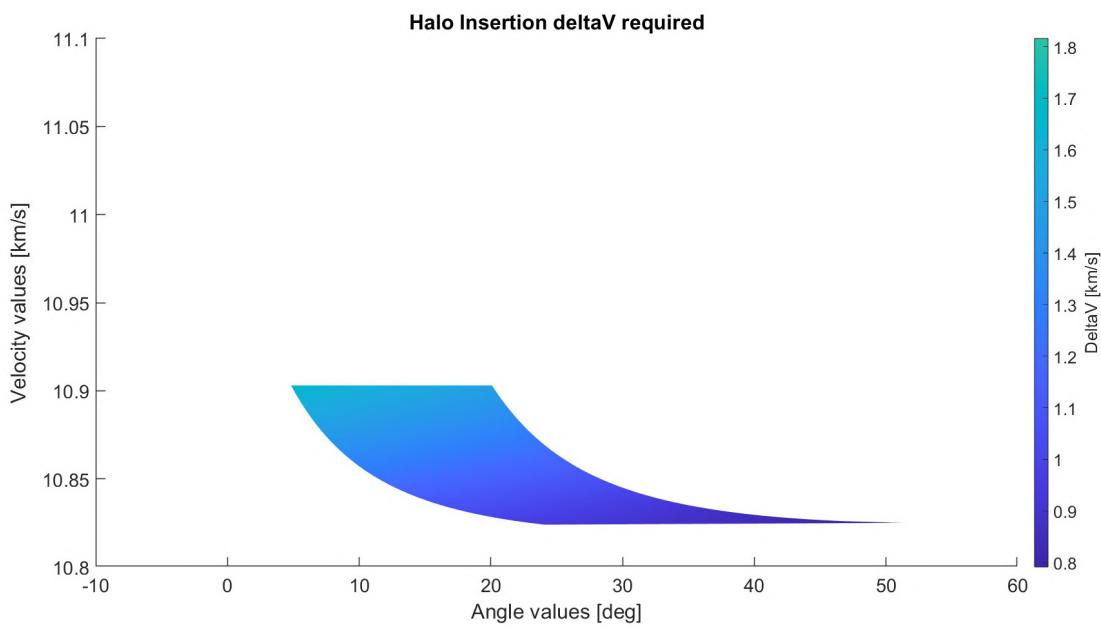


Figure B.7: "Direct" Spacecraft Propulsion System ΔV

"Selenocentri Bi-Impulse Approach"

The calculations were carried out by using the same algorithm as the "Direct" Approach, adapting it to handle the different selenocentric phase. Fig. B.8 -B.9 - B.10.

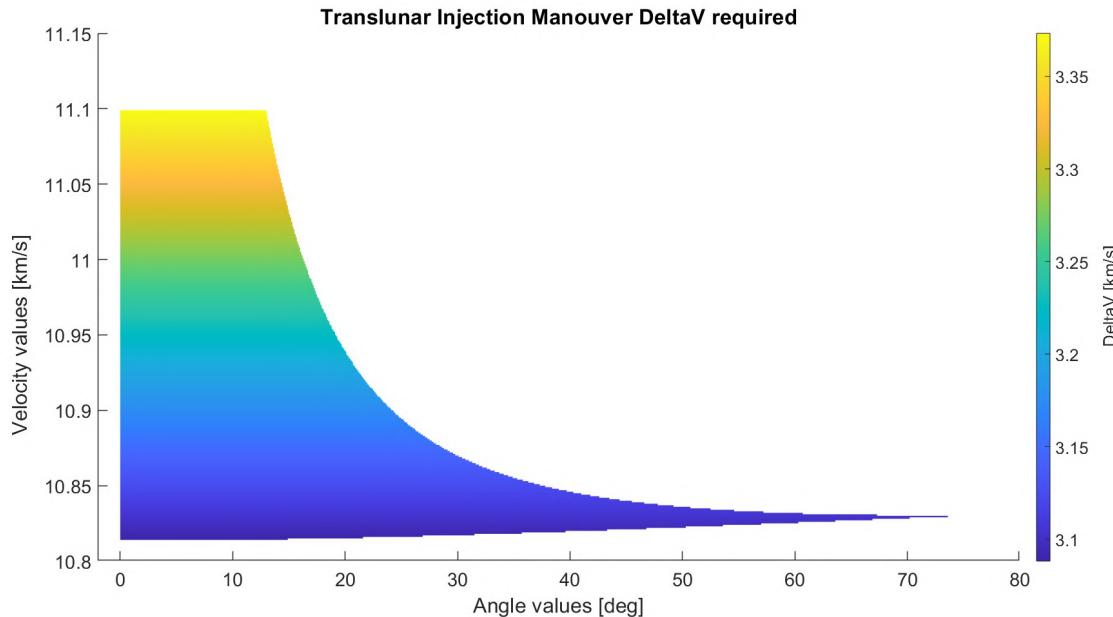


Figure B.8: "Selenocentric Bi-Impulse" Translunar Injection ΔV

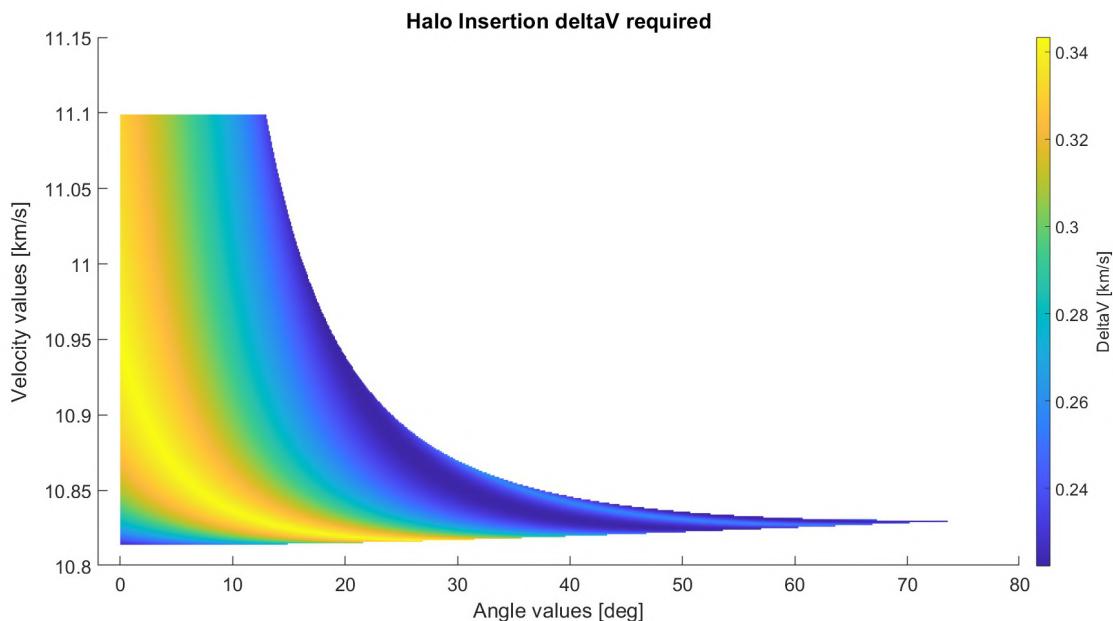


Figure B.9: "Selenocentric Bi-Impulse" Halo Orbit Injection ΔV

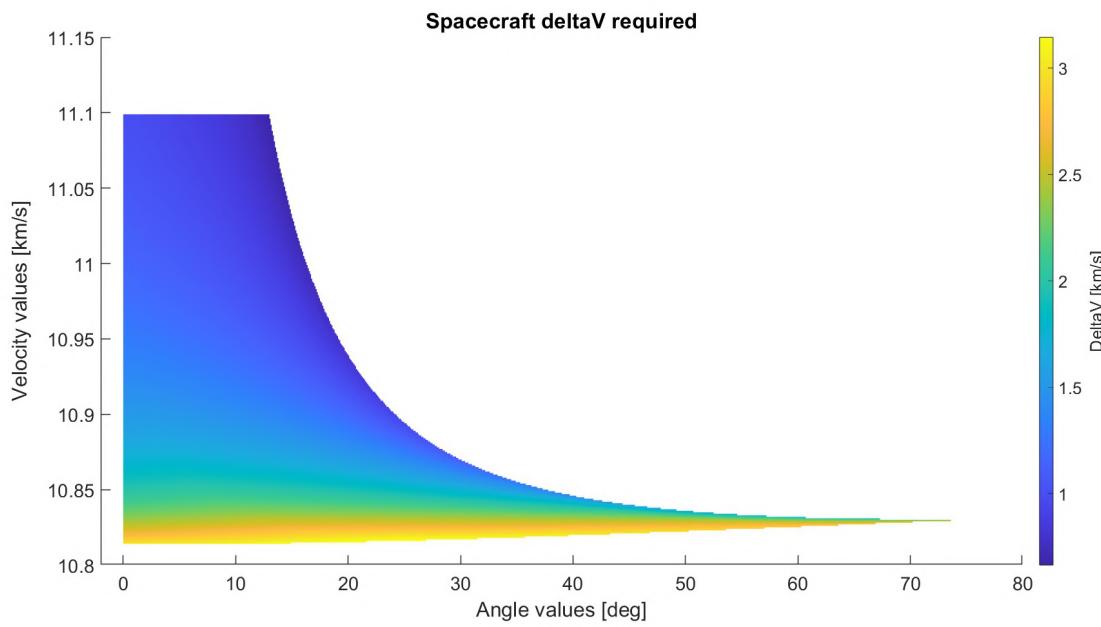


Figure B.10: "Selenocentric Bi-Impulse" Spacecraft Propulsion System ΔV

Geocentric Bi-Elliptic Approach

This approach holds the same initial assumptions as the previous methods, but this time the algorithm used for the calculations had to be completely redesigned to manage this new geocentric transfer phase.

In contrast to previous method, in this approach, the first ellipse remains fixed, while the spacecraft's velocity after the first maneuver is gradually changed between 0.4km/s and 0.7km/s with increments of 0.01km/s .

For each velocity v , entrance into the lunar soi was attempted with λ angle varying from 0° to 180° (contrary to the previous approaches, this time the angle is assumed to be positive counterclockwise and not clockwise) with increments of 0.1° .

For every angle-velocity combination tested, the algorithm look for the optimal entry angle to ensure that the hyperbolic selenocentric trajectory intersects one of the two intersection points of the halo orbit.

However, it's important to note that not all angle-velocity combinations result in a successful transfer.

Hence, the spacecraft propulsion system will execute two maneuvers: the first to insert itself into the second transfer ellipse, and the second to match the velocity of the halo orbit at the intersection point.

Only the five solutions that showed the lowest propellant consumption are shown in the figure Fig. B.11¹.

In this table, the ΔV results obtained from the five orbits that exhibited the lowest propellant consumption are shown Tab. 4.4, where the "Spacecraft ΔV " values is the sum of the two maneuvers done.

¹The graphs showing the launcher and halo insertion ΔV are not showed here because they didn't bring any more information.

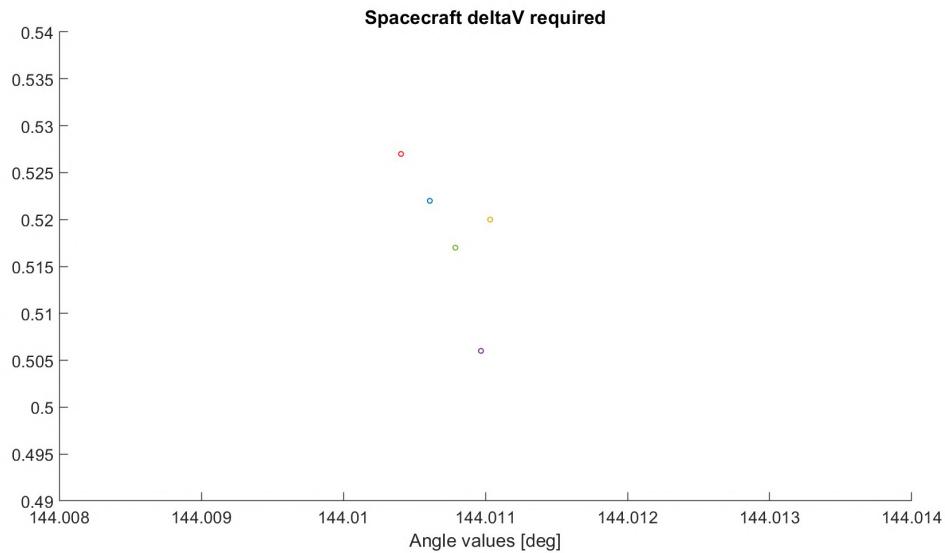


Figure B.11: "Geocentric Bi-Elliptic" working angle-velocity combination

B.2. 3 Body Dynamics

The previously used methods are all based on Keplerian dynamics, taking advantage of the patched conics approximation in order to connect the orbits between the different spheres of influence. However, for more precise results, it is essential to consider the gravitational attraction of both Earth and Moon throughout the mission, that is, by analyzing the dynamics of the three-body problem.

By studying the Circular Restricted Three-Body Problem (CR3BP), we observed the existence of families of trajectories passing close to collinear Lagrangian points. These trajectories form envelopes known as manifolds, which can be conceptualized as actual "rails" guiding the movement of spacecraft.

Originating from the collinear points (L1, L2, and L3), a spacecraft entering one of these tubular structures will subsequently be "transported" towards these points. Entry into these manifolds requires the spacecraft to meet precise position and velocity conditions.

According to the "Gordon Thesis," it is feasible to enter a halo orbit around L2 by leveraging these manifolds. The paper examines a scenario where entry into the manifold is achieved through a suitable maneuver near the lunar surface.

The paper investigates how the Δv required for manifold entry varies with the following parameters:

- Amplitude of the halo orbit's z-component
- Altitude of the circular departure orbit from Earth
- Flight path angle between the manifold trajectory and the Earth-Moon transfer trajectory

B. APPENDIX

From the analysis, several conclusions can be drawn:

- The altitude of the starting circular orbit has a negligible effect on the required Δ_v for manifold entry.
- A flight path angle of 60° appears to be the optimal choice as the z-amplitude of the halo orbit increases.
- Variations in the z-amplitude of the halo orbit result in the most significant changes in the required Δ_v for the insertion maneuver.

Based on these results (fig B.12) we can expect that the minimum Δ_v required to entry the manifold (which will subsequently guide us to our desired halo orbit with a z-amplitude of 25,000 km), should be approximately 0.4 km/s.

In order to obtain a more accurate prediction, we selected multiple points along the blue line (flight path angle 60°), to establish an interpolating function charactering its trend. Subsequently, we evaluate it with our desired z-amplitude to predict the Δ_v required for the manifold insertion Tab. B.1.

Z-Amplitude [km]	Δ_v [km/s]
1000	0.25
2000	0.27
4000	0.30
6000	0.37
7000	0.40
8000	0.39
9000	0.37
10000	0.40

Table B.1: Selected point for the interpolation

As observed from the graph fig B.14, the estimated Δ_v value is approximately 0.46km/s .

This approach was disregarded due to various intrinsic issues:

Observing the curve trend Fig. B.12 representing the results obtained between 1000km and 10000km , we hypothesized that could follow a logarithmic pattern. Consequently, we used a logarithmic function to interpolate the points we took on tabB.1.

However, the error produced in predicting the required Δ_v value is heavily reliant on the choice of interpolating function, as the amplitude of our halo orbit exceeds twice the maximum value found in the literature.

Hence, the obtained result was deemed unreliable for use in our considerations.

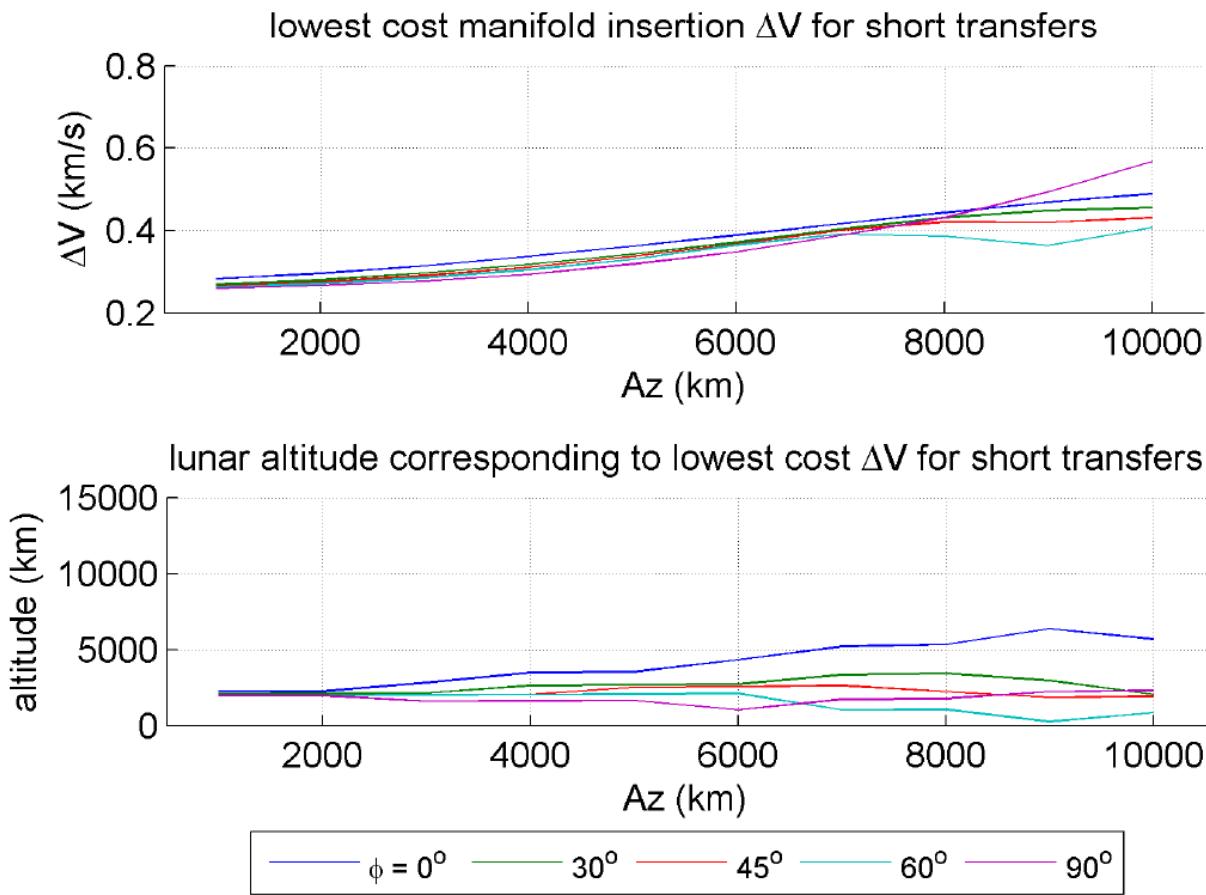


Figure B.12: Upper plot includes minimum short transfer manifold insertion cost ΔV versus halo orbit Az-amplitude for various insertion angles, ϕ_0 . Lower plot includes the corresponding lunar altitude at which the minimum ΔV occurs [72]

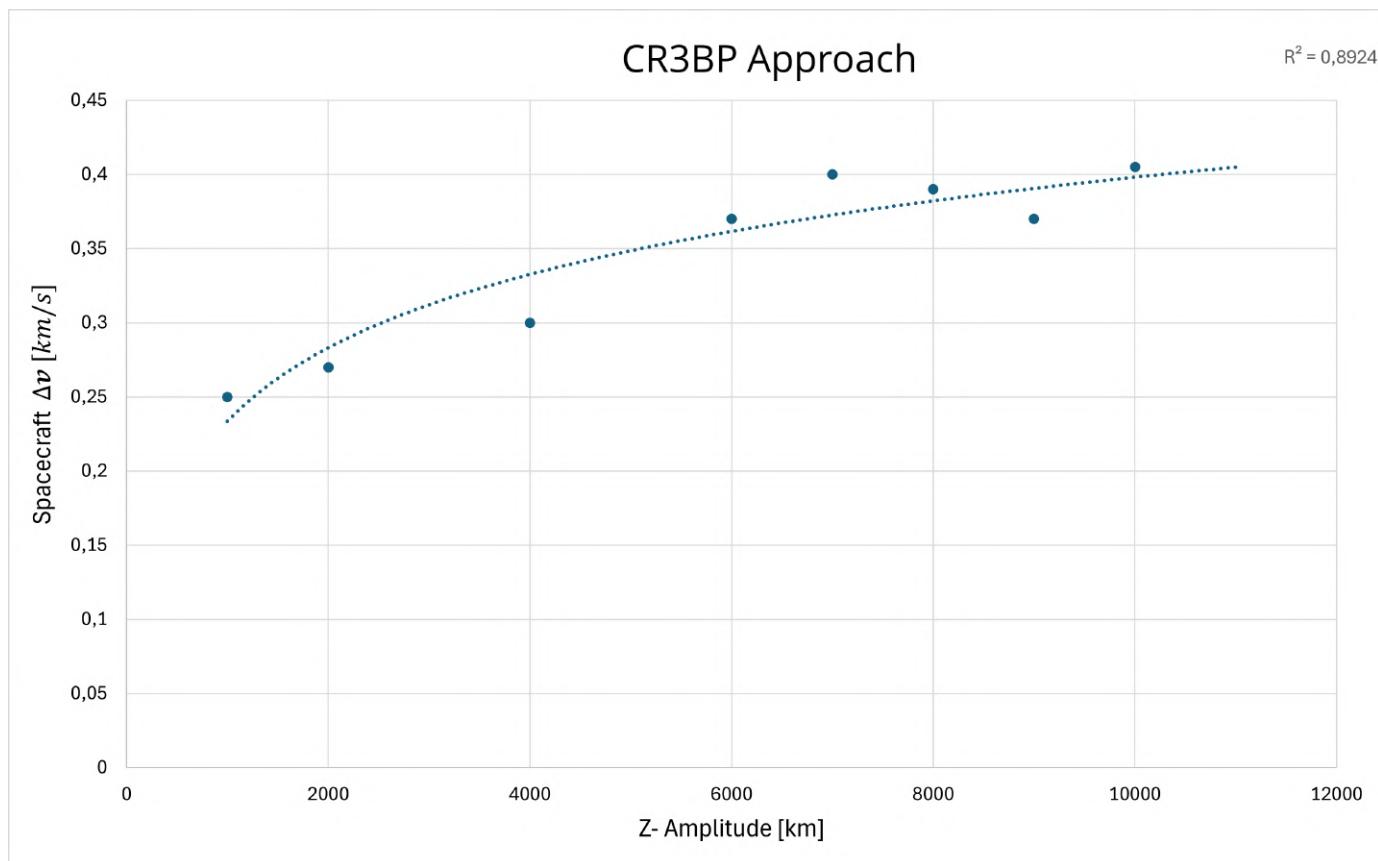


Figure B.13: Interpolation curve

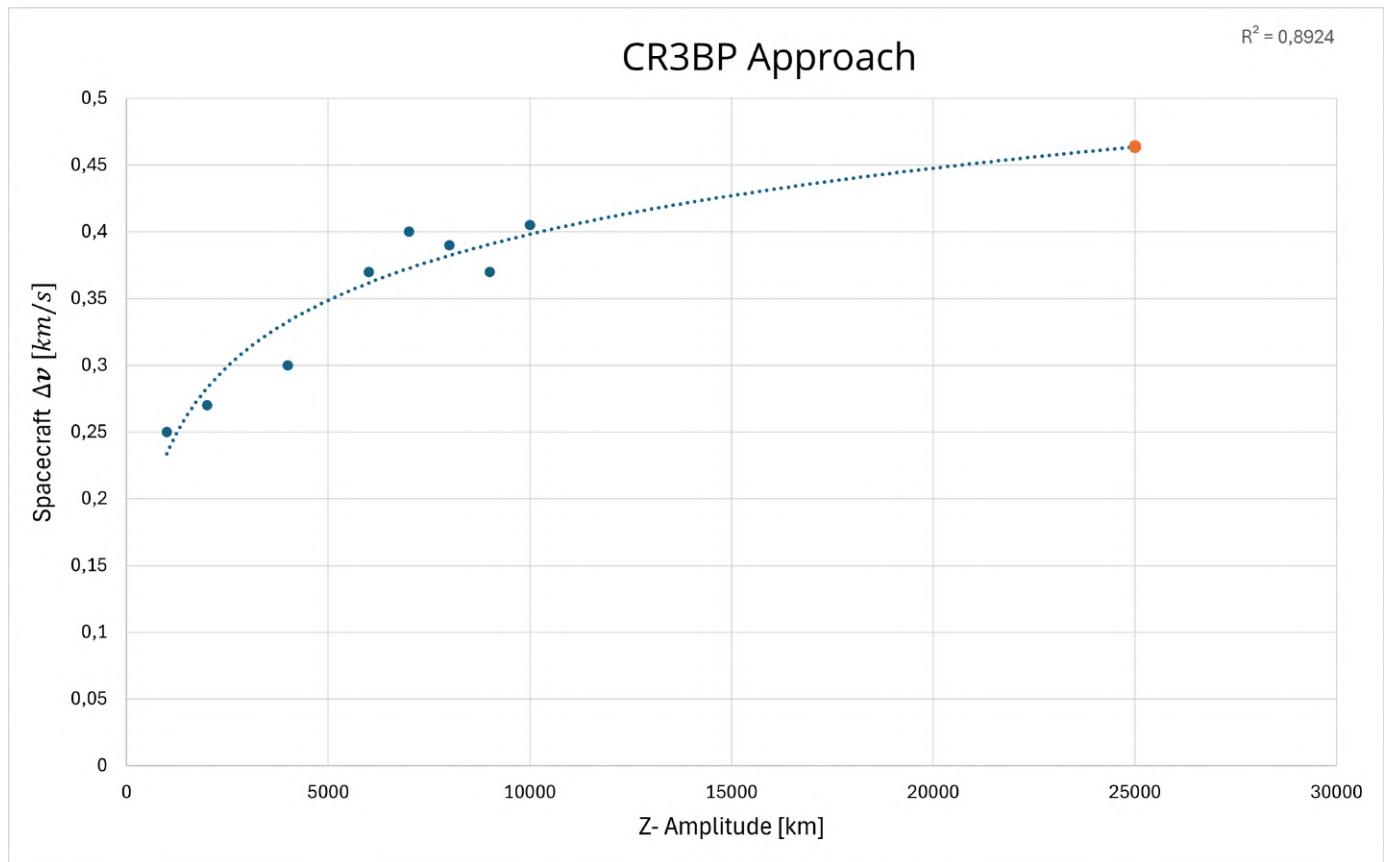


Figure B.14: Interpolation curve Δv prediction

B.3. Propulsion Modules

B.3.1. Impulsive Approximation

From $F = m \cdot a$ and $v = a \cdot t$ we could derive that $F = \frac{m \cdot v}{t}$ where F is the thrust, m is the mass to stationkeep, v is the change in velocity to be accomplished and t is the time required to accomplish it.

The formula $v = a \cdot t$ derives from uniformly accelerated motion, given the minuscule propellant consumption required to achieve velocity change, we can safely approximate the mass of the satellite to be constant during the maneuver. Consequently the formula is correctly applicable

B.3.2. Technology Comparison: Ep calculations

Known the different values of thrust required for each mass value analyzed, through those "performance factors per unit weight" of electric thrusters, we are able to estimate what might be a possible weight and energy consumption of the theoretical propulsion system needed to fulfill the tasks.

The values of this parameters are:

- AverageThrust: 10.2 mN/kg
- Averagelsp: 211s
- AveragePowerConsumption: 160 W/kg

Known T , required thrust from the "impulsive approximation hypothesis", then we have that:

$$\frac{T[mN]}{\text{AverageThrust}[mN/kg]} = \text{Thrustermass}[kg]$$

Knowing the mass of the engine, we can derive the theoretical consumption, via the other performance factor:

$$\text{ThrusterMass}[kg] \cdot \text{AveragePowerConsumption}[W/kg] = \text{PowerConsumption}[W]$$

With this last value, we are able to estimate the masses and dimensions of the solar panels required to support the use of the electric thruster.

Finally, we used the "average Isp" value to calculate the propellant mass for the stationkeeping activities, considering all weights involved.

Specifically, we have that:

- **Chemical propulsion Mass** = weight of an engine + weight of propellant tanks + propellant weight
- **Electric propulsion Mass** = weight of an engine (calculated as explained ??) + weight of required solar panels + tanks weight + propellant weight

To estimate the weight of the tanks, we did as follows:

We searched online for hydrazine tanks installable in the spacecraft bus and found a spherical tank having a capacity of 15.4L weighing 3.1kg. From Queqiao data, we calculated that we will definitely need more than 10kg of propellant for chemical stationkeeping over 10 years, therefore, choosing a tank size was deemed reasonable. Consequently, we have that:

$$\frac{15.4[L]}{3.1[kg]} = 4.97[L/kg]$$

¹

In addition, hydrazine has a density approximately equal to $1kg/L$, so we have that:

$$4.97[L/kg] \cdot 1[kg/L] = 4.97$$

This factor tells us that for every 4.97 kg of propellant, we have to "add" 1kg of "tank mass" to our mass estimations.

In the case of electric propulsion, tank weight is highly dependent on the propellant chosen; therefore, as a first approximation we used the same ratio found above for the hydrazine tanks to estimate the tank weight for the Electric propulsion technology.

So once we have calculated the first-attempt weight of the propellant needed for maneuvers, both for insertion and stationkeeping phases, we calculated the tank weight required and added it to the weight of the entire spacecraft, then we reiterated the calculations. This process was repeated until the difference between the propellant to have on board between iterations fell below a very small percentage value.

B.3.3. Thruster Selection

For the total satellite mass calculations, we used a modified version of this algorithm B.3.2. The idea was to calculate the equipment masses required for electric station-keeping for each inert mass-engine combination.

¹Liters of Hydrazine transported for each kg of tank



B. APPENDIX

To do so, this procedure was followed:

1. First, given the initial inert mass, we calculated the mass of propellant required for insertion into the halo orbit, and consequently, the mass of the hydrazine tank.
2. Then we added the hydrazine tank mass to the inert mass and with it we calculated the minimum thrust required by the stationkeeping propulsive system, according to the impulsive approximation 5.2.3
3. Next, we calculated the minimum number of engines that should be simultaneously turned on in order to generate the minimum required thrust. This value should then be rounded up to the next higher even number²
4. Once the number of engines was known, we proceeded to calculate the remaining parameters: Total engines mass, Solar panels mass, propellant needed and tanks mass³
5. Finally, we updated the inert mass value by adding the weight of the electric stationkeeping tanks, thrusters and additional solar panels required to satisfy the propulsion needs.
6. These calculations were repeated until the difference between the mass values from one iteration to the other did not fall below a tolerance value of 0.1kg

It is important to note that not all thrusters were able to meet the assumption required by the impulsive approximation. In fact, it is possible that more engines are required to meet this assumption, which, however, leads to an increase in the total weight of the spacecraft.

If the engine does not possess a sufficiently large thrust to support the resulting increase in weight, then the entire calculation procedure fails⁴

As shown in the table Tab. 5.6, the gridded ion engine BIT-3RF was unable to generate a sufficiently high thrust, so it did not produce any results compatible with the impulsive approximation.

²The thruster will be mounted on one side of the spacecraft symmetrically, so that no moment will be developed during the propulsion phase

³Once the mass of propellant required for stationkeeping was known, the tanks were sized under the assumption of performing supercritical propellant storage

⁴for each iteration made an N+1th engine is required, this increases the total spacecraft mass, so we need more engines and so on.



C. Calculations

C.1. Restricted Second Order Approaches

```
%Trajectory Design script For SpaceSystem Project  
%FRANCESCO MARRADI - Support to Human Activities from EML2
```

```
close all  
clear  
clc
```

```
%This script was build using the informations found the "lunar patched conic" pdf.  
%It is capable of calculating different kin3ds of approach trajectories to  
%the L2 point, outputting the DeltaV required for each of them.  
%The script requires us to fix 4 parameters, so to optimize the DeltaV  
%requirements we need to do a lot of calculations.
```

```
%Tagliamo l'orbita halo col piano orbitale Terra-Luna, ottenendo così i 2  
%punti di intersezione. Al fine di "entrare" nell'halo sarà necessario  
%raggiungere tali punti ed assumere la velocità richiesta (in tali punti)  
%dall'orbita halo. La differenza fra la velocità della s/c lungo la  
%traiettoria attorno alla Luna e quella richiesta alla halo fornisce la  
%DeltaV richiesta alla manovra di inserzione.  
%L'obiettivo è cercare di minimizzare tale valore.
```

```
%-----GENERAL PARAMETERS-----%  
global muT muL muS RT RM rTL rTS RsoiT RsoiL vL omegaL %#ok<*GVMIS>  
muT = 3.986*10^(5); %Earth Gravitational parameter [km^3/s^2]  
muL = 4902; %Moon Gravitational parameter [km^3/s^2]  
muS = 1.327*10^(11); %Sun Gravitational parameter [km^3/s^2]  
RT = 6378; %Earth Radius [km]  
RM = 1737; %Moon Radius [km]  
rTL = 384400; %Earth-Moon distance [km]  
rTS = 149597871; %Earth-Sun distance [km]  
RsoiT = rTS*(muT/muS)^(2/5); %Earth Soi radius [km]  
RsoiL = 66300; %Moon Soi radius [km]  
vL = 1.018; %Moon Orbital velocity [km/s]  
omegaL = 2.649*10^(-6); %Moon angular velocity along its orbit [rad/s]
```

C. CALCULATIONS

```
%Halo intercept points from Geocentric Reference system
P1T = [436467.36 -37898.616];      %[km]
P2T = [436467.36 +37898.616];      %[km]

%Halo intercept points from Moon Reference system
P1L = [P1T(1)-rTL P1T(2)];      %[km]
P2L = [P2T(1)-rTL P2T(2)];      %[km]
P1L1 = P1L(1);
P1L2 = P1L(2);
P2L1 = P2L(1);
P2L2 = P2L(2);

%Halo orbit interception points distance from the Moon
RInterceptP1 = (P1L(1)^2 + P1L(2)^2)^(0.5);    %Halo Orbital Plane Position vector
Magnitude
RInterceptP2 = (P2L(1)^2 + P2L(2)^2)^(0.5);    %Halo Orbital Plane Position vector
Magnitude

%-----FIXED INITIAL PARAMETERS-----
%
H = 300;                      %Earth circular parking orbit altitude [km]
phi0 = 0;                      %Departure Path angle [degree]
lambda0G = 0;                  %Entering Lunar Soi angle [degree] (MUST BE in [0 :180 ])
r0 = H + RT;                  %Parking Orbit Radius [km]

%-----CHOSING THE APPROACH TO USE-----
%
%Now we have to choose which approach utilize. With each of them the deltaV
requirements
%will be computed.
disp('Choose the approach to use to compute the deltaV requirements: ')
disp('Press "1" for "Direct approach"')
disp('Press "2" for "Selenocentric Bi-Impulse approach"')
disp('Press "3" for "Bi-Elliptic approach"')
choice = input('');           %Takes the user input
disp(' ')
disp('-----')

%Now we calculate the minimum and maximum velocity to intersect the Moon Soi
%for each method, in this way we could specify a better range of velocity
```



```
%evaluations
if choice == 1 || choice == 2
    disp('Input the initial departure velocity from Earth in [km/s]: ')
    v0 = input('');
    disp(' ')
elseif choice == 2
    disp('With this approach, the higher is the departure velocity the better ')
    disp('because we can fly closer to the surface of the moon, optimizing')
    disp('the breaking manouver')
    disp(' ')
    disp('Input the initial departure velocity from Earth in [km/s]: ')
    v0 = input('');
    disp(' ')
elseif choice == 3
    v1Min = (2*muT*((1/RsoiT)-(1/(RsoiT+rTL-RsoiL))))^(0.5);
    v1Max = (2*muT*((1/RsoiT)-(1/(RsoiT+rTL+RsoiL))))^(0.5);
    disp('The spacecraft will be placed in a high elliptic orbit which apoapsis')
    disp('coincides with the Earth soi.')
    disp('From there, the maximum velocity that still allows an encounter')
    disp(['with the moon soi is: ',num2str(v1Max),'km/s'])
    disp('')
    disp('Input the departure velocity from the high elliptic orbit apoapsis')
    disp('in [km/s]: ')
    v0 = input('');
    disp(' ')
end
```

```
%-----CALCULATIONS-----%
%We need to calculate how the different v0 and lambda1 parameters influence
%the deltaV requirements for the mission, so we will calculate the
%variation of the deltaV fixing one parameter and varying the other and
%viceversa.

%We define two counters for the iterations
k = 1;           %Velocity counter [v0]
q = 1;           %Entering moon soi counter [lambda1]
```



C. CALCULATIONS

```
%We initialize some variables that are going to be used later
Memory = zeros(200,1800);
yqθ=0;
DeltaVinsMag = 0;
DeltaVScMag = 0;
DeltaVtotal = 0;
DeltaV1 = 0;
DeltaV2 = 0;

%Lambda conversion from degree to radians
lambda0G = lambda0G*pi/180;

%For the sake of calculations, we need to memorize the initial value of lambda1 by
%using lambda0G
lambda1 = lambda0G;

%Circular velocity on initial the parking orbit around Earth
vcircular = (muT/r0)^0.5;

%We initialize a variable to break the iterations if the desired conditions
%are met
breaking = 0;

if choice == 1 || choice == 2 || choice == 3

    for k = 1:300      %We will analyze a range of 300 m/s

        for q = 1:1800    %We will analyze a range of 180

            if choice == 1 || choice == 2
                [E1,h1,r1,v1,phi1,gamma1,p1,a1,e1,ni0,ni1,ecc0,ecc1,gamma0,FlightTime,
                 v2,epsilon2,E2,h2,e2,p2,ni2,a2,breaking] = GeoDepOrbit(r0,v0,
                 lambda1,phi0);

            if breaking == 1
                yqθ(k,q) = 0;
                DeltaVinsMag(k,q) = 0;
                DeltaVScMag(k,q) = 0;
                DeltaVtotal(k,q) = 0;
                DeltaV1(k,q) = 0;
                DeltaV2(k,q) = 0;
```



```
break
else
end

if choice == 1 %Direct Approach
    [AimedPoint,niIntercept,breaking] = MoonPeriapsisPosition(p2,e2,
        ni2,lambda1,P1L1,P1L2,P2L1,P2L2);
    if breaking == 1
        yq0(k,q) = 0;
        DeltaVinsMag(k,q) = 0;
        DeltaVScMag(k,q) = 0;
        DeltaVtotal(k,q) = 0;

        break

    else
        [yq0,breaking,lambda1,E1,h1,r1,v1,phi1,gamma1,p1,a1,e1,ni0,ni1
            ,ecc0,ecc1,gamma0,FlightTime,v2,epsilon2,E2,h2,e2,p2,ni2,
            a2,niIntercept] = OptimalSoiEnteringAngle(r0,v0,lambda1,
            phi0,niIntercept,ni2,P1L1,P1L2,P2L1,P2L2,AimedPoint,q,yq0,
            k,e2,p2,RInterceptP1);

        if breaking == 1
            yq0(k,q) = 0;
            DeltaVinsMag(k,q) = 0;
            DeltaVScMag(k,q) = 0;
            DeltaVtotal(k,q) = 0;

            break

        else
            [DeltaVinsMag,DeltaVScMag,DeltaVtotal,yq0] =
                DeltaVCalcDirect(v0,vcircular,phi0,niIntercept,p2,
                lambda1,ni2,k,q,AimedPoint,e2,yq0,DeltaVinsMag,
                DeltaVtotal,DeltaVScMag);

        end

    end

elseif choice == 2 %Selenocentric Bi-Impulse Approach
```

C. CALCULATIONS

```
[breaking,rpmoon,vrpmoon,ramoon,amoonyellpmoon,vcircularmoon,
vellamoon,Emoon,hmoon,emoon,pmoon] = SelenocentricBiImpulse(p2
,e2,a2,RInterceptP1);

if breaking == 1
    yq0(k,q) = 0;
    DeltaVinsMag(k,q) = 0;
    DeltaVScMag(k,q) = 0;
    DeltaVtotal(k,q) = 0;
    DeltaV1(k,q) = 0;
    DeltaV2(k,q) = 0;
    break

else
    [DeltaVinsMag,DeltaVScMag,DeltaVtotal,yq0,DeltaV1,DeltaV2] =
        DeltaVCalcSeleBiImpulse(v0,vcircular,phi0,RInterceptP1,
        RInterceptP2,lambdal1,ni2,k,q,vrpmoon,yellpmoon,
        vcircularmoon,vellamoon,P1L1,P1L2,P2L1,P2L2,yq0,
        DeltaVinsMag,DeltaVtotal,DeltaVScMag,DeltaV1,DeltaV2);

end

end

elseif choice == 3 %Bi-Elliptic Approach
[E1,h1,r1,v1,phi1,gamma1,p1,a1,e1,ni0,ni1,ecc0,ecc1,gamma0,
FlightTimeTotal,v2,epsilon2,E2,h2,e2,p2,ni2,a2,breaking,v01a,a01,
v01p,lambdal1,h01,e01,p01,FlightTimeFirstEllipse,
FlightTimeSecondEllipse] = EllipseDepOrbit(r0,v0,lambdal1,phi0,
v1Max,RInterceptP1);

if breaking == 1
    yq0(k,q) = 0;
    DeltaVinsMag(k,q) = 0;
    DeltaVScMag(k,q) = 0;
    DeltaVtotal(k,q) = 0;

    break

else
    [AimedPoint,niIntercept,breaking] =
```



```
MoonPeriapsisPositionBiElliptic(p2,e2,ni2,lambda1,P1L1,P1L2,  
P2L1,P2L2);  
  
if breaking == 1  
    yq0(k,q) = 0;  
    DeltaVinsMag(k,q) = 0;  
    DeltaVScMag(k,q) = 0;  
    DeltaVtotal(k,q) = 0;  
    break  
  
else  
    [yq0,breaking,lambdaOpt,E1,h1,r1,v1,phi1,gamma1,p1,a1,e1,ni0,  
     ni1,ecc0,ecc1,gamma0,FlightTimeTotal,v2,epsilon2,E2,h2,e2,  
     p2,ni2,a2,v01a,a01,v01p,niIntercept,h01,e01,p01] =  
    OptimalSoiEnteringAngleBiElliptic(r0,v0,lambda1,phi0,  
    niIntercept,ni2,P1L1,P1L2,P2L1,P2L2,q,yq0,k,AimedPoint,  
    v1Max,e2,p2,RInterceptP1);  
  
    if breaking == 1  
        yq0(k,q) = 0;  
        DeltaVinsMag(k,q) = 0;  
        DeltaVScMag(k,q) = 0;  
        DeltaVtotal(k,q) = 0;  
        break  
  
    else  
        [DeltaVinsMag,DeltaVScMag,DeltaVtotal,yq0] =  
        DeltaVCalcBiElliptic(v0,vcircular,v01a,v01p,  
        niIntercept,p2,lambdaOpt,ni2,k,q,AimedPoint,e2,yq0,  
        DeltaVinsMag,DeltaVtotal,DeltaVScMag);  
  
    end  
  
end  
  
end  
  
%Updating the lambda1 angle to reiterate the calculations  
lambda1 = lambda0G + 0.1*(pi/180)*q;
```



```
%Saving the data used for the calculations
if breaking == 1
    %In this way we will know which velocity/angle combinations don't
    %work
    Memory(k,q) = 0;

    %resetting the "breaking" value
    breaking = 0;

else
    Memory(k,q) = 1;

end

%Resetting the angle values to reiterate the calculations
lambda1 = lambda0G;

%Storing and updating the velocity values to reiterate the calculations
xv0(k) = v0;
v0 = v0 + 0.001;

%Resetting the q value to reiterate the calculations
q = 1;

end

else
    disp('You have pressed the wrong choice''s number, retry')
end

disp(' ')
disp(' ')
disp('Calculations ended')
disp('-----')

%% -----MANAGING DATA-----%
%We need to substitute to every "zero" inside the DeltaV matrices a "nan"
```



%value, in this way they aren't gonna be plotted into the graphs

```
[A,B] = size(DeltaVinsMag);
for i = 1:A
    for j = 1:B
        if DeltaVinsMag(i,j) == 0
            DeltaVinsMag(i,j) = nan;
        end
    end
end

[A,B] = size(DeltaVScMag);
for i = 1:A
    for j = 1:B
        if DeltaVScMag(i,j) == 0
            DeltaVScMag(i,j) = nan;
        end
    end
end

[A,B] = size(DeltaVtotal);
for i = 1:A
    for j = 1:B
        if DeltaVtotal(i,j) == 0
            DeltaVtotal(i,j) = nan;
        end
    end
end

[A,B] = size(DeltaV1);
for i = 1:A
    for j = 1:B
        if DeltaV1(i,j) == 0
            DeltaV1(i,j) = nan;
        end
    end
end

[A,B] = size(DeltaV2);
for i = 1:A
    for j = 1:B
        if DeltaV2(i,j) == 0
```



C. CALCULATIONS

```
DeltaV2(i,j) = nan;
end
end
end

disp(' ')
disp(' ')
disp('DeltaV data rearranged')
disp('-----')

%% -----PLOTTING 3D CURVES FOR DELTAV CALCULATIONS-----%

disp(' ')
disp(' ')
disp('Plotting DeltaV calculations...')
disp(' ')
close all
figure(1)
title('Translunar Injection Manouver DeltaV required')
xlabel('Angle values [deg]')
ylabel('Velocity values [km/s]')
zlabel('DeltaV [km/s]')
set(gca, 'FontSize', 13, 'LineWidth', 1.2)
set(gcf, 'color', 'w')
hold on
surf(yq0,xv0,DeltaVtotal-DeltaVScMag)
shading interp
c = colorbar;
c.Label.String = 'DeltaV [km/s]';

figure(2)
title('Halo Insertion deltaV required')
xlabel('Angle values [deg]')
ylabel('Velocity values [km/s]')
zlabel('DeltaV [km/s]')
set(gca, 'FontSize', 13, 'LineWidth', 1.2)
set(gcf, 'color', 'w')
hold on
surf(yq0,xv0,DeltaVinsMag)
shading interp
c = colorbar;
```



```
c.Label.String = 'DeltaV [km/s]';

figure(3)
title('Spacecraft deltaV required')
xlabel('Angle values [deg]')
ylabel('Velocity values [km/s]')
zlabel('DeltaV [km/s]')
set(gca,'FontSize', 13, 'LineWidth', 1.2)
set(gcf,'color','w')
hold on
surf(yq0,xv0,DeltaVScMag)
shading interp
c = colorbar;
c.Label.String = 'DeltaV [km/s]';

disp('Press continue to identify the best deltaV values')
pause

%We wont to identify the five smallest deltaV requirements for our spacecraft,
%to evaluate them
%We initialize the vectors we are going to use:
%This will contain the DeltaV numbers
A = zeros(1,5);
%While this will contain the color used in the plot
color = ["#0072BD" "#D95319" "#EDB120" "#77AC30" "#7E2F8E" "#A2142F"];

%We use this function to extract the "best" 5 values inside each matrix.
[velocityA,AngleA,row,col] = ValuesExtractor3000(A,DeltaVScMag,xv0,yq0);

%Now we identify their positions on the deltaV plots for each figure
for j = 1:5
    figure(1)
    scatter3(AngleA(j),velocityA(j),DeltaVtotal(row(j),col(j)),40, 'MarkerEdgeColor',
        color(j), 'Marker', 'o', 'MarkerFaceColor', 'auto')

    figure(2)
    scatter3(AngleA(j),velocityA(j),DeltaVinsMag(row(j),col(j)),40, 'MarkerEdgeColor',
        color(j), 'Marker', 'o', 'MarkerFaceColor', 'auto')

    figure(3)
```

C. CALCULATIONS

```
scatter3(AngleA(j),velocityA(j),DeltaVScMag(row(j),col(j)),40, 'MarkerEdgeColor',
        color(j), 'Marker', 'o', 'MarkerFaceColor', 'auto')

pause(0.5)
end

disp('end')
disp('-----')

%% -----TRAJECTORY PLOTS-----
%Here we plot the five best trajectory to visualize them.

%First we need to extract the velocity and angle value from the deltaV matrix,
%that generate that solutions.
%We wont to identify the five smallest deltaV requirements for our spacecraft,
%to evaluate them
%We initialize the vectors we are going to use
A = zeros(1,5); %will contain the DeltaV numbers
%This will contain the color used in the plot
color = ["#0072BD" "#D95319" "#EDB120" "#77AC30" "#7E2F8E" "#A2142F"];

%We use this function to extract the "best" 5 values inside each matrix.
[velocityA,AngleA,row,col] = ValuesExtractor3000(A,DeltaVScMag,xv0,yq0);

%Now we need to initialize the figures we are going to use to plot the
%trajectories.
theta = linspace(0,2*pi,1000);

%In the code below we will switch to and from polar to cartesian coordinates
%and than back. Its the only way i've found to make matlab draw the
%circles in the way i wanted

figure(4)
[x,y] = pol2cart(theta,RT);
[~, Rt] = cart2pol(x,y);
%plots the Earth
polarplot(theta,Rt,'b')
rlim([0 1.4*10^4]);
set(gcf, 'color', 'w')
title('Departure orbit from Earth')
set(gca,'FontSize', 13, 'LineWidth', 1)
hold on
```



```
%plots the parking orbit around Earth
[x,y] = pol2cart(theta,r0);
[~, R0] = cart2pol(x,y);
polarplot(theta,R0,'k','LineWidth',1,'LineStyle','--')
legend('', 'Parking orbit','Autoupdate','off')

%Here we are plotting the Geocentric transfer trajectory in a reference
%system fixed with the Earth-Moon axe
figure(5)
%This code allow us to plot the Earth
[x,y] = pol2cart(theta,RT);
[~, Rt] = cart2pol(x,y);
polarplot(theta,Rt,'b')
set(gcf, 'color' , 'w')
title('Geocentric transfer trajectory')
set(gca, 'FontSize', 13, 'LineWidth', 1.2)
hold on
%This code allow us to plot the Moon Soi from Earth point of view
[x,y] = pol2cart(theta,RsoiL);
[thm, Rm] = cart2pol( x-rTL, y);
polarplot(thm,Rm,'k','LineStyle','--');
%This code allow us to plot the Moon
[x,y] = pol2cart(theta,RM);
[thmsoi, Rmsoi] = cart2pol(x-rTL,y);
polarplot(thmsoi,Rmsoi,'k')
%This code allow us to plot the Earth Soi
[x,y] = pol2cart(theta,RsoiT);
[thTsoi, RTsoi] = cart2pol(x,y);
polarplot(thTsoi,RTsoi,'k','LineStyle','--','Color','b')
legend('', 'Moon SOI', '', 'Earth SOI','Autoupdate','off')

%Here we are plotting the halo interception points, the moon and the moon
%soi and at last the selenocentric arrival trajectory of the sc
figure(6)
%This code allow us to plot the Moon
[x,y] = pol2cart(theta,RM);
[thmsoi, Rmsoi] = cart2pol(x,y);
polarplot(thmsoi,Rmsoi,'k')
rlim([0 RsoiL*1.2])
set(gcf, 'color' , 'w')
```

C. CALCULATIONS

```
title('Selenocentric arrival trajectory')
set(gca,'FontSize', 13, 'LineWidth', 1.2)
hold on
%This code allow us to plot the Moon SoI
[x,y] = pol2cart(theta,RsoIL);
[thm, Rm] = cart2pol( x, y);
polarplot(thm,Rm,'k','LineStyle','--');
%Now we plot the halo interception points
[thetaP1, rP1] = cart2pol(-P1L(1), -P1L(2));
[thetaP2, rP2] = cart2pol(-P2L(1), -P2L(2));
polarplot(thetaP1,rP1,'.m','MarkerSize',24);
polarplot(thetaP2,rP2,'.r','MarkerSize',24);
legend('', 'Moon SOI', 'P1', 'P2','Autoupdate','off')

disp(' ')
disp(' ')
disp('Press continue to plot the trajectories...')
pause

%Now we need to recalculate that trajectories and then plotting them.
for j = 1:5
    v0 = velocityA(j);
    lambda1 = AngleA(j)*pi/180;           %converting lambda1 into radians
    if choice == 1
        [E1,h1,r1,v1,phi1,gamma1,p1,a1,e1,ni0,ni1,ecc0,ecc1,gamma0,FlightTime,v2,
         epsilon2,E2,h2,e2,p2,ni2,a2,breaking] = GeoDepOrbit(r0,v0,lambda1,phi0);

    elseif choice == 2
        [E1,h1,r1,v1,phi1,gamma1,p1,a1,e1,ni0,ni1,ecc0,ecc1,gamma0,FlightTime,v2,
         epsilon2,E2,h2,e2,p2,ni2,a2,~] = GeoDepOrbit(r0,v0,lambda1,phi0);
        [breaking,rpmoon,vrpmoon,ramoon,amoon,vellpmoon,vcircularmoon,vellamoona,Emoon,
         hmoon,emoon,pmoon] = SelenocentricBiImpulse(p2,e2,a2,RInterceptP1);

    elseif choice == 3
        [E1,h1,r1,v1,phi1,gamma1,p1,a1,e1,ni0,ni1,ecc0,ecc1,gamma0,FlightTimeTotal,v2,
         epsilon2,E2,h2,e2,p2,ni2,a2,breaking,v01a,a01,v01p,lambda1,h01,e01,p01,
         FlightTimeFirstEllipse,FlightTimeSecondEllipse] = EllipseDepOrbit(r0,v0,
         lambda1,phi0,v1Max,RInterceptP1);

    end

%Here we plots the geocentric departure orbit
```



```
figure(4)
if choice == 1 || choice == 2
    theta2 = linspace(0,ni1,1000);
    polarplot(theta2+pi-gamma0-omegaL*FlightTime,(p1./(1+e1*cos(theta2))),'-',
               'LineWidth',1.2,'Color',color(j))

elseif choice == 3
    theta2 = linspace(0,pi,1000);
    polarplot(+theta2+gamma0+omegaL*FlightTimeTotal,(p01./(1+e01*cos(theta2))),'-',
               'LineWidth',1.2,'Color',color(j))

end

%Here we plots the geocentric transfer orbit
figure(5)
if choice == 1 || choice == 2
    theta3 = linspace(0,ni1,1000);
    polarplot(theta3+pi-gamma0-omegaL*FlightTime,(p1./(1+e1*cos(theta3))),'-',
               'LineWidth',1.2,'Color',color(j))

elseif choice == 3
    theta3 = linspace(0,pi,1000);
    theta4 = linspace(-pi,-ni1,1000);

    %Plot the first ellipse with the apogee coincident with RsoiT
    polarplot(theta3+gamma0+omegaL*FlightTimeTotal,(p01./(1+e01*cos(theta3))),'-',
               'LineWidth',1.2,'Color',color(j))

    %Plot the second ellipse
    polarplot(theta4+gamma0+omegaL*FlightTimeTotal,(p1./(1+e1*cos(theta4))),'-',
               'LineWidth',1.2,'Color',color(j))

end

%Here we plots the Selenocentric Arrival orbit
figure(6)
if choice == 1
    theta4 = linspace(-1.75*pi+niIntercept,ni2+0.03*pi,1000);
```



SUPREMAE
DIGNITATIS
1343

UNIVERSITÀ
DI PISA

C. CALCULATIONS

```
polarplot((theta4-(ni2+lambda1)),(p2./(1+e2*cos(theta4))), '-','LineWidth',1.2,  
'Color',color(j))  
  
elseif choice == 2  
theta1 = linspace(-pi,0,1000);  
theta4 = linspace(0*pi,ni2+0.01*pi,1000);  
theta5 = linspace(-0.135*pi,pi,1000);  
  
%Plot the arrival selenocentric trajectory  
polarplot((theta4-(ni2+lambda1)),(p2./(1+e2*cos(theta4))), '-','LineWidth',1.2,  
'Color',color(j))  
  
%Plot the selenocentric semi-ellipse  
polarplot((theta1-(ni2+lambda1)),(pmoon./(1+emoon*cos(theta1))), '-','LineWidth  
' ,1.2,'Color',color(j))  
  
%Plot the circular selenocentric intercept trajectory with the halo  
%intercept points  
polarplot((theta5-(ni2+lambda1)),(RInterceptP1./(1+0*cos(theta5))), '-','  
LineWidth',1.2,'Color',color(j))  
  
elseif choice == 3  
theta1 = linspace(pi,0.5*pi,1000);  
polarplot((-theta1+(ni2+lambda1)),(p2./(1+e2*cos(theta1))))  
  
end  
  
pause(0.2)  
  
end  
  
disp('end')  
disp('-----')  
  
%% -----FUNCTIONS USED IN THE  
SCRITP-----%
```



```
%-----GEOCENTRIC TRANSFER ORBIT -----%
function [E1,h1,r1,v1,phil,gamma1,p1,a1,e1,ni0,ni1,ecc0,ecc1,gamma0,FlightTime,v2,
epsilon2,E2,h2,e2,p2,ni2,a2,breaking] = GeoDepOrbit(r0,v0,lambda1,phi0)
global muT muL rTL RsoiL vL omegaL

%We will assume that the geocentric trajectory crosses the lunar Soi before
%the orbit's apogee.
E1 = (v0^2)/2 - muT/r0;           %Transfer Trajectory Mechanical Specific Energy
h1 = r0*v0*cos(phi0*pi/180);      %Transfer Trajectory Angular Momentum

%Now we calculate the arrival (at the Moon Soi) parameters from Earth
%reference system point of view
r1 = (rTL^2 + RsoiL^2 - 2*rTL*RsoiL*cos(lambda1))^(0.5);      %Sc Position vector
v1 = (2*(E1 + muT/r1))^(0.5);          %Sc speed
phil = acos(h1/(r1*v1));              %Sc flight path angle
    Entering the Lunar Soi
gamma1 = asin((RsoiL/r1)*sin(lambda1));        %Sc Entering Lunar Soi
    Angle from Earth view point
p1 = (h1^(2))/muT;                      %Geocentric Departure
    orbit Semilatus rectus
a1 = -muT/(2*E1);                      %Geocentric Departure
    orbit Semimajor Axis
e1 = (1-p1/a1)^(0.5);                  %Geocentric Departure
    orbit eccentricity
ni0 = real(acos((p1-r0)/(r0*e1)));      %Starting true anomaly (on
    the parking orbit)
ni1 = acos((p1-r1)/(r1*e1));          %Arriving true anomaly (at
    the moon Soi)
ecc0 = real(acos((e1 + cos(ni0))/(1 + e1*cos(ni0))));        %Starting eccentric
    anomaly (on the parking orbit)
ecc1 = acos((e1 + cos(ni1))/(1 + e1*cos(ni1)));        %Arriving eccentric
    anomaly (at the moon Soi)

%We compute both the flight time along the departure orbit and the phase
%angle
FlightTime = ((a1^3/muT)^(0.5))*((ecc1-e1*sin(ecc1))-((ecc0-e1*sin(ecc0))));
gamma0 = ni1 - ni0 - gamma1 - omegaL*FlightTime;            %Departure Phasing angle
```



C. CALCULATIONS

```
%-----CONDITION AT THE PATCH POINT -----
v2 = (v1^2 + vL^2 - 2*v1*vL*cos(phi1-gamma1))^(0.5);           %Sc entering velocity into
                                                               Moon SOI
epsilon2 = asin((vL/v2)*cos(lambda1)-(v1/v2)*cos((lambda1 + gamma1 - phi1)));
                                                              

%-----SELENOCENTRIC ARRIVAL ORBIT -----
E2 = (v2^2)/2 - muL/RsoiL;                                         %Moon Trajectory
                                                               Mechanical Specific Energy
h2 = RsoiL*v2*sin(epsilon2);                                         %Moon Trajectory Angular
                                                               Momentum
e2 = (1 + (2*E2*h2^(2))/(muL^(2)))^(0.5);                         %Moon Trajectory
                                                               Eccentricity
p2 = h2^(2)/muL;                                                 %Moon Trajectory Semilatus
                                                               rectus
ni2 = acos((1/e2)*(p2/RsoiL - 1));                                %Moon Trajectory Enter
                                                               point True Anomaly
a2 = -muL/2*E2;                                                 %Moon Trajectory semimajor
                                                               axis

%We need to check if the Sc velocity at the moon Soi interception is positive
%if not, the initial departure trajectory was not sufficiently energetic
if isreal(v1) == 0
    disp('The initial departure trajectory was not energetic enough to reach')
    disp('the moon Soi')
    breaking = 1;
    return

elseif isreal(phi1) == 0
    disp('The actual trajectory doesn't permits to reach the interception points ')
    breaking = 1;
    return
end

breaking = 0;
end

%-----
% -----DIRECT APPROACH -----
%The idea behind this is to verify if the selenocentric trajectory's
```

%periapsis is below or above the Earth-Moon axe. If it is below, we are % going to aim our trajectory at the interception point above, if it is % above, the viceversa.
 %To do so, we need to calculate if our trajectory "pass above" or "pass %below" the aimed point.
 %If it "pass below", we need to decrease the spacecraft entering velocity %in the Moon soi (maintaining the same entering angle), if we "pass above", %the viceversa.

```
%-----SELENOCENTRIC PERIAPSIS POSITION-----%
function [AimedPoint,niIntercept,breaking] = MoonPeriapsisPosition(p2,e2,ni2,lambda1,
P1L1,P1L2,P2L1,P2L2)
global RM

%Verifying the selenocentric periapsis position
if (ni2+(lambda1)) < pi
    disp('The lunar fly by trajectory periapsis is under the Earth-Moon axe')

    if (ni2+(lambda1)) < pi/2
        disp('and we are going to aim for P2 interception point')
        disp(' ')
        AimedPoint = 2;

        %We calculate the true anomaly of the P2 point with respect to the
        %selenocentri trajectory periapsis
        [thetaP2, ~] = cart2pol(-P2L1, -P2L2);
        niIntercept = (2*pi+thetaP2) + ni2 + lambda1;

    elseif (ni2+(lambda1)) > pi/2 || (ni2+(lambda1)) < 2*pi
        disp('and we are going to aim for P1 interception point')
        disp(' ')
        AimedPoint = 1;

        %We calculate the true anomaly of the P1 point with respect to the
        %selenocentri trajectory periapsis
        [thetaP1, ~] = cart2pol(-P1L1, -P1L2);
        niIntercept = thetaP1 + ni2 + lambda1;

    end

elseif (ni2+(lambda1)) > pi
```

C. CALCULATIONS

```
disp('The lunar fly by trajectory periapsis is above the Earth-Moon axe')

if -(ni2+(lambda1)) < -pi/2
    disp('and we are going to aim for P1 interception point')
    disp(' ')
    AimedPoint = 1;

    %We calculate the true anomaly of the P1 point with respect to the
    %selenocentri trajectory periapsis
    [thetaP1, ~] = cart2pol(-P1L1, -P1L2);
    niIntercept = thetaP1 + ni2 + lambda1;

elseif -(ni2+(lambda1)) > -pi/2 || -(ni2+(lambda1)) < -2*pi
    disp('and we are going to aim for P2 interception point')
    disp(' ')
    AimedPoint = 2;

    %We calculate the true anomaly of the P2 point with respect to the
    %selenocentri trajectory periapsis
    [thetaP2, ~] = cart2pol(-P2L1, -P2L2);
    niIntercept = (2*pi+thetaP2) + ni2 + lambda1;

end

end

%We need to check that the Moon Orbit Periapsis doesnt go through the Moon
%Surface.
rpmoon = p2/(1+e2);           %Moon Trajectory periapsis

if rpmoon < RM + 75
    disp('The Moon Iperbolic Trajectory intersect the moon surface')

    %We need to know which parameter not to save, to do so we
    %will use this value
    breaking = 1;

    return
else

end
```



```
breaking = 0;  
end
```

```
%-----OPTIMAL SOI ENTER ANGLE: CYCLE-----%  
%In the hypothesis that the selenocentric trajectory exiting soi point  
%coincides with the halo interception point (thats should produce a very  
%small error), we write the "angle difference" function between the two  
%points and we reiterate the precedent calculations until we achieve  
%an "angle difference" value under our error tolerance.  
%The idea is that to achieve an encounter we can try to "steer" the  
%insertion orbit pivoting it around the moon, in doing so we need to change  
%the soi interning angle lambda1. We could also mantain the same entering  
%angle and change the entering velocity, but this will increase the DeltaV  
%requirements for our mission.
```

```
function [yq0,breaking,lambda1,E1,h1,r1,v1,phi1,gamma1,p1,a1,e1,ni0,ni1,ecc0,ecc1,  
gamma0,FlightTime,v2,epsilon2,E2,h2,e2,p2,ni2,a2,niIntercept] =  
OptimalSoiEnteringAngle(r0,v0,lambda1,phi0,niIntercept,ni2,P1L1,P1L2,P2L1,P2L2,  
AimedPoint,q,yq0,k,e2,p2,RInterceptP1)  
  
tol = 0.005; %tolerance [deg]  
AngleDiff = niIntercept - (2*pi-ni2); %Angle difference definition  
i = 0; %Iteration counter  
breaking = 0;  
  
%Iterative Cycle  
while abs(AngleDiff) > ((tol*pi/180)) || i < 1  
  
    AngleDiff = niIntercept - (2*pi-acos((1/e2)*(p2/RInterceptP1 - 1)));  
    lambda0pt = (lambda1) - AngleDiff/4;  
    lambda1 = lambda0pt;  
  
%-----GEOCENTRIC TRANSFER ORBIT -----%  
[E1,h1,r1,v1,phi1,gamma1,p1,a1,e1,ni0,ni1,ecc0,ecc1,gamma0,FlightTime,v2,epsilon2,  
E2,h2,e2,p2,ni2,a2,breaking] = GeoDepOrbit(r0,v0,lambda1,phi0);  
  
if breaking == 1
```

C. CALCULATIONS

```
return
end

%Verifying the selenocentric periapsis position
if AimedPoint == 1

    %We calculate the true anomaly of the P1 point with respect to the
    %selenocentri trajectory periapsis
    [thetaP1, ~] = cart2pol(-P1L1, -P1L2);
    niIntercept = thetaP1 + ni2 + lambda1;

elseif AimedPoint == 2

    %We calculate the true anomaly of the P2 point with respect to the
    %selenocentri trajectory periapsis
    [thetaP2, ~] = cart2pol(-P2L1, -P2L2);
    niIntercept = (2*pi+thetaP2) + ni2 + lambda1;

end

i = i + 1;           %n   of iterations done

%This code underneat will exit the while cycle if the
%"AngleDiff" parameter become imaginary or the number of
%iterations starts to be too high
if isreal(AngleDiff) == 0 || i>(1500)
    disp('The geocentric transfer trajectory is not energetic enough to')
    disp('reach the required point on the moon soi')
    breaking = 1;
    break

else
end

end

%This code underneat will exit the for cycle if the
%"AngleDiff" parameter became imaginary, skipping directly
%the deltaV calculations
if isreal(AngleDiff) == 0 || breaking == 1
    disp('The actual trajectory doesnt permits to reach the interception points')
```



```
breaking = 1;
yq0(k,q) = 0;

return
else

end

%Obtained Results display
disp('to achieve an encounter, the entering angle on the moon soi should be: ')
disp([num2str(lambda1*180/pi), ' degree'])
disp(['number of orbit iterations: ',num2str(i)])
yq0(k,q) = lambda1*180/pi;

breaking = 0;

end

%-----%
%-----DELTA V CALCULATIONS DIRECT-----%
%Here we will calculate both the deltaV required for the injection on the
%departure orbit (from the parking one) and for the insertion manouver into
%the halo orbit

function [DeltaVinsMag,DeltaVScMag,DeltaVtotal,yq0] = DeltaVCalcDirect(v0,vcircular,
    phi0,niIntercept,p2,lambda1,ni2,k,q,AimedPoint,e2,yq0,DeltaVinsMag,DeltaVtotal,
    DeltaVScMag)
global muL

%Departure manouver deltaV required
DepBurnDeltaV = (v0^2 + vcircular^2 - 2*v0*vcircular*cos(phi0))^(0.5);

if AimedPoint == 1
    vP1 = [-0.0562 0.0543 0.1013]; %Halo orbit velocity
    required for entering P1
    vscp1PQWtan = ((muL/p2)^(0.5))*(-sin(niIntercept)); %Sc tangential
    velocity at the interception point with P1
    vscp1PQWrad = ((muL/p2)^(0.5))*(e2 + cos(niIntercept)); %Sc radial velocity
```

C. CALCULATIONS

```

at the interception point with P1
vscp1PQW = [vscp1PQWtan vscp1PQWrad 0]; %Sc velocity at the
                                             interception point with P1, vectorial form

%Rotation matrix from PQW
T = [cos(-(lambda1+ni2)) sin(-(lambda1+ni2)) 0
     -sin(-(lambda1+ni2)) cos(-(lambda1+ni2)) 0
     0 0 1];
vscp1XYZ = T*vscp1PQW';

%Now we can compute the DeltaV required for the Halo orbit insertion
DeltaVins = vP1 - vscp1XYZ';
DeltaVinsMag(k,q) = (DeltaVins(1)^2 + DeltaVins(2)^2 + DeltaVins(3)^2)^(0.5);
DeltaVScMag(k,q) = DeltaVinsMag(k,q);
DeltaVtotal(k,q) = (DeltaVins(1)^2 + DeltaVins(2)^2 + DeltaVins(3)^2)^(0.5) +
DepBurnDeltaV;

elseif AimedPoint == 2
vP2 = [+0.0562 -0.0543 -0.1013]; %Halo orbit velocity
                                             required for entering P2
vscp2PQWtan = ((muL/p2)^(0.5))*(-sin(niIntercept)); %Sc tangential
                                             velocity at the interception point with P2
vscp2PQWrad = ((muL/p2)^(0.5))*(e2 + cos(niIntercept)); %Sc radial velocity
                                             at the interception point with P2
vscp2PQW = [vscp2PQWtan vscp2PQWrad 0]; %%Sc velocity at the
                                             interception point with P2, vectorial form

%Rotation matrix from PQW
T = [cos(-(lambda1+ni2)) sin(-(lambda1+ni2)) 0
     -sin(-(lambda1+ni2)) cos(-(lambda1+ni2)) 0
     0 0 1];
vscp2XYZ = T*vscp2PQW';

%Now we can compute the DeltaV required for the Halo orbit insertion
DeltaVins = vP2 - vscp2XYZ';
DeltaVinsMag(k,q) = (DeltaVins(1)^2 + DeltaVins(2)^2 + DeltaVins(3)^2)^(0.5);
DeltaVScMag(k,q) = DeltaVinsMag(k,q);
DeltaVtotal(k,q) = (DeltaVins(1)^2 + DeltaVins(2)^2 + DeltaVins(3)^2)^(0.5) +
DepBurnDeltaV;

end

```



```
disp('The deltaV required for the halo orbital insertion with this transfer')
disp(['manouver is: ',num2str(DeltaVinsMag(k,q)), ' [km/s]'])
disp('The deltaV required for the translunar insertion manouver would be: ')
disp([num2str(v0 - vcircular), ' [km/s]'])
disp(['For a total deltaV of: ', num2str(DeltaVtotal(k,q)), ' [km/s]'])

end

%-----%
% -----SELENOCENTRIC BI-IMPULSE APPROACH-----%
%The idea its to try to minimize the deltaV required for the Halo orbit
%insertion by using multiple manouvers.
%First we need to try to achieve the smaller selenocentric periapsis
%possible, than we will perform an insertion manouver to achieve an orbit
%around the moon.

function [breaking,rpmoon,vrpmoon,ramoon,amoon,vellpmoon,vcircularmoon,vellamoon,Emoon
,hmoon,emoon,pmoon] = SelenocentricBiImpulse(p2,e2,a2,RInterceptP1)
global muL RM

%Selenocentric orbit values
rpmoon = p2/(1+e2);
vrpmoon = (2*(muL/rpmoon) - muL/a2)^(0.5); %Moon Trajectory periapsis
%Moon Sc Velocity at
    Periapsis
ramoon = RInterceptP1;
    radius
amoon = (ramoon+rpmoon)/2; %Moon Ellipse Trajectory
    semimajor axis
vellpmoon = (2*(muL/rpmoon) - muL/amoon)^(0.5); %Moon Sc velocity required
    for the insertion in the Ellipse at its periapsis
vcircularmoon = (muL/RInterceptP1)^(0.5); %Moon Sc required circular
    velocity
vellamoon = (2*(muL/RInterceptP1) - muL/amoon); %Moon Sc velocity required
    for the insertion in the circular orbit

Emoon = (vellpmoon^2)/2 - muL/rpmoon; %Moon Trajectory
    Mechanical Specific Energy
hmoon = rpmoon*vellpmoon; %Moon Trajectory Angular
```



C. CALCULATIONS

Momentum

```
emoon = (1 + (2*Emoon*hmoon^(2))/(muL^(2)))^(0.5);
```

%Moon Trajectory

Eccentricity

```
pmoon = hmoon^(2)/muL;
```

%Moon Trajectory Semilatus

```
rectus
```

%We need to check that the Moon Orbit Periapsis doesnt go through the Moon

%Surface

```
if rpmoon < RM + 10
```

```
    disp('The Moon Iperbolic Trajectory intersect the moon surface')
```

%We need to know which parameter not to save, to do so we

%will use this value

```
breaking = 1;
```

```
return
```

```
else
```

```
end
```

```
breaking = 0;
```

```
end
```

%-----DELTA V CALCULATIONS MOON BI-IMPULSE-----%

%Here we will calculate both the deltaV required for the injection on the

%departure orbit (from the parking one) and for the insertion manouver into

%the halo orbit

```
function [DeltaVinsMag,DeltaVScMag,DeltaVtotal,yq0,DeltaV1,DeltaV2] =
```

```
    DeltaVCalcSeleBiImpulse(v0,vcircular,phi0,RInterceptP1,RInterceptP2,lambda1,ni2,k,
```

```
    q,vrpmoon,vellpmoon,vcircularmoon,vellamoon,P1L1,P1L2,P2L1,P2L2,yq0,DeltaVinsMag,
```

```
    DeltaVtotal,DeltaVScMag,DeltaV1,DeltaV2)
```

```
global muL
```

%Departure manouver deltaV required

```
DepBurnDeltaV = (v0^2 + vcircular^2 - 2*v0*vcircular*cos(phi0))^(0.5);
```

%First Burn, we impose that the apoapsis radius of the elliptical orbit

%around the moon has to be equal to the "halo interception point" radius



```
DeltaV1(k,q) = vrpmoon - vellpmoon; %Now we compute the
deltaV required for the first manouver

%Second Burn, we circularize the orbit at the halo points interception
%radius
DeltaV2(k,q) = vcircularmoon - vellamoon; %Now we compute the
deltaV required for the first manouver

%Third Burn, required for the halo insertion, we compute both the deltv
%requirements for the insertion on P1 and P2. We will choose the cheaper
%one
%We calculate the true anomaly of the P1 point with respect to the
%selenocentri trajectory periapsis
[thetaP1, ~] = cart2pol(-P1L1, -P1L2);
niInterceptP1 = thetaP1 + ni2 + lambda1;

%We calculate the true anomaly of the P2 point with respect to the
%selenocentri trajectory periapsis
[thetaP2, ~] = cart2pol(-P2L1, -P2L2);
niInterceptP2 = (2*pi+thetaP2) + ni2 + lambda1;

vP1 = [-0.0562 0.0543 0.1013]; %Halo orbit velocity required
for entering P1
vP2 = [+0.0562 -0.0543 -0.1013]; %Halo orbit velocity required
for entering P2

vscp1PQWtan = ((muL/RInterceptP1)^(0.5))*(-sin(niInterceptP1)); %Sc
tangential velocity at the interception point with P1
vscp1PQWrad = ((muL/RInterceptP1)^(0.5))*(cos(niInterceptP1)); %Sc radial
velocity at the interception point with P1
vscp1PQW = [vscp1PQWtan vscp1PQWrad 0]; %Sc velocity
at the interception point with P1, vectorial form

vscp2PQWtan = ((muL/RInterceptP2)^(0.5))*(-sin(niInterceptP2)); %Sc
tangential velocity at the interception point with P2
vscp2PQWrad = ((muL/RInterceptP2)^(0.5))*(cos(niInterceptP2)); %Sc radial
velocity at the interception point with P2
vscp2PQW = [vscp2PQWtan vscp2PQWrad 0]; %Sc velocity
at the interception point with P2, vectorial form

%Rotation matrix from PQW for P1
```

C. CALCULATIONS

```
T = [cos(-(lambda1+ni2)) sin(-(lambda1+ni2)) 0
     -sin(-(lambda1+ni2)) cos(-(lambda1+ni2)) 0
     0 0 1];
vscp1XYZ = T*vscp1PQW';
vscp2XYZ = T*vscp2PQW';

%Now we can compute the DeltaV required for the Halo orbit insertion in P1
DeltaVinsP1 = vP1 - vscp1XYZ';
DeltaVinsMagP1 = (DeltaVinsP1(1)^2 + DeltaVinsP1(2)^2 + DeltaVinsP1(3)^2)^(0.5);

%Now we can compute the DeltaV required for the Halo orbit insertion in P2
DeltaVinsP2 = vP2 - vscp2XYZ';
DeltaVinsMagP2 = (DeltaVinsP2(1)^2 + DeltaVinsP2(2)^2 + DeltaVinsP2(3)^2)^(0.5);

%Now we choose the best entering point
if DeltaVinsMagP1 < DeltaVinsMagP2
    DeltaVinsMag(k,q) = DeltaVinsMagP1;
    DeltaVScMag(k,q) = DeltaV1(k,q) + DeltaV2(k,q) + DeltaVinsMagP1;
    DeltaVtotal(k,q) = DeltaV1(k,q) + DeltaV2(k,q) + DeltaVinsMagP1 + DepBurnDeltaV;
    disp('The total DeltaV required for the halo orbit insertion would be:')
    disp([num2str(DeltaVinsMag(k,q)), ' [km/s]'])

else
    DeltaVinsMag(k,q) = DeltaVinsMagP2;
    DeltaVScMag(k,q) = DeltaV1(k,q) + DeltaV2(k,q) + DeltaVinsMagP2;
    DeltaVtotal(k,q) = DeltaV1(k,q) + DeltaV2(k,q) + DeltaVinsMagP2 + DepBurnDeltaV;
    disp('The total DeltaV required for the halo orbit insertion would be:')
    disp([num2str(DeltaVinsMag(k,q)), ' [km/s]'])

end

yqθ(k,q) = lambda1*180/pi;
disp('The deltaV required for the translunar insertion manouver would be: ')
disp(['For a total deltaV of: ', num2str(DeltaVtotal(k,q)), ' [km/s]'])
disp(' ')

end
```



```
%-----%
%----- BI ELLIPTIC ORBIT DEFINITION -----%
function [E1,h1,r1,v1,phi1,gamma1,p1,a1,e1,ni0,ni1,ecc0,ecc1,gamma0,FlightTimeTotal,v2
,epsilon2,E2,h2,e2,p2,ni2,a2,breaking,v01a,a01,v01p,lambda1,h01,e01,p01,
FlightTimeFirstEllipse,FlightTimeSecondEllipse] = EllipseDepOrbit(r0,v0,lambda1,
phi0,v1Max,RInterceptP1)
global muT muL rTL RsoiT vL omegaL RsoiT

%From the starting circular parking orbit around Earth, the spacecraft will
%enter into an elliptic orbit that will have its apogee at the Earth's Soi
% edge. This elliptic orbit will have its dimensions fixed.

%First Ellipse characteristic definition
a01 = (r0 + RsoiT)/2; %Semimajor axis of the
first Ellipse
v01p = (2*muT*((1/r0 - 1/(2*a01)))^(0.5)); %First Ellipse Trajectory
sc velocity at periapsis
E01 = (v01p^2)/2 - muT/r0; %First Ellipse Trajectory
Mechanical Specific Energy
v01a = (2*(E01 + muT/RsoiT))^(0.5); %First Ellipse Trajectory
sc velocity at apoapsis
h01 = v01p*r0; %First Ellipse Angular
momentum at periapsis
e01 = (1 + (2*E01*h01^(2))/(muT^(2)))^(0.5); %First Ellipse
Eccentricity
p01 = h01^(2)/muT; %First Ellipse Semilatus
rectus

%Because this first ellipse has an apoapsis higher than the Moon orbital
%radius, the true anomaly would be defined from the entering point in the
%moon soi along the second transfer ellipse.

%The v0 initial velocity from the First Ellipse apoapsis would be iterated
E1 = (v0^2)/2 - muT/RsoiT; %Second Ellipse Mechanical Specific
Energy
h1 = RsoiT*v0*cos(phi0*pi/180); %Second Ellipse Angular Momentum

%Now we initialize the Sc Flight Entering Flight Fath Angle in the Moon
%Soi. We assing it an imaginary number so the while loop below can start.
phi1 = 0.1i;
i = 0;
```

C. CALCULATIONS

```
%Different from the other two methods, fixing the v0 velocities at the
%apoapsis of the second Ellipse "fix" also the entering point on the moon
%soi. We now have a minimum entering angle on the moon soi, going below it
%its not possible and this would reflected on the phi1 angle becoming
%imaginary.

%So with each velocity iteration we will also iterate the lambda1 parameter
%to achieve the right entering angle, by checking when the parameter r1
%makes the term phi1 no longer imaginary

while isreal(phi1) == 0
    if v0 >=v1Max
        breaking = 1;
        r1 = 0;
        v1= 0;
        break

    else
        %Updating the lambda1 value
        lambda1 = lambda1 + 0.001;
        r1 = (rTL^2 + RsoiL^2 - 2*rTL*RsoiL*cos(lambda1))^(0.5);      %Sc Position
        vector
        v1 = (2*(E1 + muT/r1))^(0.5);                                     %Sc speed just
        before entering the Moon Soi
        phi1 = acos(h1/(r1*v1));                                         %Sc flight path
        angle entering the Moon Soi
        i = i + 1;

        if i > 10^3
            break
        end

    end

end

%Now we calculate the arrival (at the Moon Soi) parameters from Earth
%reference system point of view
gamma1 = asin((RsoiL/r1)*sin(lambda1));                                %Sc Entering Lunar Soi
    Angle from Earth view point
p1 = (h1^(2))/muT;                                                       %Geocentric Departure
    orbit Semilatus rectus
```

```

a1 = -muT/(2*E1);                                     %Geocentric Departure
  orbit Semimajor Axis
e1 = (1-p1/a1)^(0.5);                               %Geocentric Departure
  orbit eccentricity
ni0 = real(acos((p1-RsoiT)/(RsoiT*e1)));           %Starting true anomaly at
  the apoapsis of the First Ellipse
ni1 = acos((p1-r1)/(r1*e1));                         %Arriving true anomaly (at
  the moon Soi)
ecc0 = real(acos((e1 + cos(ni0))/(1 + e1*cos(ni0)))); %Starting eccentric
  anomaly (at the apoapsis of the First Ellipse)
ecc1 = acos((e1 + cos(ni1))/(1 + e1*cos(ni1)));      %Arriving eccentric
  anomaly (at the moon Soi)

%We compute both the flight time along both first and second Ellipse
FlightTimeFirstEllipse = pi*(a01^3/muT)^(0.5);          %Flight Time along the First Ellipse
FlightTimeSecondEllipse = ((a1^3/muT)^(0.5))*((ecc0-e1*sin(ecc0))-((ecc1-e1*sin(ecc1))) );
  ); %Flight Time along the Second Ellipse
FlightTimeTotal = FlightTimeFirstEllipse + FlightTimeSecondEllipse; %Total Flight Time
gamma0 = ni1 - ni0 - gamma1 - omegaL*FlightTimeTotal;    %Departure Phasing angle

%-----CONDITION AT THE PATCH POINT -----
v2 = (v1^2 + vL^2 - 2*v1*vL*cos(phi1-gamma1))^(0.5);   %Sc entering velocity into
  Moon SOI
epsilon2 = asin((vL/v2)*cos(lambda1)-(v1/v2)*cos((lambda1 + gamma1 - phi1)));

%-----SELENOCENTRIC ARRIVAL ORBIT -----
E2 = (v2^2)/2 - muL/RsoiL;                                %Moon Trajectory
  Mechanical Specific Energy
h2 = RsoiL*v2*sin(epsilon2);                               %Moon Trajectory Angular
  Momentum
e2 = (1 + (2*E2*h2^2)/(muL^2))^(0.5);                   %Moon Trajectory
  Eccentricity
p2 = h2^2/muL;                                            %Moon Trajectory Semilatus
  rectus
a2 = -muL/2*E2;                                           %Moon Trajectory semimajor
  axis

```



C. CALCULATIONS

```
%We hypothesize that the entering point on the moon soi coincides with the
%position of the halo intercepts points
ni2 = acos((1/e2)*(p2/RInterceptP1 - 1)); %Moon Trajectory Enter
point True Anomaly

%We need to check if the Sc velocity at the moon Soi interception is positive
%if not, the initial departure trajectory was not sufficiently energetic
if isreal(v1) == 0
    disp('The initial departure trajectory was not energetic enough to reach')
    disp('the moon Soi')
    breaking = 1;

    return

end

if v0 >=v1Max
    breaking = 1;
else
    breaking = 0;
end
end

%-----SELENOCENTRIC PERIAPSIS POSITION BI-ELLIPTIC-----%
function [AimedPoint,niIntercept,breaking] = MoonPeriapsisPositionBiElliptic(p2,e2,ni2
,lambda1,P1L1,P1L2,P2L1,P2L2)
global RM

%Verifying the selenocentric periapsis position
if (ni2+(lambda1)) < pi
    disp('The lunar fly by trajectory periapsis is under the Earth-Moon axe')

    if (ni2+(lambda1)) < pi/2
        disp('and we are going to aim for P2 interception point')
        AimedPoint = 2;

        %We calculate the true anomaly of the P2 point with respect to the
```



```
%selenocentri trajectory periapsis
[thetaP2, ~] = cart2pol(-P2L1, -P2L2);
niIntercept = -(2*pi+thetaP2) + ni2 + lambda1;

elseif (ni2+(lambda1)) > pi/2 || (ni2+(lambda1)) < 2*pi
    disp('and we are going to aim for P1 interception point')
    AimedPoint = 1;

    %We calculate the true anomaly of the P1 point with respect to the
    %selenocentri trajectory periapsis
    [thetaP1, ~] = cart2pol(-P1L1, -P1L2);
    niIntercept = -thetaP1 + ni2 + lambda1;

end

elseif (ni2+(lambda1)) > pi
    disp('The lunar fly by trajectory periapsis is above the Earth-Moon axe')

if -(ni2+(lambda1)) < -pi/2
    disp('and we are going to aim for P1 interception point')
    AimedPoint = 1;

    %We calculate the true anomaly of the P1 point with respect to the
    %selenocentri trajectory periapsis
    [thetaP1, ~] = cart2pol(-P1L1, -P1L2);
    niIntercept = -thetaP1 + ni2 + lambda1;

elseif -(ni2+(lambda1)) > -pi/2 || -(ni2+(lambda1)) < -2*pi
    disp('and we are going to aim for P2 interception point')
    AimedPoint = 2;

    %We calculate the true anomaly of the P2 point with respect to the
    %selenocentri trajectory periapsis
    [thetaP2, ~] = cart2pol(-P2L1, -P2L2);
    niIntercept = -(2*pi+thetaP2) + ni2 + lambda1;

end

end
```



C. CALCULATIONS

```
%We need to check that the Moon Orbit Periapsis doesnt go through the Moon
%Surface. This would only be required if we are aiming to reach P2
rmoon = p2/(1+e2); %Moon Trajectory periapsis
```

```
if AimedPoint == 2
    if rmoon < RM + 75
        disp('The Moon Iperbolic Trajectory intersect the moon surface')
        disp(' ')
    end
```

```
%We need to know which parameter not to save, to do so we
%will use this value
breaking = 1;
```

```
return
else
end
else
end
```

```
breaking = 0;
end
```

```
%-----OPTIMAL SOI ENTER ANGLE BI-ELLIPTIC-----%
%In the hypothesis that the selenocentric trajectory exiting soi point
%coincides with the halo interception point (that's should produce a very
%small error), we write the "angle difference" function between the two
%points and we reiterate the precedent calculations until we achieve
%an "angle difference" value under our error tolerance.
%The idea is that to achieve an encounter we can try to "steer" the
%insertion orbit pivoting it around the moon, in doing so we need to change
%the soi entering angle lambda1. We could also maintain the same entering
%angle and change the entering velocity, but this will increase the DeltaV
%requirements for our mission.
```

```
function [yq0,breaking,lambda0pt,E1,h1,r1,v1,phi1,gamma1,p1,a1,e1,ni0,ni1,ecc0,ecc1,
gamma0,FlightTimeTotal,v2,epsilon2,E2,h2,e2,p2,ni2,a2,v01a,a01,v01p,niIntercept,
h01,e01,p01,FlightTimeFirstEllipse,FlightTimeSecondEllipse] =
OptimalSoiEnteringAngleBiElliptic(r0,v0,lambda1,phi0,niIntercept,ni2,P1L1,P1L2,
P2L1,P2L2,q,yq0,k,AimedPoint,v1Max,e2,p2,RInterceptP1)
```



```
tol = 0.005; %tolerance [deg]
i = 0; %Iteration counter
lambdaMin = lambda1; %Minimum value acceptable for lambda1
breaking = 0;

%Angle difference definition
AngleDiff = niIntercept - acos((1/e2)*(p2/RInterceptP1 - 1));

disp(['The minimum entering angle in the moon soi its: ',num2str(lambdaMin)])

%Iterative Cycle
while abs(AngleDiff) > ((tol*pi/180))

    AngleDiff = niIntercept - acos((1/e2)*(p2/RInterceptP1 - 1));
    lambda0pt = (lambda1) - AngleDiff/4;
    lambda1 = lambda0pt;

    %-----BI-ELLIPTIC TRANSFER ORBIT -----
    [E1,h1,r1,v1,phi1,gamma1,p1,a1,e1,ni0,ni1,ecc0,ecc1,gamma0,FlightTimeTotal,v2,
     epsilon2,E2,h2,e2,p2,ni2,a2,~,v01a,a01,v01p,lambda1,h01,e01,p01,
     FlightTimeFirstEllipse,FlightTimeSecondEllipse] = EllipseDepOrbit(r0,v0,
     lambda1,phi0,v1Max,RInterceptP1);

    %Verifying the selenocentric periapsis position
    if AimedPoint == 1

        %We calculate the true anomaly of the P1 point with respect to the
        %selenocentric trajectory periapsis
        [thetaP1, ~] = cart2pol(-P1L1, -P1L2);
        niIntercept = -thetaP1 + ni2 + lambda1;

    elseif AimedPoint == 2

        %We calculate the true anomaly of the P2 point with respect to the
        %selenocentric trajectory periapsis
        [thetaP2, ~] = cart2pol(-P2L1, -P2L2);
        niIntercept = -(2*pi+thetaP2) + ni2 + lambda1;

    end
```

C. CALCULATIONS

```
i = i + 1; %n    of iterations done

%This code underneat will exit the while cycle if the
%"AngleDiff" parameter become imaginary or the number of
%iterations starts to be too high
if isreal(AngleDiff) == 0 || i>(10^3)
    disp('The Second Ellipse apoapsis velocity is not energetic enough to')
    disp('reach the required point on the moon soi')
    breaking = 1;
    break

else
end

end

%This code underneat will exit the for cycle if the
%"AngleDiff" parameter became imaginary, skipping directly
%the deltaV calculations
if isreal(AngleDiff) == 0 || breaking == 1
    disp('The actual trajectory doesnt permits to reach the interception points')
    disp(' ')
    breaking = 1;
    yq0(k,q) = 0;
    return
else

end

%This code underneat will exit the for cycle if the
%"LambdaOpt" parameter became less than lambdaMin, skipping directly
%the deltaV calculations
if lambdaOpt < lambdaMin
    disp('The actual trajectory doesnt permits to reach the interception points')
    disp(' ')
    breaking = 1;
    yq0(k,q) = 0;
    return
else

end
```



```
%obtained Results display
disp('to achieve an encounter, the entering angle on the moon soi should be: ')
disp([num2str(lambda1*180/pi), ' degree'])
disp(['number of orbit iterations: ',num2str(i)])
yq0(k,q) = lambda1*180/pi;

breaking = 0;
end

%-----DELTA V CALCULATIONS BI-ELLIPTIC-----
function [DeltaVinsMag,DeltaVScMag,DeltaVtotal,yq0] = DeltaVCalcBiElliptic(v0,
    vcircular,v01a,v01p,niIntercept,p2,lambda1,ni2,k,q,AimedPoint,e2,yq0,DeltaVinsMag,
    DeltaVtotal,DeltaVScMag)
global muL

%Now we calculate the deltaV required for inserting in the first Ellipse
DeltaV1 = v01p - vcircular;

%Now we calculate the deltaV required for inserting in the second Ellipse
DeltaV2 = v0 - v01a;

if AimedPoint == 1
    vP1 = [-0.0562 0.0543 0.1013]; %Halo orbit velocity
    required for entering P1
    vscp1PQWtan = ((muL/p2)^(0.5))*(-sin(-niIntercept)); %Sc tangential
    velocity at the interception point with P1
    vscp1PQWrad = ((muL/p2)^(0.5))*(e2 + cos(-niIntercept)); %Sc radial velocity
    at the interception point with P1
    vscp1PQW = [vscp1PQWtan vscp1PQWrad 0]; %Sc velocity at the
    interception point with P1, vectorial form

    %Rotation matrix from PQW to XYZ
    T = [cos(-(lambda1+ni2)) sin(-(lambda1+ni2)) 0
        -sin(-(lambda1+ni2)) cos(-(lambda1+ni2)) 0
        0 0 1];
    vscp1XYZ = T*vscp1PQW';

    %Now we can compute the DeltaV required for the Halo orbit insertion
    DeltaVins = vP1 - vscp1XYZ';
    DeltaVinsMag(k,q) = (DeltaVins(1)^2 + DeltaVins(2)^2 + DeltaVins(3)^2)^(0.5);
```

C. CALCULATIONS

```
DeltaVScMag(k,q) = (DeltaVins(1)^2 + DeltaVins(2)^2 + DeltaVins(3)^2)^(0.5) +
    DeltaV2;
DeltaVtotal(k,q) = (DeltaVins(1)^2 + DeltaVins(2)^2 + DeltaVins(3)^2)^(0.5) +
    DeltaV1 + DeltaV2;

elseif AimedPoint == 2
    vP2 = [+0.0562 -0.0543 -0.1013];                                %Halo orbit velocity
    required for entering P2
    vscp2PQWtan = ((muL/p2)^(0.5))*(-sin(-niIntercept));          %Sc tangential
    velocity at the interception point with P2
    vscp2PQWrad = ((muL/p2)^(0.5))*(e2 + cos(-niIntercept));      %Sc radial velocity
    at the interception point with P2
    vscp2PQW = [vscp2PQWtan vscp2PQWrad 0];                         %Sc velocity at the
    interception point with P2, vectorial form

    %Rotation matrix from PQW to XYZ
    T = [cos(-(lambda1+ni2)) sin(-(lambda1+ni2)) 0
        -sin(-(lambda1+ni2)) cos(-(lambda1+ni2)) 0
        0 0 1];
    vscp2XYZ = T*vscp2PQW';

    %Now we can compute the DeltaV required for the Halo orbit insertion
    DeltaVins = vP2 - vscp2XYZ';
    DeltaVinsMag(k,q) = (DeltaVins(1)^2 + DeltaVins(2)^2 + DeltaVins(3)^2)^(0.5);
    DeltaVScMag(k,q) = (DeltaVins(1)^2 + DeltaVins(2)^2 + DeltaVins(3)^2)^(0.5) +
        DeltaV2;
    DeltaVtotal(k,q) = (DeltaVins(1)^2 + DeltaVins(2)^2 + DeltaVins(3)^2)^(0.5) +
        DeltaV1 + DeltaV2;

end

disp('The deltaV required for the halo orbital insertion with this transfer')
disp(['manouver is: ',num2str(DeltaVinsMag(k,q)), ' [km/s]'])
disp('The deltaV required for the translunar insertion manouever would be: ')
disp(['For a total deltaV of: ', num2str(DeltaVtotal(k,q)), ' [km/s]'])
disp('')

end
```



```
%-----%
%-----EXTRACTING VALUES-----%
function [velocityA,AngleA,row,col] = ValuesExtractor3000(A,DeltaVScMag,xv0,yq0)
%We use this cycle to extract the "best" 5 values inside each matrix.
%The "+0.001" helps to get a different results, not the same result
%obtained before

for j = 1:5
    %We use this if statement to overwrite the first element of A
    if j == 1
        A(j) = min(DeltaVScMag(DeltaVScMag>(A(j)+0.001)));
    else
        A(j) = min(DeltaVScMag(DeltaVScMag>(A(j-1)+0.001)));
    end

    %Now we need to extrapolate the coordinates of said DeltaV values from the
    %matrix
    [row(j),col(j)] = find(DeltaVScMag==A(j));

    %At last we need to determinate the velocity and angle values that
    %generate that deltaV values, knowing their coordinates we can go back
    %inside the xv0 and yq0 vectors
    velocityA(j) = xv0(row(j));           %Velocities vector
    AngleA(j) = yq0(row(j),col(j));       %Angle vector

end

end
```

C.2. Propulsion Calculations

%Propulsion and Mass Budget script For SpaceSystem Project
 %FRANCESCO MARRADI - Support to Human Activities from EML2

```
close all
clear
clc
```

%Questo script verrà usato per comparare i diversi sistemi propulsivi
 %elettrici, confrontando come diversi motori si comportano al variare della
 %massa da mantere in stationkeeping attorno alla halo orbit.
 %Le tecnologie di propulsione analizzata sono sia Gridded ion che Hall
 %Thrusters. Per quanto riguarda i propellenti, dall'analisi fatta abbiamo
 %deciso di utilizzare il krypton

```
%-----GENERAL PARAMETERS-----%
%Constants
global g0 %#ok<*GVMIS>
g0 = 9.813; %Gravity's acceleration value [m/s]

%General Parameters
MissionTime = 10; %Mission duration time [years]
Torbit = 15; %Halo Orbital period [days]
SolarPanelValue = 300/6; %Watt generated by 1kg of solar panels[W/kg]

%Propulsion Parameters
AnnualDvStkChem = 22; %Annual deltaV consumption for Chemical StationKeeping [m
/s] (from Queqiao)
AnnualDvStkEl = 40; %Annual deltaV consumption for Electrical StationKeeping
[m/s]
DeltaVInsBurn = 660; %Deltav insertion burn [m/s]
ChemEngMass = 0.65*4; %Mass of the chemical engines for insertion [kg] (From
Queqiao)
ChemEngIsp = 211; %Queqiao Hydrazine engine Isp [s]
ChemEngStkMass = 0.4*10; %Mass of the Stationkeeping engines from Queqiao
configuration, 10 engines weighing 0.4kg [kg]
MargChem = 1; %Margin coefficient
MargEl = 1; %Margin coefficient to take into account ppu's, tank and
structure masses
```



```
%Definiamo un vettore contenente i valori delle masse su cui andare ad  
%effettuare i calcoli  
ScMasstoStkOG = [100 120 140 160 180 200];
```

```
%-----CHOOSING THE PROPULSION SYSTEM TO COMPUTE-----%  
disp('Choose the propulsion system to use for the calculations: ')  
disp('Press "1" for "Hermes Hybrid propulsion")  
disp('Press "2" for "Hermes Chemical propulsion")  
disp('Press "3" to analyze both propulsion systems')  
choice = input(''); %Takes the user input  
disp('')  
disp(['Mission time set to: ',num2str(MissionTime),' years'])  
disp('')  
disp('-----')
```

```
%-----IMPULSIVE APPROXIMATION FOR ELECTRIC THRUSTERS-----%  
%Nel caso sia stata scelta la configurazione ibrida, definiamo  
%l'approssimazione impulsiva per stimare i calcoli dello stationkeeping  
%elettrico.  
if choice == 1 || choice == 3  
    [DeltaVskburn,tMaxManouver] = ImpulsiveApproximation(Torbit,AnnualDvStkEl);  
  
end
```

```
%-----CALCULATIONS-----%  
%In questa sezione andremo a fare tutti i calcoli relativi alle stime delle  
%masse. Il procedimento nei calcoli è il seguente:  
%Massa sc -> seleziono un motore per fare lo stationkeeping -> faccio i  
%calcoli con tutti i propellenti che può usare.  
%Testo tutti i motori, poi cambio massa della sc e rifaccio i conti.
```

```
%DeltaV for Electrical Stationkeeping [m/s]  
DeltaVskEl = MissionTime*AnnualDvStkEl;
```

C. CALCULATIONS

```
%Definiamo una funzione "catalogo" delle performance dei diversi motori
%Elettrici, suddivisi per tecnologia propulsiva
[ElEngThrust,ElEngIsp,ElEngPower,ElEngMass] = ElectricThrustersCharacteristics();  
  

%Inizializzazione di alcune variabili
%Definiamo il peso delle tank come inizialmente nullo, verrà poi calcolato
%successivamente
TankInsChem = zeros(length(ScMasstoStk0G),length(ElEngThrust)); %Peso dei tank per
    l'inserimento nella halo[kg]
TankInsChemChem = zeros(length(ScMasstoStk0G),length(ElEngThrust)); %Peso dei tank
    per l'inserimento nella halo[kg]
TankStkChem = zeros(length(ScMasstoStk0G)); %Peso dei tank per
    lo stationkeeping[kg]
TankStkEl = zeros(length(ScMasstoStk0G),length(ElEngThrust)); %Peso dei tank per
    lo stationkeeping[kg]
MpropInsEl = zeros(length(ScMasstoStk0G),length(ElEngThrust));
MpropInsChem = zeros(length(ScMasstoStk0G),length(ElEngThrust));
MpropChem = zeros(length(ScMasstoStk0G)); %Massa del
    propellente richiesto per lo stationkeeping [kg]
MpropEl = zeros(length(ScMasstoStk0G),length(ElEngThrust));
n_eng = zeros(length(ScMasstoStk0G),length(ElEngThrust)); %Numero di
    Electric Thursters richiesti dal sistema propulsivo
MtotEl = zeros(length(ScMasstoStk0G),length(ElEngThrust));
ChemTotalMass = zeros(length(ScMasstoStk0G),length(ElEngThrust));
EpTotalMass = zeros(length(ScMasstoStk0G),length(ElEngThrust));
EpSystMass = zeros(length(ScMasstoStk0G),length(ElEngThrust));
MinThrust = zeros(length(ScMasstoStk0G),length(ElEngThrust));
ThrustersMass = zeros(length(ScMasstoStk0G),length(ElEngThrust));
SolarPanelMass = zeros(length(ScMasstoStk0G),length(ElEngThrust));
ThrustersConsumption = zeros(length(ScMasstoStk0G),length(ElEngThrust));  
  

%Con questo if andiamo a "trasformare" il vettore ScMasstoStk0G come una
%matrice avente il numero delle righe pari al numero delle masse e quello
%delle colonne pari a quello dei motori.
ScMasstoStk0GM = zeros(length(ScMasstoStk0G),length(ElEngThrust));
ScMasstoStkEl = zeros(length(ScMasstoStk0G),length(ElEngThrust));
ScMasstoStkChem = zeros(length(ScMasstoStk0G),length(ElEngThrust));
for i = 1 : length(ElEngThrust)
ScMasstoStk0GM(:,i) = ScMasstoStk0G;
```



```
ScMasstoStkEl(:,i) = ScMasstoStk0G;
ScMasstoStkChem(:,i) = ScMasstoStk0G;
end

if choice == 1 || choice == 2 || choice == 3

    %Il ciclo for permette di reiterare i calcoli, andando a raffinare
    %i valori trovati.
    for its = 1:10

        %Con questo ciclo for andiamo a selezionare il valore della massa della
        %spacecraft da usare per il calcolo
        for i = 1 : size(ScMasstoStk0GM,1)

            %Qua dentro inseriamo la funzione per il calcolo delle masse
            %nell'ibrido
            if choice == 1

                %Con questo ciclo vado a selezionare il motore da usare nei calcoli
                for j = 1 : length(ElEngThrust)

                    %Per prima cosa si calcola il propellente per l'inserimento
                    %nella halo
                    [ScMasstoStkEl,MpropInsEl,TankInsChem] = HaloInsertionProp (
                        ScMasstoStk0GM,ScMasstoStkEl,ChemEngMass,MargChem,ChemEngIsp,
                        DeltaVInsBurn,MpropInsEl,TankInsChem,i,j);

                    %Poi calcoliamo la massa del propellente richiesto allo
                    %stationkeeping per ogni tecnologia di motori usati.
                    [EpTotalMass,EpSystMass,ThrustersMass,SolarPanelMass,MpropEl,
                     TankStkEl,ScMasstoStkEl,n_eng,MinThrust] = ElecStationkeeping(
                        ElEngThrust,ElEngIsp,ElEngPower,ElEngMass,DeltaVskEl,
                        ScMasstoStkEl,ScMasstoStk0GM,DeltaVskburn,tMaxManouver,
                        SolarPanelValue,MargEl,MtotEl,EpTotalMass,EpSystMass,
                        ThrustersMass,SolarPanelMass,MpropEl,TankStkEl,TankInsChem,
                        n_eng,MinThrust,ThrustersConsumption,i,j);

                end
            end
        end
    end
end
```



```
elseif choice == 2
    %Qua dentro inseriamo la funzione per il calcolo delle masse
    %nel full chimico.
    %Per prima cosa si calcola il propellente per l'inserimento
    %nella halo
    j = 1;
    [ScMasstoStkChem,MpropInsChem,TankInsChemChem] = HaloInsertionProp (
        ScMasstoStkOGM,ScMasstoStkChem,ChemEngMass,MargChem,ChemEngIsp,
        DeltaVInsBurn,MpropInsChem,TankInsChem,i,j);

    %Poi calcoliamo la massa del propellente richiesto allo
    %stationkeeping
    [MtotChem,ChemTotalMass,MpropChem,TankStkChem,ScMasstoStkChem] =
        ChemStationkeeping(MpropChem,TankStkChem,TankInsChemChem,
        MissionTime,AnnualDvStkChem,ScMasstoStkChem,ScMasstoStkOGM,
        ChemEngStkMass,ChemEngIsp,MargChem,ChemTotalMass,i,j);

elseif choice == 3
    %Con questo ciclo vado a selezionare il motore da usare nei calcoli
    for j = 1 : length(ElEngThrust)

        %Per prima cosa si calcola il propellente per l'inserimento
        %nella halo
        [ScMasstoStkEl,MpropInsEl,TankInsChem] = HaloInsertionProp (
            ScMasstoStkOGM,ScMasstoStkEl,ChemEngMass,MargChem,ChemEngIsp,
            DeltaVInsBurn,MpropInsEl,TankInsChem,i,j);

        %Poi calcoliamo la massa del propellente richiesto allo
        %stationkeeping per ogni tecnologia di motori usati.
        [EpTotalMass,EpSystMass,ThrustersMass,SolarPanelMass,MpropEl,
         TankStkEl,ScMasstoStkEl,n_eng,MinThrust] = ElecStationkeeping(
            ElEngThrust,ElEngIsp,ElEngPower,ElEngMass,DeltaVskEl,
            ScMasstoStkEl,ScMasstoStkOGM,DeltaVskburn,tMaxManouver,
            SolarPanelValue,MargEl,MtotEl,EpTotalMass,EpSystMass,
            ThrustersMass,SolarPanelMass,MpropEl,TankStkEl,TankInsChem,
            n_eng,MinThrust,ThrustersConsumption,i,j);

        %Qua dentro inseriamo la funzione per il calcolo delle masse
        %nel full chimico.
        %Per prima cosa si calcola il propellente per l'inserimento
```



```
%nella halo
j = 1;
[ScMasstoStkChem,MpropInsChem,TankInsChemChem] = HaloInsertionProp
(ScMasstoStk0GM,ScMasstoStkChem,ChemEngMass,MargChem,
ChemEngIsp,DeltaVInsBurn,MpropInsChem,TankInsChemChem,i,j);

%Poi calcoliamo la massa del propellente richiesto allo
%stationkeeping
[MtotChem,ChemTotalMass,MpropChem,TankStkChem,ScMasstoStkChem] =
ChemStationkeeping(MpropChem,TankStkChem,TankInsChemChem,
MissionTime,AnnualDvStkChem,ScMasstoStkChem,ScMasstoStk0GM,
ChemEngStkMass,ChemEngIsp,MargChem,ChemTotalMass,i,j);

end

end

end

else
    disp('You have pressed the wrong choice''s number')

end

%% -----PLOTTING DATA-----
%Plotting the Ep Propulsion system mass required for the impulsive approximation
figure(1)
hold on
title('Electric Propulsion Stationkeeping')
xlabel('Satellite Mass to Stationkeep in Halo orbit [kg]')
ylabel('Equipment Mass Required[kg]')
Results = [TankInsChem(:,5) TankStkEl(:,5) EpSystMass(:,5) MpropInsEl(:,5) MpropEl
(:,5)];
bar(ScMasstoStk0GM(:,5),Results, 'stacked')
legend('Chem Ins Tank Mass','Elec Tank Mass','Electric Propulsion System Mass',
'Insertion Prop Mass','Stk Prop Mass')
```



C. CALCULATIONS

hold off

```
%Plotting the Chemical Propulsion system mass required
figure(2)
hold on
title('Chemical Propulsion Stationkeeping')
xlabel('Satellite Mass to Stationkeep in Halo orbit [kg]')
ylabel('Equipment Mass Required[kg]')
Results = [TankInsChemChem(:,1) TankStkChem(:,1) (zeros(1,length(MpropChem))+ChemEngMass)' MpropInsChem(:,1) MpropChem(:,1)];
bar(ScMasstoStkOGM(:,1),Results, 'stacked')
legend('Chem Ins Tank Mass','Chem Stk Tank Mass','Chemical Propulsion System Mass','
Insertion Prop Mass','Stk Prop Mass')
hold off

%Plotting the mass differences between the two propulsion systems
figure(3)
hold on
set(gca, 'FontSize', 19, 'LineWidth', 1.4); %Set the font size and line
width for the axes
title('Electrical Prop. Stationkeeping vs Chemical Prop. Stationkeeping')
xlabel('Satellite Mass to Stationkeep in Halo orbit [kg]')
ylabel('Mass difference [kg]')

%Grafico i risultati
Chem_El_System_Mass = (TankInsChemChem(:,1) + TankStkChem(:,1) + (zeros(1,length(
MpropChem))+ChemEngMass)' - (TankInsChem(:,5) + TankStkEl(:,5) + EpSystMass(:,5))
;
Chem_El_Prop_Mass = (MpropChem(:,1) - MpropEl(:,5));
Mass_Penalty = Chem_El_System_Mass + Chem_El_Prop_Mass;
Results = [Chem_El_System_Mass'; Chem_El_Prop_Mass';Mass_Penalty'];
bar(ScMasstoStkOG,Results,1,'grouped')
correctlegend = legend('(Chem - El) System Mass [kg]', '(Chem - El) Propellant Mass [
kg]','Mass Penalty [kg]');
correctlegend.Box = "off";
hold off
```

%% -----FUNZIONI USATE NELLO SCRIPT-----&

Questo sarà lo spazio dove andremo ad inserire tutte le funzioni usate
nello script, in modo da scriverlo bello pulito e ordinato



```
%-----VALUES OF ELECTRIC ENGINES-----%
%Busek Hall Thruster Parameters: https://www.busek.com/hall-thrusters
%Safran Hall Thruster PPSX00: https://www.safran-group.com/products-services/ppsrx00-stationary-plasma-thruster
>All the parameters are loaded into vectors

function [ElEngThrust,ElEngIsp,ElEngPower,ElEngMass] =
ElectricThrustersCharacteristics()

%Hall Thrusters
%[BHT-100 BHT-200 BHT-350 BHT-600 PPSX00mod1 PPSX00mod2]
HtThrust = [7 13 17 39 15 75]; %Thrust [mN]
HtIsp = [1000 1390 1244 1500 1300 1650]; %Isp [s]
HtPower = [100 200 300 600 200 1000]; %Power consumption [W]
HtMass = [1.16 1.1 1.9 2.8 3.2 3.2]; %Engine mass [kg]

%Gridded Ion Thrusters
%https://www.busek.com/bit3
%https://www.space-propulsion.com/spacecraft-propulsion/propulsion-systems/electric-propulsion/index.html
%[BIT-3RF RIT-10EV0mod1 RIT-10EV0mod2 RIT-10EV0mod3]
GdiThrust = [1.1 5 15 25]; %Thrust [mN]
GdiIsp = [2150 1900 3000 3200]; %Isp [s]
GdiPower = [75 145 435 760]; %Power consumption [W]
GdiMass = [1.4 1.8 1.8 1.8]; %Engine mass [kg]

%Infine unisco le matrici, in modo da averne una unica per tutti i motori.
%In questo modo riesco a gestire meglio i diversi parametri.
ElEngThrust = [HtThrust,GdiThrust];
ElEngIsp = [HtIsp,GdiIsp];
ElEngPower = [HtPower,GdiPower];
ElEngMass = [HtMass,GdiMass];

end

%-----IMPULSIVE APPROXIMATION FOR ELECTRIC STATIONKEEPING-----%
%Nell'ipotesi di avere uno station keeping per l'elettrico di 33m/s annuo
%e nell'ipotesi di effettuare una manovra al giorno, avremmo che:
```

C. CALCULATIONS

```
%33/365 -> 0.0904 m/s a manovra.  
%Con un periodo orbitale di 15 giorni, avremo quindi 15 burn ogni orbita.  
  
function [DeltaVskeburn,tMaxManouver] = ImpulsiveApproximation(Torbit,AnnualDvStkEl)  
  
disp('-----Electric Thrusters StationKeeping Impulsive approximation-----')  
  
%DeltaV to achieve for each stkeeping burn [m/s]  
DeltaVskeburn = AnnualDvStkEl/365;  
  
%Nell'ipotesi di poter usare approssimazione impulsiva, posso dire che tale  
%approssimazione è valida se ttotmanovra = 1% del periodo orbitale, in  
%questo modo ogni singola manovra rappresenta lo 1%/15 del periodo orbitale totale  
ImpApproxParameter = 0.01;  
  
disp('For this impulsive approximation, we took the stationkeeping manouvers')  
disp(['to be the ',num2str(ImpApproxParameter*100),'% of the orbital period time'])  
disp(' ')  
  
%Maximum acceptable total manouver time for each orbit [s]  
ttotMaxManouver = Torbit*24*3600*ImpApproxParameter;  
  
disp(['The maximum acceptable total time to execute the manouvers is: ',num2str(  
ttotMaxManouver), ' s'])  
disp(['That's ', num2str(ttotMaxManouver/(3600)), ' hours over ', num2str(Torbit*24), '  
hours of orbital period'])  
  
%Manouver time associated with each burn [s]  
tMaxManouver = ttotMaxManouver/Torbit;  
  
disp(['The maximum acceptable time to execute the single manouver burn is: ',num2str(  
tMaxManouver), ' s'])  
disp(['That's ', num2str(tMaxManouver/(60)), ' minutes'])  
  
end  
  
%-----HALO CHEMICAL INSERTION PROPELLANT CALCULATIONS-----%  
%Funzione per il calcolo del propellente richiesto per inseririrsi nella  
%halo orbit. Questa funzione verrà richiamata sia per la configurazione  
%ibrida che per quella chimica.
```

```

function [ScMasstoStk,MpropIns,TankInsChem] = HaloInsertionProp (ScMasstoStk0GM,
    ScMasstoStk,ChemEngMass,MargChem,ChemEngIsp,DeltaVInsBurn,MpropIns,TankInsChem,i,j
)
global g0

%Col ciclo for sotto andiamo ad iterare più volte i calcoli del
%propellente, aggiungendo ad ogni iterazione la stima dei pesi delle
%tank dell'iterazione precedente
for it = 1 : 5
    %Propellant Mass Calculations using Tsiolkovsky equation rearranged
    MpropIns(i,j) = (ScMasstoStk(i,j))*(exp(DeltaVInsBurn/(ChemEngIsp*g0))-1);

    %We found that for every 71kg of hydrazine propellant we need
    %8.2kg of "propellant's tank"
    %https://www.satnow.com/products/propellant-tanks/mt-aerospace/96-1269-ptd-96
    TankInsChem(i,j) = MpropIns(i,j)/(8.65);

    %Satellite mass to Stationkeep nella halo orbit
    ScMasstoStk(i,j) = ScMasstoStk0GM(i,j) + ChemEngMass*MargChem + TankInsChem(i,j);

end
end

```

```

%-----CHEMICAL PROPULSION STATIONKEEPING-----%
%Poichè un singolo motore chimico sviluppa una spinta 10^3 volte superiore
%di quella di un motore elettrico, la manovra sarà sempre impulsiva.
%Per la massa dei motori, andremo a considerare dei
%"4N monopropellant hydrazine thrusters" con 0.4kg di peso e 211s Isp
function [MtotChem,ChemTotalMass,MpropChem,TankStkChem,ScMasstoStkChem] =
    ChemStationkeeping(MpropChem,TankStkChem,TankInsChem,MissionTime,AnnualDvStkChem,
    ScMasstoStkChem,ScMasstoStk0GM,ChemEngStkMass,ChemEngIsp,MargChem,ChemTotalMass,i,
    j)
global g0

```

```

%DeltaV for Chemical StationKeeping [m/s]
DeltaVskChem = MissionTime*AnnualDvStkChem;

```

```

%Col ciclo while sotto andiamo ad iterare più volte i calcoli del
%propellente, aggiungendo ad ogni iterazione la stima dei pesi delle
%tank

```

C. CALCULATIONS

```
for it = 1 : 5
    %Propellant Mass Calculations using Tsiolkovsky equation rearranged
    MpropChem(i,j) = (ScMasstoStkChem(i,j))*(exp(DeltaVskChem/(ChemEngIsp*g0))-1);

    %We found that for every 3.46kg (4.5/1.3) of hydrazine propellant we need
    %1kg of "propellant tank"
    TankStkChem(i,j) = MpropChem(i,j)/(4.967);

end

%Usiamo il parametro MargChem per marginare i risultati, tenendo di conto
%del peso delle tubature dei motori, della struttura dei motori...
%Total Chem propulsion system mass
ChemTotalMass(i,j) = (ChemEngStkMass)*MargChem + MpropChem(i,j) + TankStkChem(i,j)*
    MargChem;

%Total Satellite Mass
MtotChem(i,j) = ScMasstoStkChem(i,j) + ChemTotalMass(i,j);

%Aggiorniamo la massa del satellite
ScMasstoStkChem(i,j) = ScMasstoStk0GM(i,j) + ChemEngStkMass + TankStkChem(i,j) +
    TankInsChem(i,j);

end

%-----ELECTRIC PROPULSION STATIONKEEPING-----
%Calcolo il Thrust minimo necessario a ottenere una burn time accettabile
%secondo le ipotesi fatte con l'approssimazione impulsiva.

function [EpTotalMass,EpSystMass,ThrustersMass,SolarPanelMass,MpropEl,TankStkEl,
    ScMasstoStkEl,n_eng,MinThrust] = ElecStationkeeping(ElEngThrust,ElEngIsp,
    ElEngPower,ElEngMass,DeltaVskEl,ScMasstoStkEl,ScMasstoStk0GM,DeltaVskburn,
    tMaxManouver,SolarPanelValue,MargEl,MtotEl,EpTotalMass,EpSystMass,ThrustersMass,
    SolarPanelMass,MpropEl,TankStkEl,TankInsChem,n_eng,MinThrust,ThrustersConsumption,
    i,j)
global g0

%Definiamo il peso delle tank come inizialmente nullo, verrà poi calcolato
```



```
%successivamente
%Variables Definition
toll = 0.1;
ScMasstoStkOld = ScMasstoStkEl.*2; %col *2
garantisco l'entrata all'interno del ciclo while

%Al contrario del caso del chimico, una variazione della massa genera una
%variazione nelle caratteristiche richieste dal motore (e quindi nei
%pannelli solari etc).
%Il parametro di tolleranza ci permette di capire quando il risultato
%converge ad una soluzione

if toll < abs(ScMasstoStkOld(i,j)-ScMasstoStkEl(i,j))

while toll < abs(ScMasstoStkOld(i,j)-ScMasstoStkEl(i,j))
    % F=m*a   v=a*t  ->  F=m*v/t  dove F è il thrust sviluppato dal sistema
    % propulsivo
    %Calcoliamo la spinta minima richiesta dal sistema propulsivo,
    %secondo l'approssimazione impulsiva
    %Min Thrust calculation [N]
    MinThrust(i,j) = ScMasstoStkEl(i,j)*DeltaVsrburn/tMaxManouver;  %[N]

    %Al fine di permettere un corretto comparison fra le diverse
    %tecniche e i diversi motori, andiamo a calcolare il numero
    %minimo di motori che dovrebbero essere contemporaneamente accesi
    %al fine di generare la minima spinta richiesta. Tale valore dovrà
    %poi essere arrotondato al primo numero pari superiore.

    while (MinThrust(i,j) - n_eng(i,j)*(ElEngThrust(j)/10^3)) > 0
        %incrementiamo il numero di motori e reiteriamo il calcolo
        n_eng(i,j) = n_eng(i,j) + 1;

    end

    %Controlliamo che il numero di motori non diventi troppo
    %grande. In quel caso il calcolo tende a divergere, ovvero il motore
    %non è in grado di soddisfare la approssimazione impulsiva.
    %Con l'altro if sotto andiamo a controllare che il numero di motori
    %sia pari, se così non fosse, ne aggiungiamo uno.
    if n_eng(i,j) < 12
```

C. CALCULATIONS

```
if rem(n_eng(i,j),2) == 0

else
    n_eng(i,j) = n_eng(i,j) + 1;

end

else
    %Il numero di motori necessari diverge
    n_eng(i,j) = nan;
    MtotEl(i,j) = nan;
    EpTotalMass(i,j) = nan;
    EpSystMass(i,j) = nan;
    MpropEl(i,j) = nan;
    ThrustersMass(i,j) = nan;
    SolarPanelMass(i,j) = nan;
    TankStkEl(i,j) = nan;
    ScMasstoStkEl(i,j) = nan;
    return

end

%Una volta noto il numero di motori, andiamo a calcolare i
%restanti parametri.

%Hall thrusters engines total mass
ThrustersMass(i,j) = ElEngMass(j)*n_eng(i,j);

%Hall thrusters engines total consumption, considering that only half
%of them would be operative during propulsion
ThrustersConsumption(i,j) = ElEngPower(j)*n_eng(i,j)/2;

%Solar Panel extra mass required for sustain propulsion needs [kg]
SolarPanelMass(i,j) = ThrustersConsumption(i,j)/SolarPanelValue;
%SolarPanelMass(i,j) = 0;

%Propellant Mass Calculations using Tsiolkovsky equation rearranged
MpropEl(i,j) = (ScMasstoStkEl(i,j)+(EpSystMass(i,j))+TankStkEl(i,j))*(exp(
    DeltaVskEl/(ElEngIsp(j)*g0))-1);

%Valutare il peso dei diversi serbatoi, a seconda del
```



```
%propellente scelto per ogni motore.  
[TankStkEl] = TankEvaluator3000(MpropEl,TankStkEl,i,j);  
  
%Electric propulsion system mass without the propellant[kg]  
EpSystMass(i,j) = (ThrustersMass(i,j) + SolarPanelMass(i,j));  
  
%Adesso la massa della spacecraft con cui fare  
%stationkeeping è diversa a seconda del motore usato e  
%del propellente usato (aiuto)  
%Teniamo in "memoria" la massa del satellite a questa iterazione  
%per confrontarla con quella successiva  
ScMasstoStkOld(i,j) = ScMasstoStkEl(i,j);  
  
%Aggiorniamo la massa del satellite  
ScMasstoStkEl(i,j) = ScMasstoStk0GM(i,j) + EpSystMass(i,j) + TankStkEl(i,j) +  
    TankInsChem(i,j);  
  
end  
  
else  
    return  
end  
%Usiamo il parametro MargEl per marginare i risultati, tenendo di conto  
%del peso della struttura dei pannelli solari, del peso delle ppu,  
%delle tubature dei motori...  
%Total Ep propulsion system mass  
EpTotalMass(i,j) = (EpSystMass(i,j))*MargEl + MpropEl(i,j) + TankStkEl(i,j)*MargEl;  
  
end  
  
%-----ELECTRIC THRUSTERS TANK EVALUATIONS-----%  
function [TankStkEl] = TankEvaluator3000(MpropEl,TankStkEl,i,j)  
%In questa funzione andremo a definire i diversi pesi per i serbatoi, a  
%seconda del propellente usato da ogni motore.  
  
% We suppose to use carbon filament material and spherical shaped tanks:  
rho_Storage = [1800 1100 4900]; % Optimm propellant storage density [kg/m^3]  
p_Storage = [15 30 0.1].*1e6; % Optimm propellant storage pressure [Pa]  
T_Storage = [300 300 300]; % Propellant storage temperature [K]
```



C. CALCULATIONS

```
SafetyFactor_tank = 1.2;
rho_tank = [4850 1550 8440]; % Tank material density - Ti,CF,Inconel625 [kg/m^3]
sigma_Tank = [1.4 0.701 0.483].*1e9; % Yield strength of tank material [N/m^2]

%Kripton
TankStkEl(i,j) = 3./2.*p_Storage(2).*SafetyFactor_tank.*rho_tank(2)./sigma_Tank(2)./
    rho_Storage(2) .* MpropEl(i,j);
V_tank_spherical(i,j) = MpropEl(i,j)./rho_Storage(2);
r_tank_spherical(i,j)=(3/4/pi.*V_tank_spherical(i,j))^(1/3);

end
```



C.3. Propellant Evaluation

%Propulsion and Mass Budget script For SpaceSystem Project

%FRANCESCO MARRADI

```
close all  
clear  
clc
```

%Questo script verrà usato per comparare i diversi sistemi propulsivi
%elettrici, confrontando come diversi motori si comportano al variare della
%massa da mantere in stationkeeping attorno alla halo orbit.
%Le tecnologie di propulsione analizzata sono sia Gridded ion che Hall
%Thrusters. Per quanto riguarda i propellenti, dall'analisi fatta abbiamo
%deciso di utilizzare il krypton

```
%-----GENERAL PARAMETERS-----%  
%Constants  
global g0 %#ok<*GVMIS>  
g0 = 9.813; %Gravity's acceleration value [m/s]  
  
%General Parameters  
MissionTime = 10; %Mission duration time [years]  
Torbit = 15; %Halo Orbital period [days]  
SolarPanelValue = 300/6; %Watt generated by 1kg of solar panels[W/kg]  
  
%Propulsion Parameters  
AnnualDvStkChem = 20; %Annual deltaV consumption for Chemical StationKeeping [m  
/s] (from Queqiao)  
AnnualDvStkEl = 40; %Annual deltaV consumption for Electrical StationKeeping  
[m/s]  
DeltaVInsBurn = 660; %Deltav insertion burn [m/s]  
ChemEngMass = 0.65*4; %Mass of the chemical engines for insertion [kg] (From  
Queqiao)  
ChemEngIsp = 211; %Queqiao Hydrazine engine Isp [s]  
ChemEngStkMass = 0.4*10; %Mass of the Stationkeeping engines from Queqiao  
configuration, 10 engines weighing 0.4kg [kg]  
MargChem = 1; %Margin coefficient  
MargEl = 1; %Margin coefficient to take into account ppu's, tank and  
structure masses
```



C. CALCULATIONS

```
%Definiamo un vettore contenente i valori delle masse su cui andare ad  
%effettuare i calcoli
```

```
ScMasstoStk0G = 161.7;
```

```
%-----IMPULSIVE APPROXIMATION FOR ELECTRIC THRUSTERS-----%  
%Nel caso sia stata scelta la configurazione ibrida, definiamo  
%l'approssimazione impulsiva per stimare i calcoli dello stationkeeping  
%elettrico.
```

```
[DeltaVsrburn,tMaxManouver] = ImpulsiveApproximation(Torbit,AnnualDvStkEl);
```

```
%-----CALCULATIONS-----%  
%In questa sezione andremo a fare tutti i calcoli relativi alle stime delle  
%masse. Il procedimento nei calcoli è il seguente:  
%Massa sc -> seleziono un motore per fare lo stationkeeping -> faccio i  
%calcoli con tutti i propellenti che può usare.  
%Testo tutti i motori, poi cambio massa della sc e rifaccio i conti.
```

```
%DeltaV for Electrical Stationkeeping [m/s]  
DeltaVsrbEl = MissionTime*AnnualDvStkEl;
```

```
%Definiamo una funzione "catalogo" delle performance dei diversi motori  
%Elettrici, suddivisi per tecnologia propulsiva  
[ElEngThrust,ElEngIsp,ElEngPower,ElEngMass] = ElectricThrustersCharacteristics();
```

```
%Inizializzazione di alcune variabili  
%Definiamo il peso delle tank come inizialmente nullo, verrà poi calcolato  
%successivamente  
TankInsChem = zeros(length(ElEngThrust),1); %Peso dei tank per l'inserimento nella  
halo[kg]  
TankInsChemChem = zeros(length(ElEngThrust),1); %Peso dei tank per l'inserimento  
nella halo[kg]  
TankStkChem = zeros(length(ElEngThrust),1); %Peso dei tank per  
lo stationkeeping[kg]  
TankStkEl = zeros(length(ElEngThrust),1); %Peso dei tank per lo stationkeeping[  
kg]  
MpropInsEl = zeros(length(ElEngThrust),1);  
MpropInsChem = zeros(length(ElEngThrust),1);  
MpropEl = zeros(length(ElEngThrust),1);
```



```
n_eng = zeros(length(ElEngThrust),1); %Numero di Electric Thursters
richiesti dal sistema propulsivo
MtotEl = zeros(length(ElEngThrust),1);
EpTotalMass = zeros(length(ElEngThrust),1);
EpSystMass = zeros(length(ElEngThrust),1);
MinThrust = zeros(length(ElEngThrust),1);
ThrustersMass = zeros(length(ElEngThrust),1);
SolarPanelMass = zeros(length(ElEngThrust),1);

%Con questo if andiamo a "trasformare" il vettore ScMasstoStk0G come una
%matrice avente il numero delle righe pari al numero delle masse e quello
%delle colonne pari a quello dei motori.
ScMasstoStk0GM = zeros(length(ElEngThrust),1);
ScMasstoStkEl = zeros(length(ElEngThrust),1);

for i = 1 : length(ElEngThrust)
ScMasstoStk0GM(i) = ScMasstoStk0G;
ScMasstoStkEl(i) = ScMasstoStk0G;
end

ScMasstoStk = ScMasstoStk0GM;

%Il ciclo for permette di reiterare i calcoli, andando a raffinare
%i valori trovati.
for it = 1:10

    %Con questo ciclo vado a selezionare il motore da usare nei calcoli
    for i = 1 : length(ElEngThrust)

        %Per prima cosa si calcola il propellente per l'inserimento
        %nella halo
        [ScMasstoStkEl,MpropInsEl,TankInsChem] = HaloInsertionProp (ScMasstoStk0GM
            ,ScMasstoStkEl,MpropInsEl,ChemEngIsp,DeltaVInsBurn,i);

        %Poi calcoliamo la massa del propellente richiesto allo
        %stationkeeping per ogni tecnologia di motori usati.
        [MtotEl,EpTotalMass,EpSystMass,ThrustersMass,SolarPanelMass,MpropEl,
        TankStkEl,ScMasstoStkEl,n_eng,MinThrust] = ElecStationkeeping(
            ElEngThrust,ElEngIsp,ElEngPower,ElEngMass,DeltaVskEl,ScMasstoStkEl,
```



C. CALCULATIONS

```
ScMasstoStk,DeltaVsrburn,tMaxManouver,SolarPanelValue,MtotEl,  
EpTotalMass,EpSystMass,ThrustersMass,SolarPanelMass,MpropEl,TankStkEl,  
n_eng,MinThrust,i);  
  
end  
  
end  
  
%% -----FUNZIONI USATE NELLO SCRIPT-----&  
%Questo sarà lo spazio dove andremo ad inserire tutte le funzioni usate  
%nello script, in modo da scriverlo bello pulito e ordinato  
  
%-----VALUES OF ELECTRIC ENGINES-----%  
%Busek Hall Thruster Parameters: https://www.busek.com/hall-thrusters  
%Safran Hall Thruster PPSX00: https://www.safran-group.com/products-services/ppsrx00-stationary-plasma-thruster  
%All the parameters are loaded into vectors  
  
function [ElEngThrust,ElEngIsp,ElEngPower,ElEngMass] =  
    ElectricThrustersCharacteristics()  
  
%Hall Thrusters  
%[BHT-100 BHT-200 BHT-350 BHT-600 PPSX00mod1 PPSX00mod2]  
HtThrust = [7 15 17]; % Thrust [mN]  
HtIsp = [1000 1300 1244]; % Isp [s]  
HtPower = [100 200 300]; % Power consumption [W]  
HtMass = [1.16 3.2 1.9]; % Engine mass [kg]  
  
%Infine unisco le matrici, in modo da averne una unica per tutti i motori.  
%In questo modo riesco a gestire meglio i diversi parametri.  
ElEngThrust = HtThrust;  
ElEngIsp = HtIsp;  
ElEngPower = HtPower;  
ElEngMass = HtMass;  
  
end  
  
%-----IMPULSIVE APPROXIMATION FOR ELECTRIC STATIONKEEPING-----%  
%Nell'ipotesi di avere uno station keeping per l'elettrico di 33m/s annuo
```



%e nell'ipotesi di effettuare una manovra al giorno, avremmo che:

%33/365 -> 0.0904 m/s a manovra.

%Con un periodo orbitale di 15 giorni, avremo quindi 15 burn ogni orbita.

```
function [DeltaVsrburn,tMaxManouver] = ImpulsiveApproximation(Torbit,AnnualDvStkEl)

disp('-----Electric Thrusters StationKeeping Impulsive approximation-----')

%DeltaV to achieve for each stkeeping burn [m/s]
DeltaVsrburn = AnnualDvStkEl/365;

%Nell'ipotesi di poter usare approssimazione impulsiva, posso dire che tale
%approssimazione è valida se ttotmanovra = 1% del periodo orbitale, in
%questo modo ogni singola manovra rappresenta lo 1%/15 del periodo orbitale totale
ImpApproxParameter = 0.01;

disp('For this impulsive approximation, we tought the stationkeeping manouvers')
disp(['to be the ',num2str(ImpApproxParameter*100),'% of the orbital period time'])
disp(' ')

%Maximum acceptable total manouver time for each orbit [s]
ttotMaxManouver = Torbit*24*3600*ImpApproxParameter;

disp(['The maximum acceptable total time to execute the manouvers is: ',num2str(
    ttotMaxManouver), ' s'])
disp(['Thats ', num2str(ttotMaxManouver/(3600)), ' hours over ', num2str(Torbit*24),
    ' hours of orbital period'])

%Manouver time associated with each burn [s]
tMaxManouver = ttotMaxManouver/Torbit;

disp(['The maximum acceptable time to execute the single manouver burn is: ',num2str(
    tMaxManouver), ' s'])
disp(['Thats ', num2str(tMaxManouver/(60)), ' minutes'])

end

%-----HALO CHEMICAL INSERTION PROPELLANT CALCULATIONS-----
%Funzione per il calcolo del propellente richiesto per inseririrsi nella
%halo orbit. Questa funzione verrà richiamata sia per la configurazione
```

C. CALCULATIONS

```
%ibrida che per quella chimica.

function [ScMasstoStk,MpropIns,TankInsChem] = HaloInsertionProp (ScMasstoStk0GM,
    ScMasstoStk,MpropIns,ChemEngIsp,DeltaVInsBurn,i)
global g0

%Col ciclo for sotto andiamo ad iterare più volte i calcoli del
%propellente, aggiungendo ad ogni iterazione la stima dei pesi delle
%tank dell'iterazione precedente

TankInsChem=[6 6 6];

%for it = 1 : 5

    %Propellant Mass Calculations using Tsiolkovsky equation rearranged
    MpropIns(i) = (ScMasstoStk(i))*(exp(DeltaVInsBurn/(ChemEngIsp*g0))-1);

    %Satellite mass to Stationkeep nella halo orbit
    ScMasstoStk(i) = ScMasstoStk0GM(i)-TankInsChem(i);

    %We found that for every 71kg of hydrazine propellant we need
    %8.2kg of "propellant's tank"
    %https://www.satnow.com/products/propellant-tanks/mt-aerospace/96-1269-ptd-96
    TankInsChem(i) = MpropIns(i)/(8.65);

    %Satellite mass to Stationkeep nella halo orbit
    ScMasstoStk(i) = ScMasstoStk0GM(i) + TankInsChem(i);

%end

end

%-----ELECTRIC PROPULSION STATIONKEEPING-----
%Calcolo il Thrust minimo necessario a ottenere una burn time accettabile
%secondo le ipotesi fatte con l'approssimazione impulsiva.

function [MtotEl,EpTotalMass,EpSystMass,ThrustersMass,SolarPanelMass,MpropEl,TankStkEl
    ,ScMasstoStkEl,n_eng,MinThrust] = ElecStationkeeping(ElEngThrust,ElEngIsp,
    ElEngPower,ElEngMass,DeltaVskEl,ScMasstoStkEl,ScMasstoStk,DeltaVskburn,
    tMaxManouver,SolarPanelValue,MtotEl,EpTotalMass,EpSystMass,ThrustersMass,
```



```
SolarPanelMass,MpropEl,TankStkEl,n_eng,MinThrust,i)
global g0

%Definiamo il peso delle tank come inizialmente nullo, verrà poi calcolato
%successivamente
%Variables Definition
toll = 0.1;
ScMasstoStkOld = ScMasstoStkEl.*2; %col *2
    garantisco l'entrata all'interno del ciclo while

%Al contrario del caso del chimico, una variazione della massa genera una
%variazione nelle caratteristiche richieste dal motore (e quindi nei
%pannelli solari etc).
%Il parametro di tolleranza ci permette di capire quando il risultato
%converge ad una soluzione

if toll < abs(ScMasstoStkOld(i)-ScMasstoStkEl(i))

while toll < abs(ScMasstoStkOld(i)-ScMasstoStkEl(i))
    % F=m*a   v=a*t   ->   F=m*v/t   dove F è il thrust sviluppato dal sistema
    % propulsivo
    %Calcoliamo la spinta minima richiesta dal sistema propulsivo,
    %secondo l'approssimazione impulsiva
    %Min Thrust calculation [N]
    MinThrust(i) = ScMasstoStkEl(i)*DeltaVskburn/tMaxManouver;  %[N]

    %Al fine di permettere un corretto comparison fra le diverse
    %tecnicologie e i diversi motori, andiamo a calcolare il numero
    %minimo di motori che dovrebbero essere contemporaneamente accesi
    %al fine di generare la minima spinta richiesta. Tale valore dovrà
    %poi essere arrotondato al primo numero pari superiore.

    while (MinThrust(i) - n_eng(i)*(ElEngThrust(i)/10^3)) > 0
        %incrementiamo il numero di motori e reiteriamo il calcolo
        n_eng(i) = n_eng(i) + 1;

    end

    %Controlliamo che il numero di motori non diventi troppo
    %grande. In quel caso il calcolo tende a divergere, ovvero il motore
    %non è in grado di soddisfare la approssimazione impulsiva.
```

C. CALCULATIONS

```
%Con l'altro if sotto andiamo a controllare che il numero di motori  
%sia pari, se così non fosse, ne aggiungiamo uno.  
if n_eng(i) < 12
```

```
if rem(n_eng(i),2) == 0  
  
else  
n_eng(i) = n_eng(i) + 1;  
  
end
```

```
else  
%Il numero di motori necessari diverge  
n_eng(i) = nan;  
MtotEl(i) = nan;  
EpTotalMass(i) = nan;  
EpSystMass(i) = nan;  
MpropEl(i) = nan;  
ThrustersMass(i) = nan;  
SolarPanelMass(i) = nan;  
TankStkEl(i) = nan;  
ScMasstoStkEl(i) = nan;  
return
```

```
end
```

```
%Una volta noto il numero di motori, andiamo a calcolare i  
%restanti parametri.
```

```
%Hall thrusters engines total mass  
ThrustersMass(i) = ElEngMass(i)*n_eng(i);
```

```
%Hall thrusters engines total consumption, considering that only half  
%of them would be operative during propulsion  
ThrustersConsumption(i) = ElEngPower(i)*n_eng(i)/2;
```

```
%Solar Panel extra mass required for sustain propulsion needs [kg]  
SolarPanelMass(i) = ThrustersConsumption(i)/SolarPanelValue;
```

```
%Dato che usiamo le batterie per sostenere la propulsione, pongo il  
%valore dei pannelli solari extra a zero
```



```
SolarPanelMass(i) = 0;

%Electric propulsion system + extrapanell required mass without the propellant[kg]
EpSystMass(i) = (ThrustersMass(i) + SolarPanelMass(i));

%Propellant Mass Calculations using Tsiolkovsky equation rearranged
MpropEl(i) = (ScMasstoStkEl(i)+EpSystMass(i)+TankStkEl(i))*(exp(DeltaVskEl/(
    ElEngIsp(i)*g0))-1);

%Valutare il peso dei diversi serbatoi, a seconda del
%propellente scelto per ogni motore.
[TankStkEl] = TankEvaluator3000(MpropEl,TankStkEl,i);

%Adesso la massa della spacecraft con cui fare
%stationkeeping è diversa a seconda del motore usato e
%del propellente usato (aiuto)
%Teniamo in "memoria" la massa del satellite a questa iterazione
%per confrontarla con quella successiva
ScMasstoStkOld(i) = ScMasstoStkEl(i);

%Aggiorniamo la massa del satellite
ScMasstoStkEl(i) = ScMasstoStk(i) + EpSystMass(i) + TankStkEl(i);

end

else
    return
end

%Usiamo il parametro MargEl per marginare i risultati, tenendo di conto
%del peso della struttura dei pannelli solari, del peso delle ppu,
%delle tubature dei motori...

%Total Satellite Mass
MtotEl(i) = ScMasstoStkEl(i) + MpropEl(i);

end
```



C. CALCULATIONS

%-----ELECTRIC THRUSTERS TANK EVALUATIONS-----%

```
function [TankStkEl] = TankEvaluator3000(MpropEl,TankStkEl,i)
%In questa funzione andremo a definire i diversi pesi per i serbatoi, a
%seconda del propellente usato da ogni motore.
```

% We suppose to use carbon filament material and spherical shaped tanks:

```
rho_Storage = [1800 1100 4900]; % Optimm propellant storage density [kg/m^3]
p_Storage = [15 30 0.1].*1e6; % Optimm propellant storage pressure [Pa]
T_Storage = [300 300 300]; % Propellant storage temperature [K]
SafetyFactor_tank = 1.2;
rho_tank = [4850 1550 8440]; % Tank material density - Ti,CF,Inc625 [kg/m^3]
sigma_Tank = [1.4 0.701 0.483].*1e9; % Yield strenght of tank material [N/m^2]
```

% Krypton

```
TankStkEl(i) = 3./2.*p_Storage(2).*SafetyFactor_tank.*rho_tank(2)./sigma_Tank(2)./
    rho_Storage(2) .* MpropEl(i);
V_tank_spherical(i) = MpropEl(i)./rho_Storage(2);
r_tank_spherical(i)=(3/4/pi.*V_tank_spherical(i))^(1/3);
```

end



C.4. Link Budget

```
%%%%----- RADIO COMMUNICATION SYSTEM -----%%%%%
% Elia Bianchini - Space System - Support to Human Activities from EML2
% Script to obtain antennas data to use

% From the distances, the data rate to be transmitted and the frequency
% used, we compute the gain and power needed for transmitting
% (& receiving) communication systems, starting from assumptions on power
% and gain of the receiving (& transmitting) antennas.

clear; close all; clc;

k=1.38064852e-23;           % Boltzmann Constant [m^2*kg/s/K]
c=2.99792458e8;            % Speed of light [m/s]

%% SELECTION OF ORBIT

% Halo orbit with low Z-Amplitude:
r_moon=7.608643e7;          % Distance from Moon: 50'689 - 76'086.43 km
r_earth=4.646026e8;          % Distance from Earth: 439'448,8 - 464'602.6 km

La_earth=10^(0.1./r_earth/10); % Atmospheric Attenuation
La_moon=1;                   % Atmospheric Attenuation

%% SELECTION OF DATA RATE

DR_data_telescope=1.25e9;    % 1.25 Gbps -> Data Rate Moon Astronomy
                             % Data Transmission [bps]
DR_data_rover=200e6;          % 200 Mbps -> Data Rate Surface Exploration
                             % Data Transmission [bps]
DR_telemetry=300e3;           % 300 kbps -> Data Rate Telemetry & Command [bps]
DR_data_earth=100e6;           % 100 Mbps -> Data Rate sent from Earth [bps]
DR_LNSS=1500;                 % 1500 bps -> Data rate for Lunar Navigation
                             % Space System [bps]
```

%% ----- SATELLITE - EARTH LINK ----- %%

C. CALCULATIONS

```
% Ka-band: UL 22.55 - 23.15 GHz
%           DL 25.5-27.0 GHz
f_ES_Ka_UL=22.85e9; % Frequency for Ka-band Downlink [Hz]
lambda_ES_Ka_UL=c/f_ES_Ka_UL;
f_ES_Ka_DL=26.25e9; % Frequency for Ka-band Uplink [Hz]
lambda_ES_Ka_DL=c/f_ES_Ka_DL;

% X-band: UL 7.19 - 7.235 GHz
%           DL 8.45 - 8.5 GHz
f_ES_X_UL=7.2125e9; % Frequency for X-band Downlink [Hz]
lambda_ES_X_UL=c/f_ES_X_UL;
f_ES_X_DL=8.5e9; % Frequency for X-band Uplink [Hz]
lambda_ES_X_DL=c/f_ES_X_DL;

% S-band: UL 2.3 - 2.45 GHz
%           DL 2.50 - 2.650 GHz
f_ES_S_UL=2.375e9; % Frequency for S-band Uplink [Hz]
lambda_ES_S_UL=c/f_ES_S_UL;
f_ES_S_DL=2.575e9; % Frequency for S-band Downlink [Hz]
lambda_ES_S_DL=c/f_ES_S_DL;
```

%% ----- DATA UPLINK -----

```
% S-band | 400 Mbps | Datas from Earth
% From the data rate we select the bandwidth to use in order to have a
% correct signal to noise ratio:
```

```
p_ES_Data_UL=0.05; % Bandwidth percentage of frequency
Tsys_ES_Data_UL=300; % Receiver Temperature - Satellite [K]
B_ES_Data_UL=f_ES_S_UL*p_ES_Data_UL % Bandwidth [Hz]
eff=0.55; % Overall efficiency of the antennas
```

```
S_ES_Data_UL=((2.^((4*DR_data_earth./B_ES_Data_UL) - 1))); % Signal to Noise
% Ratio UL DATAS E-S
S_ES_Data_UL_db=10*log10(S_ES_Data_UL) % Signal to Noise Ratio UL DATAS
% E-S [dB]
N_ES_Data_UL=k*Tsys_ES_Data_UL.*B_ES_Data_UL; % Receiver Noise UL DATAS E-S
% [W]
```

```
% From the signal to noise ratio and the receiver noise, we select the
```

% power and gain of the satellites antenna transmitter and we calculate
 % the power and gain of the receiver needed to perform the wanted link:

```

EIRPt_ES_Data_UL_dbW=60;          % Trasmitter EIRP - ATLAS [dBW]
% Pt_ES_Data_UL=100;              % Transmitter Power [W]
% Gt_ES_Data_UL_db=72.4;          % Transmitter Gain - SDSA [dBi]
% Gt_ES_Data_UL_db=69.7;          % Transmitter Gain - Goonhilly [dBi]
% Gt_ES_Data_UL=10.^ (Gt_ES_Data_UL_db./10); % Transmitter Gain

L_s_UL_earth=(4*pi.*r_earth./lambda_ES_S_UL).^2; % Free Space Loss Earth

Gr_ES_Data_UL= ((S_ES_Data_UL.*N_ES_Data_UL)./(10.^ (EIRPt_ES_Data_UL_dbW/10))).*%
    L_s_UL_earth .* La_earth); % Reciver Gain
Gr_ES_Data_UL_db=10.*log10(Gr_ES_Data_UL)           % Reciver Gain
[dBi]
dr_ES_Data_UL=sqrt(Gr_ES_Data_UL./eff).*lambda_ES_S_UL./pi % Reciver
% Antenna Diameter [m]

% VERIFICATION TEST:
Gr1=10^(46/10);
EIRP1=10^(60/10);
SN1=Gr1*EIRP1/L_s_UL_earth/La_earth/N_ES_Data_UL;
C1=B_ES_Data_UL*log2(1+SN1);
if C1>4*DR_data_earth
    disp(['Uplink Capacity from earth is ',num2str(C1*1e-6), 'Mbps -> The link is
        VERIFIED with the selected antennas'])
else
    disp(['Uplink Capacity from earth is ',num2str(C1*1e-6), 'Mbps -> The link is NOT
        VERIFIED with the selected antennas'])
end

```

%% ----- DATA DOWNLINK -----

% X-band | 1.25 Gbps + 200 Mbps | Moon Astronomy & Surface Exploration

```

p_ES_Data_DL=0.05;          % Bandwidth percentage of frequency
Tsys_E_Data_DL=50;          % Reciver Temperature - Ground Station - G [K]
% Tsys_E_Data_DL=33.6;       % Reciver Temperature - Ground Station - SDSA [K]
B_ES_Data_DL=f_ES_X_DL*p_ES_Data_DL % Bandwidth [Hz]

```

C. CALCULATIONS

```
eff=0.55; % Overall efficiency of the antennas

S_ES_Data_DL=((2.^((DR_data_rover+DR_data_telescope)./B_ES_Data_DL) - 1));
% Signal to Noise Ratio DL DATAS S-E
S_ES_Data_DL_db=10*log10(S_ES_Data_DL) % Signal to Noise Ratio DL DATAS
% S-E [dB]
N_ES_Data_DL=k*Tsys_E_Data_DL.*B_ES_Data_DL; % Reciver Noise DL DATAS S-E
% [W]

Pt_ES_Data_DL=5; % Transmitter Power [W]
Gr_T_ES_Data_DL_db=53.7; % Reciver Gain/T - Goonhilly [dBK]
Gr_ES_Data_DL=10.^((Gr_T_ES_Data_DL_db./10)*Tsys_E_Data_DL);
% Gr_ES_Data_DL_db=72.4; % Reciver Gain - SDSA [dBi]

L_s_DL_earth=(4*pi.*r_earth./lambda_ES_X_DL).^2; % Free Space Loss Earth

Gt_ES_Data_DL= ((S_ES_Data_DL.*N_ES_Data_DL)./(Pt_ES_Data_DL.*Gr_ES_Data_DL) .* L_s_DL_earth .* La_earth); % Transmitter Gain
Gt_ES_Data_DL_db=10.*log10(Gt_ES_Data_DL) % Transmitter Gain [dBi]
dt_ES_Data_DL=sqrt(Gt_ES_Data_DL./eff).*lambda_ES_X_DL./pi % Transmitter
% Antenna Diameter [m]

figure
hold on
x1 = linspace(1,20,1000); % Transmitter Power 1 W - 20 W
y1 = linspace(30,80,1000); % Reciver Gain 30 - 80 dBi
[X1,Y1] = meshgrid(x1,y1);
Z1 = 10.*log10((S_ES_Data_DL.*N_ES_Data_DL)./(X1.*10.^(Y1/10)) .* L_s_DL_earth .* La_earth); % Transmitter Gain [dBi]
[C1,h1]=contour(X1,Y1,Z1,12,'-k','LineWidth',2,'ShowText','on');
clabel(C1,h1,'FontSize',15);
yline(Gr_T_ES_Data_DL_db,'color',[0.8500 0.3250 0.0980],'LineWidth',2)
xline(Pt_ES_Data_DL,'color',[0 0.4470 0.7410],'LineWidth',2)
legend('G_t','G_r = 53.7 dBi','P_t = 5 W','FontSize',15)
xlabel('Transmitter Power [W]','FontSize',12)
ylabel('Reciver Gain [dBi]','FontSize',12)
title('Earth-Spacecraft Datas Downlink Transmitter Gain [dBi]')
legend
grid on
hold off
```



% VERIFICATION TEST:

%Rover datas:

```
Pt2_1=10;
Gr2_1=10^(26/10)*Tsys_E_Data_DL;
Gt2=10^((19+23)/10);
SN2_1=Gr2_1*Gt2*Pt2_1/L_s_DL_earth/La_earth/N_ES_Data_DL;
C2_1=B_ES_Data_DL*log2(1+SN2_1);
if C2_1>DR_data_rover
    disp(['Rover Data Downlink Capacity to earth is ',num2str(C2_1*1e-6),'Mbps -> The
          link is VERIFIED with the selected antennas only if we use a Pt > ',num2str(
          Pt2_1),' W, and ground station antennas with Gr/T > 26 dB/K'])
else
    disp(['Rover Data Downlink Capacity to earth is ',num2str(C2_1*1e-6),'Mbps -> The
          link is NOT VERIFIED with the selected antennas'])
end
```

% Telescope datas:

```
Pt2_2=15;
Gr2_2=10^(53.7/10);
Gt2=10^((19+23)/10);
SN2_2=Gr2_2*Gt2*Pt2_2/L_s_DL_earth/La_earth/N_ES_Data_DL;
C2_2=B_ES_Data_DL*log2(1+SN2_2);
if C2_2>DR_data_telescope
    disp(['Telescope Data Downlink Capacity to earth is ',num2str(C2_2*1e-6),'Mbps ->
          The link is VERIFIED with the selected antennas only if we use a Pt > ',
          num2str(Pt2_2),' W'])
else
    disp(['Telescope Data Downlink Capacity to earth is ',num2str(C2_2*1e-6),'Mbps ->
          The link is NOT VERIFIED with the selected antennas'])
end
```

% Both datas:

```
Pt2_3=21;
Gr2_3=10^(53.7/10);
Gt2=10^((19+23)/10);
SN2_3=Gr2_3*Gt2*Pt2_3/L_s_DL_earth/La_earth/N_ES_Data_DL;
C2_3=B_ES_Data_DL*log2(1+SN2_3);
if C2_3>DR_data_telescope+DR_data_rover
    disp(['Rover+Telescope Data Downlink Capacity to earth is ',num2str(C2_3*1e-6),''
          Mbps -> The link is VERIFIED with the selected antennas only if we use a Pt > '
          ',num2str(Pt2_3),' W'])
```

C. CALCULATIONS

```
else
    disp(['Rover+Telescope Data Downlink Capacity to earth is ',num2str(C2_3*1e-6),'
        Mbps -> The link is NOT VERIFIED with the selected antennas'])
end

%% ----- TT&C -----
% S-band | 300 kbps | Telemetry & Command

p_ES_Telemetry=0.00004; % Bandwidth percentage of frequency
Tsys_E_Telemetry=300; % Reciver Temperature - Satellite [K]
B_ES_Telemetry=f_ES_S_UL*p_ES_Telemetry % Bandwidth [Hz]
eff=0.55; % Overall efficiency of the antennas

S_ES_Telemetry=((2.^ (DR_telemetry./B_ES_Telemetry) - 1)); % Signal to
% Noise Ratio UL T&C E-S
S_ES_Telemetry_db=10*log10(S_ES_Telemetry) % Signal to Noise Ratio UL
% T&C E-S [dB]
N_ES_Telemetry=k*Tsys_E_Telemetry.*B_ES_Telemetry; % Reciver Noise UL T&C
% E-S [W]

EIRPt_ES_Telemetry_UL_dbW=48; % Trasmitter EIRP - ATLAS [dBW]
% Pt_ES_Data_UL=100; % Transmitter Power [W]
% Gt_ES_Data_UL_db=72.4; % Transmitter Gain - SDSA [dBi]
% Gt_ES_Data_UL_db=69.7; % Transmitter Gain - Goonhilly [dBi]
% Gt_ES_Data_UL=10.^ (Gt_ES_Data_UL_db./10); % Transmitter Gain

L_s_earth=(4*pi.*r_earth./lambda_ES_S_UL).^2; % Free Space Loss Earth

Gr_ES_Telemetry= ((S_ES_Telemetry.*N_ES_Telemetry)./10.^ (EIRPt_ES_Telemetry_UL_dbW/10)
    .* L_s_earth .* La_earth); % Reciver Gain
Gr_ES_Telemetry_db=10.*log10(Gr_ES_Telemetry) % Reciver Gain [dBi]
dr_ES_Telemetry=sqrt(Gr_ES_Telemetry./eff).*lambda_ES_S_UL./pi % Reciver
% Antenna Diameter [m]

% VERIFICATION TEST:

%UPLINK:
Gr3=10^(46/10);
EIRP3=10^(60/10);
```



```
SN3=Gr3*EIRP3/L_s_UL_earth/La_earth/N_ES_Telemetry;
C3=B_ES_Telemetry*log2(1+SN3);
if C3>DR_telemetry
    disp(['TT&C Uplink Capacity from earth is ',num2str(C3*1e-3),'kbps -> The link is
        VERIFIED with the selected antennas'])
else
    disp(['TT&C Uplink Capacity from earth is ',num2str(C3*1e-3),'kbps -> The link is
        NOT VERIFIED with the selected antennas'])
end

%DOWNLINK:
Gt3_1=10^(30/3);
Pt3_1=1;
Gr3_1=10^(12/10)*Tsys_E_Telemetry;
SN3_1=Gr3_1*Gt3_1*Pt3_1/L_s_UL_earth/La_earth/N_ES_Telemetry;
C3_1=B_ES_Telemetry*log2(1+SN3_1);
if C3_1>DR_telemetry
    disp(['TT&C Downlink Capacity to earth is ',num2str(C3_1*1e-3),'kbps -> The link
        is VERIFIED with the selected antennas'])
else
    disp(['TT&C Downlink Capacity to earth is ',num2str(C3_1*1e-3),'kbps -> The link
        is NOT VERIFIED with the selected antennas'])
end

%% ----- SATELLITE - EARTH LINK -----
% Ka-band: UL 22.55 - 23.15 GHz
%          DL 22.5-23.5 GHz
f_MS_Ka_UL=22.85e9; % Frequency for Ka-band Downlink [Hz]
lambda_MS_Ka_UL=c/f_MS_Ka_UL;
f_MS_Ka_DL=23e9; % Frequency for Ka-band Uplink [Hz]
lambda_MS_Ka_DL=c/f_MS_Ka_DL;

% X-band: UL 7.19 - 7.235 GHz
%          DL 8.45 - 8.5 GHz
f_MS_X_UL=7.2125e9; % Frequency for X-band Downlink [Hz]
lambda_MS_X_UL=c/f_MS_X_UL;
f_MS_X_DL=8.45e9; % Frequency for X-band Uplink[Hz]
lambda_MS_X_DL=c/f_MS_X_DL;

% S-band: UL 2.025 - 2.104 GHz
%          DL 2.200 - 2.290 GHz
```



C. CALCULATIONS

```
f_MS_S_UL=2.0645e9; % Frequency for S-band Uplink [Hz]
lambda_MS_S_UL=c/f_MS_S_UL;
f_MS_S_DL=2.2e9; % Frequency for S-band Downlink [Hz]
lambda_MS_S_DL=c/f_MS_S_DL;

% L-band: 1.559 - 1.591 GHz
f_MS_L=1.57542e9; % Frequency for L-band Downlink [Hz]
lambda_MS_L=c/f_MS_L;

Tsys_M=300; % Space antennas temperature [K]
```

%% ----- DATA UPLINK -----

```
% X-band | 1.25 Gbps + 200 Mbps | Moon Astronomy & Surface Exploration

    % p_MS_Data_rover_UL=0.04; % Bandwidth percentage of frequency
% Surface Exploration
p_MS_Data_telescope_UL=0.04; % Bandwidth percentage of frequency
% Moon Astronomy
    % B_MS_Data_rover_UL=f_MS_X_UL*p_MS_Data_rover_UL % Bandwidth
% Surface Exploration [Hz]
B_MS_Data_telescope_UL=f_MS_X_UL*p_MS_Data_telescope_UL % Bandwidth
% Moon Astronomy [Hz]
eff=0.55; % Overall efficiency of the antennas

    % S_MS_Data_rover_UL=((2.^((DR_data_telescope)./B_MS_Data_rover_UL) - 1)); % Signal to Noise Ratio UL DATAS - Surface Exploration M-S
    % S_MS_Data_rover_UL_db=10*log10(S_MS_Data_rover_UL) % Signal to Noise Ratio UL DATAS - Surface Exploration M-S [dB]
S_MS_Data_telescope_UL=((2.^((DR_data_telescope)./B_MS_Data_telescope_UL) - 1)); % Signal to Noise Ratio UL DATAS - Moon Astronomy M-S
S_MS_Data_telescope_UL_db=10*log10(S_MS_Data_telescope_UL) % Signal to Noise Ratio UL DATAS - Moon Astronomy M-S [dB]
    % N_MS_Data_rover_UL=k*Tsys_MS_Data_UL.*B_MS_Data_rover_UL; % Reciver Noise UL DATAS M-S - Surface Exploration [W]
N_MS_Data_telescope_UL=k*Tsys_M.*B_MS_Data_telescope_UL; % Reciver Noise % UL DATAS M-S - Moon Astronomy [W]

    % Pt_MS_Data_rover_UL=10; % Transmitter Power - Surface Exploration [W]
    % Gt_MS_Data_rover_UL_db=20; % Transmitter Gain -
```



```
Surface Exploration [dBi]
% Gt_MS_Data_rover_UL=10.^ (Gt_MS_Data_rover_UL_db./10); % Transmitter Gain -
    Surface Exploration
Pt_MS_Data_telescope_UL=100; % Transmitter Power - Moon Astronomy [W]
Gt_MS_Data_telescope_UL_db=50; % Transmitter Gain - Moon Astronomy [dBi]
Gt_MS_Data_telescope_UL=10.^ (Gt_MS_Data_telescope_UL_db./10); % Transmitter
% Gain - Moon Astronomy

L_s_UL_moon=(4*pi.*r_moon./lambda_MS_X_UL).^2; % Free Space Loss Moon

% Gr_MS_Data_UL=((S_MS_Data_rover_UL.*N_MS_Data_rover_UL)./(Pt_MS_Data_rover_UL
.%Gt_MS_Data_rover_UL)).* L_s_UL_moon .* La_moon;
% Reciver Gain
Gr_MS_Data_UL=((S_MS_Data_telescope_UL.*N_MS_Data_telescope_UL)./
    Pt_MS_Data_telescope_UL.*Gt_MS_Data_telescope_UL)).* L_s_UL_moon .* La_moon;
% Reciver Gain
Gr_MS_Data_UL_db=10.*log10(Gr_MS_Data_UL) % Reciver Gain [dBi]
dr_MS_Data_UL=sqrt(Gr_MS_Data_UL./eff).*lambda_MS_X_UL./pi % Reciver Antenna Diameter
[m]

figure
hold on
x2 = linspace(10,1000,1000); % Transmitter Power 10 W - 1 kW
y2 = linspace(20,70,1000); % Transmitter Gain 20 - 70 dBi
[X2,Y2] = meshgrid(x2,y2);
Z2 = 10.*log10( ((S_MS_Data_telescope_UL.*N_MS_Data_telescope_UL)./(X2.*10.^ (Y2/10)))
    ).* L_s_UL_moon .* La_moon; % Reciver Gain [dBi]
[C2_1,h2]=contour(X2,Y2,Z2,15,'-k','LineWidth',2,'ShowText','on');
clabel(C2_1,h2,'FontSize',15);
yline(Gt_MS_Data_telescope_UL_db,'color',[0.8500 0.3250 0.0980],'LineWidth',2)
xline(Pt_MS_Data_telescope_UL,'color',[0 0.4470 0.7410],'LineWidth',2)
xlabel('Transmitter Power - Moon Astronomy [W]', 'Fontsize',12)
ylabel('Transmitter Gain - Moon Astronomy [dBi]', 'Fontsize',12)
title('Moon-Spacecraft Datas Uplink Reciver Gain - Moon Astronomy[dBi]')
legend('G_r','G_t = 50 dBi','P_t = 100 W','FontSize',15)
grid on
hold off

% VERIFICATION FOR ROVER:

% I select the minumun bandwidth to satisfy a sufficiently high S/N (>6db)
```

C. CALCULATIONS

```
% for smaller applications. I use this value for every other applications
% to be sure that my system will be able to communicate with every smaller
% user on the moon (even the slightly "bigger").

p_MS_Data_rover_UL=[0.000005 0.0002 0.012];           % Bandwidth percentage of
% frequency - Surface Exploration
B_MS_Data_rover_UL=(7.41e9).*p_MS_Data_rover_UL      % Bandwidth
% Surface Exploration [Hz]
N_MS_Data_rover_UL=k.*Tsys_M.*B_MS_Data_rover_UL; % Receiver Noise UL DATA
% M-S - Surface Exploration [W]

Pt_MS_Data_rover_UL=[5 15 20];           % Transmitter Power
% Surface Exploration [W]
Gt_MS_Data_rover_UL_db=[8 20 36];       % Transmitter Gain
% Surface Exploration [dBi]
Gt_MS_Data_rover_UL=10.^ (Gt_MS_Data_rover_UL_db./10); % Transmitter Gain
% Surface Exploration

Gr_M_UL_db=40;                         % Receiver Gain [dBi]
Gr_M_UL=10^(Gr_M_UL_db/10);            % Receiver Gain

N_MS_Data_rover_UL=k.*Tsys_M.*B_MS_Data_rover_UL; % Receiver Noise UL DATAS
% M-S - Surface Exploration [W]
S_N_M_UL=10.*log10( Gr_M_UL .* Pt_MS_Data_rover_UL .* Gt_MS_Data_rover_UL ./ (
    N_MS_Data_rover_UL) ./ (L_s_UL_moon*La_moon));
C_M_UL=B_MS_Data_rover_UL.*log2(1+10.^ (S_N_M_UL./10));

N_MS_Data_rover_UL_final=k.*Tsys_M.*B_MS_Data_rover_UL(1); % Receiver Noise
% UL DATAS M-S - Surface Exploration [W]
S_N_M_UL_final=10.*log10( Gr_M_UL .* Pt_MS_Data_rover_UL .* Gt_MS_Data_rover_UL ./ (
    N_MS_Data_rover_UL_final) ./ (L_s_UL_moon*La_moon));
C_M_UL_final=B_MS_Data_rover_UL(3).*log2(1+10.^ (S_N_M_UL_final./10))

% x3=linspace(14,50,1000);
% y3=(10.^ (x3./10).*Gr_M_UL./ (L_s_UL_moon*La_moon*k*Tsys_M*10^(6/10))).* log2
% (1+10^(6/10)).*1e-6;
% figure
% hold on
% plot(x3,y3,'k','LineWidth',2)
% yline(200,'color',[0.8500 0.3250 0.0980],'LineWidth',2)
% xline(10*log10(5*10^(8/10)),'color',[0 0.4470 0.7410],'LineWidth',2)
% xlabel('EIRP - Surface Exploration Module [dBW]','FontSize',12)
```



```
% ylabel('Channel Capacity [Mbps]', 'FontSize', 12)
% title('Data Rate [Mbps] based on the user')
% grid on
% hold off

%% ----- DATA DOWNLINK -----

% Ka-band | 150 Mbps | Surface Exploration
%           | 250 Mbps | Moon Astronomy

p_MS_Data_DL=0.003;
B_MS_Data_DL=f_MS_Ka_DL*p_MS_Data_DL % Bandwidth [Hz]
eff=0.55;                                % Overall efficiency of the antennas

S_MS_Data_DL=((2.^((2.5*DR_data_earth)./B_MS_Data_DL) - 1)); % Signal to
% Noise Ratio DL DATAS S-M
S_MS_Data_DL_db=10*log10(S_MS_Data_DL) % Signal to Noise Ratio DL DATAS
% S-M [dB]
N_MS_Data_DL=k*Tsys_M.*B_MS_Data_DL;    % Reciver Noise DL DATAS S-M [W]

Pt_MS_Data_DL=1;                         % Trasmitter Power - Moon Astronomy [W]
Gr_MS_Data_DL_db=50;                     % Reciver Gain - Moon Astronomy [dBi]
Gr_MS_Data_DL=10.^((Gr_MS_Data_DL_db./10)); % Reciver Gain - Moon Astronomy

L_s_DL_moon=(4*pi.*r_moon./lambda_MS_Ka_DL).^2;      % Free Space Loss Moon

Gt_MS_Data_DL= ((S_MS_Data_DL.*N_MS_Data_DL)./(Pt_MS_Data_DL.*Gr_MS_Data_DL) .* 
    L_s_DL_moon .* La_moon); % Transmitter Gain
Gt_MS_Data_DL_db=10.*log10(Gt_MS_Data_DL) % Transmitter Gain [dBi]

dt_MS_Data_DL=sqrt(Gt_MS_Data_DL_db./eff).*lambda_MS_Ka_DL./pi
% Transmitter Antenna Diameter [m]

figure
hold on
x4 = linspace(1,20,1000);                  % Transmitter Power 1 W - 20 W
y4 = linspace(20,70,1000);                  % Reciver Gain 20 - 80 dBi
[X4,Y4] = meshgrid(x4,y4);
Z4 = 10.*log10((S_MS_Data_DL.*N_MS_Data_DL)./(X4.*10.^(Y4/10)) .* L_s_UL_moon .* 
    La_moon); % Transmitter Gain [dBi]
[C4,h4]=contour(X4,Y4,Z4,15, '-k', 'LineWidth', 2, 'ShowText', 'on');
```

C. CALCULATIONS

```
clabel(C4,h4,'FontSize',15);
yline(Gr_MS_Data_DL_db,'color',[0.8500 0.3250 0.0980],'LineWidth',2)
xline(Pt_MS_Data_DL,'color',[0 0.4470 0.7410],'LineWidth',2)
xlabel('Transmitter Power [W]', 'Fontsize',12)
ylabel('Receiver Gain [dBi]', 'Fontsize',12)
title('Moon-Spacecraft Datas Downlink Transmitter Gain [dBi]')
legend('G_t','G_r = 50 dBi','P_t = 1 W','FontSize',15)
grid on
hold off
```

% VERIFICATION FOR ROVER:

```
Pt_MS_Data_rover_DL=[20 20 15];% Trasmitter Power - Surface Exploration [W]
Gr_MS_Data_rover_DL_db=[6 12 16];% Reciver Gain - Surface Exploration [dBi]
Gr_MS_Data_rover_DL=10.^ (Gr_MS_Data_rover_DL_db./10); % Reciver Gain
% Surface Exploration
Gr_M_DL_db=70; % Reciver Gain [dBi]
Gr_M_DL=10^(Gr_M_DL_db/10); % Reciver Gain

B_M_DL=[60 65 69]*1e6;
N_M_DL=k*Tsys_M.*B_M_DL;
S_N_M_DL=10.*log10(( Gr_M_DL .* Pt_MS_Data_rover_DL .* Gr_MS_Data_rover_DL ./ N_M_DL) ...
./ (L_s_DL_moon*La_moon));
C_M_DL=B_M_DL.*log2(1+10.^ (S_N_M_DL/10));

B_M_DL_final=69*1e6;
N_M_DL_final=k*Tsys_M.*B_M_DL_final;
S_N_M_DL_final=10.*log10(( Gr_M_DL .* Pt_MS_Data_rover_DL .* Gr_MS_Data_rover_DL ./ ...
N_M_DL_final) ./ (L_s_DL_moon*La_moon))
C_M_DL_final=B_M_DL_final.*log2(1+10.^ (S_N_M_DL_final/10))

figure
hold on
x5=linspace(10,25,1000); % Transmitter Power 10 - 25 W
y5=linspace(5,20,1000); % Reciver Gain 5 - 20 dBi
[X5,Y5] = meshgrid(x5,y5);
Z5 = B_M_DL_final .* log2(1+ ( Gr_M_DL .* X5 .* 10.^ (Y5./10) ./ N_M_DL_final) ./ ( ...
L_s_DL_moon*La_moon) ) *1e-6; % Channel Ccapacity [Mbps]
[C5, h5] = contour(Y5, X5, Z5, [150 150], 'color',[0.8500 0.3250 0.0980], 'linewidth',
2, 'ShowText', 'on');
clabel(C5, h5, 'FontSize', 15);
```



```
[C5,h5]=contour(Y5,X5,Z5,10,'-k','LineWidth',2,'ShowText','on');
clabel(C5,h5,'FontSize',15);
xlabel('Transmitter Power [W]','FontSize',12)
ylabel('Gain - Surface Exploration Module [dBi]','FontSize',12)
title('Moon-Spacecraft Datas Downlink Data-Rate based on the user [Mbps]')
grid on
hold off

%% ----- LNSS DOWNLINK -----

% L-band | 1500 bps | LNNS

p_MS_LNSS_DL=0.0000004;
B_MS_LNSS_DL=f_MS_L*p_MS_LNSS_DL % Bandwidth [Hz]
eff=0.55; % Overall efficiency of the antennas

S_MS_LNSS_DL=((2.^((DR_LNSS)./B_MS_LNSS_DL)-1)); % Signal to Noise Ratio
% DL LNSS S-M
S_MS_LNSS_DL_db=10*log10(S_MS_LNSS_DL) % Signal to Noise Ratio
% DL LNSS S-M [dB]
N_MS_LNSS_DL=k*Tsys_M.*B_MS_LNSS_DL; % Reciver Noise DL LNSS S-M [W]

Pt_MS_LNSS_DL=2; % Reciver Power [W]
Gr_MS_LNSS_DL_db=7; % Reciver Gain [dBi]
Gr_MS_LNSS_DL=10.^((Gr_MS_LNSS_DL_db./10)); % Reciver Gain

L_s_DL_moon_LNSS=(4*pi.*r_moon./lambda_MS_L).^2; % Free Space Loss Moon

Gt_MS_LNSS_DL= ((S_MS_LNSS_DL.*N_MS_LNSS_DL)./(Pt_MS_LNSS_DL.*Gr_MS_LNSS_DL)).*
L_s_DL_moon_LNSS .* La_moon); % Transmitter Gain
Gt_MS_LNSS_DL_db=10.*log10(Gt_MS_LNSS_DL) % Transmitter Gain [dBi]
dt_MS_LNSS_DL=sqrt(Gt_MS_LNSS_DL./eff).*lambda_MS_L./pi % Transmitter
% Antenna Diameter [m]

figure
hold on
x6 = linspace(1,10,1000); % Reciver Power 1 - 10 W
y6 = linspace(1,30,1000); % Reciver Gain 10 - 30 dBi
[X6,Y6] = meshgrid(x6,y6);
Z6 = 10.*log10((S_MS_LNSS_DL.*N_MS_LNSS_DL)./(X6.*10.^((Y6/10))).*L_s_DL_moon_LNSS .* La_moon); % Transmitter Gain [dBi]
```

C. CALCULATIONS

```
[C6,h6]=contour(X6,Y6,Z6,15,'-k','LineWidth',2,'ShowText','on');
clabel(C6,h6,'FontSize',15);
yline(Gr_MS_LNSS_DL_db,'color',[0.8500 0.3250 0.0980],'LineWidth',2)
xline(Pt_MS_LNSS_DL,'color',[0 0.4470 0.7410],'LineWidth',2)
xlabel('Transmitter Power [W]','FontSize',12)
ylabel('Receiver Gain [dBi]','FontSize',12)
title('Moon-Spacecraft LNSS Downlink Transmitter Gain [dBi]')
legend('G_t','G_r = 7 dBi','P_t = 2 W','FontSize',15)
grid on
hold off
```

% VERIFICATION TEST:

```
Gr4=10^(17/10);
Gt4=10^(6/10);
Pt4=2;
SN4=Gr4*Gt4*Pt4/L_s_DL_moon_LNSS/La_moon/N_MS_LNSS_DL;
C4=B_MS_LNSS_DL*log2(1+SN4);
if C4>DR_LNSS
    disp(['LNSS Downlink Capacity to moon is ',num2str(C4),'bps -> The link is
        VERIFIED with the selected antennas'])
else
    disp(['LNSS Downlink Capacity to moon is ',num2str(C4),'bps -> The link is NOT
        VERIFIED with the selected antennas'])
end
```

%% ----- TT&C -----

```
% S-band | 100 kbps | Telemetry & Command

p_MS_Telemetry_UL=0.000028;
B_MS_Telemetry_UL=f_MS_S_UL*p_MS_Telemetry_UL % Bandwidth [Hz]
eff=0.55; % Overall efficiency of the antennas

S_MS_Telemetry_UL=((2.^((DR_telemetry/3)./B_MS_Telemetry_UL))-1));
% Signal to Noise Ratio UL T&C M-S
S_MS_Telemetry_UL_db=10*log10(S_MS_Telemetry_UL) % Signal to Noise Ratio
% UL T&C M-S [dB]
N_MS_Telemetry_UL=k*Tsys_M.*B_MS_Telemetry_UL; % Receiver Noise UL T&C M-S
% [W]
```



```
Pt_MS_Telemetry_UL=5;      % Transmitter Power [W]
Gt_MS_Telemetry_UL_db=8;   % Transmitter Gain [dBi]
Gt_MS_Telemetry_UL=10.^((Gt_MS_Telemetry_UL_db./10)); % Transmitter Gain

L_s_TC_moon=(4*pi.*r_moon./lambda_MS_S_UL).^2;           % Free Space Loss Moon

Gr_MS_Telemetry_UL= ((S_MS_Telemetry_UL.*N_MS_Data_telescope_UL)./(Pt_MS_Telemetry_UL
    .*Gt_MS_Telemetry_UL) .* L_s_TC_moon .* La_moon); % Reciver Gain
Gr_MS_Data_UL_db=10.*log10(Gr_MS_Telemetry_UL)           % Reciver Gain [dBi]
dr_MS_Telemetry_UL=sqrt(Gr_MS_Telemetry_UL./eff).*lambda_MS_S_UL./pi
% Reciver Antenna Diameter [m]

figure
hold on
x7 = linspace(1,50,1000);          % Transmitter Power 1 W - 50 W
y7 = linspace(1,30,1000);          % Transmitter Gain 1 - 30 dBi
[X7,Y7] = meshgrid(x7,y7);
Z7 = 10.*log10((S_MS_Telemetry_UL.*N_MS_Telemetry_UL)./(X7.*10.^((Y7/10)) .* 
    L_s_TC_moon .* La_moon));    % Reciver Gain [dBi]
[C7,h7]=contour(X7,Y7,Z7,15, '-k','LineWidth',2,'ShowText','on');
clabel(C7,h7,'FontSize',15);
yline(Gt_MS_Telemetry_UL_db,'color',[0.8500 0.3250 0.0980],'LineWidth',2)
xline(Pt_MS_Telemetry_UL,'color',[0 0.4470 0.7410],'LineWidth',2)
xlabel('Transmitter Power [W]', 'FontSize',12)
ylabel('Transmitter Gain [dBi]', 'FontSize',12)
title('Moon-Spacecraft T&C Uplink Reciver Gain [dBi]')
legend('G_r','G_t = 8 dBi','P_t = 5 W','FontSize',15)
grid on
hold off

% VERIFICATION TEST:

%UPLINK:
Gr5=10^(32/10);
Gt5=10^(5/10);
Pt5=2;
SN5=Gr5*Gt5*Pt5/L_s_TC_moon/La_moon/N_MS_Telemetry_UL;
C5=B_MS_Telemetry_UL*log2(1+SN5);
if C5*2>DR_telemetry/3
    disp(['TT&C Uplink Capacity from moon is ',num2str(C5*2e-3),' kbps -> The link is
        VERIFIED with the selected antennas'])
end
```

C. CALCULATIONS

```
else
    disp(['TT&C Uplink Capacity from moon is ',num2str(C5*2e-3),' kbps -> The link is
          NOT VERIFIED with the selected antennas'])
end

%DOWNLINK:
Gt5_1=10^(30/10);
Pt5_1=3;
Gr5_1=10^(5/10);
SN5_1=Gr5_1*Gt5_1*Pt5_1/L_s_TC_moon/La_moon/N_MS_Telemetry_UL;
C5_1=B_MS_Telemetry_UL*log2(1+SN5_1);
if C5_1*2>(DR_telemetry/3)
    disp(['TT&C Downlink Capacity to moon is ',num2str(C5_1*2e-3),' kbps -> The link is
          VERIFIED with the selected antennas'])
else
    disp(['TT&C Downlink Capacity to moon is ',num2str(C5_1*2e-3),' kbps -> The link is
          NOT VERIFIED with the selected antennas'])
end
%% ----- BACKUP -----
% EARTH BACKUP UPLINK
Gr_backup_E_UL=10^(45.5/10);
EIRP_backup_E_UL=10^(48/10);
SN_backup_E_UL=Gr_backup_E_UL*EIRP_backup_E_UL/L_s_UL_earth/La_earth/N_ES_Telemetry;
C_backup_E_UL=B_ES_Telemetry*log2(1+SN_backup_E_UL);

disp(['Backup Uplink Capacity from earth is: ',num2str(C_backup_E_UL.*1e-3),' kbps'])

% EARTH BACKUP DOWNLINK
Gt_backup_E_DL=10^(40.5/10);
Gr_backup_E_DL=10^(12/10)*Tsys_E_Telemetry;
Pt_backup_E_DL=5;
SN_backup_E_DL=Gr_backup_E_DL.*Gt_backup_E_DL.*Pt_backup_E_DL./L_s_DL_earth/La_earth/
N_ES_Telemetry;
C_backup_E_DL=B_ES_Telemetry.*log2(1+SN_backup_E_DL);

disp(['Backup Downlink Capacity to earth is: ',num2str(C_backup_E_DL.*1e-3),' kbps'])

% MOON BACKUP UPLINK
Gr_backup_M_UL=10^(45.5/10);
```

```

Gt_backup_M_UL=10^(5/10);
Pt_backup_M_UL=2;
SN_backup_M_UL=Gr_backup_M_UL.*Gt_backup_M_UL.*Pt_backup_M_UL./L_s_UL_moon/La_moon/
N_MS_Telemetry_UL;
C_backup_M_UL=B_MS_Telemetry_UL.*log2(1+SN_backup_E_UL);

disp(['Backup Uplink Capacity from moon is: ',num2str(C_backup_M_UL.*1e-3), ' kbps'])

% MOON BACKUP DOWNLINK
Gt_backup_M_DL=10^(40.5/10);
Gr_backup_M_DL=10^(5/10);
Pt_backup_M_DL=3;
SN_backup_M_DL=Gr_backup_M_DL.*Gt_backup_M_DL.*Pt_backup_M_DL./L_s_DL_earth/La_earth/
N_MS_Telemetry_UL;
C_backup_M_DL=B_MS_Telemetry_UL.*log2(1+SN_backup_M_DL);

disp(['Backup Downlink Capacity to moon is: ',num2str(C_backup_M_DL.*1e-3), ' kbps'])

```



C.5. Inertia Matrix

```
% made by Mauro Furfari, Lucrezia Podestà, Mario Nocilli
% this script is called calcoloInerzia.m
% calculus inertia main body spacecraft

lxb = 1; %m
lyb = 1; %m
lzb = 1; %m
m_body = 200; %kg
Inerzia_body = 1/12*m_body*[lzb^2 + lxb^2 ,0,0 ; 0, lyb^2+lxb^2,0 ; 0,0, lzb^2+lyb^2];

lxsp = 1; %m
lysP = 1.25; %m
lzsp = 0.2; %m
m_sp = 7.50; %kg
Inerzia_sp = 1/12*m_body*[lzsp^2 + lxsp^2 ,0,0 ; 0, lysP^2+lxsp^2,0 ; 0,0, lzsp^2+lysP
^2];
spostamento_y = 1+lysP/2; %m
Inerzia_spostamento_sp = m_sp*spostamento_y^2*[1 0 0; 0 0 0 ; 0 0 1];

Inerzia_tot = Inerzia_body + 2*(Inerzia_sp + Inerzia_spostamento_sp);
sc_inertia = Inerzia_tot;
```



C.6. Attitude Perturbations

```
clear all
close all
clc

% made by Mauro Furfari, Lucrezia Podestà, Mario Nocilli
% this script is called disturbiAttitude.m
massEarth = 5.94*10^24; %kg
MassMoon = 7.348*10^22; %kg
massTot = MassMoon+massEarth;
G = 6.6726 * 10^-11;
mu = G*massTot;

distance_earth_moon = 384000000; %metri

CG_EarthMoon = (MassMoon*0+massEarth*distance_earth_moon)/(MassMoon+massEarth);
disance_eml2_moon = 65637000; %m
min_distX = 46540000; %m
max_distX = 74021000; %m
min_distZ = 16855000; %m

min_dist = sqrt(min_distZ^2+(CG_EarthMoon+min_distX)^2);

calcoloInerzia

M_gravity = 3/2*mu/min_dist^3 *[Inerzia_tot(3,3)-Inerzia_tot(2,2) ; Inerzia_tot(3,3)-
    Inerzia_tot(1,1) ; Inerzia_tot(1,1)-Inerzia_tot(2,2) ; ];

period_orbit = 1294922.63837404; %seconds
time = linspace(0,10*period_orbit,1000);
H_gravity = (M_gravity * time)';

P_earth = 4.5632 * 10^-6; % pressione solare, pascal
r_sun_to_earth = 149597871000; %metri
P_sat = P_earth * (r_sun_to_earth / (r_sun_to_earth-distance_earth_moon-max_distX))^2;

apertura_alare = 4.5; %m
vector_pressure_center = [0 ,apertura_alare/100 , 0 ];
ro_specularity = 0.15;
ro_absorbed = 0.7;
ro_diffusively = 0.15;
```



C. CALCULATIONS

```
A_sp = 3.5; %m^2
F_solar = [0,0,P_sat * A_sp * (ro_absorbed+ro_diffusively+2*ro_specularity+2/3*
    ro_diffusively)];
M_solar = cross(vector_pressure_center, F_solar)';
H_solar = (M_solar*time)'

M_tot = M_solar + M_gravity
H_tot = (H_solar+H_gravity);

plot(time,H_tot);
xticks(0:15*24*60*60:10*15*24*60*60);
xticklabels({'0', '15', '30', '45', '60', '75', '90', '105', '120', '135', '150'});
title('Momentum to be stored in the reaction wheels');
xlabel('[Days]');
ylabel('[Nms]');
legend('h_x','h_y','h_z')

lifeTime = 10*365*24*60*60; %seconds % 10 years
lifeOrbits = 10*365/15 %days
H_tot_10orb = 11.705655725052267 %Nms
H_tot_life = H_tot_10orb/10*lifeOrbits
N_dumping = H_tot_life / 4;

I_max = 130 %kg*m^2 inerzia max
w_sat = H_tot_life/I_max %rad/s
F_thrusters = 1 %N
t_desat_tot = H_tot_life/F_thrusters
t_desat_1dump = t_desat_tot/N_dumping
nominal_flow = 2* 0.44 %g/s % has to be multiplied
prop_to_dump = t_desat_tot * nominal_flow %grams

%%%%%%%%%%%%% orbit maneuvers %%%%%%%%%%%%%%
M_gyro = 0.3 %Nm
alpha_sat = M_gyro/I_max
t_180 = 2*sqrt(2*pi/alpha_sat)
%%%%%%%%%%%%% SLEW %%%%%%%%%%%%%%
w_max = 1.03*10^-3 % deg /s
alpha_max = 5.275*10^-9 %deg /s^2
w_max = w_max*pi/180 %rad/s
alpha_max = alpha_max*pi/180
h_slew = w_max*I_max %Nns momento angolare
```



M_slew = alpha_max*I_max %Nm torque



SUPREMAE
DIGNITATIS
1343

UNIVERSITÀ
DI PISA

C.7. Camera Orbit (used in Attitude Plot and Calculation)

```
%this script moves the camera of the script plotDHaloOrbit.m file , so that a moving
%animation can be made
%made by Mauro Furfari
% this script is called cameraOrbit.m
plotBool = false; % put this to true to see the camera trajectory
dimentionsss = 461;
if plotBool
    clear all
    close all
    clc
    plotBool = true;
    dimentionsss = 461/5;
end

r = 40000;
x_c = r;
y_c = 0;
t = linspace(0,pi,dimentionsss);
clear dimentionsss;
a = [r*cos(t) + x_c]';
b = [r*sin(t) + y_c]';
c = [zeros(1,length(a))]'';

VCam1 = [c,-b,a];
VCam2 = [flipud(a),-b,c];

VCam = VCam1+VCam2;
VCam = VCam./vecnorm(VCam,2,2)*2*r;
VCamDir = VCam;

offset = 40000;
rate = offset/pi;
pippo = offset - rate*t';
pippoVet = [pippo, zeros(length(pippo),2)];
VCam = VCam + pippoVet;

freeze = (0.78 * length(VCam));
```



```
freeze = round(0.728 * length(VCam));
for i = freeze:length(VCam)
    VCam(i,:) = VCam(freeze,:);
    VCamDir(i,:) = VCamDir(freeze,:);
end

if plotBool
    figure
    axis equal
    grid on
    hold on
    xlabel('x')
    ylabel('y')
    zlabel('z')
    view([30 35])
    for i = 1:length(VCam)
        plot3(VCam1(i,1), VCam1(i,2), VCam1(i,3), 'ro')
        plot3(VCam2(i,1), VCam2(i,2), VCam2(i,3), 'go')
        plot3(VCam(i,1), VCam(i,2), VCam(i,3), 'bo')
        getframe;
        if mod(i, 2) > 0
            drawnow
        end
    end
end
```

C.8. Attitude Plot and Calculation

```

close all
clear all
clc

% made by Mauro Furfari, Lucrezia Podestà, Mario Nocilli
% this script is called plot3DHaloOrbit.m
##### USER CONFIG #####
##### Initialize all variables to true or false to show different things
plotHaloOrbit      = true;
plotTransferOrbit   = false;
plotEarth           = true;
plotMoon            = true;
plotEML2             = true;
plotSpecialPoints   = true;
plotEarthMoonPlane  = true;
plotSatellites      = true;
plotAntennaVectors  = true;
plotAttitudeControl = false;
plotInertialFrame   = true;
plotOrbitalFrame    = true;
plotW_OI             = true;
plotAnimation        = false;
plotVectorsToMoonEarth = false;
setCameraControl     = false;

%%%%%%%%%%%%% LOAD DATA %%%%%%
% Load csv file with position and velocity data of 1 halo orbit from the
% website 'https://ssd.jpl.nasa.gov/tools/periodic\_orbits.html#/periodic'
dataHalo = readmatrix('propagationOfOneHaloOrbit.csv');
% load values from other script; make sure to disable all plot and remove
% 'close all, clear all, clc' from those other script
cameraOrbit;
calcoloInerzia;

%%%%%%%%%%%%% CONSTANTS %%%%%%
% Length unit of the moon
LU = 389703; % km
% Time unit of the moon
TU = 382981; % s

```



% Distance from the Earth to the Moon

a_moon = 384400; % km

%%%%%%%%%%%%% DATA ELABORATION %%%%%%%%%%%%%%

% Convert propagation0fOneHaloOrbit.csv data from lunar units to km and s

```
dataHalo(:,1) = dataHalo(:,1) * TU;  
dataHalo(:,2) = dataHalo(:,2) * LU - 384400;  
dataHalo(:,3:4) = dataHalo(:,3:4) * LU;  
dataHalo(:,5:7) = dataHalo(:,5:7) * LU / TU;
```

halo_time = dataHalo(:,1); % Halo orbit time

halo_position = [dataHalo(:,2:4)]; % Halo orbit position

halo_velocity = [dataHalo(:,5:7)]; % Halo orbit velocity

arraySize = size(halo_position, 1);

last_time = halo_time(end,1); % Select last value of time

period_orbit = last_time / (60 * 60 * 24); % Period of orbit in days

%%%%%%%%%%%%% STATIC OBJECTS %%%%%%%%%%%%%%

% Earth

earth_radius = 6371; % km

earth_distance = -384400; % km

[x, y, z] = sphere;

x_earth = x * earth_radius + earth_distance;

y_earth = y * earth_radius;

z_earth = z * earth_radius;

% Moon

moon_radius = 1737.4; % km

[x, y, z] = sphere;

y_moon = y * moon_radius;

x_moon = x * moon_radius;

z_moon = z * moon_radius;

% EML2 point

x_eml2 = 65737; % km

%##### ANTENNA

VECTORS #####

% Calculate set of vectors pointing from the position of the halo orbit to the moon

moon_position = [0, 0, 0]; % Moon position at origin

vector_to_moon = moon_position - halo_position; % Vector pointing from halo orbit to

C. CALCULATIONS

moon

```
% Calculate set of vectors pointing from the position of the halo orbit to the Earth
earth_position = [earth_distance, 0, 0]; % Earth position at origin
vector_to_earth = earth_position - halo_position; % Vector pointing from halo orbit to
Earth

%##### SPECIAL POINTS #####
% Find coordinates where x is minimum
[min_x, min_x_index] = min(halo_position(:,1));
min_distance = halo_position(min_x_index, :);

% Find coordinates where x is maximum
[max_x, max_x_index] = max(halo_position(:,1));
max_distance = halo_position(max_x_index, :);

% Find coordinates where y is maximum
[max_y, max_y_index] = max(halo_position(:,2));
far_right = halo_position(max_y_index, :);

% Find coordinates where y is minimum
[min_y, min_y_index] = min(halo_position(:,2));
far_left = halo_position(min_y_index, :);

%##### CALCULATION ANGLES #####
% Calculate the angles between vector_to_moon and the coordinate axes
anglesYZ = atan2d(vector_to_moon(:,3), vector_to_moon(:,2)); % Angle on the yz-plane (
normal to x vector)
toMoonAnglesXZ = atan2d(vector_to_moon(:,3), vector_to_moon(:,1)); % Angle on the xz-
plane (normal to -y vector)
toMoonAnglesXY = atan2d(vector_to_moon(:,2), vector_to_moon(:,1)); % Angle on the xy-
plane (normal to z vector)
% Calculate the angles between vector_to_moon and the coordinate axes
toEarthAnglesXZ = atan2d(vector_to_earth(:,3), vector_to_earth(:,1)); % Angle on the
xz-plane (normal to -y vector)
toEarthAnglesXY = atan2d(vector_to_earth(:,2), vector_to_earth(:,1)); % Angle on the
xy-plane (normal to z vector)

% Convert angles to radians for unwrap
```



```
anglesYZ_rad = deg2rad(anglesYZ);

% Apply unwrap to angles in radians
anglesYZ_rad_unwrapped = unwrap(anglesYZ_rad);

% Convert the unwrapped angles back to degrees
anglesYZ = rad2deg(anglesYZ_rad_unwrapped);

changeYZ = (linspace(0,360,arraySize))';
lineYZ = 90 - changeYZ;
anglesYZadj = anglesYZ + changeYZ +90;

toMoonAnglesXZ(toMoonAnglesXZ < 0) = toMoonAnglesXZ(toMoonAnglesXZ < 0) + 360;
toMoonAnglesXY(toMoonAnglesXY < 0) = toMoonAnglesXY(toMoonAnglesXY < 0) + 360;
toEarthAnglesXZ(toEarthAnglesXZ < 0) = toEarthAnglesXZ(toEarthAnglesXZ < 0) + 360;
toEarthAnglesXY(toEarthAnglesXY < 0) = toEarthAnglesXY(toEarthAnglesXY < 0) + 360;

##### SYSTEMS OF REFERENCE
#####
% INERTIAL FRAME
i_inerz = [1 0 0];
j_inerz = [0 1 0];
k_inerz = [0 0 1];

% Define in which direction a face of the satellite should point: towards the moon?
% towards the earth? towards a midpoint?
middle_vector = (vector_to_earth./vecnorm(vector_to_earth,2,2) + vector_to_moon./
vecnorm(vector_to_moon,2,2));

%antennaPointingVector = vector_to_moon; % posso scegliere tra vector_to_moon ,
% vector_to_earth, middle_vector
antennaPointingVector = middle_vector;
% ORBITAL FRAME
k_orb = antennaPointingVector ./ vecnorm(antennaPointingVector, 2, 2);
j_orb = - cross(halo_velocity,antennaPointingVector) ./ vecnorm(cross(halo_velocity,
antennaPointingVector), 2, 2);
i_orb = cross(j_orb,k_orb);

% Check if versors are of length 1
checkNorm = [vecnorm(i_orb,2,2), vecnorm(j_orb,2,2), vecnorm(k_orb,2,2)];

% Scale the versors
```



C. CALCULATIONS

```
i_inerz_scaled = i_inerz * 10000;
j_inerz_scaled = j_inerz * 10000;
k_inerz_scaled = k_inerz * 10000;
i_orb_scaled = i_orb * 100;
j_orb_scaled = j_orb * 100;
k_orb_scaled = k_orb * 100;

%##### ANGULAR
    VELOCITY #####
% Calculate the finite differences of the orbital versors
di_orb_dt = diff(i_orb) ./ diff(halo_time);
dj_orb_dt = diff(j_orb) ./ diff(halo_time);
dk_orb_dt = diff(k_orb) ./ diff(halo_time);

% For the last point, duplicate the penultimate point or use another method to
    approximate the derivative
di_orb_dt(end+1, :) = di_orb_dt(end, :);
dj_orb_dt(end+1, :) = dj_orb_dt(end, :);
dk_orb_dt(end+1, :) = dk_orb_dt(end, :);

% Preallocation for angular velocity
W_OI_inerz = zeros(size(i_orb));
W_OI_orb = zeros(size(i_orb));

% Calculate the angular velocity for each point in time, excluding the last point
    which is approximated
for i = 1:arraySize
    % Calculate the angular velocity W_OI using the cross product
    W_OI_inerz(i,:) = cross(di_orb_dt(i,:), i_orb(i,:)) + cross(dj_orb_dt(i,:), j_orb(
        i,:)) + cross(dk_orb_dt(i,:), k_orb(i,:));
    % Calculate the components of W_OI in the orbital reference frame
    W_OI_orb(i,:) = [dot(W_OI_inerz(i,:), i_orb(i,:)), dot(W_OI_inerz(i,:), j_orb(i,:))
        , dot(W_OI_inerz(i,:), k_orb(i,:))];
    % Calculate W_OI_orb written in the inertial reference frame

    % Transform W_OI_orb into the inertial reference frame for plotting
    R_orb_to_iner = [i_orb(i,:)', j_orb(i,:)', k_orb(i,:)']; % Rotation matrix from
        the orbital to the inertial reference frame

    W_OI_orb_inerz(i,:) = R_orb_to_iner * W_OI_orb(i, :)';
end
```



```
%##### ANGULAR ACCELERATION #####
#####
% Calculate the finite differences of the angular velocities
dW_OI_orb_dt = diff(W_OI_orb) ./ diff(halo_time);
dW_OI_inerz_dt = diff(W_OI_inerz) ./ diff(halo_time);

% For the last point, duplicate the penultimate point or use another method to
% approximate the derivative
dW_OI_orb_dt(end+1, :) = dW_OI_orb_dt(end, :);
dW_OI_inerz_dt(end+1, :) = dW_OI_inerz_dt(end, :);

%##### CALCULUS TORQUE OF CONTROL #####
for i = 1:arraySize
    M_orb(i,1) = sc_inertia(1,1) * dW_OI_orb_dt(i,1) + (sc_inertia(3,3) - sc_inertia
        (2,2)) * W_OI_orb(i,2) * W_OI_orb(i,3);
    M_orb(i,2) = sc_inertia(2,2) * dW_OI_orb_dt(i,2) + (sc_inertia(1,1) - sc_inertia
        (3,3)) * W_OI_orb(i,3) * W_OI_orb(i,1);
    M_orb(i,3) = sc_inertia(3,3) * dW_OI_orb_dt(i,3) + (sc_inertia(2,2) - sc_inertia
        (1,1)) * W_OI_orb(i,1) * W_OI_orb(i,2);

    M_inerz(i,1) = sc_inertia(1,1) * dW_OI_inerz_dt(i,1) + (sc_inertia(3,3) -
        sc_inertia(2,2)) * W_OI_inerz(i,2) * W_OI_inerz(i,3);
    M_inerz(i,2) = sc_inertia(2,2) * dW_OI_inerz_dt(i,2) + (sc_inertia(1,1) -
        sc_inertia(3,3)) * W_OI_inerz(i,3) * W_OI_inerz(i,1);
    M_inerz(i,3) = sc_inertia(3,3) * dW_OI_inerz_dt(i,3) + (sc_inertia(2,2) -
        sc_inertia(1,1)) * W_OI_inerz(i,1) * W_OI_inerz(i,2);
end

%##### KINETIC ROTATIONAL ENERGY #####
E_cin_rot_orb = 0.5 * (sc_inertia(1,1) * W_OI_orb(:,1).^2 + sc_inertia(2,2) * W_OI_orb
    (:,2).^2 + sc_inertia(3,3) * W_OI_orb(:,3).^2);
E_cin_rot_inerz = 0.5 * (sc_inertia(1,1) * W_OI_inerz(:,1).^2 + sc_inertia(2,2) *
    W_OI_inerz(:,2).^2 + sc_inertia(3,3) * W_OI_inerz(:,3).^2);

deltaEcin_orb = max(E_cin_rot_orb) - min(E_cin_rot_orb);
deltaEcin_inerz = max(E_cin_rot_inerz) - min(E_cin_rot_inerz);

%
```



```
%##### FIGURE 1
PLOT #####
% Plot 3D halo orbit
figureSpaceObjects = figure('Name', 'plot animation of halo orbit in 3D', 'Units', 'normalized', 'OuterPosition', [0.1 0.1 0.7 0.7]);
hold on;
grid on;
axis equal;

%totalBlackFigure(figureSpaceObjects); % <--- use this to ma the figure total black;
% but you will need to change the color of the labels of earth,moon and eml2n
if setCameraControl
    f_scale = 10000;
    xlim([-moon_radius max_distance(:,1)+f_scale ]);
    ylim([far_left(:,2)-f_scale far_right(:,2)+f_scale]);
    zlim([max_distance(:,3)-f_scale min_distance(:,3)+f_scale ]);
else
    f_scale = 10000;
    %xlim([earth_distance-earth_radius, max_distance(:,1)+f_scale]);
    xlim([-earth_radius+earth_distance, max_distance(:,1)+f_scale]);
    ylim([far_left(:,2)-f_scale, far_right(:,2)+f_scale]);
    zlim([max_distance(:,3)-f_scale, min_distance(:,3)+f_scale ]);
end

if plotHaloOrbit
    % plot orbit
    plot3(halo_position(:,1), halo_position(:,2), halo_position(:,3), 'r', 'LineWidth', 3);
end

if plotTransferOrbit
    % plot transfer orbit
    plot3(transfer_position(:,1), transfer_position(:,2), transfer_position(:,3), 'k')
    ;
end

if plotMoon
    % plot moon
    mesh(x_moon, y_moon, z_moon, 'FaceColor', [0.8 0.8 0.8], 'EdgeColor', 'k');
    text(3000, 3000, 3000, 'Moon', 'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom', 'Color', 'black');

```



```
end

if plotEML2
    % plot EML2 point
    plot3(x_eml2, 0, 0, 'ro', 'MarkerSize', 5);
    text(x_eml2+2000, 2000, 2000, 'EML2 point', 'HorizontalAlignment', 'center', ...
        'VerticalAlignment', 'bottom', 'Color', 'red');
end

if plotEarth
    % plot Earth
    mesh(x_earth, y_earth, z_earth, 'FaceColor', [0 0.5 0], 'EdgeColor', [0 0 1]);
    text(earth_distance+2*earth_radius, 8000, 8000, 'Earth', 'HorizontalAlignment', ...
        'center', 'VerticalAlignment', 'bottom', 'Color', 'b');
end

if plotSpecialPoints
    % plot circular special points
    plot3(min_distance(1), min_distance(2), min_distance(3), 'ko', 'MarkerSize', 5, ...
        'MarkerFaceColor', 'magenta');
    plot3(max_distance(1), max_distance(2), max_distance(3), 'ko', 'MarkerSize', 5, ...
        'MarkerFaceColor', 'magenta');
    plot3(far_right(1), far_right(2), far_right(3), 'ko', 'MarkerSize', 5, ...
        'MarkerFaceColor', 'magenta');
    plot3(far_left(1), far_left(2), far_left(3), 'ko', 'MarkerSize', 5, ...
        'MarkerFaceColor', 'magenta');
end

if plotEarthMoonPlane
    % plot earth-moon plane
    [X, Y] = meshgrid((3/2)*earth_distance:10000:-(1/3)*earth_distance, earth_distance ...
        :10000:-earth_distance);
    Z = zeros(size(X)); % Set z-coordinate to 0 for all points
    surf(X, Y, Z, 'FaceColor', [0 0 1], 'FaceAlpha', 0.3, 'EdgeColor', [0 0 0], ...
        'EdgeAlpha', 0.3);
end

if plotInertialFrame
    %plot inertial frame
    iIndexVersors = quiver3(0, 0, 0, i_inerz_scaled(1), i_inerz_scaled(2),
        i_inerz_scaled(3), 'r', 'LineWidth', 2, 'DisplayName', 'X_inerz');
    jIndexVersors = quiver3(0, 0, 0, j_inerz_scaled(1), j_inerz_scaled(2),
```

C. CALCULATIONS

```
j_inerz_scaled(3), 'g', 'LineWidth', 2, 'DisplayName', 'Y_inerz');  
kIndexVersors = quiver3(0, 0, 0, k_inerz_scaled(1), k_inerz_scaled(2),  
    k_inerz_scaled(3), 'b', 'LineWidth', 2, 'DisplayName', 'Z_inerz');  
end  
if plotW_OI  
    %plot angular velocity  
    quiver3(halo_position(:,1), halo_position(:,2), halo_position(:,3), W_OI_orb(:,1),  
        W_OI_orb(:,2), W_OI_orb(:,3), 'color', '#FF00FF', 'LineWidth', 1, 'DisplayName'  
        , 'W_OI');  
end  
  
% Set legend  
legend('Halo Orbit', 'Moon', 'EML2 Point', 'Earth', 'Min Distance', 'Max Distance'  
    , 'Far Right', 'Far Left', 'Earth-Moon Plane', 'Satellites', 'Location', '  
northwest');
```

%##### FIGURE 2

```
PLOT #####
```

```
%figure for pointing vectors  
figurePointingVectors = figure('Name', 'pointing vectors');
```

%##### SUBPLOT 1

```
#####
```

```
%subplot to moon  
subplot(2,2,1);  
hold on  
grid on  
title('directions to the moon');  
xlabel('Equivalent time where 2pi is 1 full orbit');  
ylabel('Angle [degrees]');
```

```
% Imposta gli intervalli per l'asse x (da 0 a 4*pi con intervalli di pi/2)  
xticks(0:pi/2:4*pi);  
% etichette personalizzate in formato pi  
xticklabels({'0', '\pi/2', '\pi', '3\pi/2', '2\pi', '5\pi/2', '3\pi', '7\pi/2', '4\pi'  
});  
% Imposta i limiti per l'asse Y  
xlim([0 4*pi]);  
  
% Imposta gli intervalli per l'asse y (da -270 a +270 con intervalli di 90)
```

```

yticks(-360:10:360);
% Imposta i limiti per l'asse y
ylim([140 220]);

% Plot angles
%plot(linspace(0, 2*pi, arraySize) , anglesYZ, 'r', 'DisplayName', 'Angle on the YZ-
% plane');
plot(linspace(0, 2*pi, arraySize) , anglesYZadj, 'r', 'DisplayName', 'Angle on the YZ-
plane');
plot(linspace(0, 2*pi, arraySize) , toMoonAnglesXZ, 'G', 'DisplayName', 'Angle on the
XZ-plane');
plot(linspace(0, 2*pi, arraySize) , toMoonAnglesXY, 'B', 'DisplayName', 'Angle on the
XY-plane');
%plot(linspace(2*pi, 4*pi, arraySize) , anglesYZ, 'r');
plot(linspace(2*pi, 4*pi, arraySize) , anglesYZadj, 'r');
plot(linspace(2*pi, 4*pi, arraySize) , toMoonAnglesXZ, 'G');
plot(linspace(2*pi, 4*pi, arraySize) , toMoonAnglesXY, 'B');
% Set legend
legend( 'Angle on the YZ-plane', 'Angle on the XZ-plane', 'Angle on the XY-plane', '
Location', 'southoutside');

#####
##### SUBPLOT 3 #####
#####

%subplot to earth
subplot(2,2,3);
hold on
grid on
title('directions to the earth');
xlabel('Equivalent time where 2pi is 1 full orbit');
ylabel('Angle [degrees]');
% Imposta gli intervalli per l'asse x (da 0 a 4*pi con intervalli di pi/2)
xticks(0:pi/2:4*pi);
% etichette personalizzate in formato pi
xticklabels({'0', '\pi/2', '\pi', '3\pi/2', '2\pi', '5\pi/2', '3\pi', '7\pi/2', '4\pi'
});
% Imposta i limiti per l'asse Y
xlim([0 4*pi]);

% Imposta gli intervalli per l'asse y (da -270 a +270 con intervalli di 90)
yticks(-360:10:360);
% Imposta i limiti per l'asse y

```



C. CALCULATIONS

```
ylim([140 220]);\n\n% Plot angles\nplot(linspace(0, 2*pi, arraySize) , anglesYZadj, 'r', 'DisplayName', 'Angle on the YZ-\n    plane');\nplot(linspace(0, 2*pi, arraySize) , toEarthAnglesXZ, 'G', 'DisplayName', 'Angle on the\n    XZ-plane');\nplot(linspace(0, 2*pi, arraySize) , toEarthAnglesXY, 'B', 'DisplayName', 'Angle on the\n    XY-plane');\nplot(linspace(2*pi, 4*pi, arraySize) , anglesYZadj, 'r');\nplot(linspace(2*pi, 4*pi, arraySize) , toEarthAnglesXZ, 'G');\nplot(linspace(2*pi, 4*pi, arraySize) , toEarthAnglesXY, 'B');\n% Set legend\nlegend( 'Angle on the YZ-plane', 'Angle on the XZ-plane', 'Angle on the XY-plane', '\n    Location', 'southoutside');\n\n%##### SUBPLOT 2\n#####\n\n%subplot to earth\nsubplot(2,2,2);\nhold on\ngrid on\ntitle('Comparison Y and Z angles');\nxlabel('Equivalent time where 2pi is 1 full orbit');\nylabel('Angle [degrees]');\n% Imposta gli intervalli per l'asse x (da 0 a 4*pi con intervalli di pi/2)\nxticks(0:pi/2:4*pi);\n% etichette personalizzate in formato pi\nxticklabels({'0', '\pi/2', '\pi', '3\pi/2', '2\pi', '5\pi/2', '3\pi', '7\pi/2', '4\pi'\n});\n% Imposta i limiti per l'asse Y\nxlim([0 4*pi]);\n\n% Imposta gli intervalli per l'asse y (da -270 a +270 con intervalli di 90)\nyticks(0:5:360);\n% Imposta i limiti per l'asse y\nylim([140 220]);\n\n% Plot angles\nplot(linspace(0, 2*pi, arraySize) , toMoonAnglesXZ, 'g', 'DisplayName', 'Moon Angle on\n    the XZ-plane');
```



```
plot(linspace(0, 2*pi, arraySize) , toMoonAnglesXY, 'b', 'DisplayName', 'Moon Angle on
the XZ-plane');
plot(linspace(0, 2*pi, arraySize) , toEarthAnglesXZ, 'g.', 'DisplayName', 'Earth Angle
on the XY-plane');
plot(linspace(0, 2*pi, arraySize) , toEarthAnglesXY, 'b.', 'DisplayName', 'Earth Angle
on the XY-plane');
plot(linspace(2*pi, 4*pi, arraySize) , toMoonAnglesXZ, 'g');
plot(linspace(2*pi, 4*pi, arraySize) , toMoonAnglesXY, 'b');
plot(linspace(2*pi, 4*pi, arraySize) , toEarthAnglesXZ, 'g.');
plot(linspace(2*pi, 4*pi, arraySize) , toEarthAnglesXY, 'b.');
% Set legend
legend('Moon Angle on the XZ-plane', 'Moon Angle on the XY-plane', 'Earth Angle on the
XZ-plane', 'Earth Angle on the XY-plane', 'Location', 'southoutside');

%##### SUBPLOT 4
#####

%subplot to earth
subplot(2,2,4);
hold on
grid on
title('Comparison X angles');
xlabel('Equivalent time where 2pi is 1 full orbit');
ylabel('Angle [degrees]');
% Imposta gli intervalli per l'asse x (da 0 a 4*pi con intervalli di pi/2)
xticks(0:pi/2:4*pi);
% etichette personalizzate in formato pi
xticklabels({'0', '\pi/2', '\pi', '3\pi/2', '2\pi', '5\pi/2', '3\pi', '7\pi/2', '4\pi'});
% Imposta i limiti per l'asse Y
xlim([0 4*pi]);

% Imposta gli intervalli per l'asse y (da -270 a +270 con intervalli di 90)
yticks(-360:90:360);
% Imposta i limiti per l'asse y
ylim([140 220]);

% Plot angles
plot(linspace(0, 2*pi, arraySize) , anglesYZ, 'r', 'DisplayName', 'Moon Angle on the
YZ-plane');
plot(linspace(0, 2*pi, arraySize) , anglesYZadj, 'g', 'DisplayName', 'Moon Angle on
the YZ-plane adjusted');
```

C. CALCULATIONS

```
plot(linspace(0, 2*pi, arraySize) , lineYZ, 'b', 'DisplayName', 'Line YZ-plane');

plot(linspace(2*pi, 4*pi, arraySize) , anglesYZ, 'r');
plot(linspace(2*pi, 4*pi, arraySize) , anglesYZadj, 'g');
plot(linspace(2*pi, 4*pi, arraySize) , lineYZ, 'b');
% Set legend
legend('Moon Angle on the YZ-plane', 'Moon Angle on the YZ-plane adjusted', 'Line YZ-
plane', 'Location', 'southoutside');

%##### FIGURE 3
PLOT #####
%figure for attitude control
figureAttitudeControl = figure('Name', 'attitude control');
%##### SUBPLOT 1
#####

%subplot to moon
subplot(2,2,1);
hold on
grid on
title('Angular Velocity of the orbital reference system');
xlabel('Equivalent time where  $2\pi$  is 1 full orbit');
ylabel('[degrees]/[s]');

% Imposta gli intervalli per l'asse x (da 0 a  $4\pi$  con intervalli di  $\pi/2$ )
xticks(0:pi/2:4*pi);
% etichette personalizzate in formato pi
xticklabels({'0', '\pi/2', '\pi', '3\pi/2', '2\pi', '5\pi/2', '3\pi', '7\pi/2', '4\pi'});
% Imposta i limiti per l'asse Y
xlim([0 4*pi]);

% Imposta gli intervalli per l'asse y (da -270 a +270 con intervalli di 90)
%yticks(-360:10:360);
% Imposta i limiti per l'asse y
%ylim([140 220]);

% Plot angles
plot(linspace(0, 2*pi, arraySize) , W_OI_inerz(:,1)*180/pi, 'r', 'DisplayName', 'w_{
    orb}_x');
plot(linspace(0, 2*pi, arraySize) , W_OI_inerz(:,2)*180/pi, 'g', 'DisplayName', 'w_{


```



```
orbY});  
plot(linspace(0, 2*pi, arraySize) , W_OI_inerz(:,3)*180/pi, 'b', 'DisplayName', 'w_{  
orb}_z');  
plot(linspace(2*pi, 4*pi, arraySize) , W_OI_inerz(:,1)*180/pi, 'r');  
plot(linspace(2*pi, 4*pi, arraySize) , W_OI_inerz(:,2)*180/pi, 'g');  
plot(linspace(2*pi, 4*pi, arraySize) , W_OI_inerz(:,3)*180/pi, 'b');  
% Set legend  
legend( 'w_x', 'w_y', 'w_z', 'Location', 'best');  
  
##### SUBPLOT 2 #####  
  
%subplot to moon  
subplot(2,2,2);  
hold on  
grid on  
title('Angular Acceleration of the orbital reference system');  
xlabel('Equivalent time where 2\pi is 1 full orbit');  
ylabel('[degrees]/[s^2]');  
  
% Imposta gli intervalli per l'asse x (da 0 a 4*pi con intervalli di pi/2)  
xticks(0:pi/2:4*pi);  
% etichette personalizzate in formato pi  
xticklabels({'0', '\pi/2', '\pi', '3\pi/2', '2\pi', '5\pi/2', '3\pi', '7\pi/2', '4\pi'  
});  
% Imposta i limiti per l'asse Y  
xlim([0 4*pi]);  
  
% Imposta gli intervalli per l'asse y (da -270 a +270 con intervalli di 90)  
%yticks(-360:10:360);  
% Imposta i limiti per l'asse y  
%ylim([140 220]);  
  
% Plot angles  
plot(linspace(0, 2*pi, arraySize) , dW_OI_inerz_dt(:,1)*180/pi, 'r', 'DisplayName', '  
w_{orb}_x');  
plot(linspace(0, 2*pi, arraySize) , dW_OI_inerz_dt(:,2)*180/pi, 'g', 'DisplayName', '  
w_{orb}_y');  
plot(linspace(0, 2*pi, arraySize) , dW_OI_inerz_dt(:,3)*180/pi, 'b', 'DisplayName', '  
w_{orb}_z');  
plot(linspace(2*pi, 4*pi, arraySize) , dW_OI_inerz_dt(:,1)*180/pi, 'r');  
plot(linspace(2*pi, 4*pi, arraySize) , dW_OI_inerz_dt(:,2)*180/pi, 'g');
```



SUPREMAE
DIGNITATIS
1343

UNIVERSITÀ
DI PISA

C. CALCULATIONS

```
plot(linspace(2*pi, 4*pi, arraySize) , dW_0I_inerz_dt(:,3)*180/pi, 'b');
% Set legend
legend( '\alpha_x', '\alpha_y', '\alpha_z', 'Location', 'best');

##### SUBPLOT 3 #####
#####

%subplot to moon
subplot(2,2,3);
hold on
grid on
title('Momentum');
xlabel('Equivalent time where 2pi is 1 full orbit');
ylabel('[Nm]');

% Imposta gli intervalli per l'asse x (da 0 a 4*pi con intervalli di pi/2)
xticks(0:pi/2:4*pi);
% etichette personalizzate in formato pi
xticklabels({'0', '\pi/2', '\pi', '3\pi/2', '2\pi', '5\pi/2', '3\pi', '7\pi/2', '4\pi'});
% Imposta i limiti per l'asse Y
xlim([0 4*pi]);

% Imposta gli intervalli per l'asse y (da -270 a +270 con intervalli di 90)
%yticks(-360:10:360);
% Imposta i limiti per l'asse y
%ylim([140 220]);

% Plot angles
plot(linspace(0, 2*pi, arraySize) , M_inerz(:,1), 'r', 'DisplayName', 'w_orb_x');
plot(linspace(0, 2*pi, arraySize) , M_inerz(:,2), 'g', 'DisplayName', 'w_orb_y');
plot(linspace(0, 2*pi, arraySize) , M_inerz(:,3), 'b', 'DisplayName', 'w_orb_z');
plot(linspace(2*pi, 4*pi, arraySize) , M_orb(:,1), 'r');
plot(linspace(2*pi, 4*pi, arraySize) , M_orb(:,2), 'g');
plot(linspace(2*pi, 4*pi, arraySize) , M_orb(:,3), 'b');
% Set legend
legend( 'w_orb_x', 'w_orb_y', 'w_orb_z', 'Location', 'best');
##### SUBPLOT 4 #####
#####

%subplot to moon
subplot(2,2,4);
```



```
hold on
grid on
title('Kinetic Rotational Energy');
xlabel('Equivalent time where 2pi is 1 full orbit');
ylabel('[joules]');

% Imposta gli intervalli per l'asse x (da 0 a 4*pi con intervalli di pi/2)
xticks(0:pi/2:4*pi);
% etichette personalizzate in formato pi
xticklabels({'0', '\pi/2', '\pi', '3\pi/2', '2\pi', '5\pi/2', '3\pi', '7\pi/2', '4\pi'
    });
% Imposta i limiti per l'asse Y
xlim([0 4*pi]);

% Imposta gli intervalli per l'asse y (da -270 a +270 con intervalli di 90)
%yticks(-360:10:360);
% Imposta i limiti per l'asse y
%ylim([140 220]);

% Plot angles
plot(linspace(0, 2*pi, arraySize) , E_cin_rot_inerz(:,1), 'r', 'DisplayName', 'w_orb_x
    ');
%plot(linspace(0, 2*pi, arraySize) , E_cin_rot_inerz(:,2), 'g', 'DisplayName', 'w_orb_y');
%plot(linspace(0, 2*pi, arraySize) , E_cin_rot_inerz(:,3), 'b', 'DisplayName', 'w_orb_z');
plot(linspace(2*pi, 4*pi, arraySize) , E_cin_rot_orb(:,1), 'r');
%plot(linspace(2*pi, 4*pi, arraySize) , E_cin_rot_orb(:,2), 'g');
%plot(linspace(2*pi, 4*pi, arraySize) , E_cin_rot_orb(:,3), 'b');
% Set legend
legend( 'w_orb_x', 'w_orb_y', 'w_orb_z', 'Location', 'best');

%% animation -- use this to view the satellites moving and save cool animations
##### FOR LOOP
    ANIMATION OF FIGURE 1
#####
% create animation with for loop
figure(figureSpaceObjects);
% Ottieni la data e l'ora corrente
currentTime = now;
% Converti la data e l'ora corrente in una stringa con il formato desiderato
```

C. CALCULATIONS

```
timestamp = datestr(currentTime, 'yyyyymmdd_HHMM');

%axis tight manual % assicura che getframe() restituisca un frame coerente
%axis vis3d
%set(figureSpaceObjects, 'Position', [100 100 1200 800]); % Dimensioni della finestra
    di visualizzazione
%view(3); % Imposta la vista 3D iniziale
timestamp = datestr(now, 'yyyy-mm-dd_HH-MM-SS
');
% Crea il nome del file con il timestamp
videoFilename = ['generatedAnimations/animazione_'
    timestamp '.mp4'];
% Usa il nome del file per creare un oggetto VideoWriter
Video = VideoWriter(videoFilename, 'MPEG-4');
open(Video); % Apri il file video per la scrittura

% Inizializza la vista iniziale
%view(30, 10); % Angoli azimutali ed elevazione
%camproj('orthographic'); % Imposta la proiezione della camera 'orthographic'

loopDuration = 1.1*arraySize;
for i = 1:loopDuration
    % define indexs for animation
    global indexs;
    indexs = [mod(round(i+0.00*arraySize), arraySize)+1, mod(round(i+0.25*arraySize),
        arraySize)+1, mod(round(i+0.50*arraySize), arraySize)+1, mod(round(i+0.75*
        arraySize), arraySize)+1];
    j = mod(round(i+0.00*arraySize), arraySize)+1;

    % plot squares in blue
    squares = plot3(halo_position(indexs, 1), halo_position(indexs, 2), halo_position(
        indexs, 3), 'ks', 'MarkerSize', 15, 'MarkerFaceColor', 'blue'); % plot
    vector to moon in green
    if plotVectorsToMoonEarth
        % plot vector to moon in green
        vectors_to_moon = plotVector(vector_to_moon, halo_position, 'g');
        % plot vector to Earth in red
        vectors_to_earth = plotVector(vector_to_earth, halo_position, 'r');
    end

    % plot translated velocity vectors in blue
    velocity_vectors = plotVector(halo_velocity, halo_position, 'b');

    if plotOrbitalFrame
```



```
%plot orbital frame
iOrbVersors = plotVector(i_orb_scaled, halo_position, 'r', 'LineWidth', 2, 'DisplayName', 'X_orb');
jOrbVersors = plotVector(j_orb_scaled, halo_position, 'g', 'LineWidth', 2, 'DisplayName', 'Y_orb');
kOrbVersors = plotVector(k_orb_scaled, halo_position, 'b', 'LineWidth', 2, 'DisplayName', 'Z_orb');
end

if plotW_OI
    %plot angular velocity O-I wrt orbital frame
    W_OI_orb_handler = plotVector(W_OI_orb, halo_position, 'y', 'LineWidth', 1, 'DisplayName', 'W_OI_orb');
    %plot angular velocity O-I wrt inertial frame
    W_OI_inerz_handler = plotVector(W_OI_inerz, halo_position, 'm', 'LineWidth', 1, 'DisplayName', 'W_OI_inerz');
end
if setCameraControl
    ##### camera control #####
    % Muovi la camera se necessario
    % camorbit(0,0,1); % ruota la camera di 1 grado attorno all'asse z
    % Posizione della camera
    cameraPosition = VCam(j,:); %[x_eml2+10000,-10000,10000];%[halo_position(j,1)
        *1.3, halo_position(j,2)*1.1, halo_position(j,3)*1.1]; % Posizione della
        camera
    cameraTarget = -VCamDir(j,:)+[40000,0,0]; % -i_inerz; %k_orb(j,:);
        % Target della camera
    cameraUpVector = k_inerz; %-j_orb(j,:); % Orientamento della
        camera verso l'alto
    %viewAngle = 120; % Campo visivo in gradi
    % Chiamata alla funzione
    cameraControl(cameraPosition, cameraTarget, cameraUpVector);

    % Aggiorna il grafico
    drawnow;
    % Cattura il frame corrente
    frame = getframe(figureSpaceObjects); % Cattura il frame corrente
    writeVideo(Video, frame); % Scrivi il frame nel video

    % mantieni ultima posizione camera prima del nuovo loop
    if j == arraySize-1
        VCamDir = repmat(VCam(j,:), size(VCamDir, 1), 1);

```

C. CALCULATIONS

```
    VCam = repmat(VCam(j,:), size(VCam, 1), 1);
end
end

pause(0.01);
% delete objects
delete(squares);
if plotVectorsToMoonEarth    delete(vectors_to_moon);    delete(vectors_to_earth);
    end
delete(velocity_vectors);
if plotOrbitalFrame    delete(i0rbVersors);    delete(j0rbVersors);    delete(
    k0rbVersors);    end
if plotW_OI        delete(W_OI_orb_handler);        delete(W_OI_inerz_handler);
    end
end

close(Video); % Chiudi il file video

% Ask the user if they want to save the video
prompt = 'Do you want to save the video? (Y/N): ';
saveVideo = input(prompt, 's');
if strcmpi(saveVideo, 'N')
    delete(videoFilename);
    disp('Video Eliminated');
else
    disp('Video Saved')
end

%
#####
##### FUNCTIONS #####
#####

% function to plot vectors for the animation
function plotVectors = plotVector(vector, position, varargin)
global indexs;
vector = vector ./ vecnorm(vector, 2, 2) * 10000;
plotVectors = quiver3(position(indexs, 1),position(indexs, 2),position(indexs, 3),
    vector(indexs, 1), vector(indexs, 2), vector(indexs, 3), varargin{:}, '
    AutoScale', 'off');
end
```



```
function V_rel = transformVectorToRelativeFrame(i_iner, j_iner, k_iner, u, v, w, V)
    % Creazione della matrice di trasformazione dal sistema relativo all'inerziale
    % u, v, w sono i vettori del sistema di riferimento relativo scritti nel sistema
    % inerziale
    R = [u; v; w]'; % Trasposta per allineare i vettori come colonne

    % Calcolo della matrice inversa
    R_inv = inv(R); % Inverte la matrice di trasformazione

    % Trasformazione del vettore V nel sistema di riferimento relativo
    V_rel = R_inv * V';

    % Ritorna V_rel come vettore colonna
    V_rel = V_rel';
end

function cameraControl(cameraPosition, cameraTarget, cameraUpVector)
    % Imposta la posizione della camera
    campos(cameraPosition);

    % Imposta il punto su cui la camera deve essere puntata
    camtarget(cameraTarget);

    % Imposta il vettore "up" della camera per definire l'orientamento
    camup(cameraUpVector);

    % Imposta il campo visivo della camera
    %camva(viewAngle);
end

function totalBlackFigure(fig)
    % Ottiene l'handle degli assi correnti nella figura
    fig.Color = 'k';
    ax = get(fig, 'CurrentAxes');

    ax.XGrid = 'on'; % Attiva la griglia sull'asse X
    ax.YGrid = 'on'; % Attiva la griglia sull'asse Y
    ax.ZGrid = 'on'; % Attiva la griglia sull'asse Z
    ax.XMinorGrid = 'on'; % Attiva la griglia minore sull'asse X
    ax.YMinorGrid = 'on'; % Attiva la griglia minore sull'asse Y
    ax.ZMinorGrid = 'on'; % Attiva la griglia minore sull'asse Z
```



SUPREMAE
DIGNITATIS
1343

UNIVERSITÀ
DI PISA

C. CALCULATIONS

```
ax.GridLineStyle = '--'; % Imposta lo stile della griglia principale
ax.MinorGridLineStyle = ':'; % Imposta lo stile della sottogriglia

ax.Color = 'k'; % Imposta il colore di sfondo
ax.XColor = 'w'; % Imposta il colore dell'asse X
ax.YColor = 'w'; % Imposta il colore dell'asse Y
ax.ZColor = 'w'; % Imposta il colore dell'asse Z
ax.Title.Color = 'w';
ax.XLabel.Color = 'w';
ax.YLabel.Color = 'w';
ax.ZLabel.Color = 'w';
ax.GridColor = 'w'; % Imposta il colore della griglia
ax.MinorGridColor = 'y'; % Imposta il colore della griglia minore
ax.GridAlpha = 0.5; % Imposta la trasparenza della griglia principale (0
    trasparente, 1 opaco)
ax.MinorGridAlpha = 0.5; % Imposta la trasparenza della griglia minore

% Cambia il colore di tutto il testo aggiuntivo nella figura
allText = findall(fig, 'Type', 'text');
set(allText, 'Color', 'w');

end
```



C.9. Power Generation battery load shift evaluation

```
%Power weight evaluation script For SpaceSystem Project
%FRANCESCO MARRADI - Support to Human Activities from EML2
clc
close all
clear

%l'obiettivo di questo script è capire se è possibile "spostare" una parte
%della richiesta energetica dei propulsori elettrici dai pannelli solari
%verso le batterie. Così facendo possiamo usare dei pannelli più piccoli
%ipoteticamente andando a risparmiare in massa.

%Inizialmente avevamo previsto di soddisfare l'intero apporto energetico
%tramite i pannelli solari

%-----GENERAL PARAMETERS-----
%General Parameters
SolarPanelValue = 300/6; %Watt generated by 1kg of solar panels[W/kg]
HeaterConsumption = 20; %Heater consumption for each battery pack[W]

%Propulsion Parameters
EngineConsumption = 200; %[W]
N_Engines = 2; %Numero di motori richiesti alla manovra di
    stationkeeping
BurnTime = 864; %[s]
DoD = 0.7; %Battery dept of discharge
CellCapacity = 16; %[W/h]
EOLCellCapacityfactor = 0.72; %Percentage of the total cell capacity after 4000
    cycles
CellWeight = 0.182; %[kg]

%-----CALCULATIONS-----
%Calcolo la capacità della batteria richiesta nel caso dovesse sopportare
%"da sola" la propulsione
BatteryCapacityTotal = (EngineConsumption*N_Engines*(BurnTime/3600));
SolarPanelWeight = (EngineConsumption*N_Engines)/SolarPanelValue;
```



C. CALCULATIONS

%Definisco un ciclo il cui sposta progressivamente il carico dai pannelli solari verso la batteria, infine calcola il peso dell'intero sistema.

```
for i = 1:10
    SolarPanelLoad(i) = (1-i/10)*(EngineConsumption*N_Engines);
    BatteryLoad(i) = (i/10)*EngineConsumption*N_Engines;
    TotalBatteryWeight(i) = ((BatteryLoad(i)*(BurnTime/3600))/(CellCapacity*
        EOLCellCapacityfactor))*CellWeight;
    TotalSolarPanelWeight(i) = (SolarPanelLoad(i)/SolarPanelValue);

end

TotalPowerGenerationWeight = 2*TotalBatteryWeight/DoD + TotalSolarPanelWeight;
p = polyfit([10 20 30 40 50 60 70 80 90 100],TotalPowerGenerationWeight,1);
P = polyval(p,linspace(0,100,1000));
figure(1)
hold on
ylabel('Power Generation Mass [kg]')
xlabel('Power load to the batteries')
plot(linspace(0,100,1000),P)
hold off
```

C.10. Temperature cold face

```
% Made by Michela Maestripieri, Mario Nocilli, Mauro Furfari, Lucrezia Podestà
close all
clear
clc

T_sun = 5772; % K
sigma = 5.67*10^-8; % W/(m^2K^4)

% Power emitted by the sun
Radius_sun = 6.9634*10^8; % m
P_sun = 4*pi*Radius_sun^2*sigma*T_sun^4;

% Distanze e flusso solare
distance_sun_earth = 1.4959787*10^11; % m
distance_earth_moon = 3.844*10^8; % m
distance_moon_eml2 = 6.1*10^7; % m
d_min = distance_sun_earth - distance_earth_moon - distance_moon_eml2;
d_max = distance_sun_earth + distance_earth_moon + distance_moon_eml2;
distance = linspace(d_min,d_max,1000);
I_sun = P_sun./(4*pi*distance.^2);

% Dati spacecraft
l_sc = 1; % m (faccia di interesse)
P_sc = 500; % W
T_space = 3; % K
T_sc = 277; % K - Value taken from previous calculations - (Aluminum Paint = 304K) (Al Black Kapton = 293K) (Aluminized Kapton = 277K)

% Calcoli
alpha_sc = 0.35; %(AlPa=0.3)(AlBlKa=0.9)(AlKa=0.35)
rapportoAlphaEpsilon = 0.72; %(AlPa=1)(AlBlKa=1.1)(AlKa=0.72)

% Frazione della potenza termica generata internamente dallo spacecraft che si irradia verso la faccia di interesse
Q_f =(P_sc/6).*linspace(0.50,0.70,11)';

Q_s = alpha_sc*l*I_sun;
T_f = (((Q_f./ (sigma*(alpha_sc./rapportoAlphaEpsilon)) + T_space^4 + T_sc^4))./2)
.^^(1/4);
```



C. CALCULATIONS

```
% Plot Temperature
figure
plot(Q_f,T_f-273)
xlabel('Thermal power received by the surface of interest [W]')
ylabel('Temperature [ C ]')
title ('With Aluminized Kapton Film')
```



C.11. Temperature hot face

```
% Made by Michela Maestripieri, Mario Nocilli, Mauro Furfari, Lucrezia Podestà
close all
clear
clc

T_sun = 5772; % K
sigma = 5.67*10^-8; % W/(m^2K^4)

% Power emitted by the sun
Radius_sun = 6.9634*10^8; % m
P_sun = 4*pi*Radius_sun^2*sigma*T_sun^4;

% Distanze e flusso solare
distance_sun_earth = 1.4959787*10^11; % m
distance_earth_moon = 3.844*10^8; % m
distance_moon_eml2 = 6.1*10^7; % m
d_min = distance_sun_earth - distance_earth_moon - distance_moon_eml2;
d_max = distance_sun_earth + distance_earth_moon + distance_moon_eml2;
distance = linspace(d_min,d_max,1000);
I_sun = P_sun./(4*pi*distance.^2);

% Dati spacecraft
l_sc = 1; % m (faccia di interesse)
P_sc = 500; % W
T_space = 3; % K
T_sc = 277; % K - Value taken from previous calculations - (Aluminum Paint = 304K) (Al Black Kapton = 293K) (Aluminized Kapton = 277K)

% Frazione della potenza termica generata internamente dallo spacecraft che si irradia verso la faccia di interesse
Q_f =(P_sc/6).*linspace(0.50,0.70,11)';

% Calculation
alpha_sc = 0.35; %(AlPa=0.3)(AlBlKa=0.9)(AlKa=0.35)
rapportoAlphaEpsilon = 0.72; %(AlPa=1)(AlBlKa=1.1)(AlKa=0.72)
Q_s = alpha_sc*1*I_sun;
T_f = (((Q_s+Q_f) ./ (sigma*(alpha_sc./rapportoAlphaEpsilon)) + T_space^4 + T_sc^4))
./2).^^(1/4);
```



C. CALCULATIONS

```
% Plot Temperature
figure
plot(distance/10^3,T_f-273)
xlabel('Distance from the sun [km]')
ylabel('Temperature [ C ]')
title ('With Aluminized Kapton Film')
```



C.12. Thermal equilibrium temperature

% Made by Michela Maestripieri, Mario Nocilli, Mauro Furfari, Lucrezia Podestà

```
close all  
clear  
clc
```

```
T_sun = 5772; % K  
sigma = 5.67*10^-8; % W/(m^2K^4)
```

```
% Power emitted by the sun  
Radius_sun = 6.9634*10^8; % m  
P_sun = 4*pi*Radius_sun^2*sigma*T_sun^4;
```

```
% Distanze e flusso solare  
distance_sun_earth = 1.4959787*10^11; % m  
distance_earth_moon = 3.844*10^8; % m  
distance_moon_eml2 = 6.1*10^7; % m  
d_min = distance_sun_earth - distance_earth_moon - distance_moon_eml2;  
d_max = distance_sun_earth + distance_earth_moon + distance_moon_eml2;  
distance = linspace(d_min,d_max,1000);  
I_sun = P_sun./(4*pi*distance.^2);
```

```
% Dati spacecraft  
lato_sc = 1; % m  
A_abs = (2^0.5)*lato_sc; % m^2  
P_sc = 500; % W  
T_space = 3; % K  
A_sp = 6; % m^2
```

```
% Potenza termica generata internamente dallo spacecraft  
Q_i = P_sc.*linspace(0.50,0.70,11)';
```

```
% For Aluminum Paint  
alpha_sc_AlPa = 0.3;  
rapportoAlphaEpsilon_AlPa = 1;  
Q_s_AlPa = alpha_sc_AlPa*A_abs*I_sun;  
T_sc_AlPa = (((Q_s_AlPa+Q_i) ./ (sigma*(alpha_sc_AlPa/rapportoAlphaEpsilon_AlPa)*A_sp)  
) + T_space^4 ).^(1/4);
```



C. CALCULATIONS

```
% For Aluminized Black Kapton
alpha_sc_bp = 0.9;
rapportoAlphaEpsilon_bp = 1.1;
Q_s_bp = alpha_sc_bp*A_abs*I_sun;
T_sc_bp = (((Q_s_bp+Q_i) ./ (sigma*(alpha_sc_bp/rapportoAlphaEpsilon_bp)*A_sp)) +
T_space^4 ).^(1/4);

% For Aluminized Kapton
alpha_sc_AlKa = 0.35;
rapportoAlphaEpsilon_AlKa = 0.72;
Q_s_AlKa = alpha_sc_AlKa*A_abs*I_sun;
T_sc_AlKa = (((Q_s_AlKa+Q_i) ./ (sigma*(alpha_sc_AlKa/rapportoAlphaEpsilon_AlKa)*A_sp))
+ T_space^4 ).^(1/4);

% Plot flux
figure
plot(distance/(10^3),I_sun)
xlabel('Distance from the sun [km]')
ylabel('Flux [W/m^2]')
title('Flux vs distance from the sun')

% Plot temperature
figure
t = tiledlayout(2,2);

nexttile
plot(distance/10^3,T_sc_AlPa-273)
xlabel('Distance from the sun [km]')
ylabel('Temperature [ C ]')
title('With Aluminum Paint')
title(legend, 'Thermal power generated internally to the S/C')
legend('50% of utilized power', '52% of utilized power', '54% of utilized power', '56%
of utilized power', '58% of utilized power', '60% of utilized power', '62% of
utilized power', '64% of utilized power', '66% of utilized power', '68% of
utilized power', '70% of utilized power')

nexttile
plot(distance/10^3,T_sc_bp-273)
xlabel('Distance from the sun [km]')
ylabel('Temperature [ C ]')
title('With Aluminized Black Kapton film')
```



```
nexttile
plot(distance/10^3,T_sc_AlKa-273)
xlabel('Distance from the sun [km]')
ylabel('Temperature [ C ]')
title('With Aluminized Kapton film')
```

Bibliography

- [1] D. R. Williams, "Lunar prospector science results," *Science - NASA*, 2005.
- [2] Y. L. Y. W. L. L. X. Z. H. L. Hui Zhang, Jinbin Cao and L. Xie, "Key questions of solar wind-moon interaction," *SPJ*, 2023.
- [3] M. T. Zuber, D. E. Smith, M. M. Watkins, S. W. Asmar, A. S. Konopliv, F. G. Lemoine, H. J. Melosh, G. A. Neumann, R. J. Phillips, S. C. Solomon, M. A. Wieczorek, J. G. Williams, S. J. Goossens, G. Kruizinga, E. Mazarico, R. S. Park, and D.-N. Yuan, "Gravity field of the moon from the gravity recovery and interior laboratory (grail) mission," *Science*, vol. 339, no. 6120, pp. 668–671, 2013.
- [4] J. P. L. NASA, "Photojournal," <https://photojournal.jpl.nasa.gov/target/moon?sort=DESC>, 2023.
- [5] M. H. Jamey R. Szalay, "Detecting meteoroid streams with an in-situ dust detector above an airless body," *ELSEVIER*, 2016.
- [6] F. Z. Y. Y. Q. H. J. Y. Y. Q. Y. Z. J. Zhao, L. Qiao and L. Xiao, "Volcanism and deep structures of the moon," *SPJ*, 2023.
- [7] J. D. M. D. S. Colburn, R. G. Currie and C. P. Sonett, "Diamagnetic solar-wind cavity discovered behind moon," *Science*, 2023.
- [8] M. F. J. G. J. K. J. K. M. L. I. M. D. P. K. R. M. R. A. S. D. S. H. S. P. S. S. A. S. M. Z. Gordon Chin, Scott Brylow, "Lunar reconnaissance orbiter overview: The instrument suite and mission," *Springer Science*, 2007.
- [9] B. M. F. Grant H. Heiken, David T. Vaniman, *Lunar Sourcebook*. Cambridge University Press, 1991.
- [10] L. Q. T. M. B. L. Z. F. C. X. Yingzhuo Jia, Zhanlan Zhang and Y. Zou, "Research of lunar water-ice and exploration for china's future lunar water-ice exploration," *SPJ*, 2023.
- [11]
- [12] NASA, -.
- [13] K. C. H. Emily M. Zimovan and D. C. Davis, "Near rectilinear halo orbits and their application in cis-lunar space," *ResearchGate*, 2017.
- [14] R. Funase, S. Ikari, K. Miyoshi, Y. Kawabata, S. Nakajima, S. Nomura, N. Funabiki, A. Ishikawa, K. Kakihara, S. Matsushita, R. Takahashi, K. Yanagida, D. Mori, Y. Murata, T. Shibukawa, R. Suzumoto, M. Fujiwara, K. Tomita, H. Aohama, and T. Hashimoto, "Mission to earth-moon lagrange point by a 6u cubesat: Equuleus," *IEEE Aerospace and Electronic Systems Magazine*, vol. 35, pp. 30–44, 03 2020.

- [15] S. B. A. R. V. F. F. T. Paolo Panicucci, Felice Piccolo, "Current status of the lumio autonomous optical navigation experiment," *12th International Conference on Guidance, Navigation & Control Systems (GNC) and 9th International Conference on Astrodynamics Tools and Techniques (ICATT)*, 2023.
- [16] ASI, 2023.
- [17] P. Ye, Z. Sun, H. Zhang, and F. Li, "An overview of the mission and technical characteristics of change'4 lunar probe," *Science China Technological Sciences*, 2017.
- [18] J. Haruyama, S. Sawai, T. Mizuno, T. Yoshimitsu, S. Fukuda, and I. Nakatani, "Exploration of lunar holes, possible skylights of underlying lava tubes, by smart lander for investigating moon (slim)," *Transactions of The Japan Society for Aeronautical and Space Sciences, Aerospace Technology Japan*, vol. 10, no. ists28, pp. Pk_7-Pk_10, 2012.
- [19] L. D. C. Y. e. a. Zeng, X., "Landing site of the chang'e-6 lunar farside sample return mission from the apollo basin," *Nature Astronomy*, 2023.
- [20] J. Trimble, "Viper lunar rover agile mission systems," in *17th International Conference on Space Operations (SpaceOps 2023)*, 2023.
- [21] M. Banks, "India launches chandrayaan-3 mission to the moon," *Physics World*, vol. 36, no. 8, p. 12ii, 2023.
- [22] T. D. Haws, J. S. Zimmerman, and M. E. Fuller, "Sls, the gateway, and a lunar outpost in the early 2030s," in *2019 IEEE aerospace conference*, pp. 1–15, IEEE, 2019.
- [23] I. O. F. Kristian Zarb Adami, "Low-frequency technology for a lunar interferometer," *The Royal Society Publishing*, 2021.
- [24] G. Gupta, M. Arya, A. Goel, S. Bandyopadhyay, P. Goldsmith, P. McGarey, J. Lazio, and N. Chahat, "Detector development for the lunar crater radio telescope," in *2022 IEEE Wireless Antenna and Microwave Symposium (WAMS)*, pp. 1–5, IEEE, 2022.
- [25] G. Gupta, N. Chahat, A. Goel, M. Arya, D. Pisanti, P. Goldsmith, J. Lazio, and S. Bandyopadhyay, "Reflector antenna for lunar crater radio telescope," in *2023 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (USNC-URSI)*, pp. 1725–1726, IEEE, 2023.
- [26] P. Amaro-Seoane, L. Bischof, J. J. Carter, M.-S. Hartig, and D. Wilken, "Lion: laser interferometer on the moon," *Classical and Quantum Gravity*, vol. 38, no. 12, p. 125008, 2021.
- [27] S. D. Bale, N. Bassett, J. O. Burns, J. D. Jones, K. Goetz, C. Hellum-Bye, S. Hermann, J. Hibbard, M. Maksimovic, R. McLean, et al., "Lusee'night': The lunar surface electromagnetics experiment," *arXiv preprint arXiv:2301.10345*, 2023.



C. BIBLIOGRAPHY

- [28] E. Tamura, J. Fried, P. O'Connor, and S. Herrmann, "Design and characterization of the engineering model of the spectrometer onboard lusee-night," in *2023 XXXVth General Assembly and Scientific Symposium of the International Union of Radio Science (URSI GASS)*, pp. 1–4, IEEE, 2023.
- [29] I. Handayani, A. Juliana, and G. Gumelar, "Artemis base camp and exploitation of outer space: Problems and the needs of legal framework," *Padjadjaran Journal of International Law*, vol. 7, no. 1, 2023.
- [30] L. Zhang, "Development and prospect of chinese lunar relay communication satellite," *Space: Science & Technology*, 2021.
- [31] Y. L. Yongliao Zou and Y. Jia, "Overview of china's upcoming chang'e series and the scientific objectives and payloads for chang'e-7 mission," in *51st Lunar and Planetary Science Conference*, 2020.
- [32] L. Demers, R. Weiss-Lambrou, and B. Ska, "Quebec user evaluation of satisfaction with assistive technology versione 2.0," *The Institute for Matching Persons and Technology: Webster, NY*, 2000.
- [33] R. Reinhart, "Lunar communications and navigation: Integrating government, commercial and international partners solutions," in *Global Satellite Servicing Forum*, 2023.
- [34] S. Marcuccio, "Space system course - msc in space engineering, università di pisa." Telecommunications.
- [35] SSC, 2024.
- [36] Atlas, 2024.
- [37] "Goonhilly brochure."
- [38] R. G. A. S. G. S. E. U. G. M. S. M. S. W. A. T. T. P. J. D. V. GIUSEPPE VALENTE, MARIA NOEMI IACOLINA and S. VIVIANO, "The sardinia space communication asset: Performance of the sardinia deep space antenna x-band downlink capability," *IEEAccess*, 2022.
- [39] Erzia, 2024.
- [40] "Datasheet: Iq spacecom, s-band patch antenna."
- [41] "Datasheet: Spaceteq, x-horn 23."
- [42] "Datasheet: Iq spacecom, x-band patch antenna."
- [43] "Datasheet: Printech, k-band patch antenna."
- [44] "Datasheet: Printech, l-band circulary polarized patch antenna."
- [45] "Datasheet: Planewave, pw2020-000."

- [46] "Datasheet: Planewave, pw2222-101."
- [47] "Datasheet: Erz-hpa-0500-2300-20."
- [48] "Datasheet: Erz-hpa-1700-4300-22-e."
- [49] "Datasheet: Erz-hpa-0100-1800-21."
- [50] "Datasheet: Erz-hpa-0200-0400-24."
- [51] "Datasheet: Erz-Ina-0100-4000-45-5."
- [52] "Datasheet: Erz-Ina-0600-1200-28-1.5."
- [53] "Datasheet: Planewave, pw2020-110."
- [54] "Datasheet: Planewave, pw2222-110."
- [55] "Datasheet: Erz-hpa-0200-0400-30."
- [56] L. Zhang, "Development and prospect of chinese lunar relay communication satellite," *Space: Science & Technology*, 2021.
- [57] Z. Lou and Y. Wang, "Robust station-keeping control of sun-earth/moon libration point orbits using electric propulsion," *Journal of Aerospace Engineering*, vol. 32, no. 2, p. 04018142, 2019.
- [58] L. Guerreschi, *LIBRATION POINT ORBIT STATION-KEEPING: CONTROL STRATEGIES FOR IMPROVED SCIENTIFIC MISSION TIME*. PhD thesis, POLITECNICO DI MILANO, 2020.
- [59] R. P. Welle, "Propellant storage considerations for electric propulsion," in *22nd International Electric Propulsion Conference*, pp. 91–107, 1991.
- [60] "Three-body periodic orbits."
- [61] Astrofein, "Datasheet: Astrofein, reaction wheel family," 2023.
- [62] S. GmbH, 2024.
- [63] Saft, 2024.
- [64] Saft, 2024.
- [65] Dunmore, 2024.
- [66] R. Urban, "How much does it cost to launch a rocket?," NA, 2023.
- [67] NASA, 2020.
- [68] TE-Connectivity, 2024.
- [69] AstroSpace, 2022.



C. BIBLIOGRAPHY

- [70] ESA, 2024.
- [71] L. Space, 2020.
- [72] D. P. Gordon, *TRANSFERS TO EARTH-MOON L2 HALO ORBITS USING LUNAR PROXIMITY AND INVARIANT MANIFOLDS*. PhD thesis, Purdue University, 2008.
- [73]
- [74]
- [75] P. A. ILES, “Photovoltaic conversion: Space applications,” *Arcadia California, United States*, 2004.
- [76] C. LARSON, “European space agency mulls extra ariane 6 cash,” *NA*, 2023.