POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Station keeping algorithm implementation for the simulation of a coupled GNC and ADCS model for the lunar CubeSat LUMIO

Tesi di Laurea Magistrale in
Space Engineering - Ingegneria Spaziale

Author: **Andrea Babato**

Student ID: 944271
Advisor: Prof. Francesco Topputo
Co-advisors: Gianmario Merisio, Antonio Rizza
Academic Year: 2020-21

"Per aspera ad astra"

Virgilio (Eneide IX, 641)

Abstract

The LUnar Meteoroid Impacts Observer (LUMIO) is a CubeSat mission to a halo orbit at Earth–Moon L2 that shall observe, quantify, and characterise meteoroid impacts on the lunar far-side, by detecting their flashes. In this way, LUMIO is expected to significantly contribute to Lunar Situational Awareness and to the current knowledge on the evolution of meteoroids in the cislunar space. In this work the methodologies to perform an optimal station keeping in the selected halo orbit are investigated, making use of MATLAB® and Simulink® programming platforms. Indeed, when dealing with a chaotic n-body dynamics, each minimum source of disturbance and error determines a potential divergence of the orbit at a future epoch. Being not feasible to account and model all the perturbations affecting the trajectory, the only solution is the development of a station keeping algorithm which provides the required corrections. To this aim, the S/K conceptual operations are consolidated and the attitude correction manoeuvres to be applied during the mission are simulated. The main goals of the thesis, related to the modelling and simulation of a coupled ADCS-GNC system for the station keeping of LUMIO are addressed, with the aim of fulfilling the mission requirements. In detail, a station keeping algorithm is first implemented for the computation of the manoeuvres characteristics to execute all over the mission, considering the LUMIO operations. Then, to properly simulate the manoeuvres execution, a Simulink model is developed, which includes the attitude corrections pre and post-manoeuvre and the actuators dynamics. The final work features a simulation tool able to carry out autonomously the ADCS-GNC operations of LUMIO for the whole mission lifetime, on the basis of the $\Delta\mathbf{v_{SK}}$ computed. In particular, the tool has been developed exploiting high-fidelity models for the dynamics systems included, and features a clear and versatile interface. The results are critically discussed and future developments are reported, having in mind the complex scenario the mission has to face and the multitude of problematic involved.

Keywords: LUMIO, halo, GNC, ADCS, station keeping, n-body problem, CubeSat, meteoroids.

Sommario

Il LUnar Meteoroid Impacts Observer (LUMIO) è una missione CubeSat in un'orbita halo Terra-Luna L2 con l'obbiettivo di osservare, quantificare e caratterizzare gli impatti di meteoroidi sul lato lunare nascosto, rilevando i loro bagliori di luce. In questo modo, LUMIO dovrebbe contribuire in modo significativo alla consapevolezza della situazione lunare e alle attuali conoscenze sull'evoluzione dei meteoroidi nello spazio cislunare. In questo lavoro vengono studiate le metodologie per eseguire uno station keeping ottimale nell'orbita halo selezionata, facendo uso dei programmi di calcolo MATLAB® e Simulink®. Per l'appunto, nel contesto di dinamiche caotiche con n corpi, orgni minima perturbazione orbitale o altro tipo di errore determina una probabile divergenza della traiettoria in un istante futuro. Dunque, non essendo possibile prevedere e modellare tutte le perturbazioni presenti, la soluzione consiste nel sviluppare un algoritmo per il mantenimento dell'orbita nominale, attraverso le dovute manovre di correzione. A questo scopo vengono consolidate le operazioni concettuali di S/K e simulate le manovre di correzione dell'assetto da applicare durante la missione. Gli obiettivi principali della tesi, relativi alla modellazione e simulazione dell'accoppiamento tra i sistemi di ADCS e GNC per il controllo orbitale di LUMIO, vengono sviluppati, finalizzado il lavoro al soddisfacimento dei requisiti di missione. Nel dettaglio, è stato prima implementato un algoritmo di station keeping per il calcolo delle manovre da eseguire nel corso della missione, sulla base delle operazioni concettuali di LUMIO. Successivamente è stato sviluppato un modello Simulink in grado di simulare le fasi di pre e post manovra e la dinamica degli attuatori. Il lavoro di tesi arriva ad ottenere un simulatore capace di eseguire autonomamente le operazioni di ADCS-GNC di LUMIO, durante l'intera durata della missione, sulla base dei $\Delta\mathbf{v_{SK}}$ calcolati. In particolare, il simulatore presenta modelli ad alta fedeltà per i sistemi dinamici inclusi e una interfaccia chiara e versatile. I risultati sono discussi in modo critico e vengono riportati gli sviluppi futuri, tenendo presente il complesso scenario che la missione deve affrontare e la moltitudine di problematiche coinvolte.

Parole chiave: LUMIO, halo, GNC, ADCS, station keeping, problema di n corpi, CubeSat, meteoriti.

Acknowledgements

I still do not realise I have arrived at the end of this long, challenging journey. It is passed five years since I have opened my first engineering book, and approached the first exam. I have to thank all the University mates who helped me to reduce the tension during these years, enriching them with beautiful moments. After the Bachelor's degree, I decided to move in Milan for the last two years of my Master, at Politecnico. I still remember the day I left home, even if I was not sure about the result of my enrolment, but with the usual spirit of pushing myself a little beyond my limits and securities. These last two years have been plenty of satisfaction and fatigue, probably more intense than the Bachelor's degree years. That being said, Politecnico di Milano allowed me to improve as a person and as an engineer, and I will bring this experience into my heart forever.

Moving to a less dramatic and philosophic part, I want to express my sincere gratitude to my advisor Francesco Topputo, and my co-advisors Gianmario Merisio and Antonio Rizza. You have revealed to be available and professional people, which made my journey during this thesis less harsh and more exciting. With patience and capability, you have guided me when I needed it, with no hesitation.

My family, who has always been to my side, supporting my choices and encouraging me when I was more fragile, has always been, and it is, my reference point and guide. To my grandparents, I extend a big thank you for always being lovingly present in my life.

Concluding, thanks to all the people who have always been close to me and who don't need to read their name between these lines, this milestone is even a little bit yours. The moments passed with all of you are indelibly fixed in my mind.

Before I leave, I would like to mention my grandfather Flavio. I know I promised you that you would have participated in my degree; this has not been possible, but I am sure you would have been proud of me, and that day, in some way, you will be there listening to me.

Andrea, Padova, Italy – November 18th, 2021

List of Figures

List of Tables

Nomenclature

Acronyms

ADCS Attitude Determination & Control System

AOCS Attitude & Orbit Control System

CoM Centre of Mass

ConOps Conceptual Operations

CR3BP Circular Restricted 3 Body Problem

CUBORG CUBesat Orbit and GNC tool

DCM Direction Cosine Matrix

DOF Degree Of Freedom

ECI Earth-Centred Inertial

EPS Electric Power Subsystem

ET Ephemeris Time

GNC Guidance Navigation & Control

IMU Inertial Measurement Unit

LQR Linear Quadratic Regulator

LUMIO LUnar Meteoroid Impacts Observer

MCI Moon-Centred Inertial

MIB Minimum Impulse Bit

NEO Near-Earth Objects

NRHO Near-Rectilinear halo Orbit

OD Orbit Determination

OI Orbit Insertion

PS Propulsion Subsystem

RCS Reaction Control System

RPF Roto-Pulsating Frame

rpm Revolutions Per Minute

RW Reaction Wheel

S/K Station Keeping

SADA Solar Array Drive Assembly

SRP Solar Radiation Pressure

STM State Transition Matrix

TCS Thermal Control Subsystem

TMTCS TeleMetry and TeleCommand Subsystem

TPM Target Point Method

Physical Constants

$AU = 1.4959 \times 10^8$ Astronomical unit $[km]$

$G = 6.6741 \times 10^{-11}$ Gravitational constant $[m^3/kg/s^2]$

$g_0 = 9.8066$ Gravitational acceleration $[m/s^2]$

$P_{SRP1AU} = 4.5732 \times 10^{-6}$ Solar pressure at 1 AU $[Pa]$

Physical Quantities

$\boldsymbol{\omega}$ Angular velocity vector $[rad/s]$

$\delta\mathbf{r_c}$ Position deviation at t_c $[km]$

$\delta\mathbf{v_c}$ Velocity deviation at t_c $[km/s]$

$\Delta\mathbf{v_{SK}}$ S/K impulsive manoeuvre $[km/s]$

Δt_c Cut-off time $[days]$

Δt_f Firing duration $[s]$

ΔT_i GNC operations timings $[s]$

Δt_i Target points epochs $[days]$

ϵ_i Random Gaussian error $[km - km/s]$

μ_i Gravitational parameter $[km^3/s^2]$

Φ State Transition Matrix $[-]$

ρ_d Diffusive reflection coefficient $[-]$

ρ_s Specular reflection coefficient $[-]$

σ_i Standard deviation $[km - km/s]$

$\mathbf{F_g}$ Gravitational force $[N]$

$\mathbf{F_{SRP}}$ SRP force $[N]$

$\mathbf{F}$ External forces $[N]$

$\mathbf{h}$ Angular momentum $[kg\ m^2/s]$

$\mathbf{M_c}$ Ideal control torque $[Nm]$

$\mathbf{M_r}$ Real control torque $[Nm]$

$\mathbf{q}$ Quaternion vector $[-]$

$\mathbf{r_i}$ Body distance wrt frame origin $[km]$

$\mathbf{T_{SRP}}$ SRP torque $[Nm]$

$\mathbf{T}$ Thrust $[N]$

$\mathbf{x}$ Trajectory state $[km - km/s]$

A RWs configuration matrix $[-]$

$A_{\Delta v}$ Manoeuvre attitude matrix $[-]$

$A_{d/p}$ Surface area $[m^2]$

A_e Error attitude matrix $[-]$

A_{nom} Nominal attitude matrix $[-]$

c_e Exhaust velocity $[m/s]$

c_r Reflectivity coefficient $[-]$

I RWs inertia $[kgm^2]$

I_{sp} Specific impulse $[s]$

J Jacobian $[-]$

$J_{d/p}$ Inertia tensor $[kgm^2]$

Kd_i Derivative gain $[-]$

Kp_i Proportional gain $[-]$

m_{sc} LUMIO mass $[kg]$

n Eclipse binary index $[-]$

$n_{d/p}$ Surfaces normals $[-]$

Q Weighting matrix $[-]$

R_i Weighting matrix $[-]$

$r_{d/p}$ Surface centre wrt CoM $[m]$

t_i GNC simulation times $[s]$

T_{max} Maximum thruster force $[N]$

T_{tot} Maximum 2-thrusters force $[N]$

Contents

1 | Introduction

"The journey of a thousand miles
begins with one step."

- Lao Tzu

1.1. Context

The Earth–Moon system is constantly bombarded by meteoroids of different sizes, in a great number, originated by fragments of asteroids and comets, that date back to planetary formation times. Observations of meteor showers on Earth have been studied for at least 50 years, in order to construct Solar System meteoroid models. These models can be useful in predicting the small-meteoroid flux that deteriorates space equipment or when the next large meteoroid will impact Earth itself. As meteoroids originate from asteroids and comets, meteoroid models can also be used to understand the spatial distribution of those objects near the Earth–Moon system [1].

By observing the lunar surface impacts, whose flux is similar to that of the Earth, we could obtain detailed information regarding their magnitudes, velocities, temporal and spatial distributions. These information can be used to increase confidence of meteoroid models, to validate the existing lunar impact models, to contribute to lunar seismology studies and interior modelling, and to initiate a Lunar Situational Awareness programme for future exploration missions. Earth-based optical observations of the light flashes produced by lunar meteoroid impacts have revealed useful in the validation and improvement of meteoroid models. Monitoring the Moon for meteoroid impact flashes allows for the observation of larger areas than those covered by traditional surveys of Earth's upper atmosphere. Thus, theoretically, more meteoroid impacts can be detected in shorter periods of time. Moreover, Earth-based lunar observations are restricted by weather, geometric, and illumination conditions. As such, a lunar CubeSat can improve the detection rate of lunar meteoroid impact flashes, as it would allow for longer monitoring periods. Moreover, it being closer to the Moon surface, a lunar CubeSat could also allow for the detection of

meteoroids smaller than millimetres. A lunar orbiter improves the quality and quantity of lunar meteoroid impact detection [1].

LUMIO (LUnar Meteoroid Impacts Observer) is a CubeSat mission to a halo orbit at Earth–Moon L2 that shall observe, quantify, and characterise meteoroid impacts on the lunar far-side by detecting their flashes, complementing Earth-based observations on the lunar nearside, to provide global information on the lunar Meteoroid Environment and contribute to Lunar Situational Awareness [2].

LUMIO was one of the proposals submitted to the SysNova LUnar CubeSats for Exploration call by the European Space Agency. The mission was awarded ex-aequo winner of the challenge, and its scientific relevance and technical feasibility were confirmed by an independent study conducted by the ESA Concurrent Design Facility [2]. The LUMIO Phase A study has been successfully concluded at the beginning of 2021 and Phase B is expected to start in early 2022.

1.2. Motivation

Deepen the knowledge of the meteoroids environment surrounding the Earth–Moon system is becoming fundamental for the security of the life on the Earth, other than delivering significant scientific advances. The study of such phenomena reveals to be particularly fruitful if performed from halo orbits, particular solutions of the n-body problem. These orbits are characterised by advantageous conditions in terms of viewing geometry, thermal and radiation environment, and $\Delta\mathbf{v}$ expenditure [3].

The chance of performing science on the far-side of the Moon, together with the possibility of testing the technology of a CubeSat equipped with a miniaturised GNC system in a halo orbit, is of great importance for space and science progresses. In particular, the mission utilises a $12U$ form-factor CubeSat which carries the LUMIO-Cam, an optical instrument capable of detecting light flashes in the visible spectrum to continuously monitor and process the data. Moreover, the mission implements a novel orbit design and latest CubeSat technologies to serve as a pioneer in demonstrating how CubeSats can become a viable tool for deep space science and exploration [1]. This offers not only the opportunity to deepen our knowledge on the Moon environment, studying the flashlights of meteoroids, and allowing to develop predicting algorithms for Earth impacts, but also to retrieve fundamental information for a future autonomous GNC system for CubeSats, which can potentially set the bases of the future space missions.

1.3. Research questions

Halo orbits require particular station keeping strategies to allow the satellite orbiting close to the nominal path all over the mission. The high instability of such orbits imposes some strict requirements and conditions for the GNC system to work. Furthermore, LUMIO mission objectives, see Appendix A, impose a peculiar station keeping strategy, necessary to carry out the Science cycles properly. That being said, the first research question is:

What are the requirements of the guidance, navigation, and control (GNC) system needed to control the trajectory of LUMIO?

LUMIO is planned to perform an autonomous navigation experiment, applied during a short time window to perform a scientific experiment, while testing this new technology. Radiometric navigation will instead represent the baseline method, based on radiometric data coming from Earth. The system is required to accomplish the mission operations in accordance with the ADCS, to correctly prepare and execute the manoeuvres all over the mission. To this aim, the following research question shall be answered:

To what extent can we model and simulate the coupling between GNC and ADCS systems of LUMIO in closed-loop during the operational phase of the mission?

1.4. Research objectives

The first research question of the thesis imposes the drafting of the GNC requirements as the initial goal of the work:

Identify the GNC requirements for the LUMIO mission, in terms of constraints and needs to have a correct operations execution during the mission.

The work is then based on the implementation of a station keeping algorithm which computes the manoeuvres magnitude, direction and timing, necessary to correct the LUMIO trajectory. This aims at answering the second research question, with the goal of:

Implement an algorithm able to compute the manoeuvres magnitude and direction at the scheduled points, and validate the station keeping process over the 1 year operative mission duration, relying on the mission requirements.

Once the station keeping strategy provides the control manoeuvres, a model should simulate, after having obtained the LUMIO nominal attitude, the rotation of the CubeSat to align the thrusters in the $\Delta\mathbf{v_{SK}}$ direction, the firing itself, and the re-orientation to restore the nominal attitude. Hence, an additional goal is:

Model the GNC system which shall simulate, accordingly to the manoeuvre to perform, the attitude corrections before and after the manoeuvre execution and the actuators behaviour, in a closed-loop process with the orbit propagation.

1.5. Structure of the thesis

The thesis aims at answering the research questions, hence at fulfilling the research goals listed.

The fulfilment of the requirements is a delicate trial and error process which imposes, at the beginning of the work, to define the high-level requirements and then implement the related algorithms and models for their verification. Subsequently, the results are compared with the listed requirements to identify the following step: modify the requirements in accordance with the simulations outputs, or perform new analyses to arrive at a better conclusion. Indeed, this is the procedure adopted in this thesis: the list of GNC requirements is presented in the final form, obtained through iteration of constraints and results; the version reported is commented and the requirements modified during the process are highlighted.

In doing so, the first parts of the work present the fundamental information and guidelines to consciously enter the most peculiar topics of the thesis. First an introduction of the mission itself and general information related to LUMIO and GNC systems are provided. Then an overview of the computer programs involved, the baseline interfaces used, and data set employed to develop the thesis are reported, with the necessary details to initialise the codes and introduce the reader to the core of the work. Before presenting the actual implementation of the algorithms, a general explanation of the mathematics and logic behind the models has been inserted. The last chapters provide the final results for both the development of a station keeping algorithm and the simulation of GNC operations, plus some important remarks on the solutions presented and hints for the future work.

In detail, the thesis is organised as follows:

- **CHAPTER 1**: it constitutes the introduction of the thesis; it includes a general overview of the work, the research questions and goals, the summary.
- **CHAPTER 2**: this chapter presents the State Of the Art of the LUMIO mission itself, of the station keeping algorithms, the GNC systems for CubeSats in such orbits, an introduction to the n-body problem and an overview of similar missions.

- **CHAPTER 3**: here the conventions introduced in the thesis, the programming platforms used, the initialisation of the codes with the input data, and the implementation logic are discussed.
- **CHAPTER 4**: the main topic of the thesis, the GNC system for LUMIO in the halo orbit, is here addressed in a theoretical and mathematical fashion, as it has been implemented.
- **CHAPTER 5**: this chapter introduces the GNC requirements, the guidance implementation strategy, and the algorithm itself.
- **CHAPTER 6**: similarly to chapter 5 but with respect to attitude determination and control implementations.
- **CHAPTER 7**: the results of the thesis are here reported and commented, together with considerations on parameters sensibility and Monte Carlo simulations.
- **CHAPTER 8**: in conclusion, some final remarks and recommendations for the future work are investigated.

2 | State of the art

"If I have seen further, it is by standing on the shoulders of Giants."

- Isaac Newton

2.1. Literature survey

A mission near a Lagrangian point requires particular attention to topics related to the n-body problem and Station keeping methods. The former determines the constraints for the selection of the operative orbit, the latter the operations to carry out in order to maintain the reference orbit during the mission timeline. In this chapter, the influence of these topics in the context of LUMIO will be analysed and compared to other case studies.

Last, few similar missions will be introduced, with the aim of getting in closer contact with this kind of space missions, similarities and relative problematic.

2.1.1. CubeSat scenario

CubeSats are a class of spacecraft called nanosatellites. They are built to standard dimensions (Units or "U") of 10 $cm \times 10\ cm \times 10\ cm$. They can be $1U$, $2U$, $3U$, $6U$ or $12U$ in size, and typically weigh less than 1.33 kg per U.

CubeSats offer numerous advantages with respect to conventional satellites[1]:

- **Higher accessibility**: for a long time, only a few countries and multinationals had the financial muscle and the technical and human capacity to access space. This has changed. The introduction and widespread use of CubeSats and small satellites has meant a total reduction in space barriers for all types of companies and countries. Advances in technology, the reduction in prices, the appearance of new providers

[1]Some of the information are retrieved from https://info.alen.space/advantages-of-cubesats-vs-conventional-satellites [last accessed: 01/11/2021].

and the reduction in waiting times for the implementation of new projects in space have had a decisive influence on this transformation.

- **Lower cost**: this is a basic criterion for any company. A conventional satellite, as a reference, can cost between 100 and 300 million euros. A small CubeSat can arrive to less than 500,000 euros, although the exact price depends on several factors, such as its size, the complexity of the type of service you want to carry out.
- **Shorter development times**: the possibility to start and finish the projects quickly is a great incentive to advance in the space sector. Just to give an idea, for a conventional satellite the waiting time can be up to 15 years.
- **Smaller size and weight**: two criteria that are closely related to the price of satellites. The lightweight heavily impacts in the cost of the launch, and similarly for the size.
- **Risk distribution**: Generally speaking, the loss caused by a CubeSat failure is way lower with respect to a conventional satellite, allowing to save money on security tests or similar.

From these and other characteristics that the CubeSats offer, it is clear that they represent a valuable option to test new space technologies.

2.1.2. LUMIO mission overview

LUMIO was one of the proposals submitted to the SysNova LUnar CubeSats for Exploration call by the European Space Agency. The mission was awarded ex-aequo winner of the challenge, and its scientific relevance and technical feasibility were confirmed by an independent study conducted by the ESA Concurrent Design Facility [2]. The phase A has been successfully concluded at the beginning of 2021 and Phase B is expected to begin in early 2022.

Scientific relevance The main objective of the mission is the study of the impact flashes caused by the impact of micro meteoroids on the Moon surface. These bodies, in particular, are represented by Near Earth Objects (NEOs), asteroids or comets with a perielion of less then 1.3 AU, listed as more than 23,900 discovered [2]. Their impact with Earth can cause a wide range of catastrophic events, hence it is very important to invest resources to better understand and, possibly, predict these events [2].

Meteoroids are hardly detectable in a direct way, but can be observed indirectly from other phenomena, such as their impact with a celestial body, represented in this case

by the Moon. Indeed, a large amount of meteoroids and micro meteoroids continuously enter the Earth–Moon system, causing substantial modifications of the lunar surface. In particular, it is theorised that the lunar nearside has approximately 0.1% more impacts than the far-side, due to the influence of the Earth gravity field, while the equatorial flux is expected to be 10–20% larger than the polar regions, due to the higher number of large meteoroids in low orbital inclinations [2].

The impacts on the lunar surface cause emission of radiation through flashes; the detection of these flashes has been selected by LUMIO as the most advantageous method to study this phenomenon.

Based on these facts, the science question that LUMIO intends to answer is: *what are the spatial and temporal characteristics of meteoroids impacting the lunar surface?*; the corresponding science goal will be to *advance the understanding of how meteoroids evolve in the cislunar space by observing the flashes produced by their impacts with the lunar surface* [2].

Mission analysis The LUMIO nominal trajectory is represented by an Earth–Moon L2 halo orbit. The duration of the operative phase of the mission, in the halo, is planned to last 1 year.

The mission phases from the launch to the end of life disposals are illustrated in Fig. 2.1.

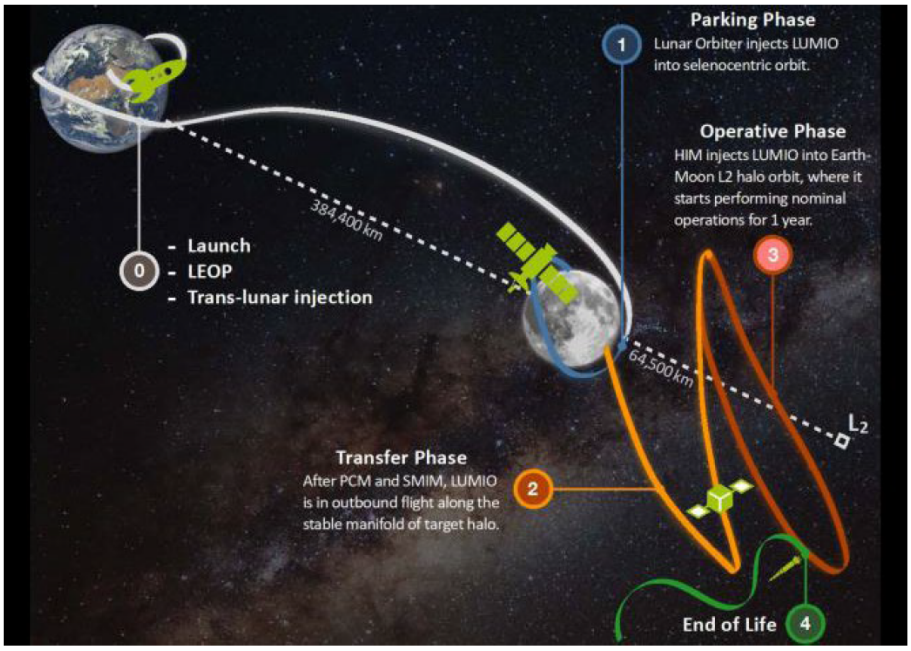
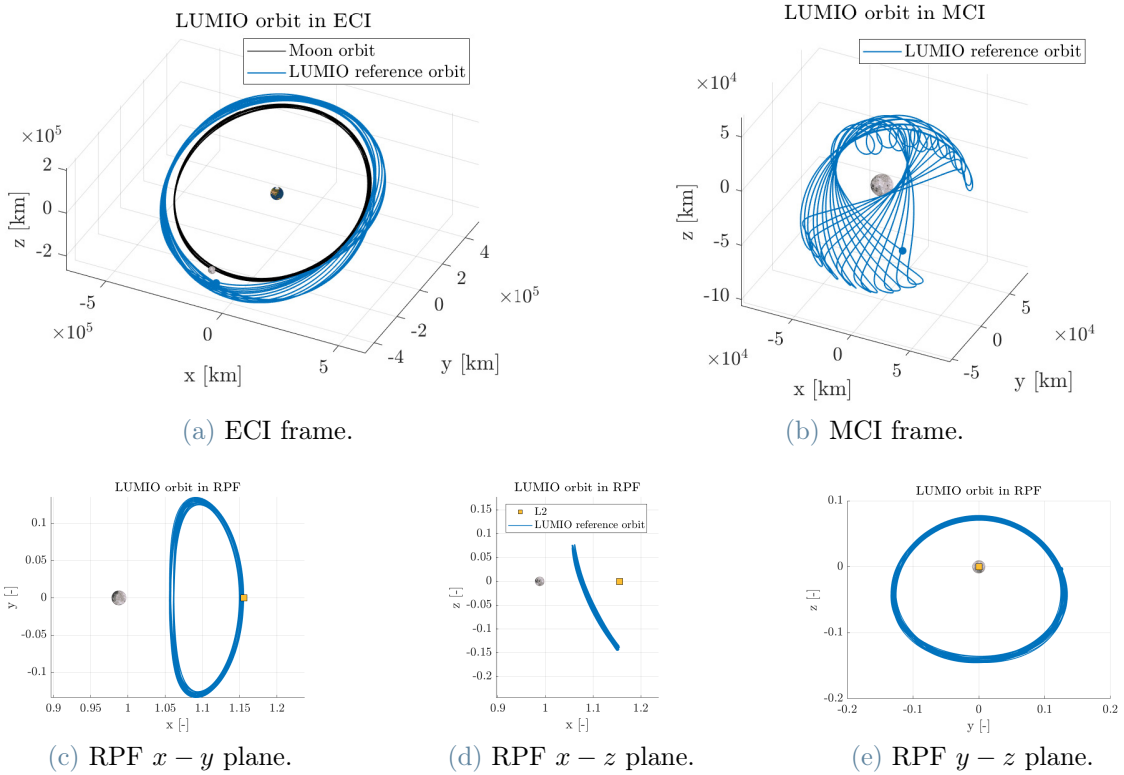


Figure 2.1: LUMIO mission concept and phases [2].

The selected LUMIO operative orbit is characterised by a Jacobi constant $C_j = 3.09$ [1], which identifies the level of energy of that particular trajectory. Such an orbit has been selected from a set of 14 candidates by means of an optimisation routine which identified this solution to be the most $\Delta\mathbf{v}$ saving and with absence of any eclipse during the complete 1 year nominal mission lifetime [2]. Fig. 2.2 shows the nominal orbit in the Earth and Moon-centred reference frames, and in the Roto-Pulsating Frame (RPF), during the 1 year operational phase of the mission.



(a) ECI frame.

(b) MCI frame.

(c) RPF $x-y$ plane.

(d) RPF $x-z$ plane.

(e) RPF $y-z$ plane.

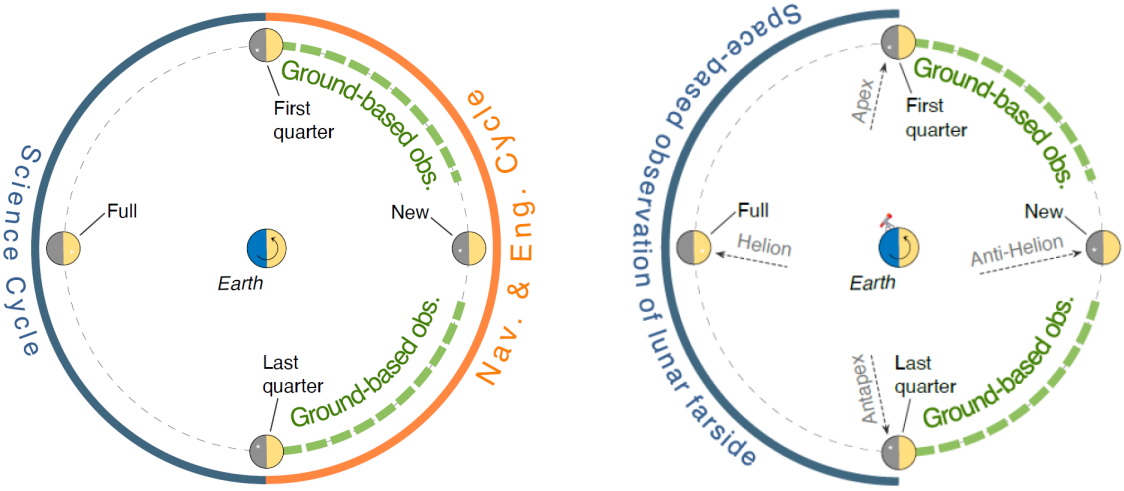
Figure 2.2: Nominal orbit of LUMIO during the 1 year mission operations.

From the plot of the operative trajectory it can be noticed the slight oscillation of the orbit path during time, which is representative of the quasi-periodic behaviour of many libration point orbits.

It is important to notice some peculiar features of the reference orbit to better understand the mission operations.

Since the Moon spin-orbit is locked into a 1:1 resonance with respect to Earth, just one face of the Moon is always visible from ground. This is also the face that can be observed by ground-based assets during night-time. Similarly, the science experiments on the far-

side of the Moon have to be conducted during the lunar night-time, which happens every half of the Moon revolution period around the Earth, lasting 29.53 days; see Fig. 2.3b. This leads to the subdivision of the LUMIO mission in two phases alternating one after the other: a Science cycle of around 15 days during the lunar night-time, and the Nav&Eng cycle, of approximately the same duration and happening during the lunar light-time. Fig. 2.3a shows these two phases, in blue and orange respectively.



(a) LUMIO concept of operations. (b) Moon phases ad main meteoroids directions.

Figure 2.3: Moon phases and mission ConOps [1, 2].

In particular, during the Nav&Eng cycle, the CubeSat will perform the station keeping manoeuvres and the related activities, in such a way to correct the trajectory for the next orbit, the Science cycle, to properly conduct the experiments. The planning of the GNC operations must take into account this timeline, together with other aspects such as: sensors acquisition latency, manoeuvres duration, station keeping errors, uncertainties, etc. These aspects will be analysed in detail in Chap. 4.

System design The design of the spacecraft has been iterated from Phase 0 to Phase A, including the response by ESA's CDF studies. Some peculiar characteristics are presented in the following; further details can be found in [1, 2, 4–7].

LUMIO is a 12U CubeSat of 22.8 kg; it is equipped with the LUMIO-cam, a custom payload developed by Leonardo S.p.A. This optic is fundamental to conduct the Science cycles collecting the lunar flashlights images, other than serve to carry out the experiment on autonomous optical navigation. Fig. 2.4 and Fig. 2.5 show the configuration of the CubeSat and the LUMIO-cam design, respectively.

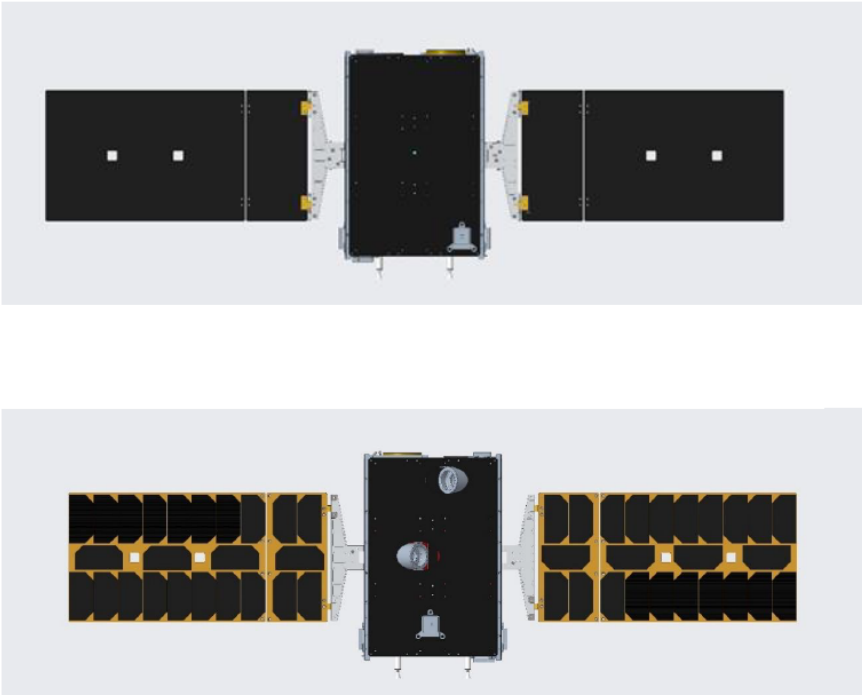


Figure 2.4: LUMIO configuration [2].

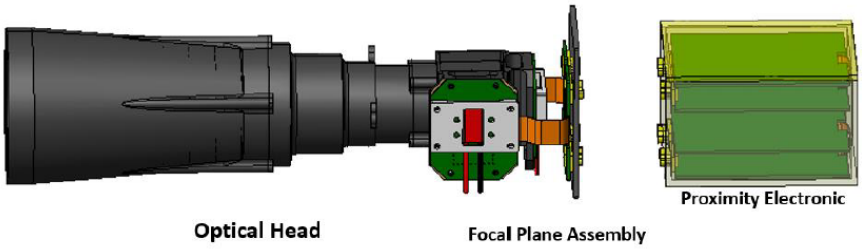


Figure 2.5: LUMIO-cam proposed design [2].

A brief overview of each subsystem is hereafter reported.

The ADCS needs to meet strict requirements on the camera pointing towards the Moon, the antenna towards the Earth, and the solar panels towards the Sun, which move by means of a SADA system. It is equipped with 2 star trackers, 6 sun sensors, 1 IMU for orbit determination, 4 RWs and a RCS for attitude control.

The PS is equipped with a mono-propellant main propulsion system, with a maximum thrust level of 1 N; it consists of two thrusters placed along the optic direction, but on the opposite face with respect to it. Both cold gas and electrothermal thrusters are being considered for the RCS part, with a thrust level in the range 1-10 mN.

The TMTCS is primarily based in radiometric navigation, with direct-to-Earth link (in S

or X band).

One of the main features of the OBDHS is the presence of an On-Board Payload Data Processing unit (OBPDP), that allows to significantly reduce the amount of data to be sent to ground by limiting them to the scientifically significant data only. To this aim, the OBPDP is designed in such a way to: 1. detect and keep only the camera images in which impact flashes are present; 2. cut from the whole image a smaller "tile", including the flash area and the information on where this area is located on the lunar far-side surface as seen by the spacecraft. This data processing strategy allows for a reduction by a factor in the order of 10^6 on the amount of data to be stored and sent to ground.

The Electrical Power System (EPS) will consist of a power generation unit in the form of 2 movable solar arrays, a battery pack, and a power distribution unit to regulate and distribute the power to the various subsystems. Each solar array presents 24 solar cells with area of 0.003 m^2 each. The cell efficiency is 0.2831. Overall, 48 solar cells are equipped on the two solar arrays, for a total area of 0.144 m^2. The ISISpace Modular Electrical Power Subsystem is the second-generation modular EPS and is designed as a flexible EPS for CubeSats from $3U$ upwards. The modular architecture allows the EPS to be tailored to the needs of the platform without needing customisation. The distributed design philosophy that underpins the IMEPS design allows unprecedented flexibility in output bus count and voltage, and enables tailorable redundancy to be applied for selectable parts of the platform, depending on the needs of the mission.

The chosen structure is the ISIS $12U$ structure. The primary outer structure acts as the payload mounting interface and an internal secondary structure accommodates avionics boards and components in a stack. The Total Ionizing Dose over the mission lifetime is predicted to be 8.547 ± 1.461 $kRad$ assuming a solid sphere Aluminium shielding of 1.5 mm which corresponds to the minimum shielding offered by the ISIS structure and solar panels. The simulation showed the thickness of 1.5 mm represents a sufficient value for LUMIO. Eventually, the fluence is further reduced by a factor of 3 by increasing the shielding to 2.5 mm.

Considering that the Hot and Cold cases are quite similar, it has been deemed reasonable to apply a completely passive design strategy for the Thermal Control System, identifying proper coatings for the external panels of the spacecraft and verifying that the temperature reached by representing nodes remain in the admissible operative range in both cases. The TCS proposes a Black Anodized Aluminum coating for the main CubeSat body, and the back faces of solar panels, not covered by cells. The hinges are planned to present an Anodized Aluminum coating. No eclipse occurs during the mission lifetime,

hence the cold case is represented by the condition in which the least amount of surfaces is illuminated; in this case the TCS requirements are met apart for few components which may require a dedicated heating.

Some modification are likely to verify in the next phase, however the design presented constitutes a solid baseline, confirmed at the end of the Phase A.

2.1.3. n-body problem

The orbit of LUMIO is obtained in the context of the n-body problem. It consists in the general gravitational case in which the dynamics of the smallest body, the spacecraft, is computed considering the influence that n external bodies have on it. The complex and non-linear dynamics which originates, determines the formation of highly unstable orbits, which can be controlled to originate periodic trajectories as the halos.

The simpler version of the n-body problem is the Circular Restricted 3 Body Problem (CR3BP) [8], which will be briefly analysed here for its high relevance in this mission context. Indeed, it consists in a simplified gravitational problem, with respect to the real n-body case, which allows to identify the presence of five solutions, called Lagrange points, in the dynamics equations. To obtain their derivation, have a look at Fig. 2.6.

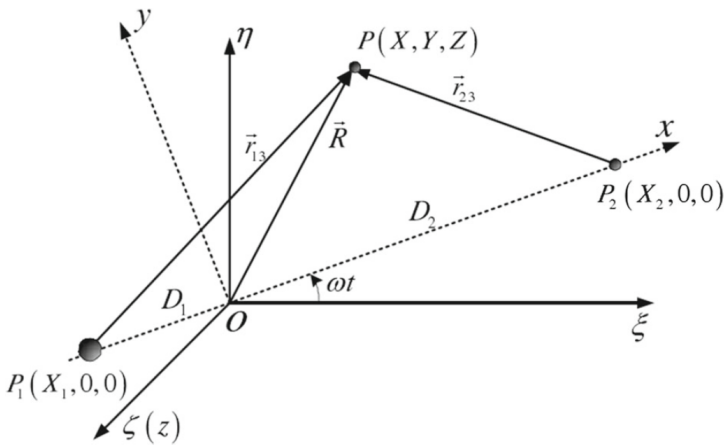


Figure 2.6: Geometry of the CR3BP [9].

The CR3BP assumes two primary bodies P1 and P2 moving in a circular orbit about their centre of mass. The mass of the third body, P, is negligible compared to the masses of the two primaries. The two primaries in this case are the Earth and the Moon, the third body, of course, is LUMIO. A rotating coordinate system with the centre of mass at the origin and both primaries fixed on the x-axis is chosen to describe the motion of the

third body. For convenience, the angular velocity of the rotating frame, the total mass and the distance between the two primaries are normalised to 1. The parameter μ is the ratio of the mass of P2 to the sum of masses of P1 and P2. Then, masses of P1 and P2 become $1-\mu$ and μ in the dynamical model. The coordinates of the first and secondary bodies are $[-\mu, 0, 0]$ and $[1-\mu, 0, 0]$ respectively, as shown in Fig. 2.6. The equations of motion of the third body P are [9]:

$$\begin{cases} \ddot{x} - 2\dot{y} = \frac{\partial \Omega}{\partial x} \\ \ddot{y} + 2\dot{x} = \frac{\partial \Omega}{\partial y} \\ \ddot{z} = \frac{\partial \Omega}{\partial z} \end{cases} \tag{2.1}$$

where, Ω is the effective potential of the system:

$$\Omega = \frac{1}{2}\left(x^2 + y^2\right) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} \tag{2.2}$$

Here, r_1 and r_2 are the scalar relative distances of the third body from the primary and the secondary:

$$r_1 = \sqrt{(x+\mu)^2 + y^2 + z^2} \tag{2.3}$$

$$r_2 = \sqrt{(x-1+\mu)^2 + y^2 + z^2} \tag{2.4}$$

While the energy of P is not constant in traditional two-body sense due to the rotating formulation of the differential equations, the system does admit an energy-like integral labelled the Jacobi constant C, that is:

$$\begin{cases} 2\omega - v^2 = C \\ v^2 = \dot{x}^2 + \dot{y}^2 + \dot{z}^2 \end{cases} \tag{2.5}$$

Given an energy level C, the third body can only move in the region described by the inequality:

$$2\omega - C \geq 0 \tag{2.6}$$

which is called the Hill's region, and the boundary of this region is called zero velocity surface. With the decrease in Jacobi constant (corresponding to the increase in energy level), the regions of allowable motion gradually increase and connect with each other. The five Lagrange points are stationary solutions of the equation:

$$\nabla \Omega = 0 \tag{2.7}$$

Three of these points are collinear and can be reached with increasing energy from L1 to L2 to L3. The remaining two points are in equilateral positions and require even more energy to be reached, corresponding to an higher level of stability; see Fig. 2.7.

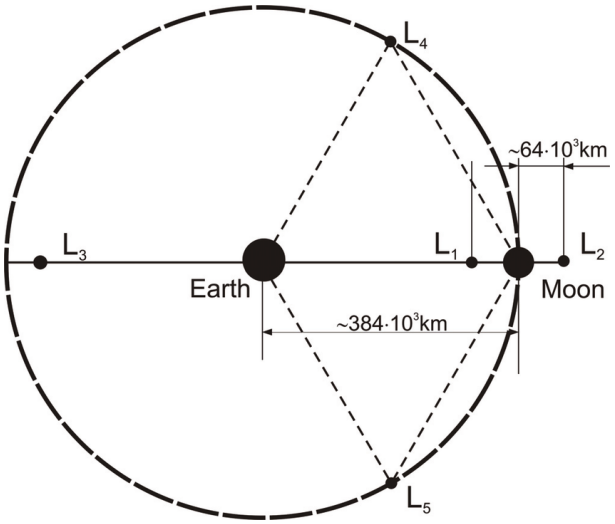
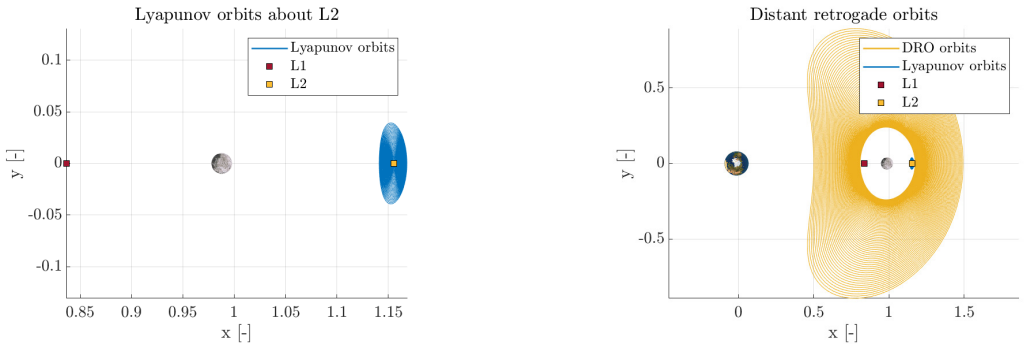


Figure 2.7: Libration points in the Earth-Moon system [10].

The trajectories which can be originated from the CR3BP are of different kinds: the halos (HO), Distant Retrograde Orbits (DRO), Lyapunov Orbits (LO), see Fig. 2.8, and some sub-categories, as the Near Rectilinear halo Orbits (NRHO) which are a particular case of halo orbits which get very near to the second primary body at the closest approach, and assume a vertical and elongated shape.



(a) Lyapunov orbits. (b) DRO orbits.

Figure 2.8: Example of libration point orbits.

However, it is important to remember that these orbits represent ideal solutions arising from the CR3BP: in the case in which a real scenario has to be simulated, gravitational disturbances coming from the other solar system bodies must be included. This gives rise

to the n-body problem, based on which the LUMIO orbit has been computed. Indeed, the reason of the "quasi" halo orbit of LUMIO, is precisely due to the introduction of perturbations which destabilise the ideal path. This clearly increases the complexity of the dynamics, making more clear why the use of such unstable trajectories requires particular attention to problems of station keeping and orbit control.

In the context of LUMIO, and more specifically of this thesis, the n-body problem is faced by means of a 6 DOF, ephemeris based approach. To give a better understanding of the model employed, equations used and other insights, refer to the dedicated chapter at Sec. 3.3.1. Indeed, the model used is associated to the Simulink tool which embeds it; therefore its architecture is discussed before providing its functioning.

2.1.4. Station keeping problem

The high instability of halo orbits and the numerous factors which contribute to disturb the CubeSat trajectory require the development of a station keeping strategy to be applied by the spacecraft during the mission. The challenges of station keeping control emerge from high accuracy, low computation burden, and minimal fuel cost control requirements under the condition of spacecraft dynamic uncertainties, unmodeled perturbations, and initial orbit injection errors. There exist several kinds of station keeping techniques applicable to libration point orbits; they will be introduced in Sec. 2.2.1, where the mathematical insights of some notorious methods are discussed.

Dealing with station keeping performed by CubeSats has also to face with the technology state of the art currently employed in these satellites. Considerations about the common trade-off reached in the developments of some subsystems, applied to obtain a faster, cheaper but performing space product, must be done and kept in mind. Thus, faults like tumbling, loss of power unit and eclipse can utmost result in failure of these mission. Furthermore, placing a CubeSat in such an unstable orbits of Earth–Moon system can result into a challenging task to accomplish for station keeping once some anomaly is encountered. As highlighted by Langer [11], the three main subsystems causing CubeSat failures are OBDHS, EPS and TMTCS. As the thruster that will be employed in these spacecrafts are fully power driven, this could lead to major failure of such spacecraft. It can be understood the importance of a fault tolerant control for the station keeping of CubeSat's serves the foundation of these control to achieve a successful mission. In case in which no particular fault-tolerant strategy is taken into account, a suitable selection of the spacecraft components and subsystems functioning must be carried out, especially for those parts which can heavily compromise the station keeping.

2.1.5. Similar missions and case studies

The use of halo orbits or, generally speaking, of orbits around a Lagrangian point, offers unique opportunities in terms of scientific outcomes, as discussed before. Few examples are briefly introduced hereafter. They have been selected for their similarities with LUMIO, in terms of technology used, scientific outcomes and station keeping needs.

Lunar Flashlight (NASA) The Lunar Flashlight (LF) CubeSat team, managed by JPL, for NASA, is a mission based on a similar concept to LUMIO [12].

The LF consists on a 6U, 14 kg spacecraft whose objective is to demonstrate new technology and measure potential surface water ice deposits in the permanently shadowed region of the Moon, in preparation of future possible lunar exploration. The mission duration will be eight months, but is extendable up to an eighteen months period.

The trajectory includes three gravity-assist lunar flybys; after 6 months, LF is then inserted into a lunar near-rectilinear halo orbit (NRHO), achieving a perilune distance of approximately 15 km above the lunar south pole, which allows for 6 min of observations at each passage. The NRHO will have a period of about 5 to 7 days, and it is eclipse-free, allowing no limited downlink opportunities.

The GNC system relies on three orthogonal reaction wheels of 50 $mNms$, by Blue Canyon Technology (BCT), and 100 mN chemical thrusters using proven green monopropellant. The system will work based on input from ground station, which will be used to compare the nominal and actual state to asses the manoeuvre characteristics. Four BCT sun sensors and one BCT star tracker along with an high precision IMU are also present.

The main ADCS-GNC requirements for LF mission are briefly listed below [12]:

- GNC system is capable of handling up to 10 deg/sec spin rate out of launch vehicle's dispenser.
- GNC system is able to detumble from post-separation spin rate 10 deg/sec within 10 min.
- GNC system is able to achieve milliradian level of Moon pointing accuracy.
- GNC system is capable of arcsecond level of pointing knowledge.
- GNC system can communicate properly with other systems and command thrusters of the Propulsion system.
- Good level of autonomy: automatic transition between modes and an on-board

lunar ephemeris propagator.

- A sun-pointing mode to ensure the safety of the spacecraft after launch separation or after a fault detection.

The GNC subsystem is fundamental to accomplish the functions of transporting the spacecraft from Earth to the Moon, and once in lunar orbit, of pointing the payload instrument toward the centre of the Moon for scientific measurements.

LF GNC system has two control modes: Safe Mode (SFM) and Fine Pointing Mode (FPM). The SFM also comprehends the sun-pointing attitude, achieved at GNC activation and during possible malfunctions.

EQUULEUS EQUULEUS is a 6U CubeSat planned to be launched by the NASA's SLS EM-1 vehicle and aims to reach and stay around the Earth–Moon L2 point with a purpose of scientific observation. In particular, the mission aims to demonstrate trajectory control technology in the Earth–Moon region via CubeSat and to perform scientific observation assessing the flux of meteors impacting on the lunar dark side [13].

The CubeSat is placed on a quasi-halo orbit, selected from an optimisation routine in the context of the n-body problem.

The station keeping strategy relies, as for LUMIO, on a Target Point Method approach, with 2 target points. The mean annual station keeping cost sets around 15 m/s [13].

OMOTENASHI OMOTENASHI is a JAXA 6U cubesat that aims to perform a semi-hard landing on the Moon's surface after being deployed into a lunar fly-by orbit by the American Space Launch System, Exploration Mission-1 [14]. One of the main challenges comes from the trajectory, which is characterised by a single deceleration manoeuvre instead of multiple ones for orbit insertion, descent, hovering and landing. After the deceleration manoeuvre, there is no time to correct for navigation and execution errors, so the robustness of the trajectory is of key importance [14].

OMOTENASHI is a challenging mission executed by a CubeSat, which represents another example of technology development in this field.

All the mission described have a common denominator represented by the space exploration with CubeSats in the context of the Earth–Moon, n-body scenario. Another example is provided by [15], which proposes a CubeSat mission for the exploration of the far-side of the Moon. In detail, a low-cost mission concept that consists of four CubeSats

(3 or 6U) in an Earth-Moon L2 (EML2) halo orbit is investigated. The mission objective is to provide real-time positioning service for lunar far-side assets, taking advantage of the visibility of EML2 halo orbits to both the Earth and lunar far-side [15]. The study also presents a station keeping analysis, based on x-axis crossing control, for the selected operative trajectory, and provides the subsystems components selected for the mission.

2.2. CubeSats GNC system

Now that an overview of the topics of the thesis has been provided, a dedicated section on the GNC system for CubeSats is reported, still in a theoretical and general manner.

The use of CubeSats in space missions is in continuous growth. However, to increase the potentialities and the outcomes arising from their use, reliable and highly innovative GNC systems are required. The possibility to perform autonomous navigation in space with such small and cheap satellites would open the doors to new horizons in the space exploration.

CubeSats GNC system is a very actual subject of study. It is the case of the Lunar Flashlight mission described in Sec. 2.1.5 with a similar target to LUMIO, or the use of GNC for In-Orbit Demonstration of Active Debris Removal with CubeSats [16], or the studies of Swarm of CubeSats for low Earth orbit debris removal experiments in constellations [17], and many others. Each mission with CubeSats requires its particular methodologies and technologies, although general similarities exist; in the following a brief overview of the typical GNC operations and technology is reported.

2.2.1. Station keeping algorithms

Different algorithms can be used to determine the control needed to correct the trajectory.

The *X-axis Crossing Control* algorithm applies a manoeuvre at a specified point along a trajectory to target a given set of parameters further downstream. Such algorithms have been, and are currently, successfully used to maintain various spacecraft in halo orbits; examples include ARTEMIS in the Earth–Moon system and WIND currently in a Sun–Earth halo orbit. In the case of HRHO, as for the lunar gateway, to avoid introducing errors at the sensitive region near perilune, which can lead to algorithm divergence, manoeuvres are placed at or near apolune. The target is placed at the x-z plane crossing near perilune, or at the nearby perilune itself, as this target location is observed to lead to lower orbit maintenance costs. A single component of the reference trajectory is selected as the target: the x component of rotating velocity in the Earth-

Moon rotating frame, v_x. Targeting the single component results in a trajectory that remains near the reference for low cost. A longer targeting horizon tends to lead to lower manoeuvre costs. However, especially in the presence of large perturbations, the targeter may converge less reliably as the horizon increases [18]. Further information on the *X-axis crossing control* can be found in [19].

Other methods consist in variants of the *LQR*, which differentiate for the selection of the weighting matrices and of the discrete or continuous state: the *Averaged-in-time LQR (ALQR)*, the *Fixed-in-time LQR (FLQR)*, the *Periodic LQR (PLQR)*, which can be delve into [18, 20–24]. Similarly holds for *Model Predictive Control* methods, which is a feedback control scheme based on minimising a quadratic cost function subject to point-wise-in-time constraints using a prediction finitely many time-steps into the future. At the end of the prediction horizon, *MPC* imposes a terminal constraint along with a terminal cost to ensure stability. It can be found in different versions: *Averaged-in-time*, *Fixed-in-time*, *Periodic*.

2.2.2. Current technology

In [25], a survey of small satellites[2] GNC components has been performed. The state of the art currently presents products with characteristics as those in Table 2.1.

Component	Performance	Status
Star Trackers	0.007° pointing knowledge	TRL 9
Magnetometers	6.5 nT resolution	TRL 9
Sun Sensors	0.1° accuracy	TRL 9
Earth Sensors	0.25° accuracy	TRL 9
Gyroscopes	1° h^{-1} bias stab., 0.1° $h^{-1/2}$ rand. walk	TRL 9
GPS Receivers	1.5 m position accuracy	TRL 9
Reaction Wheels	0.1 Nm peak torque, 1.5 Nms storage	TRL 9
Magnetorquers	5 Am^2 peak dipole	TRL 9
Control Moment Gyros	0.2 Nm peak torque	TRL 6
Integrated Units	0.007° pointing capability	TRL 6

Table 2.1: Typical GNC components for small satellites.

Small spacecraft GNC is a mature area, with many previously flown and high TRL components offered by several different vendors [25].

[2]Intended as spacecrafts with a launch mass below 50 kg, hence of the class of LUMIO.

3 | Programming interface

"Simplicity is the soul of efficiency."

- Austin Freeman

The thesis work has been entirely developed in MATLAB® and Simulink®. The version employed is the R2020a. Simulink in particular has been selected as the primary simulation framework, where the final model is supposed to be tested. MATLAB has been used to write the code for those parts which can be computed out of the Simulink closed-loop simulation, in a more efficient and time-saving manner, or to save and plot the results. In the following the MATLAB and Simulink frameworks are described, to give a clear idea of how they have been set up and how they work.

As previously mentioned, the main work has been developed in Simulink and, more in detail, with CUBORG[1], CUBesat Orbit and GNC tool, an n-body ephemeris-based model, trajectory and AOCS simulator. This tool allows to perform simulations in a 6 DOF environment, in which it is supposed to be validated the S/K strategy for LUMIO.

3.1. Conventions

Some notations used in this work are hereafter introduced.

3.1.1. Vectors notation

The vectors used in the thesis, especially the position ones, are written following a precise notation, able to indicate the bodies involved and the reference frame in which the vector is described. This convention is applied in those cases in which a vector going from a body[2] to another is used; a typical example is represented by the position vectors from ephemeris.

[1] It consists in a numerical toolbox developed by the Deep-space Astrodynamics Research & Technology (DART) team at Politecnico di Milano [26].

[2] Generally represented by a planet, moon, spacecraft.

Considering a vector with magnitude M, direction from a body i to a body j, described in the reference frame F, the notation applied is:

$$\mathbf{M_{ij_F}}$$

where the bold font indicates the vector quantity. Moreover, the *body frame* is indicated with B and is referred to the spacecraft, the *inertial frame* with N. To indicate the bodies, b is used for the spacecraft, the small initial letter of the name for the planets, the Moon and the Sun.

Hence, to indicate the position vector which goes from the spacecraft to the Sun, in the inertial frame, it is used:

$$\mathbf{R_{bs_N}}$$

Moreover, normalised vectors are expressed with the hat symbol:

$$\hat{\mathbf{v}}$$

and the matrices are written written in regular font:

$$\Phi$$

3.1.2. Quaternions notation

The use of quaternions is of fundamental importance in attitude estimation and control. Their convenient mathematical notation for representing spatial orientations and rotations of elements in three dimensional space determines one of the reasons for their large application. They are especially appealing because no singularities are present and it is possible to express a sequence of two consecutive rotations, combining quaternion components in the reversed order of rotations. Unfortunately, quaternions have no physical meaning and therefore their use is not intuitive.

In this thesis the quaternions are written using the scalar-first vector-last notation, in which the scalar term q_s is in the first position, followed by the vector part $\mathbf{q_v}$:

$$\mathbf{q} = \begin{bmatrix} q_s \\ \mathbf{q_v} \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{3.1}$$

Hence q_1 is the scalar term. This notation has been selected being used both in CUBORG and generally in Simulink (Hamiltonian notation [27]) to handle quaternions operations.

3.1.3. State transition matrix notation

The state transition matrix is widely employed through all the thesis, especially during the discussion of the station keeping algorithm. The notation used in this work is the following:

$$\mathbf{x}(t) = \Phi(t, t_0)\mathbf{x}(t_0)$$

where the *STM*, which propagates the vector $\mathbf{x}$ from t_0 to t, is written as $\Phi(t, t_0)$ and not as $\Phi(t_0, t)$, which is a common alternative notation found in literature.

3.2. MATLAB framework

The MATLAB interface contains all the information, in form of data or lines of code, which are required for the simulation with CUBORG or for auxiliary necessities. Three main folders constitute the core of the program:

- **spice**: it is the SPICE Toolkit - NASA / NAIF folder, used to retrieve the information related to ephemeris, bodies constant, frames transformation functions and everything related to geometrical properties in space.
- **cuborg_data**: this folder includes two types of data:
  - *kernels*: they are subdivided in *spk*, *pck*, *lsk*, *fk* kernels used for the computations with SPICE; they are collected in the meta-kernel *mk* to be all loaded simultaneously by SPICE.
  - *input*: this folder contains all the `.txt` files necessary for the simulation to run. All the data used in the tool are here reported, grouped by category, to achieve the most orderly layout. They are briefly described in the following, to provide a comprehensive understanding of their content.
- **cuborg**: this is the principal folder. It embeds the files `Init.m` and `Main.m` which allow to open the data files and run the simulation. Moreover it contains the subfolders:
  - *GNC_Matlab*: it represents the folder of the all brand new scripts and functions written specifically for this thesis.
  - *simulink*: it contains the Simulink main file `CUBORG.slx` and the libraries of

the tool.

- *script*: it is the folder containing all the MATLAB scripts written for the tool, except for those developed for the current thesis.
- *functions*: it includes a series of sub-folders with MATLAB functions, each of which has a specific role in the simulation, as reading the input `.txt` files or the ephemeris. As for the scripts folder, the functions developed for the current thesis are not saved here but in *GNC_Matlab*.
- *doc*: it is a documentation repository, containing some useful readings to start approaching the CUBORG tool.

In details, the *input* folder contains `.txt` files with all the data required by the simulation with CUBORG. In the following, only the input files used in the thesis will be described; subsequently their content will be shown in detail in the initialisation of the tool in Sec. 3.4.

- `spacecraft.txt`: information related to the spacecraft itself, as the inertia, the surfaces area, data for SRP etc.
- `simulation_parameters.txt`: data related to the integration scheme and simulation settings adopted in CUBORG. It can be specified the solver type, the step size, the solver to use, the tolerances etc.
- `scenario.txt`: data related to start and final epochs of the simulation, SPICE identification numbers of the reference frame to use, the body to consider in the n-body problem and others.
- `ICs.txt`: initial conditions for the propagation, in position, velocity, angular velocity, quaternions and mass.
- `constants.txt`: all the masses of the solar system bodies, gravitational constant, astronomical unit, solar pressure constant.
- `GNC.txt`: data related to the GNC simulation specifically; sensors, actuators and control data are here reported. It has been written specifically for this thesis.

3.3. Simulink framework

The `CUBORG.slx` Simulink file appears, at the upper layer, as in Fig. 3.1.

It can be noticed the presence of 3 main sections:

- ***WORLD***: it is responsible of simulating the orbit of the spacecraft, by introducing

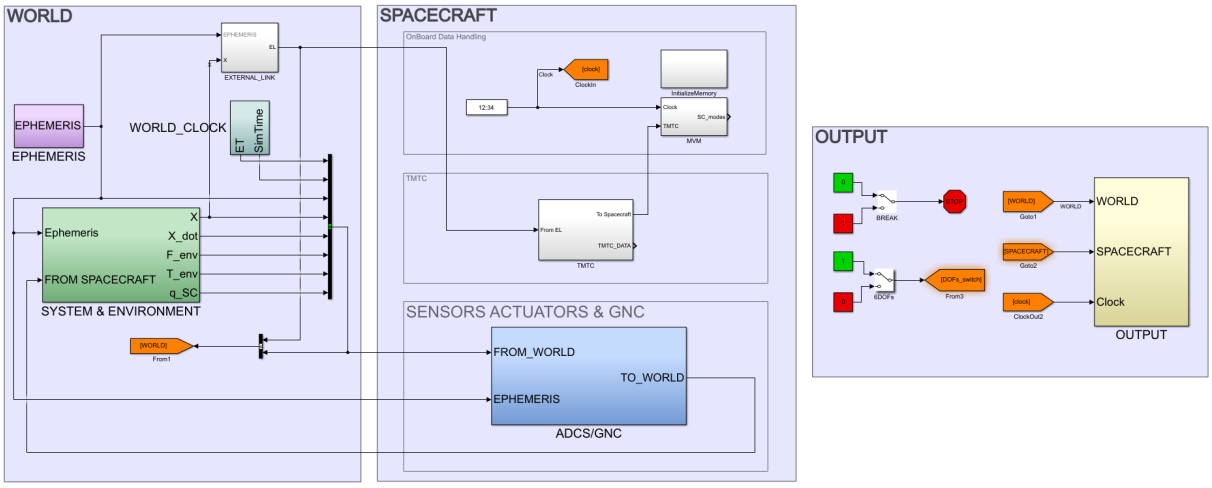


Figure 3.1: `CUBORG.slx` interface.

the perturbations computed, using the ephemeris generated and the data uploaded. All the outputs are collected with the *WORLD* tag.

- ***SPACECRAFT***: it contains the models of the OBDH, TMTC, GNC subsystems. It takes the required output from the *WORLD* section and collects its own under the *SPACECRAFT* tag.
- ***OUTPUT***: this portion of the tool is just in charged in saving the whole outputs from *WORLD* and *SPACECRAFT* models, at the desired frequencies.

As far as the work developed in this thesis is concerned, not all the blocks of the tool appearing in Fig. 3.1 have been used. In particular, *External_Link*, *OBDH*, *TMTC* and *Sensors* models have not been neither modelled nor modified and present a white background, to be distinguishable from the blocks used in the thesis, which are coloured. That being said, the following pages of this chapter will focus on the *WORLD* section. This in fact has been adapted in this thesis to be compliant with the LUMIO case and its settings require a dedicated description.

Once the *WORLD* section has been analysed, the AOCS components in the *ADCS-GNC* block will take the foreground for the rest of the work, representing the main model developed for this thesis.

3.3.1. *WORLD* model: orbit propagation

In Fig. 3.2 the *WORLD* section is reported. The chosen reference frame for CUBORG simulation has been selected as the J2000 frame, centred in the Earth; all the details on the simulation settings are reported in Sec. 3.4, together with the data used, collected for

each variable.

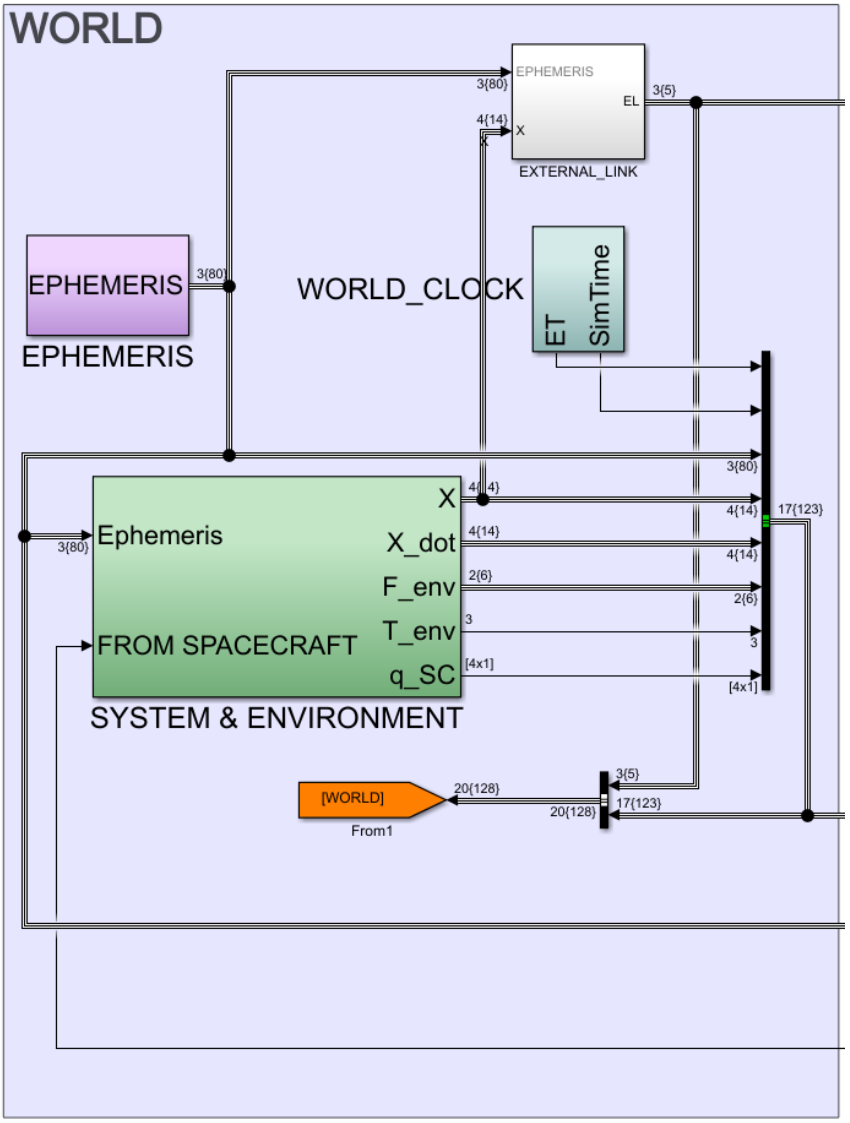


Figure 3.2: `CUBORG.slx` *WORLD* area.

The 3 main blocks which form this model are:

- ***Ephemeris***: this subsystem uploads in Simulink the ephemeris computed with SPICE according to the simulation settings.
- ***World_ Clock***: this block generates the Ephemeris Time (ET) adding the simulation time to the starting epoch of the simulation, still in ET.
- ***System and Environment***: It represents the core of the propagation. It contains the attitude and state propagators, with input the ephemeris and the control coming from ADCS and GNC and output the propagated quantities.

In Fig. 3.3 it is depicted the *System and Environment* block: it can be noticed the presence of the *State Propagator* itself and the *Environment* block, which takes into account about the gravitational forces and solar radiation pressure.

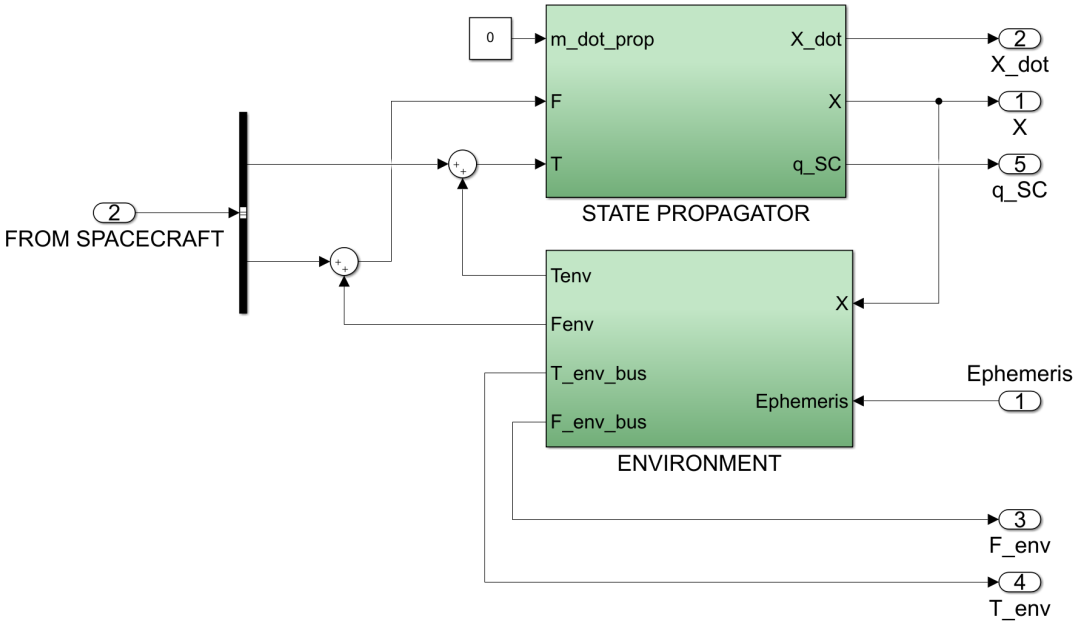


Figure 3.3: CUBORG *System and Environment* architecture.

- ***State Propagator***: it contains both the trajectory and attitude propagators. The *Trajectory Integrator* takes as input the spacecraft mass m_{sc}, the state components and the resultant force $\mathbf{F}$ acting on the spacecraft, sum of the environmental contributions and control actions, and integrates the dynamics equations:

$$\begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \\ \dot{v_x} = \frac{F_x}{m_{sc}} \\ \dot{v_y} = \frac{F_y}{m_{sc}} \\ \dot{v_z} = \frac{F_z}{m_{sc}} \end{cases} \tag{3.2}$$

  The *Attitude Integrator* propagates both the attitude dynamics and kinematics. It takes as input the angular velocity components $\boldsymbol{\omega}$, the quaternion vector $\mathbf{q}$, the resultant applied torque $\mathbf{T}$ and the inertia tensor J, to provide as outputs the integrated quantities.

The dynamics equations read:

$$\begin{cases} \dot{\omega_1} = \frac{J_2 - J_3}{J_1}\omega_2\omega_3 + \frac{T_1}{J_1} \\ \dot{\omega_2} = \frac{J_3 - J_1}{J_2}\omega_1\omega_3 + \frac{T_2}{J_2} \\ \dot{\omega_3} = \frac{J_1 - J_2}{J_3}\omega_2\omega_1 + \frac{T_3}{J_3} \end{cases} \tag{3.3}$$

where J_1, J_2 and J_3 represent the diagonal terms of J, assumed diagonal.

The kinematics equations derive from:

$$\dot{\mathbf{q}} = \frac{1}{2}\Omega(\omega)\mathbf{q} \tag{3.4}$$

$$\Omega(\omega) = \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} \tag{3.5}$$

- ***Environment***: two environmental contributions are considered for the simulation: the gravitational forces and the disturbances, provided, in the case of LUMIO, by the solely SRP (it is at least by far the most relevant, considering there is not drag and the magnetic disturbances are really low). The two are here briefly analysed; indeed, the accurate modelling of these events is of great importance in the context of n-body problem trajectories, to properly simulate the correct orbital path. This, in particular, without station keeping, will lead the spacecraft to some unstable manifolds moving away from L2, due to the action of disturbances.

  The force resulting from the gravitational attraction on the spacecraft of N bodies, in the current reference frame, is given by:

$$\mathbf{F_g} = -m_{sc}\sum_{k=1}^{N}\frac{\mu_k}{|\mathbf{x} - \mathbf{r_k}|^3}(\mathbf{x} - \mathbf{r_k}) - m_{sc}\sum_{i=1}^{n}\frac{\mu_i}{|\mathbf{r_i}|^3}(\mathbf{r_i}) \tag{3.6}$$

  where n is the number of external bodies with respect to central one, to account for non-null accelerations of the reference frame. μ_k is the gravitational parameter of the planet considered, $\mathbf{x}$ and $\mathbf{r}$ are respectively the positions of the spacecraft and of the external bodies with respect to the selected reference frame origin[3].

  As for the SRP disturbance, the force provided is computed through the cannonball

[3]The Earth, in the model employed; further details are provided in Sec. 3.4.

model:

$$\mathbf{F_{srp}} = \frac{P_{SRP1AU} AU^2 A \; c_R \times 10^{-3}}{||\mathbf{x} - \mathbf{r_{sun}}||^2} \frac{(\mathbf{x} - \mathbf{r_{sun}})}{||\mathbf{x} - \mathbf{r_{sun}}||} n \tag{3.7}$$

where n can be 0 or 1 according to the presence of eclipse[4] or not. In particular:

$$P_{SRP1AU} = \frac{I_R}{c}, \quad I_R = 1371 \; W/m^2, \quad c = 2.9979 \times 10^8 \; m/s \tag{3.8}$$

is the solar pressure at 1 AU in Pa, A is the reflection area in m^2 and c_R is the reflectivity coefficient.

The total force is then obtained and passed as input to the propagator:

$$\mathbf{F} = \mathbf{F_g} + \mathbf{F_{SRP}} \tag{3.9}$$

it is expressed in kN, having utilised the kilometres for the position vectors.

The SRP torque $\mathbf{T_{SRP}}$ is obtained by means of an high fidelity model, taking into account about the surfaces disposition and their optical properties. The reason why this accurate model has not been employed even to compute $\mathbf{F_{SRP}}$ in Eq. (3.9), is related to the original settings of the LUMIO trajectory. Indeed, to obtain a simulated orbit coherent with that in the ephemeris file, the same models and settings are used, hence also the cannonball model to compute $\mathbf{F_{SRP}}$. Then, being the torque not affecting the trajectory state, it has been developed with an higher accuracy, employing the following model:

$$\mathbf{T_{SRP}} = \sum_{i=1}^{n} \mathbf{r_i} \times (\mathbf{F_i} \; n) \tag{3.10}$$

$$\mathbf{F_i} = -P_{SRP1AU} A_i (\hat{\mathbf{R}}_{\mathbf{bs_B}} \cdot \hat{\mathbf{n}}_\mathbf{B}) \left[(1 - \rho_s) \hat{\mathbf{R}}_{\mathbf{bs_B}} + (\mathbf{2}\rho_\mathbf{s} (\hat{\mathbf{R}}_{\mathbf{bs_B}} \cdot \hat{\mathbf{n}}_\mathbf{B}) + \frac{\mathbf{2}}{\mathbf{3}} \rho_\mathbf{d}) \hat{\mathbf{n}}_\mathbf{B} \right] \tag{3.11}$$

where $\mathbf{r_i}$ is the arm of the surface-i's centre with respect to the centre of gravity of the spacecraft, ρ_s and ρ_d are the specular reflection and diffusivity coefficients of the i-th surface, $\hat{\mathbf{n}}_\mathbf{B}$ is the vector of normals exiting from the i-th surface and $\hat{\mathbf{R}}_{\mathbf{bs_B}}$ is the normalised spacecraft to Sun distance, expressed in the body frame. n is the eclipse binary indicator. The profile of $\mathbf{T_{SRP}}$ during a 30 days period is reported in Fig. 3.4. For sake of completeness, and as a proof the two models for $\mathbf{F_{SRP}}$ are not providing too different results, Fig. 3.5 shows the comparison of the force norm

[4]The eclipse is computed using an umbra cylindrical eclipse model [26].

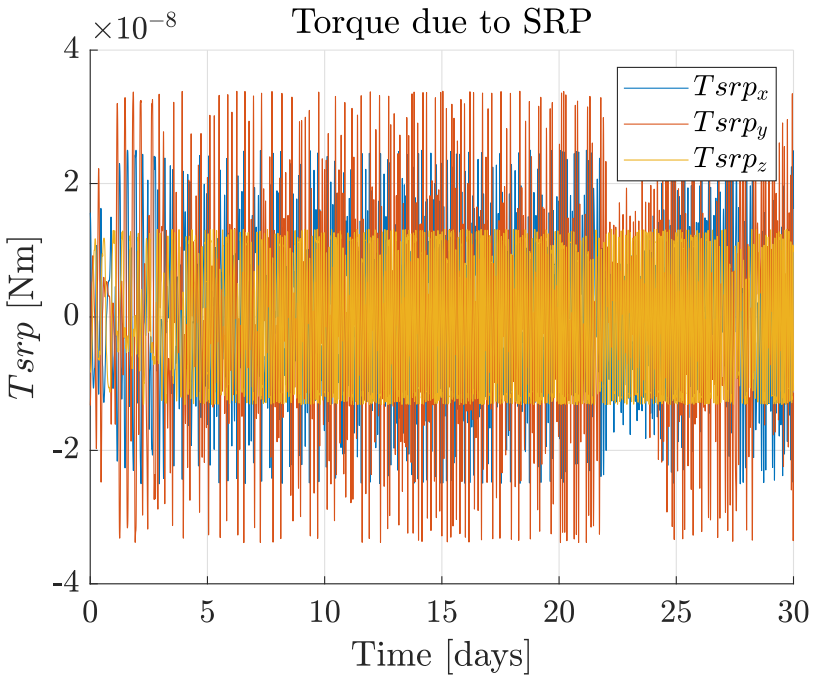


Figure 3.4: Solar radiation pressure torque acting on LUMIO.

in both cases. It can be noticed that they feature the same order of magnitude; moreover the cannonball model provides a sort of average solution with respect to the high fidelity model, coherently with the expectations.

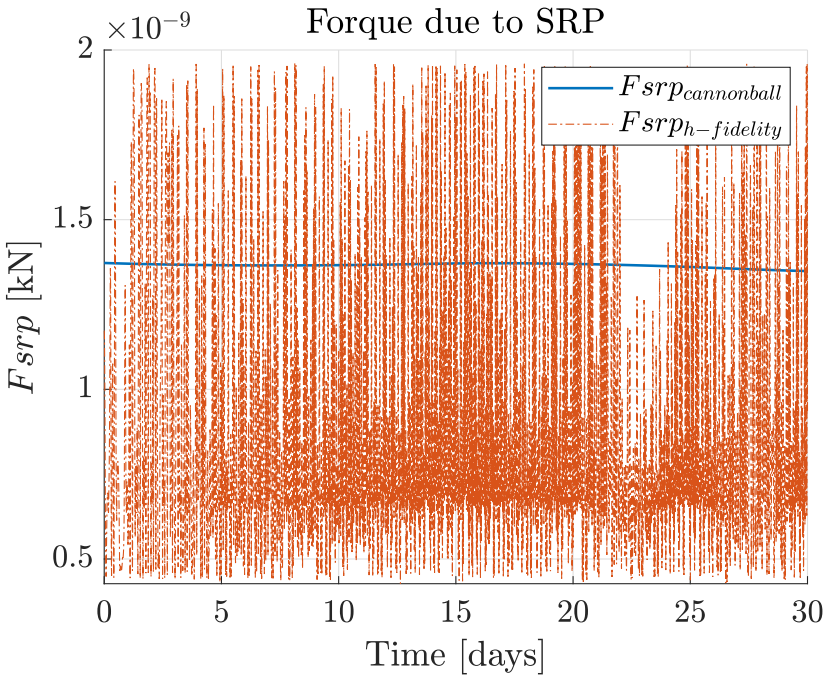


Figure 3.5: Solar radiation pressure force acting on LUMIO.

3.4. Tool initialisation

In the context of the LUMIO mission, the MATLAB and Simulink interfaces are initialised with the parameters reported in the tables below. The data related to GNC, summarised in the file `GNC.txt`, will be discussed in detail in Chap. 6, representing the specific topic of the thesis.

Input data	Description
$Depl\_unDepl = 1$ or 0	Deployed (d) or packed (p) configuration flag; set to 1 for deployed, to 0 for packed.
n_d/p	Normals to the surfaces for deployed and packed cases, α is the inclination of the solar panels. See Appendix B for the corresponding values.
r_d/p	Arms of the surface-i's centre with respect to the centre of gravity of the spacecraft for deployed and packed cases. See Appendix B for the corresponding values.
A_d/p	Surfaces areas for deployed and packed cases. See Appendix B for the corresponding values.
J_d/p	Tensor of inertia for deployed and packed cases. See Appendix B for the corresponding values.
$\rho_s = 0.6$	Avarage specular reflection coefficient of the surfaces, from [5].
$\rho_d = 0.1$	Avarage diffusivity coefficient of the surfaces, from [5].
$A\_cross\_SRP = 0.23\ [m^2]$	Cross sectional area employed in the cannonball model.
$Cr = 1.3$[5]	Reflectivity coefficient employed in the cannonball model.

Table 3.1: Data from `spacecraft.txt`.

[5]This is the value originally used to obtain the LUMIO reference orbit, hence it is maintained unchanged to simulate correctly the trajectory, although not being properly consistent with ρ_s and ρ_d.

Input data	Description
$SolverType = Fixed\text{-}step$	A fixed step has been chosen to perform the short simulations of the thrusters, and for computational time reasons.
$FixedStep = 0.15\ [s]$	Required step size to simulate thrusters dynamics of less than a second[6].
$Solver = ode2$	The Heun integration scheme has been selected for computational time efficiency and correctness.
$world\_save\_delta\_t = -1$	It determines a continuous save of the spacecraft trajectory.
$ephemeris\_delta\_t = 15\ s$	It determines an ephemeris discretization with intervals of 15 s, accordingly to the step size.

Table 3.2: Data from `simulation_parameters.txt`.

Input data	Description
$epoch\_format = ET$	The time unit required by SPICE is the Ephemeris Time (ET), in seconds.
$epoch\_start = 7.642512691856151 \times 10^8\ [s]$	It corresponds to the ET format of the date 2024-MAR-21-00:00:00, which is half day before the first manoeuvre during the orbit insertion, to better visualise the first orbit correction.
$epoch\_end = 7.694352691856151 \times 10^8\ [s]$	It is set as 60 days after the $epoch\_start$, but it will be changed accordingly to the duration of the simulation.
$trajectory\_origin\_ID = 399$	It is the identification number of the Earth in SPICE; indeed, the simulation has been set, as basis, using an inertial frame with the Earth in the origin.

[6]Further details related to integration settings are provided in the next chapters.

$trajectory\_reference\_frame = J2000$	Reference frame chosen as basis for the trajectory propagation.
$attitude\_reference\_frame = J2000$	Reference frame chosen as basis for the attitude propagation.
$ID\_bodies = [10\ 399\ 301\ 1\ 2\ 4\ 5\ 6\ 7\ 8\ 9]$	SPICE identification numbers of the bodies[7] involved in the n-body problem.
$ephemeris\_correction = NONE$	Possibility to account for the aberration corrections to apply to the state of the target body to account for one-way light time and stellar aberration; in this thesis it has not been considered.
$ID\_external\_bodies = [10\ 301\ 1\ 2\ 4\ 5\ 6\ 7\ 8\ 9]$	SPICE identification numbers of the external bodies with respect to the central one ($trajectory\_origin\_ID$), responsible for non null accelerations of the selected reference frame.

Table 3.3: Data from `scenario.txt`.

Input data	Description
$position = 10^5[-3.1277\ 2.4759\ 1.7273]'\ [km]$	Initial position with the conditions in Table 3.3.
$velocity = [-0.9773\ -0.7879\ -0.5212]'\ [km/s]$	Initial velocity with the conditions in Table 3.3.
$stm = I_{6\times6}$	STM initial condition.
$angular\_velocity = [0\ 0\ 0]'\ [rad/s]$	Initial angular velocity.
$quaternion = [0.6926\ -0.5827\ 0.4078\ -0.1201]'$	Initial quaternion.
$mass = 24\ [kg]$	Initial spacecraft mass, from [5].

Table 3.4: Data from `ICs.txt`.

[7] In order from left to right: Sun, Earth, Moon, Mercury, Venus, Mars, Jupiter, Saturn, Uranus, Neptune, Pluto.

Input data	Description
$G = 6.6741 \times 10^{-11}$ $[\frac{m^3}{kgs^2}]$	Gravitational constant.
$P_{SRP1AU} = 4.5732 \times 10^{-6}$ $[Pa]$	Solar pressure at 1 AU.
$AU = 1.495978706136889 \times 10^{8}$ $[km]$	Astronomical unit.
$obj\_ID = 10$ $obj\_name = Sun$ $mass = 1.988475415966536 \times 10^{30}$ $[kg]$ $radius = 696340$ $[km]$	Sun constants
$obj\_ID = 399$ $obj\_name = Earth$ $mass = 5.972365261370795 \times 10^{24}$ $[kg]$ $radius = 6371$ $[km]$	Earth constants
$obj\_ID = 301$ $obj\_name = Moon$ $mass = 7.346031312426276 \times 10^{22}$ $[kg]$ $radius = 1737.4$ $[km]$	Moon constants
$obj\_ID = 1$ $obj\_name = Mercury$ $mass = 3.301096181046679 \times 10^{23}$ $[kg]$ $radius = 2439.7$ $[km]$	Mercury constants
$obj\_ID = 2$ $obj\_name = Venus$ $mass = 4.867466257521636 \times 10^{24}$ $[kg]$ $radius = 6051.8$ $[km]$	Venus constants
$obj\_ID = 4$ $obj\_name = Mars$ $mass = 6.417120444166090 \times 10^{23}$ $[kg]$ $radius = 3389.5$ $[km]$	Mars constants
$obj\_ID = 5$ $obj\_name = Jupiter$ $mass = 1.898580250761156 \times 10^{27}$ $[kg]$ $radius = 69911$ $[km]$	Jupiter constants
$obj\_ID = 6$ $obj\_name = Saturn$ $mass = 5.684766319852324 \times 10^{26}$ $[kg]$ $radius = 58232$ $[km]$	Saturn constants

$obj\_ID = 7$ $obj\_name = Uranus$ $mass = 8.682168328818365 \times 10^{25}\ [kg]$ $radius = 25362\ [km]$	Uranus constants
$obj\_ID = 8$ $obj\_name = Neptune$ $mass = 1.024339999008106 \times 10^{26}\ [kg]$ $radius = 24622\ [km]$	Neptune constants
$obj\_ID = 9$ $obj\_name = Pluto$ $mass = 1.463872174142355 \times 10^{22}\ [kg]$ $radius = 1188\ [km]$	Pluto constants

Table 3.5: Data from `constants.txt`.

Input data	Description
$DT1 = 1000\ [s]$	Time before the firing which determines the start of the thrusters alignment.
$DT2 = 500\ [s]$	Time after the firing which determines the recover of the nominal attitude.
$point\_err\_margin = 1\ [deg]$	Allowable error margin for the thrusters orientation.
$w\_dot\_margin = 10^{-5}\ [rad/s]$	Allowable error margin for the spacecraft rotation.
$Kpx = Kpy = Kpz = 1.8 \times 10^{-4}$	Proportional control gains.
$Kdx = Kdy = Kdz = 7 \times 10^{-3}$	Derivative control gains.
$rate\_Mc = 0.6\ [s]$	Output sample time for actuators command.
$Bearingnoise\_mean = 0\ [Nm]$	Noise mean for RWs bearings.
$Bearingnoise\_variance = 10^{-15}\ [Nm]$	Noise variance for RWs bearings.
$saturation\_lim = 2 \times 10^{-3}\ [Nm]$	RWs saturation limit.
$A\_ = \begin{bmatrix} -a & a & a & -a \\ -a & -a & a & a \\ a & a & a & a \end{bmatrix} \quad a = \frac{1}{\sqrt{3}}$	RWs configuration matrix.

$A^* = \begin{bmatrix} -b & -b & b \\ b & -b & b \\ -b & b & b \end{bmatrix} \quad b = \frac{\sqrt{3}}{4}$	Pseudo-transponse of $A\_$.
$I = 4 \times 10^{-5}\ [kgm^2]$	RWs inertia, computed considering a mass of 0.2 kg and a radius of 0.02 m^2.
$rate\_simulation = 0.15\ [s]$	Output sample time for thrusters command.
$Isp = 204\ [s]$	Specific impulse of the thrusters selected.
$g0 = 9.8066\ [m/s^2]$	Gravity acceleration.
$Tmax = 0.245 \times 2\ [N]$	Thrust provided by the two thrusters.
$th\_range = 0.3 \times 2\ [N]$	Maximum thrust provided by the selected thrusters [5].
$th\_rate = 0.1\ [N/s]$	Rising and falling slew rates for thrust.
$th\_noisemean = 0\ [N]$	Noise mean for thrusters.
$th\_noisevariance = 10^{-9}\ [N]$	Noise variance for thrusters.

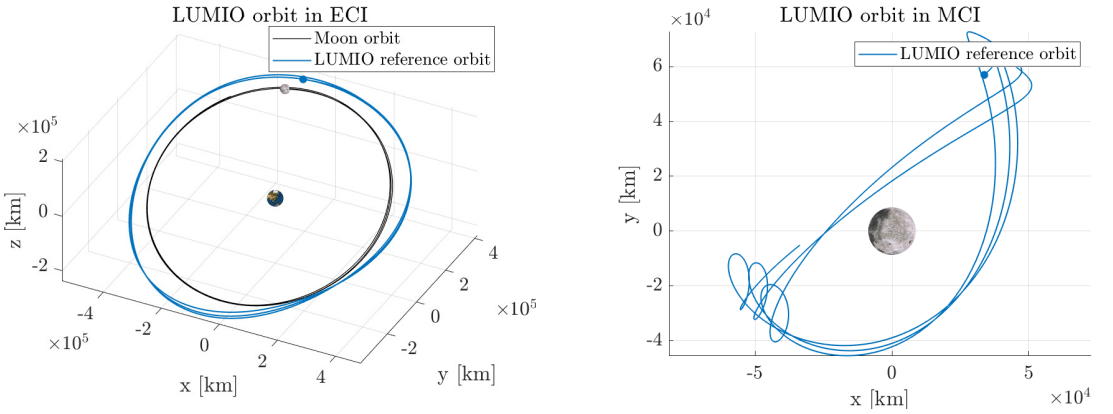
Table 3.6: Data from `GNC.txt`.

Once the parameters reported in the tables above have been set, the CUBORG tool is ready to be used. As mentioned, the selection of some of the data reported in the tables, especially for those of `GNC.txt`, will be described in the following chapters, to provide a detailed explanation behind their value.

3.4.1. Ideal and simulated orbit

The most basic functionality of the tool is the propagation of the orbit of the spacecraft. To validate the propagation, the result provided by CUBORG is compared to the ideal orbit, saved in the LUMIO ephemeris file `halo_Cj3p09.bsp`. The ideal orbit is represented in Fig. 3.6a and Fig. 3.6b, in the ECI and MCI inertial frames respectively, featuring the Earth and the Moon in the origin, in J2000. Earth and Moon are drawn magnified, to make them more visible. The orbit are drawn for a period of 80 days.

Using CUBORG with the parameters reported in the tables of Sec. 3.4, the plots of Fig. 3.7a and Fig. 3.7b are obtained. To be precise, the settings reported in Table 3.2 have been slightly modified to obtain those results. Indeed, the values in the table have



(a) Orbit in Earth centred inertial frame.

(b) Orbit in Moon centred inertial frame.

Figure 3.6: LUMIO reference orbit, in Earth and Moon centred inertial frames.

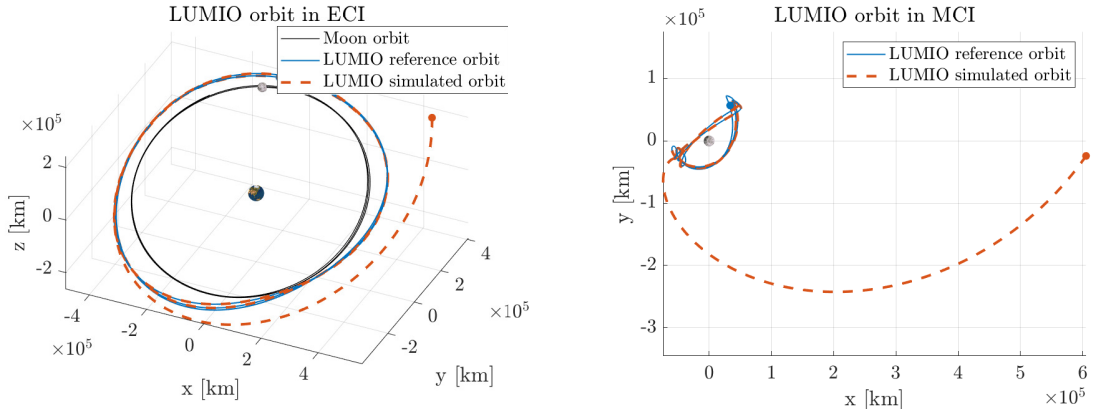
been selected as baseline to perform even the the AOCS simulations; whenever it is just enough to propagate the trajectory, the integration settings in Table 3.7 can be employed, resulting in a more fast and precise propagation.

SolverType	Solver	RelTol	AbsTol	ephemeris_delta_t
Variable-step	*ode113*	2.5×10^{-11}	2.5×10^{-11}	15 s

Table 3.7: CUBORG integration settings to perform a fast orbit propagation.

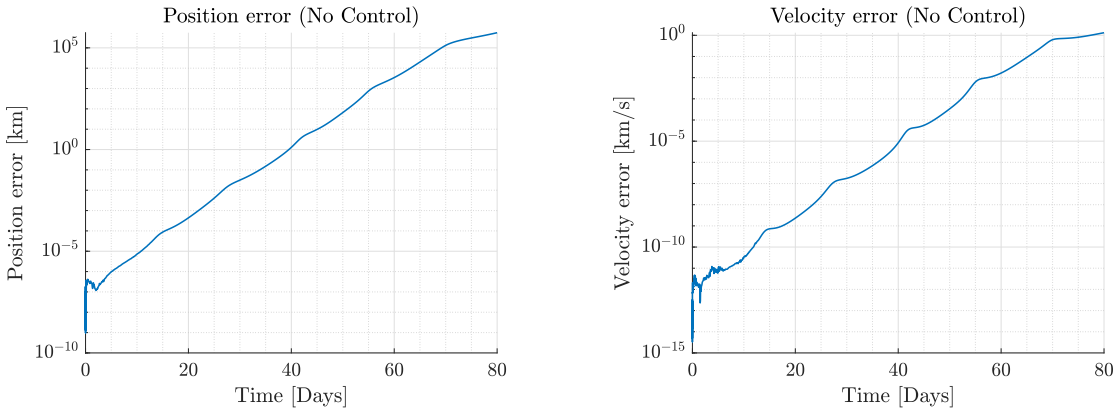
In particular, the *ephemeris_ delta_ t* has been set to such a low value to reduce the introduction of errors coming from ephemeris discretization. It has been proved that both *ephemeris_ delta_ t* and the selection of a different solver determine a drastic worsening in the propagation errors, which can increase even by 1 or 2 orders of magnitude.

In Fig. 3.8a and Fig. 3.8b the position and velocity deviations are plotted. It can be noticed that the nominal and simulated paths slightly move away one to each other, reaching a distance of thousands of kilometres after a couple of months. This is mainly due to numerical integration problems, amplified by the nature of the chaotic dynamics: even the smallest source of error is likely to increase during the time, such to lead to a complete divergence from the nominal path at a certain future epoch. This phenomenon shows well the problem of maintaining the reference path during a mission in a halo orbit. Indeed, it is not possible to perfectly model or completely predict all the sources of disturbances, hence the only solution is to develop a station keeping system able to continuously correct the trajectory.



(a) Orbit in Earth centred inertial frame. (b) Orbit in Moon centred inertial frame.

Figure 3.7: LUMIO simulated orbit with no control applied.



(a) Simulation position error. (b) Simulation velocity error.

Figure 3.8: Position and velocity errors in the uncontrolled orbit.

3.5. Implementation choices

In order to increase the computational efficiency and to improve the overall tool performances, some development choices have been made during the progress of the work, hereafter discussed. As mentioned in Sec. 1.4, the purpose of the thesis shall be to model and test a station keeping algorithm suitable for LUMIO. To achieve the desired solution, some trails have been performed before arriving to the current version.

At first, the possibility of developing an on-board guidance algorithm in Simulink was considered. This wuold have allowed to compute the S/K manoeuvres within the closed-loop simulation. Anyway, considering that for LUMIO the guidance is performed entirely on ground, the algorithm can be developed off-line in MATLAB, out of the simulation

loop, saving time and reducing the computational effort.

Hence, a separate simulation is carried out in MATLAB to compute the $\Delta\mathbf{v_{SK}}$ for all the manoeuvres, and then use these as input in the Simulink tool. To this aim, the function `orbit_propagation.m` has been created to represent a MATLAB version of the CUBORG propagator in Simulink. It has been built to allow a fast orbit propagation each time it is required, without the need to run the whole CUBORG. It includes the *STM* differential equation in Eq. (3.12) and the state integrator. It consists in a 3 DOF propagator model: this speeds up the integration and returns just the useful information.

In particular, the *STM* differential equation, added to the state propagator, undergoes the rule:

$$\begin{cases} \dot{\Phi}(t,t_0) = J\Phi(t,t_0) \\ \Phi(t_0,t_0) = I_{6\times 6} \end{cases} \tag{3.12}$$

with J being the Jacobian of the linearized dynamics:

$$J = \begin{bmatrix} 0_{3\times 3} & I_{3\times 3} \\ \frac{d\mathbf{a_g}}{d\mathbf{x}} + \frac{d\mathbf{a_{srp}}}{d\mathbf{x}} & 0_{3\times 3} \end{bmatrix} \tag{3.13}$$

then, considering the equations which give rise to the total force acting on the spacecraft, Eq. (3.6) and Eq. (3.7), the derivative of the corresponding accelerations read:

$$\frac{d\mathbf{a_g}}{d\mathbf{x}} = -\sum_{k=1}^{N} \frac{\mu_k}{|\mathbf{x}-\mathbf{r_k}|^5}(I|\mathbf{x}-\mathbf{r_k}|^2 - 3(\mathbf{x}-\mathbf{r_k})(\mathbf{x}-\mathbf{r_k})^T) \tag{3.14}$$

$$\frac{d\mathbf{a_{srp}}}{d\mathbf{x}} = \frac{P_{SRP1AU}AU^2Ac_R}{|\mathbf{x}-\mathbf{r_{sun}}|^5}(I|\mathbf{x}-\mathbf{r_{sun}}|^2 - 3(\mathbf{x}-\mathbf{r_{sun}})(\mathbf{x}-\mathbf{r_{sun}})^T)\frac{10^{-3}}{m_{sc}} \tag{3.15}$$

The computation of the *STM* following this process has two main problems: first, it needs to be propagated each time it is required in the algorithm, second, the propagation error accumulates over the time, becoming too large for long simulations and hence leading to a complete wrong result.

The reason behind the wrong results provided by the propagated *STM* deserves a dedicated explanation. This issue in fact, has represented one of the main obstacles encountered during the thesis, and it still determines a source of inaccuracy in the algorithm. The main problem, noticed during the validation and utilisation of the *STM*, was related to the high propagation error derived from its used. Indeed, as much the propagation time grew, as much the quantities propagated with the *STM* resulted to differentiate from the correct ones. This, from one side, is absolutely normal: the *STM* is obtained via lin-

earization of the n-body dynamics, hence it is intrinsically affected by an error. From the other side, however, the quantities propagated with it, should still provide a reasonable result, not a completely no sense solution. This means another source of error is present other than that derived from the linearization. This second source of error accumulates as the propagation goes on, and it is related to many factors: from the integration scheme and related settings, from little offsets in the input data and perturbations model, with respect to those used for the nominal orbit, or similar others. Indeed, the original *STM* corresponding to the reference orbit saved in the ephemeris `halo_Cj3p09.bsp`, was not available for this thesis. This means not having the exact same settings in the propagator as those used originally. As [28] says about the use of the *STM*: inaccurate derivatives generate suboptimal steps for the search, yielding a higher number of iterations to solve the problem or leading the algorithm to diverge. These problems are accentuated when the trajectory is highly nonlinear, as is the case for multiple-body orbits or multirevolution spiral orbits [28]. Having clarified how much the precision plays a fundamental role in this context, the goal for future developments is for sure the refinement of *STM* computation, or better the use, if possible, of the original *STM*. This point will be further discussed in Chap. 8.

A possibility to solve or at least reduce the problem of error accumulation during the propagation, is easily accomplished by shorter propagations, exploiting the rule [29]:

$$\Phi(t_2, t_0) = \Phi(t_2, t_1)\Phi(t_1, t_0) \tag{3.16}$$

therefore, it is sufficient to create a set of many sub-*STM* computed in short periods of time, as $\Phi(t_2, t_1)$ or $\Phi(t_1, t_0)$, to retrieve the desired one in a longer Δt, as $\Phi(t_2, t_0)$. In detail, this has been obtained following Alg. 1.

The start and final dates, t_{start} and t_{end}, span more than 1 year, covering all the operative mission duration.

In the algorithm, the variable S indicates that each day has been partitioned in 2 steps, of half day each. Therefore, for a total of n days of time span, a total of $2n$ *STM* will be saved. This step size has been selected being sufficiently short to prevent the propagation error to grow excessively, but long enough to avoid the computational time being too large. Having saved the state transition matrices with such a time step, in the cases in which an intermediate *STM* is needed, the function `STM_extract.m` allows to obtain it via interpolation of adjacent *STMs*. This function also multiplies the sub-*STM* in the Δt to return the desired one, following the rule in Eq. (3.16).

Algorithm 1 *STM* history computation.

function $[\Phi(t_{end}, t_{start})] = STM\_history(ICs, Others)$
$t_{start} =$ 2024-MAR-10-14:00:54
$t_{end} =$ 2025-MAY-20-10:59:59
$d = 3600 \times 24$ ▷ seconds in a day
$S = 2$ ▷ steps in a day
$days = (t_{end} - t_{start})/d$
for $i \leftarrow 1 : round(S \times days)$ **do**
  $t_1 \leftarrow t_{start}$
  $t_2 \leftarrow t_{start} + (1/S)d$
  $get\ ICs\ at\ t_1,\ t_2\ from\ SPICE$
  $[\mathbf{t}, \mathbf{x}] = orbit\_propagation(ICs, Others)$
  $\Phi(:, :, i) = reshape(\mathbf{x}(end, 7:42), 6, 6);$
  $t_1 \leftarrow t_2$ ▷ update time start
end for

With Alg. 1, a sort of history of the *STM* in a large period of time, extended during the 1 year operative mission phase, is obtained. This allows to extract the required *STM* in the desired Δt without performing the propagation of Eq. (3.12), allowing to have the desired *STM* as an instantaneous output.

In order to have an idea of the error associated to the propagation with the *STM*, the propagation error can be analysed. It is intended as the difference between a perturbation in the state propagated with the *STM*, and that obtained via integration of the dynamics. It is computed as follows:

$$from: \quad \mathbf{x}(t_0) \rightarrow \mathbf{x}(t) \tag{3.17}$$

$$from: \quad \mathbf{x}(t_0) + \delta\mathbf{x}(t_0) \rightarrow \mathbf{x_e}(t) \tag{3.18}$$

so the perturbation at time t is given by:

$$\delta\mathbf{x_{prop}}(t) = \mathbf{x_e}(t) - \mathbf{x}(t) \tag{3.19}$$

while the one obtained with the *STM* is:

$$\delta\mathbf{x_{STM}}(t) = \Phi(t, t_0)\delta\mathbf{x}(t_0) \tag{3.20}$$

therefore, the magnitude of the position error is provided by:

$$\delta\mathbf{x_{error}}(t) = ||\delta\mathbf{x_{STM}}(1:3) - \delta\mathbf{x_{prop}}(1:3)|| \tag{3.21}$$

Fig. 3.9 shows the error derived using the history of the state transition matrix, following Eq. (3.21).

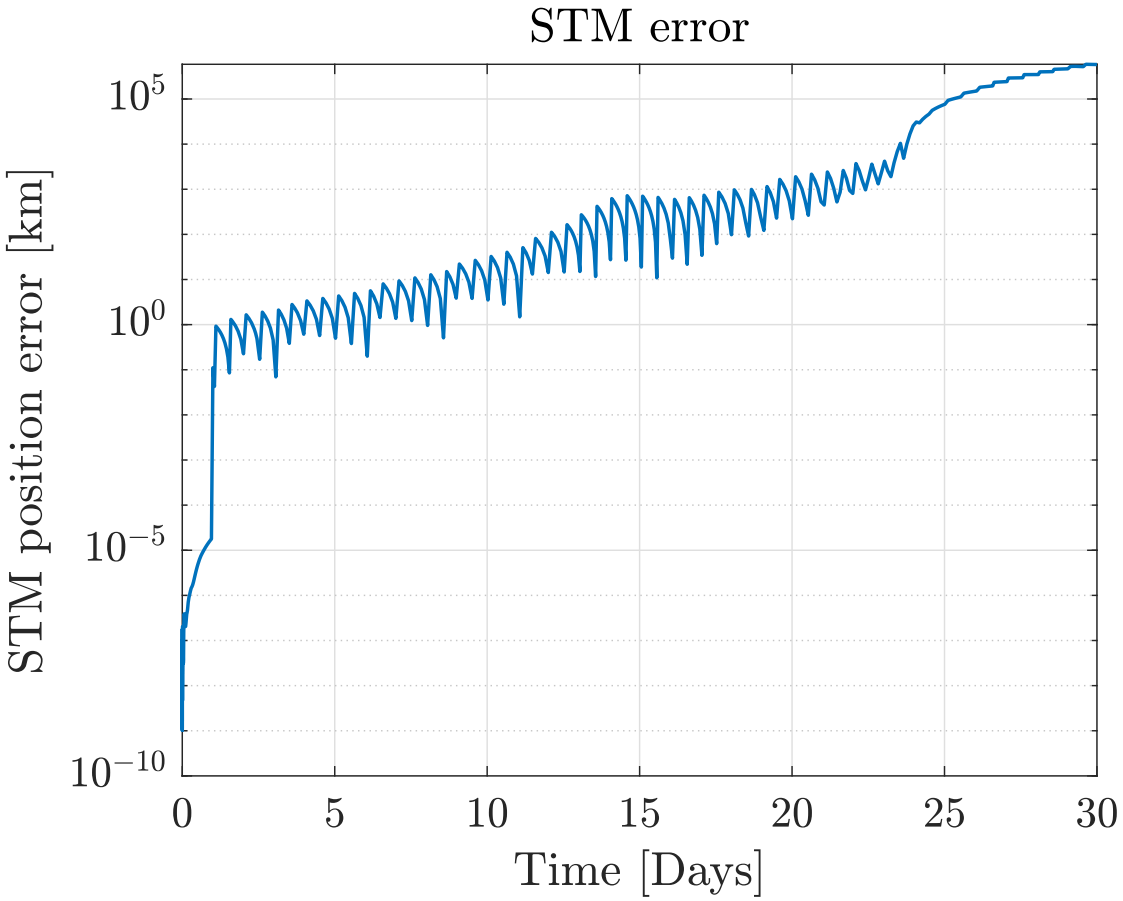


Figure 3.9: Position error with the saved *STM*.

The steps in the plot have a duration of $1/S$ days and are due to the reduction of the error each time a new saved *STM* is used.

Considering the two methods to derive the *STM* discussed, hence the direct propagation and the extraction from the saved history, Fig. 3.10 is derived. It shows the absolute value of the difference between the errors related to the two methods. The difference is sufficiently large to determine that one method is more accurate than the other.

To conclude, the best option for a future development still remains, as mentioned, the use of the original state transition matrix associated to the nominal trajectory.

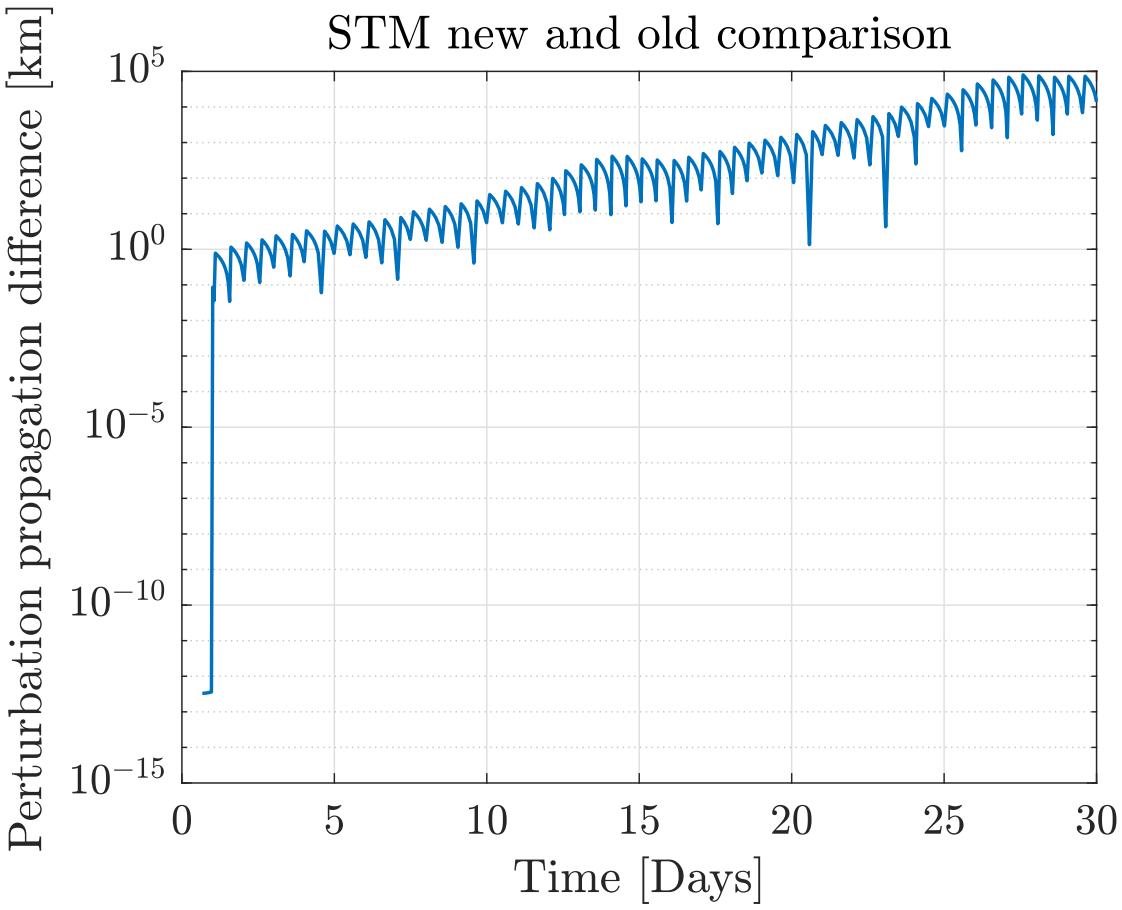


Figure 3.10: Position error difference with the two STM approaches.

4 | Guidance, Navigation & Control theory

"Guidance requires action,
but it does not guarantee safety."

- Caroline Myss

In this chapter the core methodology of the thesis will be introduced, from a theoretical point of view, in the context of LUMIO. For each of the three macro-categories of Guidance, Navigation and Control, the mathematics behind the models will be described in association with the choices made in the implementation strategy, which will be further analysed in Chap. 5 and Chap. 6.

4.1. Guidance: station keeping algorithm

The *guidance* is intended as the determination of the trajectory the spacecraft is required to follow. Once the nominal orbit is determined, it is desired to maintain the spacecraft within some regions close the reference path. Indeed, non-modelled perturbations and simulation errors will cause the departure of the spacecraft from the nominal trajectory, even amplified for the unstable nature of the libration point orbits.

The guidance problem consists in determining the magnitude, direction and timing of the corrective manoeuvres to execute during the orbit. An optimal station keeping problem is faced, therefore the $\Delta\mathbf{v_{SK}}$ budget is minimised.

In particular the algorithm is developed through the Target Points Method (*TPM*), first introduced by Dwidedi in the case of deterministic optimal manoeuvre strategy for multi-target missions [30], then adapted to the problem of LPOs Station-Keeping by Howell and Pernicka [31], and finally used for JAXA's EQUULEUS mission analysis [13].

First of all some initial choices have been made, in accordance with the strategy validated

in [1] after the conclusion of the Phase A:

- The station keeping is planned to be performed only during the Nav&Eng cycles, hence at alternated orbits.
- During each Nav&Eng cycle, 3 correction manoeuvres are planned: the first at the beginning, the second in the middle of the orbit, and the last one at the end.
- Before the manoeuvre execution, a cut-off time time Δt_c has to pass after the end of the orbit determination campaign. This gives enough time to compute the $\Delta \mathbf{v_{SK}}$, to prepare and perform the manoeuvre.
- The number of selected target points in the *TPM* is 2.
- The orbit insertion is assumed to present an initial error.
- Each OD campaign is assumed to be completed with a certain error.
- Each manoeuvre is assumed to be performed with a margin of error.

Some of the listed assumptions or parameters are likely to be slightly modified in the development of the S/K algorithm, however they represent an initial valid option, thanks to the analysis conducted up to Phase A.

Fig. 4.1 shows the concept of operations, where each orbit is assumed to have a fixed duration of 14 days, for sake of simplicity. The three main phases are:

- *OI stabilisation*: it is the first phase in the halo orbit, it is practically equal to an engineering orbit in terms of number of manoeuvres and duration.
- *Nav&Eng cycle*: during the duration of this orbit, 3 station keeping manoeuvres are executed. It is selected as the first phase after the orbit insertion, to increase the control in the firsts orbits which are the most subjected to errors.
- *Science cycle*: it is the orbit during which the science is performed, hence it is supposed to be executed without any manoeuvre.

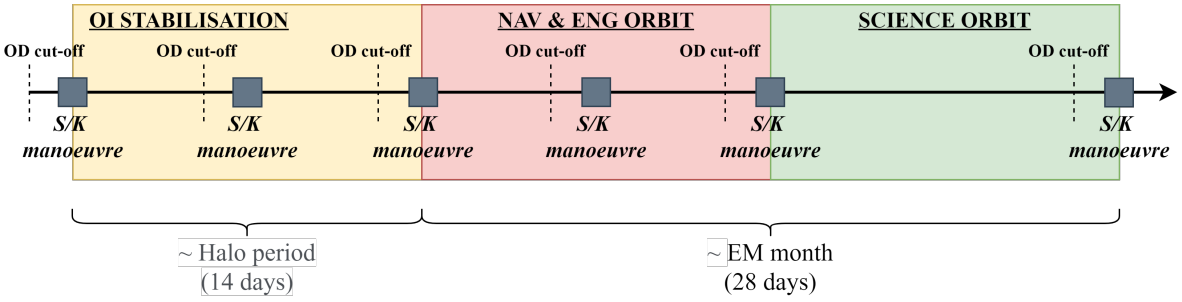


Figure 4.1: Station keeping events timeline.

The schedule of the Nav&Eng and Science cycles, obtained through the mission analysis study during Phase A [1], is reported in Table 4.1. This, in particular, has been selected as baseline after a detailed analysis of other possible solutions, as cycles of fixed duration of 14 days. The schedule reported has demonstrated to be characterised by the smallest performance degradation, although presenting some losses in terms of scientific outcome [1].

N°_{cycles}	**Cycle**	**Start**	**End**	**Duration** $[day]$
1	OI stab.	2024-MAR-21-12:00:00	2024-APR-03-14:00:54	13.1
2	Nav&Eng	2024-APR-03-14:00:54	2024-APR-16-19:30:52	13.2
3	Science	2024-APR-16-19:30:52	2024-MAY-01-16:34:03	14.9
4	Nav&Eng	2024-MAY-01-16:34:03	2024-MAY-15-06:56:28	13.6
5	Science	2024-MAY-15-06:56:28	2024-MAY-29-20:02:06	14.5
6	Nav&Eng	2024-MAY-29-20:02:06	2024-JUN-12-06:19:46	13.4
7	Science	2024-JUN-12-06:19:46	2024-JUN-26-17:58:10	14.5
8	Nav&Eng	2024-JUN-26-17:58:10	2024-JUL-10-19:20:31	14.1
9	Science	2024-JUL-10-19:20:31	2024-JUL-25-03:28:07	14.3
10	Nav&Eng	2024-JUL-25-03:28:07	2024-AUG-08-05:06:13	14.1
11	Science	2024-AUG-08-05:06:13	2024-AUG-22-07:18:30	14.1
12	Nav&Eng	2024-AUG-22-07:18:30	2024-SEP-06-08:01:30	15
13	Science	2024-SEP-06-08:01:30	2024-SEP-29-04:16:02	22.8
14	Nav&Eng	2024-SEP-29-04:16:02	2024-OCT-14-14:56:42	15.4
15	Science	2024-OCT-14-14:56:42	2024-OCT-28-11:39:32	13.9
16	Nav&Eng	2024-OCT-28-11:39:32	2024-NOV-12-09:16:35	14.9
17	Science	2024-NOV-12-09:16:35	2024-NOV-25-21:15:02	13.5
18	Nav&Eng	2024-NOV-25-21:15:02	2024-DEC-10-07:51:47	14.4
19	Science	2024-DEC-10-07:51:47	2024-DEC-24-04:03:47	13.8
20	Nav&Eng	2024-DEC-24-04:03:47	2025-JAN-07-14:35:50	14.4
21	Science	2025-JAN-07-14:35:50	2025-JAN-21-14:45:13	14.0
22	Nav&Eng	2025-JAN-21-14:45:13	2025-FEB-04-10:57:04	13.8
23	Science	2025-FEB-04-10:57:04	2025-FEB-18-17:14:57	14.3
24	Nav&Eng	2025-FEB-18-17:14:57	2025-MAR-04-15:22:31	13.9
25	Science	2025-MAR-04-15:22:31	2025-MAR-19-09:11:46	14.7
26	Nav&Eng	2025-MAR-19-09:11:46	2025-MAR-21-10:59:59	2.1

Table 4.1: Cycles during the operative mission lifetime [1].

4.1.1. Manoeuvre computation

As anticipated, the optimal station keeping problem is formulated with the Target Points Method. It provides the optimal $\Delta\mathbf{v_{SK}}$ as a solution of a LQR problem that minimises a weighted cost function. The cost function is defined in terms of a corrective manoeuvre as well as position and velocity deviations from the nominal orbit, at a number of specified future times t_i. These identify the so called *target points*, downstream of the manoeuvre [32].

The cost function reads:

$$J_{S/K} = \Delta\mathbf{v_{SK}}^T Q \Delta\mathbf{v_{SK}} + \sum_{i=1}^{N_{tp}} \mathbf{d_i}^T R_i \mathbf{d_i} \tag{4.1}$$

where Q and R_i are the weighting matrices associated to the cost of the manoeuvre and to the position deviation $\mathbf{d_i}$, respectively. $\mathbf{d_i}$ in particular can be predicted making use of the state transition matrix related to the dynamics, Φ:

$$\mathbf{d_i} = \Phi_{rr}(t_i, t_c)\delta\mathbf{r_c} + \Phi_{rv}(t_i, t_c)\delta\mathbf{v_c} + \Phi_{rv}(t_i, t_v)\Delta\mathbf{v_{SK}} \tag{4.2}$$

where $\delta\mathbf{r_c}$ and $\delta\mathbf{v_c}$ represent the position and velocity deviations at the cut-off time t_c, t_v indicates the manoeuvre epoch, t_i the target points epochs and Φ_{rr} and Φ_{rv} are the sub-matrices of the STM:

$$\Phi = \begin{bmatrix} \Phi_{rr} & \Phi_{rv} \\ \Phi_{vr} & \Phi_{vv} \end{bmatrix} \tag{4.3}$$

The solution of the minimisation problem yields the analytic expression of the $\Delta\mathbf{v_{SK}}$

$$\Delta\mathbf{v_{SK}} = A \sum_{i=1}^{N_{tp}} (\alpha_i \delta\mathbf{r_c} + \beta_i \delta\mathbf{v_c}) \tag{4.4}$$

where:

$$A = -\left[(Q^T + Q) + \sum_{i=1}^{N_{tp}} \Phi_{rv}^T(t_i, t_v)(R_i^T + R_i)\Phi_{rv}(t_i, t_v) \right]^{-1} \tag{4.5}$$

$$\alpha_i = \Phi_{rv}^T(t_i, t_v)(R_i^T + R_i)\Phi_{rr}(t_i, t_c) \tag{4.6}$$

$$\beta_i = \Phi_{rv}^T(t_i, t_v)(R_i^T + R_i)\Phi_{rv}(t_i, t_c) \tag{4.7}$$

At this point, there exist two main approaches to this problem:

- Using **dimensional quantities**: the variables are maintained with their units, hence, being $\delta\mathbf{r_c}$ in km, $\delta\mathbf{v_c}$ in km/s and the result $\Delta\mathbf{v_{SK}}$ in km/s, the units for the weighting matrices are derived consequently. In particular, the components of the state transition matrix have the following units of measure:

$$\Phi_{rr} : [-]; \quad \Phi_{rv} : [s];$$

this implies that $R_i : [1/s^2]$ and Q is dimensionless, leading to: $\alpha_i : [1/s]$, $\beta_i : [-]$ and $A : [-]$.

The dimensional approach is usually characterised by higher values of the weighting matrices, necessary to counteract the order of magnitude of the other quantities. Typical values [31, 32] are reported in Table 4.2.

Weighting matrix	Value
Q $[-]$	$10^{13} \times I_{3\times3}$
R_1 $[1/s^2]$	$10^{-1} \times I_{3\times3}$
R_2 $[1/s^2]$	$10^{-2} \times I_{3\times3}$

Table 4.2: Typical values of Q and R in the dimensional case [31, 32].

- Using **dimensionless quantities**: the variables are divided by reference values to obtain a dimensionless $\Delta\mathbf{v_{SK}}$, which is finally reconverted in km/s. In this case, the reference quantities selected are:

$$\mu_{ref} = 4.9028 \times 10^3 \left[\frac{km^3}{s^2}\right] \tag{4.8}$$

$$L_{ref} = 65000 \; [km] \tag{4.9}$$

$$T_{ref} = \sqrt{\frac{L_{ref}^3}{\mu_{ref}}} \; [s] \tag{4.10}$$

$$V_{ref} = \frac{L_{ref}}{T_{ref}} \left[\frac{km}{s}\right] \tag{4.11}$$

where μ_{ref} is the reference gravitational parameter of the Moon, L_{ref} is the Moon to L2 reference distance, T_{ref} is the reference time and V_{ref} is the reference velocity.

Dividing each quantity in Eq. (4.4) for the respective reference value, and using the weighting matrices in Table 4.3, the optimal $\Delta\mathbf{v_{SK}}$ formula is modified as:

$$V_{ref}\Delta\mathbf{v_{SK}} = A\sum_{i=1}^{N_{tp}}\left(\alpha_i\frac{\delta\mathbf{r_c}}{L_{ref}} + \beta_i\frac{\delta\mathbf{v_c}}{V_{ref}}\right) \tag{4.12}$$

$$A = -\left[(Q^T+Q) + \sum_{i=1}^{N_{tp}}\frac{\Phi_{rv}^T(t_i,t_v)}{T_{ref}}(R_i^T+R_i)\frac{\Phi_{rv}(t_i,t_v)}{T_{ref}}\right]^{-1} \tag{4.13}$$

$$\alpha_i = \frac{\Phi_{rv}^T(t_i,t_v)}{T_{ref}}(R_i^T+R_i)\Phi_{rr}(t_i,t_c) \tag{4.14}$$

$$\beta_i = \frac{\Phi_{rv}^T(t_i,t_v)}{T_{ref}}(R_i^T+R_i)\frac{\Phi_{rv}(t_i,t_v)}{T_{ref}} \tag{4.15}$$

which is characterised by a reduced order of magnitude of the quantities, as it can be noticed from the values of the weighting matrices.

Weighting matrix	Value
Q $[-]$	$10^{-1}\times I_{3\times3}$
R_1 $[-]$	$10^{-2}\times I_{3\times3}$
R_2 $[-]$	$10^{-2}\times I_{3\times3}$

Table 4.3: Typical values of Q and R in the dimensionless case [1].

As for the duration of the cut-off time and the Δt of the target points, the first has been initially chosen of 12 hours: this time is sufficiently short to prevent the spacecraft state knowledge from growing excessively, and long enough to schedule manoeuvre execution operations [1]. At the end of Phase A, however, its value has been modified to 2 days [4]; in Chap. 5 and Chap. 7 the effect of both values in the S/K will be investigated. The target points instead, are located 35 and 42 days after orbit insertion and any subsequent station keeping manoeuvres; this ensures approximately one month of operations in case of manoeuvre execution failure [1]. To conclude, considering the values in Table 4.3, having the eigenspectrum of Q a larger magnitude than that R_i means the optimisation weighs the deviation with respect to reference position more than the $\Delta\mathbf{v_{SK}}$ cost.

Further information on the settings used and their influence in the computation of the $\Delta\mathbf{v_{SK}}$ will be analysed in Chap. 5.

4.2. Navigation: attitude guidance

With *navigation*, in control theory, it is intended the acquisition of the spacecraft translational or rotational state, by means of sensors or radiometric signals coming from ground, as in case of LUMIO. In the context of this thesis, the navigation process is not considered, but replaced with the model of attitude guidance. This refers to the definition of

the attitude LUMIO must take in the different phases of the mission. Therefore, once the spacecraft state will be provided by ground signals, the internal software will construct the current attitude of the spacecraft, comparing it to the desired one to retrieve the control action to apply.

LUMIO is equipped with 6 MAUS CubeSat Sun sensors, 2 AURIGA star trackers and 1 ISISpace IMU [5].

The attitude of LUMIO, during the operational lifetime in the halo orbit, presents two main configurations:

- **Nominal attitude**: it is the attitude maintained during all the Science cycles and for the largest part of the Nav&Eng ones. It consists in the pointing of the LUMIO-cam in the Moon direction, the tracking of the Sun with the solar panels, by means of a SADA system, and the antenna pointing towards the Earth. This attitude allows to perform the science operations while remaining in communication with the Earth and do not risk to have an energy deficiency.
- **Manoeuvre attitude**: it is the attitude reached to perform the station keeping manoeuvres, which is characterised by the orientation of the thrusters in the $\Delta\mathbf{v_{SK}}$ direction, for the amount of time planned. This operation, of course, foresees a time window in which the rotation is performed and the thrusters pointing accuracy is met.

4.2.1. Nominal attitude

As anticipated, this represents the attitude LUMIO is going to maintain during most of the mission.

Fig. 4.2 shows the LUMIO coordinate frame, also named *body* frame.

The x-axis corresponds to the direction of the LUMIO-cam and the two main thrusters used to perform the correction manoeuvres, the y-axis is aligned with the solar panels axis of symmetry and the z-axis completes the tern. For this reason, to achieve the pointing or tracking conditions for the nominal attitude, the following constraints must be respected:

- **x-axis**: it must be directed towards the Moon direction $\mathbf{R_{bm_N}}$, to fulfil the requirements of the LUMIO-cam.
- **y-axis**: it must be orthogonal to the Sun, $\mathbf{R_{bs_N}}$, and Moon, $\mathbf{R_{bm_N}}$, directions, to allow the Sun tracking by means of the additional degree of freedom offered by the SADA system.

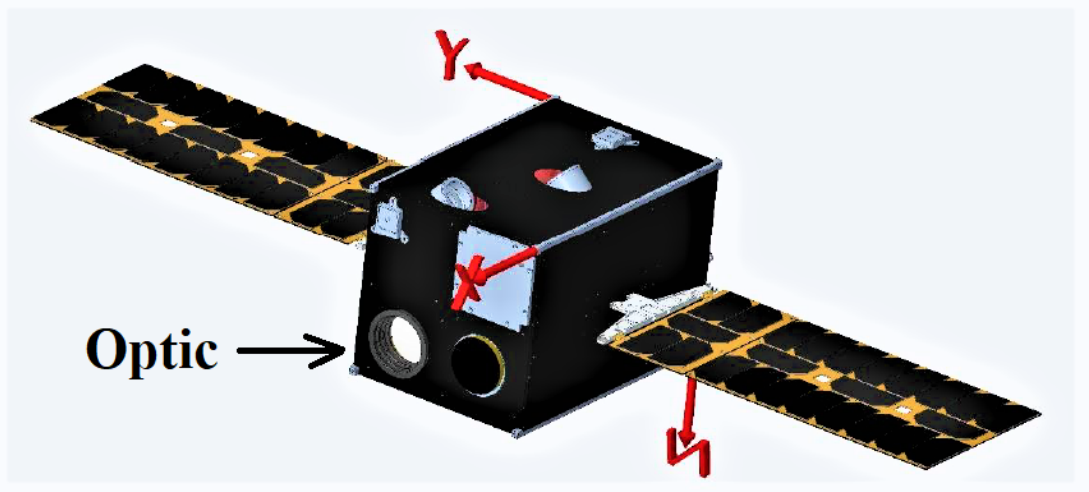


Figure 4.2: LUMIO coordinate frame [5].

- z-**axis**: it must be orthogonal to the x and y axes to complete the tern.

Therefore, the nominal attitude of LUMIO is provided by the matrix A_{nom}:

$$\begin{cases} \hat{\mathbf{x}}_\mathbf{B} = \hat{\mathbf{R}}_{\mathbf{bm_N}} \\ \hat{\mathbf{y}}_\mathbf{B} = \hat{\mathbf{x}}_\mathbf{B} \times \hat{\mathbf{R}}_{\mathbf{bs_N}} \\ \hat{\mathbf{z}}_\mathbf{B} = \hat{\mathbf{x}}_\mathbf{B} \times \hat{\mathbf{y}}_\mathbf{B} \end{cases} \tag{4.16}$$

$$A_{nom} = \begin{bmatrix} \hat{\mathbf{x}}_\mathbf{B} \\ \hat{\mathbf{y}}_\mathbf{B} \\ \hat{\mathbf{z}}_\mathbf{B} \end{bmatrix} \tag{4.17}$$

4.2.2. Manoeuvre attitude

This attitude is reached whenever a manoeuvre is planned to be performed and the thrusters need to be pointed towards the direction of the $\Delta\mathbf{v_{SK}}$. In particular, the nozzles are directed towards the negative x-axis, Fig. 4.3, hence the thrust they apply will be directed along the positive direction. That being said, the only new constraint imposed by this attitude is given by the alignment of the x-axis with the $\Delta\mathbf{v_{SK}}$ of the manoeuvre.

In particular, considering the very short impulses the thrusters must provide for the station keeping, the manoeuvres spreading is not considered in this work. Indeed, keeping in mind the duration of each manoeuvre, its introduction would be irrelevant. This is the reason why it has been imposed a constant pointing towards the $\Delta\mathbf{v_{SK}}$ direction during the

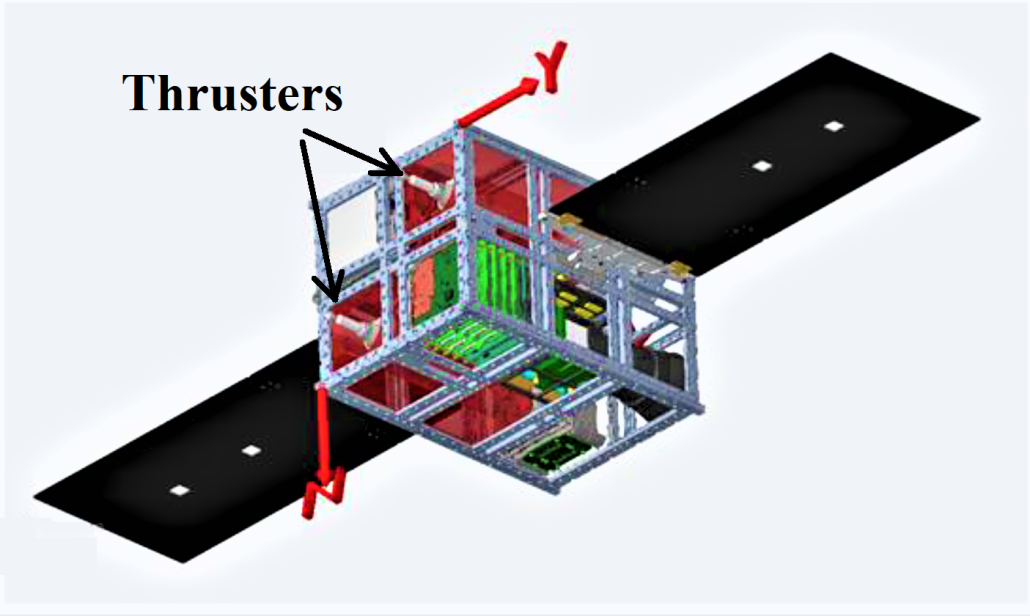


Figure 4.3: Thrusters direction in LUMIO coordinate frame [5].

firing.

Therefore, the manoeuvre attitude is expressed by $A_{\Delta v}$:

$$\begin{cases} \hat{\mathbf{x}}_{\mathbf{B}} = \Delta\hat{\mathbf{v}}_{\mathbf{SK}} \\ \hat{\mathbf{y}}_{\mathbf{B}} = \hat{\mathbf{x}}_{\mathbf{B}} \times \hat{\mathbf{R}}_{\mathbf{bs_N}} \\ \hat{\mathbf{z}}_{\mathbf{B}} = \hat{\mathbf{x}}_{\mathbf{B}} \times \hat{\mathbf{y}}_{\mathbf{B}} \end{cases} \tag{4.18}$$

$$A_{\Delta v} = \begin{bmatrix} \hat{\mathbf{x}}_{\mathbf{B}} \\ \hat{\mathbf{y}}_{\mathbf{B}} \\ \hat{\mathbf{z}}_{\mathbf{B}} \end{bmatrix} \tag{4.19}$$

The navigation process will then consist on the switch between A_{nom} and $A_{\Delta v}$ during the Nav&Eng cycles of the mission, to perform the station keeping manoeuvres.

4.3. Control: actuators command

The term *control*, in the GNC context, is referred to the actions the spacecraft must accomplish, by means of actuators, to follow the desired trajectory.

LUMIO is equipped with 4 reaction wheels ISISpace RW25 SW50 and 2 mono-propellant HPGP 1 N thrusters from Bradford-ECAPS, which are used in the AOCS system. Their properties are listed in Table 4.4 and Table 4.5.

Characteristic	Data
Producer	$ISISpace$
Producer nationality	$The\ Netherlands\ (EU)$
Momentum storage $[mNms]$	30
Maximum torque $[mNm]$	2
Maximum rotation speed $[rpm]$	5000
Mass $[g]$	200
Power nominal $[W]$	0.8
Power peak $[W]$	2.5
Volume $[mm^3]$	68000
Data connection	$I2C, SPI$
Reliability	$1/4\ allowed\ to\ fail$
Flight heritage	Yes
TRL	9

Table 4.4: ISISpace RW25 SW50 data sheet [5].

Characteristic	Data
Producer	$Bradford - ECAPS$
Producer nationality	$The\ Netherlands\ (EU)$
Propellant	$LMP - 103S$
Primary operational mode	$RCS + \Delta V$
Inlet pressure range $[bar]$	$4.5 - 22$
Thrust range $[N]$	$0.25 - 1$
Nozzle expansion ratio	$100 : 1$
Steady state Isp $[s]$	$204 - 231$
Minimum impulse bit $[mNs]$	$\leq 70\ mNs$
Overall length $[mm]$	178
Mass $[kg]$	0.38
Flight heritage	Yes
TRL	9

Table 4.5: HPGP Bradford-ECAPS 1 N thrusters data sheet [5].

Moreover, the presence of a Reaction Control System (RCS) is considered, for wheels desaturation.

4.3.1. Control law

The reaction wheels are activated in specific moments along of the trajectory, when the spacecraft is close to the manoeuvre point. Their role is to rotate the spacecraft, such to pass from the attitude A_{nom} to $A_{\Delta v}$ in a short amount of time, in preparation of the firing. Once the manoeuvre has been executed, the inverse process is performed, passing from $A_{\Delta v}$ to A_{nom}, which is maintained until the following trajectory correction.

The torque the reaction wheels must furnish to perform the rotations is obtained through a quaternion-based control law.

To derive the law, first consider the attitude error matrix A_e, representing the current error with respect to the desired attitude:

$$A_e = A_S A_T^T = \begin{bmatrix} a_{e_{11}} & a_{e_{12}} & a_{e_{13}} \\ a_{e_{21}} & a_{e_{22}} & a_{e_{23}} \\ a_{e_{31}} & a_{e_{32}} & a_{e_{33}} \end{bmatrix} \tag{4.20}$$

where A_S is the current attitude matrix, provided by the sensors, while A_T is the target attitude, hence either A_{nom} or $A_{\Delta v}$. In case of zero error between the two:

$$A_e = I \tag{4.21}$$

The control torque is applied to drive to zero the extra-diagonal components of A_e, to reach a condition close to Eq. (4.21). This means that the torque components will be at least proportional to the extra-diagonal terms. In this thesis a PD controller is employed, hence a derivative term is added to drive to zero the angular velocities:

$$\begin{cases} M_x = Kp_x a_{e_{23}} + Kd_x \omega_x \\ M_y = Kp_y a_{e_{31}} + Kd_y \omega_y \\ M_z = Kp_z a_{e_{12}} + Kd_z \omega_z \end{cases} \tag{4.22}$$

It can be noticed that the control law can be written using the terms on the top or below the diagonal; otherwise, an intermediate situation can be obtained using a sort of mean value:

$$\begin{cases} M_x = Kp_x \frac{a_{e_{23}} - a_{e_{32}}}{2} + Kd_x \omega_x \\ M_y = Kp_y \frac{a_{e_{31}} - a_{e_{13}}}{2} + Kd_y \omega_y \\ M_z = Kp_z \frac{a_{e_{12}} - a_{e_{21}}}{2} + Kd_z \omega_z \end{cases} \tag{4.23}$$

It can be demonstrated that the differences of the error terms in Eq. (4.23) can be substituted by equivalent expressions using the quaternion error $\mathbf{q_e}$, obtained with the estimated quaternion $\mathbf{q_S}$ and the target one $\mathbf{q_T}$ [33]:

$$\mathbf{q_e} = \begin{bmatrix} q_{T_1} & q_{T_2} & q_{T_3} & q_{T_4} \\ -q_{T_2} & q_{T_1} & q_{T_4} & -q_{T_3} \\ -q_{T_3} & -q_{T_4} & q_{T_1} & q_{T_2} \\ -q_{T_4} & q_{T_3} & -q_{T_2} & q_{T_1} \end{bmatrix} \mathbf{q_S} \tag{4.24}$$

Eq. (4.23) reads, with the use of $\mathbf{q_e}$:

$$\begin{cases} M_x = Kp_x \frac{4q_{e_2}q_{e_1}}{2} + Kd_x\omega_x \\ M_y = Kp_y \frac{4q_{e_3}q_{e_1}}{2} + Kd_y\omega_y \\ M_z = Kp_z \frac{4q_{e_4}q_{e_1}}{2} + Kd_z\omega_z \end{cases} \tag{4.25}$$

which finally assumes the form:

$$\begin{cases} M_x = 2Kp_x q_{e_2} q_{e_1} + Kd_x\omega_x \\ M_y = 2Kp_y q_{e_3} q_{e_1} + Kd_y\omega_y \\ M_z = 2Kp_z q_{e_4} q_{e_1} + Kd_z\omega_z \end{cases} \tag{4.26}$$

which provides the torque components to drive the reaction wheels in the AOCS model of LUMIO. As Sec. 3.1.2 reports, q_{e_1} is the scalar part of the quaternion vector, hence the first component.

The selection of the gains is a topic of Sec. 6.3, where the details of the implementation are discussed.

5 | Station keeping

"The moon is the first
milestone on the road to the stars."

- Arthur C. Clarke

In this chapter the details of the algorithm developed to compute the $\Delta\mathbf{v_{SK}}$ for the station keeping of LUMIO are reported. Starting from the requirements of the GNC, the chapter will present the MATLAB implementation and a detailed analysis of the algorithm.

5.1. GNC requirements

Table C.1, in Appendix C, shows the requirements related to the GNC of LUMIO. Their initial version was listed starting from general necessities of the system itself, from requirements of similar missions and considering the results obtained up to Phase A. Then, as the work proceeded, the results obtained allowed to modify some of the requirements, or to carry out changes to the simulations. The version proposed represents the final version reached.

A critical analysis is reported for the requirements which have not completely fulfilled with the work presented in this thesis, or those which require further analyses.

5.2. Station keeping algorithm

The implementation of the station keeping algorithm, here focused to guidance itself, hence the solely computation of the manoeuvres, i.e. the $\Delta\mathbf{v_{SK}}$, is the first step to develop the GNC system for LUMIO.

To properly understand how the algorithm works, it is essential to well clarify the ConOps of the mission, schematised in Fig. 5.1.

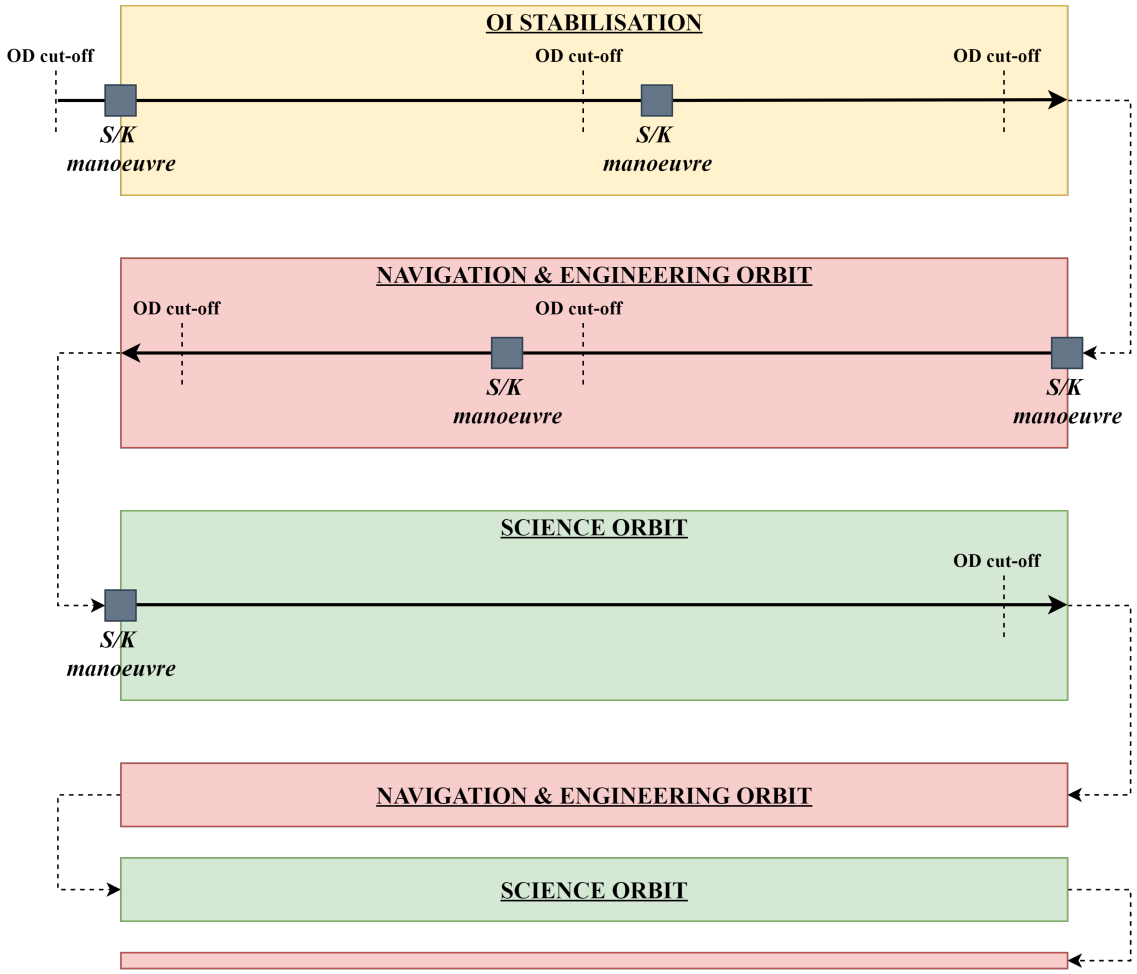


Figure 5.1: GNC cycles concept.

As anticipated in Sec. 4.1, the operative mission in the halo orbit starts with a $\simeq$ 14 days phase called OI stabilisation, with the aim of recovering the large errors which are expected at the injection in the halo. This phase is characterise by 3 station keeping manoeuvres, disposed as for the Nav&Eng orbits, i.e one at the entry, one at the middle of the orbit, and the third at the end. Each manoeuvre is scheduled and prepared with a margin equal to the cut-off time Δt_c. In particular, the position and velocity deviations at t_c are utilised to predict their development at the target points epoch and compute the $\Delta \mathbf{v_{SK}}$ to be applied at the manoeuvre date.

Once the OI stabilisation is terminated, a good correction of the initial trajectory error is expected. However, to ensure a more safe scenario, the following orbit is planed to be a Nav&Eng one, to adjust even more the path in preparation for the first Science cycle. The Nav&Eng orbit exhibits operations equal to the OI stabilisation, although being characterised by smaller navigation errors. Following this procedure, LUMIO is required

to perform 5 correction manoeuvres in a row, before performing any scientific measurement of the Moon, ensuring a higher initial stability of the trajectory. The Science cycles are free of any orbit correction, they are just focused in performing measurements. They are alternated with the Nav&Eng ones for the entire duration of the mission operations.

Considering Table 4.1, the average duration of the Science cycles is 14.8 days, while the average duration of the Nav&Eng cycles is 13.3 days. The overall duration of the observation windows is the 89.6% of the total time of the Science cycles. The Science cycle number 13 is longer than the others because of the seasonal behaviour of the selected operational orbit. The period from the 6 September, 2023, to the 29 September, 2023, is entirely dedicated to science also if the geometrical condition on the Moon illumination is triggered. That because the illumination conditions are favourable for most of the period covered by the Science cycle number 13. The Operative phase terminates with a very short Nav&Eng cycle, which can be eventually extended increasing the 1 year minimum time on the operative orbit [1].

5.2.1. Implementation

The equation for the optimal $\Delta\mathbf{v_{SK}}$ has been already introduced in Sec. 4.1.1. As mentioned in Sec. 4.1.1, the equations can be used in a dimensionless form, which gives the chance to resize the problem reducing the orders of magnitude. Applying this procedure, the values in Table 5.1 have been used for the weighting matrices.

Weighting matrix [–]	**Value**
Q	$10^{-1} \times I_{3\times3}$
R_1	$10^{-2} \times I_{3\times3}$
R_2	$10^{-2} \times I_{3\times3}$

Table 5.1: Values of Q and R employed in the S/K algorithm.

The Δ times for the cut-off and target points epochs are reported in Table 5.2, in accordance with [1].

Δt $[days]$	**Value**
Δt_c	0.5
Δt_1	35
Δt_2	42

Table 5.2: Values of Δt_c, Δt_1 and Δt_2 employed in the S/K algorithm.

The algorithm is built taking into account the orbit insertion error ϵ_{OI}, orbit determination ones ϵ_{OD}, and manoeuvre execution error ϵ_{EX}, to account for thrusters misalignment. Moreover, an additional real displacement ϵ_{ICs} in the starting condition of the spacecraft is introduced. The respective standard deviations used in the algorithm are reported in Table 5.3. In particular the values used are obtained from the firsts simulations in [1], apart for the OD ones which have been reduced for the reasons presented in Chap. 7, related to the convergence of the simulation and quality of the results.

	ICs	**OI**	**OD**	**EX**
$\sigma_{x,y,z}$ $[km]$	1	1	0.1	-
$\sigma_{u,v,w}$ $[m/s]$	0.01	0.01	0.001	2%

Table 5.3: Standard deviations of the errors employed in the S/K algorithm.

The errors are then modelled as zero-mean Gaussian distributions, hence:

$$\epsilon_{ICs} \sim \mathcal{N}(0, \sigma^2_{ICs}) \tag{5.1}$$
$$\epsilon_{OI} \sim \mathcal{N}(0, \sigma^2_{OI}) \tag{5.2}$$
$$\epsilon_{OD} \sim \mathcal{N}(0, \sigma^2_{OD}) \tag{5.3}$$
$$\epsilon_{EX} \sim \mathcal{N}(0, \sigma^2_{EX}) \tag{5.4}$$

where σ^2 is the covariance of the error (square of the standard deviation).

The algorithm must also take into account about the *MIB* of the thrusters selected, to discard the manoeuvres with a magnitude too small to be executed. The thrusters, Table 4.5, are able to provide a thrust up to 0.3 N, although it has been set to $T_{max} = 0.245$ N after some preliminary analyses conducted in [5]. Then, for a non impulsive manoeuvre:

$$\Delta t_f = \frac{c_e \Delta m}{T_{max}} \tag{5.5}$$
$$\Delta m = |m_f - m_i| = |m_i \left(e^{-||\Delta \mathbf{v_{SK}}||/c_e} - 1\right)| \tag{5.6}$$

then, considering the MIB of the thrusters in Table 4.5:

$$MIB \leq 70 \ mNs$$

the minimum firing time, considering the thrusters are two, is:

$$\Delta t_{f_{min}} = \frac{MIB}{2T_{max}} = 0.1429 \ s \tag{5.7}$$

substituting in Eq. (5.5) and solving for $||\Delta\mathbf{v_{SK}}||$, it yields:

$$||\Delta\mathbf{v_{SK\ min}}|| = 0.0029\ m/s \tag{5.8}$$

This value is the minimum threshold imposed for the determination of the $\Delta\mathbf{v_{SK}}$: in the cases in which the required $\Delta\mathbf{v_{SK}}$ is lower than $\Delta\mathbf{v_{SK\ min}}$, the manoeuvre is not executed.

The station keeping algorithm has been written adopting the implementation strategy showed in Alg. 2, a simplified and intuitive version of `DVsk_algorithm.m`. It utilises, as inputs, the data reported in Table 5.2, Table 5.3, Table 5.1 and the vector *dates*, which includes the date of each single manoeuvre during the mission, and it is reported in Table 5.4. These are obtained starting from the initial and final dates of the Nav&Eng phases in Table 4.1, and adding one approximately in the middle of the two.

$N^{\circ}_{S/K}$	Date	$N^{\circ}_{S/K}$	Date
1	2024-MAR-21-12:00:00	**22**	2024-OCT-07-04:16:02
2	2024-MAR-27-14:00:54	**23**	2024-OCT-14-14:56:42
3	2024-APR-03-14:00:54	**24**	2024-OCT-28-11:39:32
4	2024-APR-09-14:00:54	**25**	2024-NOV-04-11:39:32
5	2024-APR-16-19:30:52	**26**	2024-NOV-12-09:16:35
6	2024-MAY-01-16:34:03	**27**	2024-NOV-25-21:15:02
7	2024-MAY-08-16:34:03	**28**	2024-DEC-02-21:15:02
8	2024-MAY-15-06:56:28	**29**	2024-DEC-10-07:51:47
9	2024-MAY-29-20:02:06	**30**	2024-DEC-24-04:03:47
10	2024-JUN-05-20:02:06	**31**	2024-DEC-31-04:03:47
11	2024-JUN-12-06:19:46	**32**	2025-JAN-07-14:35:50
12	2024-JUN-26-17:58:10	**33**	2025-JAN-21-14:45:13
13	2024-JUL-03-17:58:10	**34**	2025-JAN-28-14:45:13
14	2024-JUL-10-19:20:31	**35**	2025-FEB-04-10:57:04
15	2024-JUL-25-03:28:07	**36**	2025-FEB-18-17:14:57
16	2024-AUG-01-03:28:07	**37**	2025-FEB-26-17:14:57
17	2024-AUG-08-05:06:13	**38**	2025-MAR-04-15:22:31
18	2024-AUG-22-07:18:30	**39**	2025-MAR-19-09:11:46
19	2024-AUG-30-07:18:30	**40**	2025-MAR-20-09:11:46
20	2024-SEP-06-08:01:30	**41**	2025-MAR-21-10:59:59
21	2024-SEP-29-04:16:02		

Table 5.4: Vector of dates (*dates*) used in the S/K algorithm.

The algorithm produces as output the $\Delta\mathbf{v_{SK}}$ for the 41 manoeuvres, and the state of the corrected trajectory during each orbit, to be then easily plotted by the apposite function. The results of the simulation will be presented in Chap. 7.

Algorithm 2 Algorithm for station keeping.

function $[\Delta\mathbf{v_{SK}}] = DVSK(dates, \Delta t_c, \Delta t_i, \sigma_{ICs}, \sigma_{OI}, \sigma_{OD}, \sigma_{EX}, Q, R_i)$
$\epsilon_{\mathbf{ICs}} \sim \mathcal{N}(0, \sigma^2_{ICs})$
$\epsilon_{\mathbf{OI}} \sim \mathcal{N}(0, \sigma^2_{OI})$
for $i \leftarrow 1 : length(dates)$ **do**
  $\epsilon_{\mathbf{OD}} \sim \mathcal{N}(0, \sigma^2_{OD})$ ▷ initialise variables
  $\epsilon_{\mathbf{EX}} \sim \mathcal{N}(0, \sigma^2_{EX})$
  $date_{\Delta v} \leftarrow dates(i)$
  $t_v \leftarrow date_{\Delta v}$
  $t_c \leftarrow date_{\Delta v} - \Delta t_c$
  $t_1 \leftarrow date_{\Delta v} + \Delta t_1$
  $t_2 \leftarrow date_{\Delta v} + \Delta t_2$
  if $i = 1$ **then** ▷ from t_v to t_c
    $\delta\mathbf{x_c} \leftarrow \epsilon_{\mathbf{OI}}$
  else
    $[\mathbf{t}, \mathbf{x}] = orbit\_propagation(t_{v-1} \rightarrow t_c, \mathbf{ICs})$
    $\delta\mathbf{x_c} = \mathbf{x}(end) - \mathbf{x}_{ephemeris}(end) + \epsilon_{\mathbf{OD}}$
    if $||\delta\mathbf{x_c}(1:3)|| > 1000$ **then** ▷ stop simulation for too high errors
      return
    end if
    $\mathbf{ICs} \leftarrow \mathbf{x}(end)$
  end if
  $Extract: \ \Phi(t_1, t_c), \Phi(t_2, t_c), \Phi(t_1, t_v), \Phi(t_2, t_v)$ ▷ build $\Delta\mathbf{v_{SK}}$ formula
  $\Delta\mathbf{v_{SK}} = A \sum_{i=1}^{N_{tp}} (\alpha_i \delta\mathbf{r_c} + \beta_i \delta\mathbf{v_c}) + \epsilon_{\mathbf{EX}}$
  if $\Delta\mathbf{v_{SK}} < 2.9 \times 10^{-6}$ **then** ▷ avoid executing too small manoeuvres
    $\Delta\mathbf{v_{SK}} \leftarrow [0\ 0\ 0]'$
  end if
  if $i = 1$ **then** ▷ from t_c to t_v
    $\mathbf{ICs} \leftarrow \mathbf{x}_{ephemeris} + \epsilon_{\mathbf{ICs}}$
  else
    $[\mathbf{t}, \mathbf{x}] = orbit\_propagation(t_c \rightarrow t_v, \mathbf{ICs})$
    $\mathbf{ICs} \leftarrow \mathbf{x}(end) + [0_{3x1};\ \Delta\mathbf{v_{SK}}]$
  end if
end for

6 | Attitude guidance & Control

"I think our need to control comes from
our fear of the unknown and our own limitations."

- Rob LeBow

After the development of the algorithm used to compute the manoeuvres to be executed during the mission, the model which simulates the GNC operations during the station keeping is developed. The goal consists in modelling the delicate coupling between the ADCS and GNC system, to reach the correct attitude before and after any orbit correction manoeuvre, fulfilling the LUMIO requirements.

The model devoted to this role has been developed in Simulink; in the following the description of each part of the *ADCS-GNC* subsystem will be described, to give a complete understanding of the logic behind it, simplifying the work for future developers of the model.

The dedicated block appears as in Fig. 6.1, coming from the central section called *SPACECRAFT* in Fig. 3.1, with two inputs coming from the *World* and *Ephemeris* blocks, and the outputs being the control actions, in torque and thrust, going back to the *World* block.

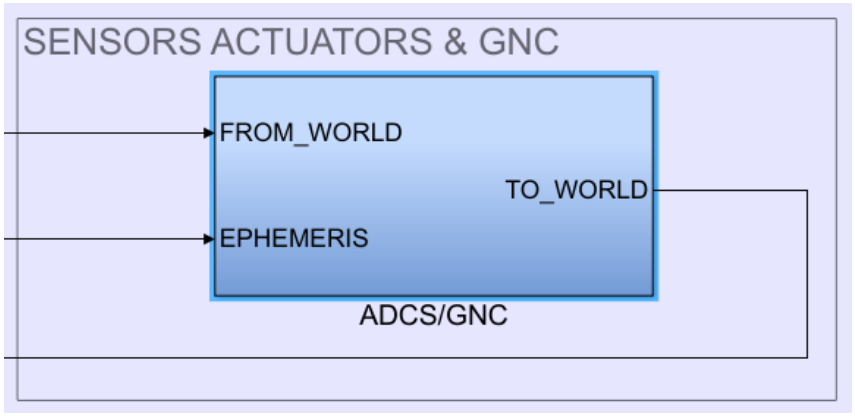


Figure 6.1: *ADCS-GNC* block in Simulink.

The content of the block is reported in Fig. 6.2. It is subdivided in 4 main sections, divided by functionality: *Data collection and control logic*, *Attitude guidance*, *Control* and *Output*.

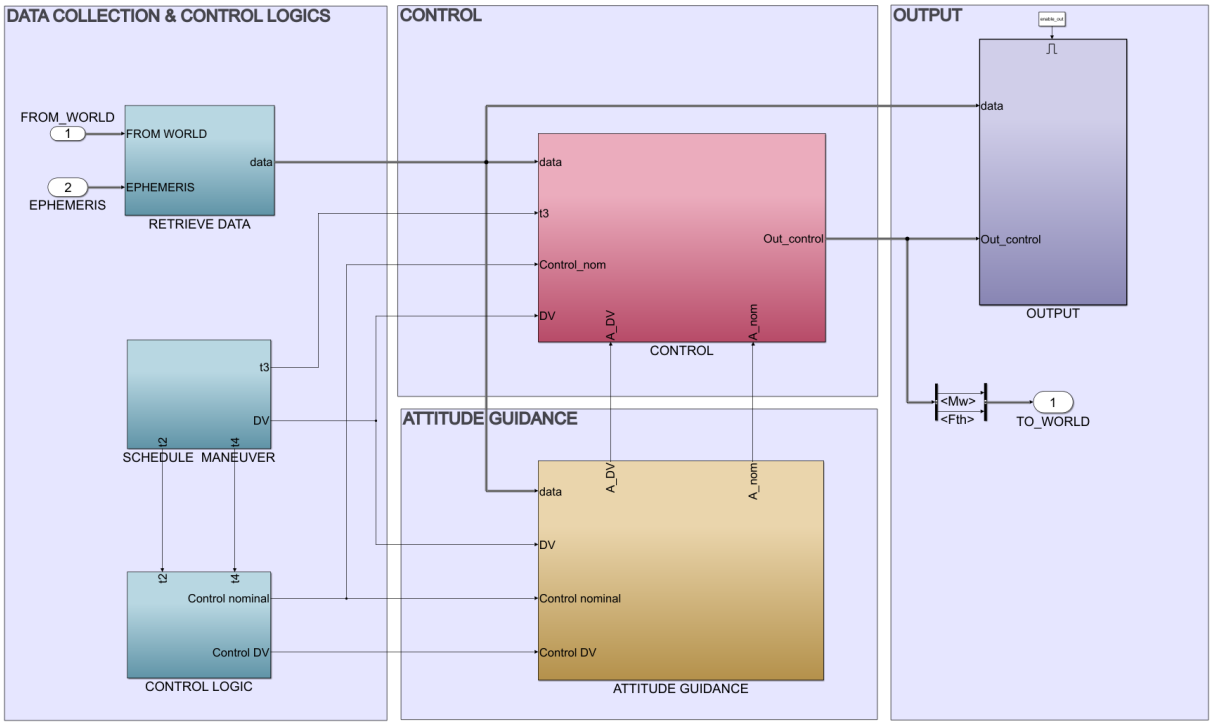


Figure 6.2: *ADCS-GNC* block content in Simulink.

6.1. Data collection and control logic

The left-hand side area presents 3 main blocks.

The first, *RETRIEVE DATA*, is the one connected to the input ports. It serves at extracting the following variables, required by the ADCS-GNC system: quaternions $\mathbf{q}$ (and respective Direction Cosine Matrix A_{BN}), angular velocities $\boldsymbol{\omega}$, trajectory state $\mathbf{x}$, Sun and Moon position $\mathbf{R_{es_N}}$, $\mathbf{R_{em_N}}$.

The second block is called *SCHEDULE MANOEUVRE* and it is in charge to furnish the times at which the GNC operations must be programmed, other than the correct $\mathbf{\Delta v_{SK}}$. The logic behind it deserves a detailed explanation. First of all, Fig. 6.3 shows the function contained in that block.

The idea used to develop the entire GNC model, relies on 4 times which define the overall AOCS operations: t_1, t_2, t_3 and t_4. Looking at Fig. 6.4, time t_1 is defined as the start of

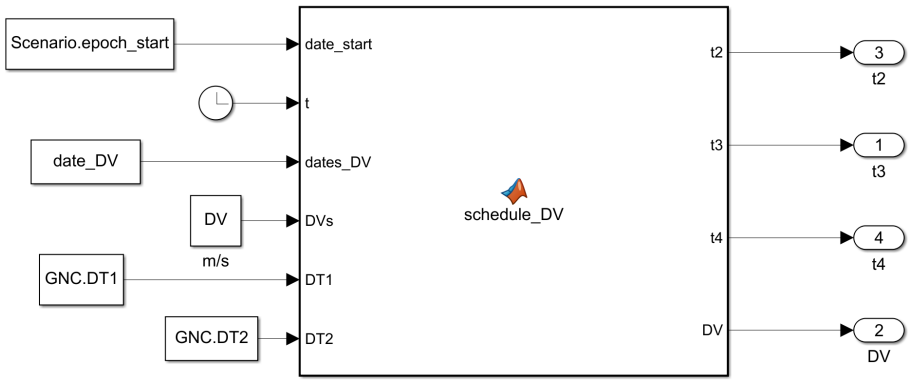


Figure 6.3: *schedule_DV* function.

the AOCS operations, which consist in performing a slew manoeuvre, at the entrance of the halo, to acquire the nominal LUMIO attitude, i.e. pointing the Moon with the optic and tracking the Sun with the solar panels. This time is set equal to zero, assuming the manoeuvre to be instantly computed, and then executed, to correct the attitude. This means that once the spacecraft enters the halo it will immediately start to position itself nominally. For these reasons, it is an event which occurs only at the beginning, contrarily to the other attitude corrections which verify every manoeuvre, and therefore are each time planned by the the function *schedule_DV*.

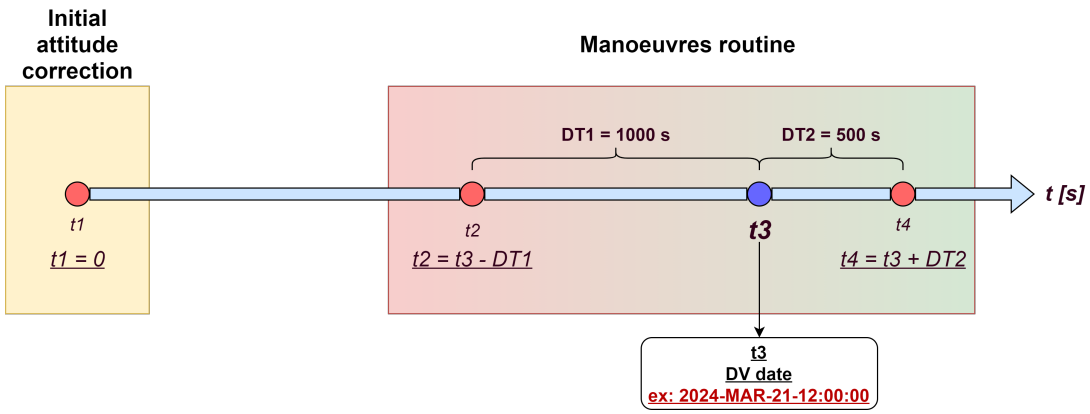


Figure 6.4: GNC operations timeline.

The three main events which occur during each GNC operation, happening at t_2, t_3 and t_4, are now described.

- t_2: is defined as the time at which the system is prepared to change the attitude for the firing phase. It is obtained as a function of t_3, the manoeuvre date:

$$t_2 = t_3 - \Delta T_1$$

where ΔT_1 is in fact the time before the firing, at which the attitude correction, to position the thrusters on the correct direction, begins. It has be set $\Delta T_1 = 1000\ s$, about 17 min before the firing. This duration, at this stage, is completely arbitrary; whenever precise constraints will be formulated its value will be modified, accordingly to the requirements. Anyway, the choice of 1000 s is due to two factors: first, its a sufficient time to notice a settling of the attitude error, with the working assumptions, meaning that the manoeuvre is performed from a stable attitude, second, the use of a short time interval allows to have plots more easy to visualise.

- t_3: it is the manoeuvre date, in Table 5.4, and coincides with the firing itself. Both t_2 and t_4 are determined in function of it.
- t_4: it represents the time at which the spacecraft restores the nominal attitude, initially obtained at t_1, and then after the conclusion of each S/K manoeuvre. It is planned to happen at:

$$t_4 = t_3 + \Delta T_2$$

where $\Delta T_2 = 500\ s$. This value has been selected for similar reasons to ΔT_1, and considering that, passed that time, the firing has been terminated for quite a while, hence the nominal attitude can be restored.

In detail, what the function *schedule_DV* does is reported in Alg. 3.

Algorithm 3 *schedule_DV* algorithm for manoeuvres planning.

function $[t_2, t_3, t_4, \Delta\mathbf{v_{SK}}] = schedule\_DV(date_{start}, t, dates_{\Delta v}, \Delta\mathbf{vs_{SK}}, \Delta T_1, \Delta T_2)$
for $i \leftarrow 1 : size(\Delta\mathbf{vs_{SK}}, 2)$ **do**
  $t_{start} = dates_{\Delta\mathbf{v}}(i) - date_{start} - \Delta T_1$ ▷ reference start t_2
  $t_{end} = t_{start} + \Delta T_1 + \Delta T_2$
  if $t \geq t_{start}$ & $t < t_{end}$ **then** ▷ define S/K timings and $\Delta\mathbf{v_{SK}}$
    $\Delta\mathbf{v_{SK}} \leftarrow \Delta\mathbf{vs_{SK}}(:, i)$
    $t_2 \leftarrow t_{start}$
    $t_3 \leftarrow t_{start} + \Delta T_1$
    $t_4 \leftarrow t_3 + \Delta T_2$
    return
  else ▷ define dummy S/K timings and $\Delta\mathbf{v_{SK}}$
    $\Delta\mathbf{v_{SK}} \leftarrow [0\ 0\ 0]'$
    $t_2 \leftarrow \Delta T_1/10000$
    $t_3 \leftarrow \Delta T_1/10000$
    $t_4 \leftarrow \Delta T_1/10000$
  end if
end for

where $date_{start}$ is the starting date of the simulation (see Table 3.3), $date_{\Delta\mathbf{v}}$ are reported

in Table 5.4 and $\Delta\mathbf{vs_{SK}}$ is the matrix of all the $\Delta\mathbf{v_{SK}}$ of the mission, obtained with Alg. 2. The dummy quantities, assigned out of the manoeuvre operations time window, are selected such to do not interfere with the algorithm during nominal attitude operations.

To conclude, in the case in which more precise requirements will be imposed for the GNC operations, it will be sufficient to modify the values of ΔT_1 and ΔT_2 in the `GNC.txt` data file.

The third block of the left-hand side section is called *CONTROL LOGIC*. It gets the GNC timings from *SCHEDULE MANOEUVRE* and determines if the spacecraft must be controlled to follow the nominal attitude or the manoeuvre one. Its internal architecture is reported in Fig. 6.5.

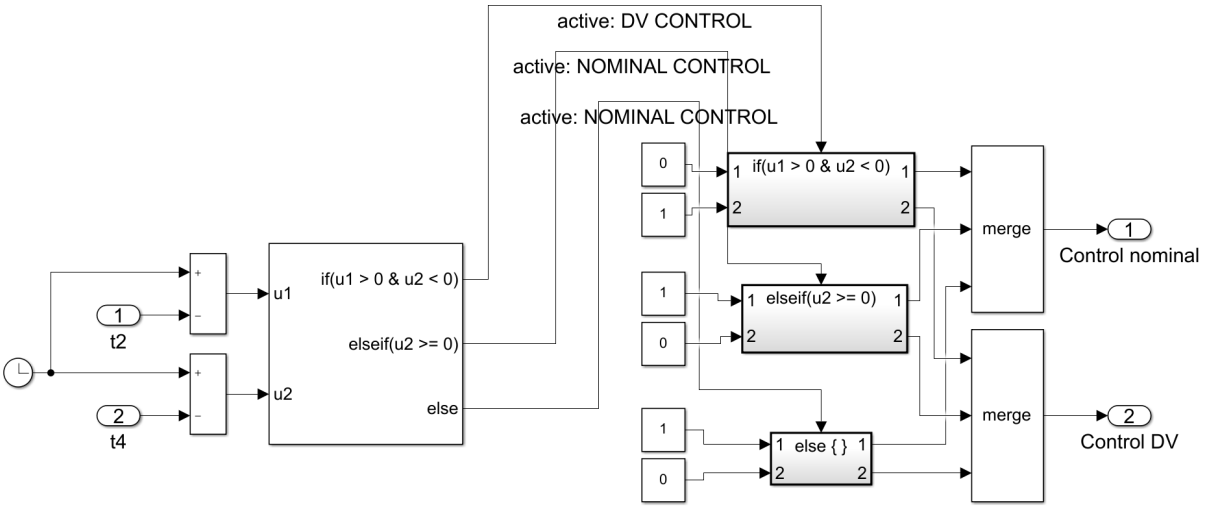


Figure 6.5: Attitude control logic in Simulink.

From the knowledge of the exact times t_2, t_3 and t_4 at which the GNC phases must occur, the logic compares them with the actual simulation time, to establish which control must be applied. This very simple working principle activates the nominal attitude control for $0 < t < t_2$ and for $t \geq t_4$, deactivating the other one, and vice versa for $t_2 < t < t_4$.

6.2. Attitude guidance model

This model, as mentioned in Chap. 4, is dedicated at the selection of the attitude to be acquired. Therefore, the *ATTITUDE GUIDANCE* block is in charged to apply the rules described in Sec. 4.2. As it can be noticed looking at that section, the input variables

required are the Sun, the Moon and the spacecraft position in the inertial frame. From their knowledge, the nominal and manoeuvre attitude matrices A_{nom} and $A_{\Delta v}$ are built. Fig. 6.6 shows the internal content of the block.

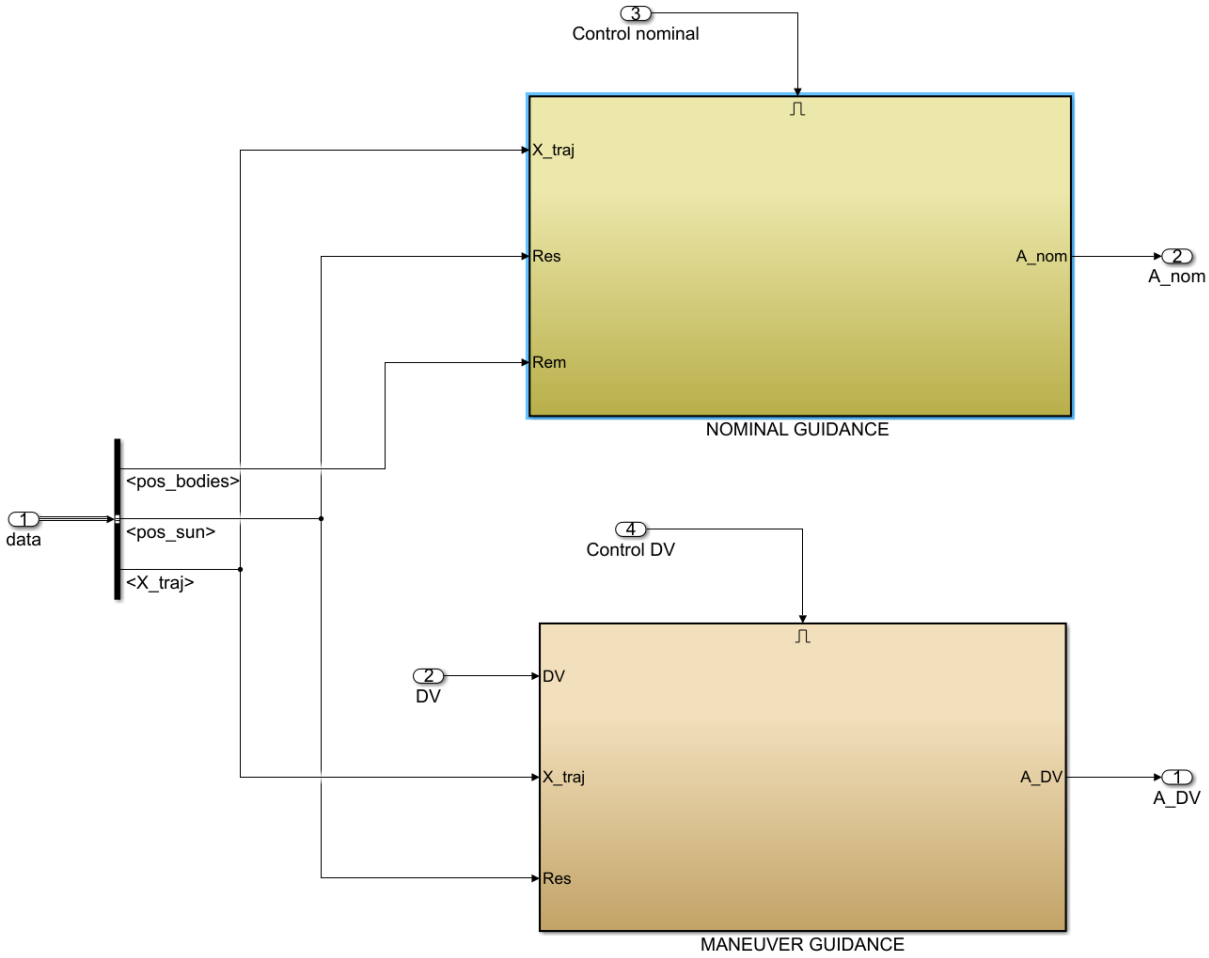


Figure 6.6: *ATTITUDE GUIDANCE* block in Simulink.

The attitude matrices are computed accordingly to the control logic coming from the homonym block: according to the phase the GNC is undergoing, the corresponding attitude matrix is computed while the other one is disabled.

6.3. Control model

The *CONTROL* block takes as inputs the attitude matrices coming from the attitude guidance, the control logic binary outputs, the vector $\Delta\mathbf{v_{SK}}$ and the time t_3 exiting from the manoeuvre planning block, indicating the firing execution date. All these inputs are processed to determine the control actions to apply during that particular S/K manoeuvre.

From Fig. 6.7 it is possible to see the internal subdivision of the block, in *Control law* and *Actuators* parts.

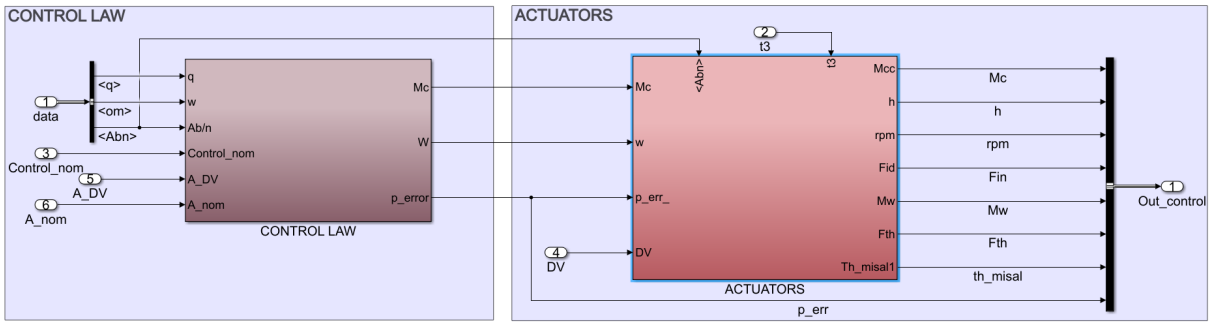


Figure 6.7: *CONTROL* block architecture in Simulink.

In *Control law*, Fig. 6.8, according to the control logic, which indicates the attitude LUMIO must have at that instant, the correct matrix is selected between A_{nom} and $A_{\Delta v}$. Subsequently the pointing error p_{err} in the x-axis is computed between the desired attitude A_{ref}, between A_{nom} and $A_{\Delta v}$, and the current one A_{BN}:

$$p_{err} = acos\left(\frac{A_{BN}(1,:)\cdot A_{ref}(1,:)}{||A_{BN}(1,:)||||A_{ref}(1,:)||}\right) \tag{6.1}$$

The control law is derived as in Eq. (4.26), with quaternions and angular velocities. These are firsts discretized to simulate a more realistic behaviour: indeed, the control action determined by the on-board software is certainly going to be discrete, even though the final actual control will be continuous. The rate transition applied provides an output frequency for $\boldsymbol{\omega}$ and $\mathbf{q}$ of 1.67 Hz, see Table 3.2.

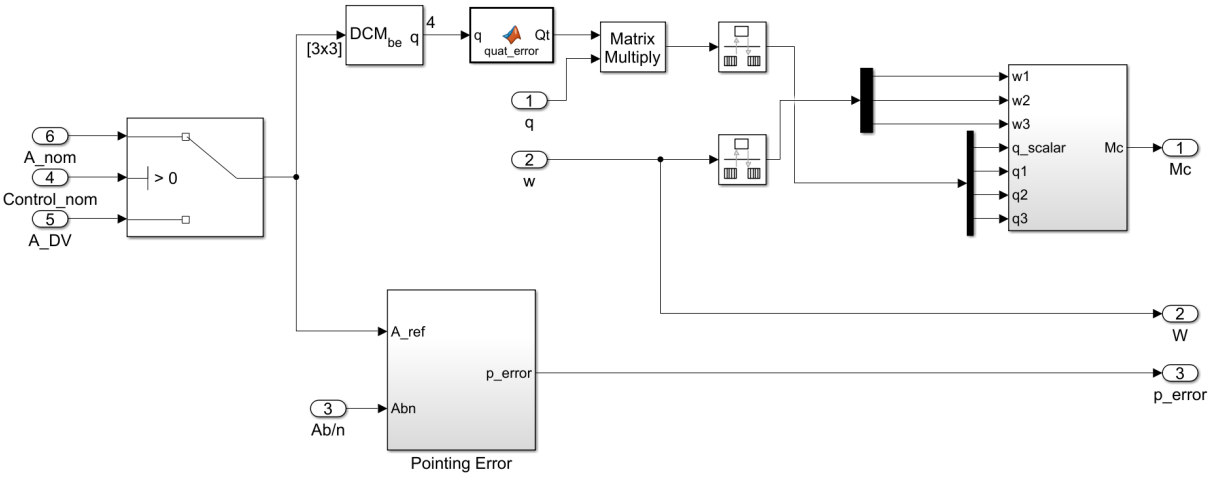


Figure 6.8: *Control law* model in Simulink.

The control gains applied are:

$$Proportional : Kp_x = Kp_y = Kp_z = 1.8 \times 10^{-4} \tag{6.2}$$

$$Derivative : Kd_x = Kd_y = Kd_z = 7 \times 10^{-3} \tag{6.3}$$

which have been selected after a trial and error process which has determined a fast settling time and low steady-state error.

The control torque $\mathbf{M_c}$ exiting from the *Control law* block is then inserted in the *Actuators* block, which has the role of simulating the behaviour of reaction wheels and thrusters, as depicted in Fig. 6.9.

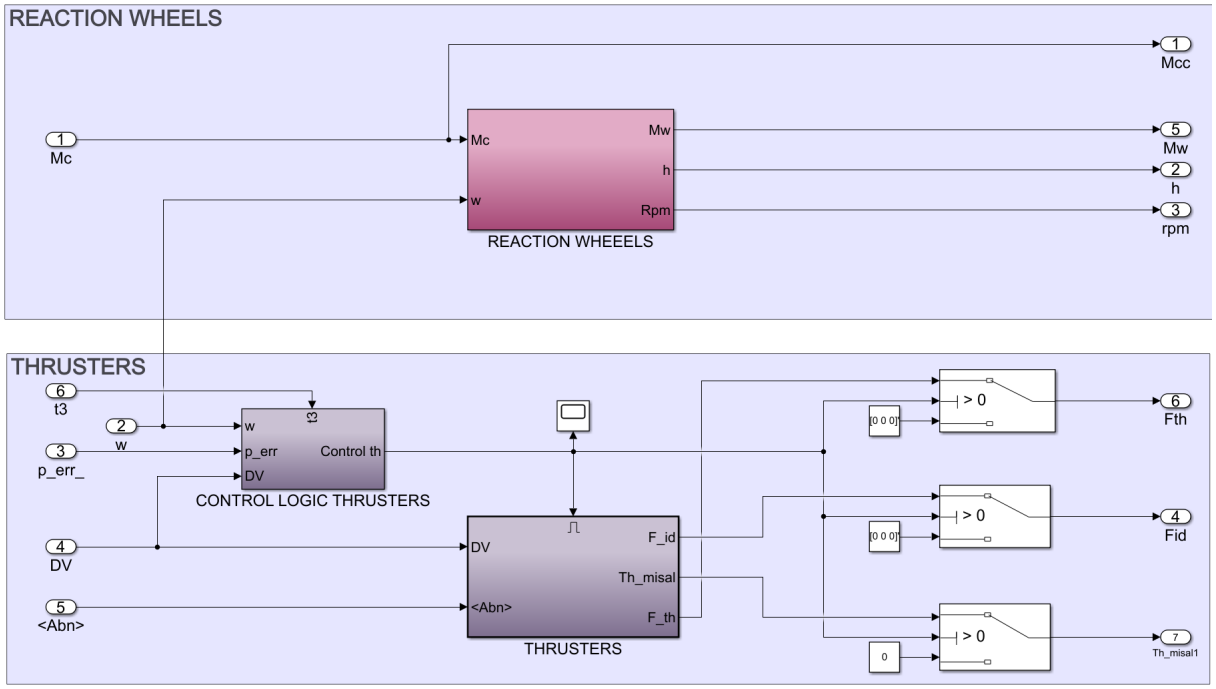


Figure 6.9: *Actuators* block content in Simulink.

Let's see in separate paragraphs how the two kind of actuators have been modelled.

Reaction wheels The reaction wheels operate on the basis of the input torque and are modelled to simulate the exact distribution they have in LUMIO, represented by 4 wheels in pyramid configuration. As depicted in Fig. 6.10, the model includes the introduction of noise, considered as provided by the bearings, and a saturation block. Their settings are reported in Table 6.1, accordingly to their data sheet in Table 4.4. In particular, the noise variance presents a very low value, which anyway has revealed to be consistent with the output torque.

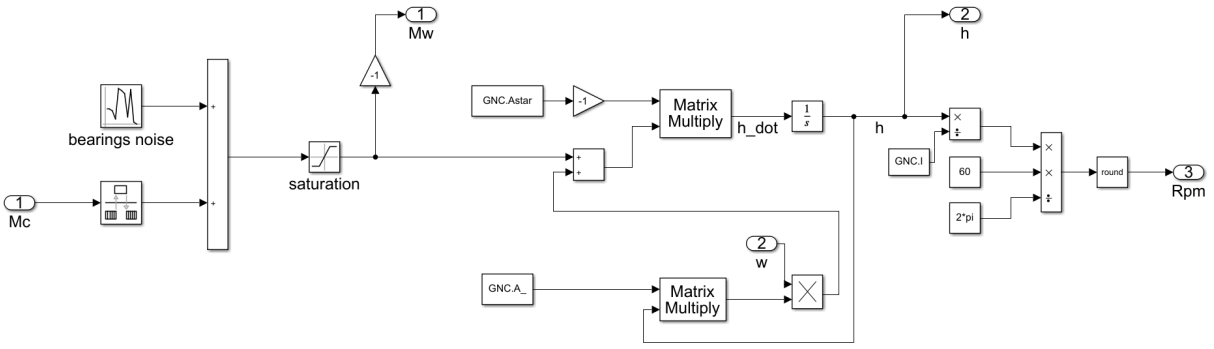


Figure 6.10: *Reaction wheels* model in Simulink.

Parameter	**Value** $[Nm]$
Noise mean	0
Noise variance	10^{-15}
Saturation limits	$\pm 2 \times 10^{-3}$

Table 6.1: Settings for the reaction wheels model in Simulink.

The real momentum $\mathbf{M_r}$ which provides each reaction wheel is obtained from the integration of the equation:

$$\mathbf{M_r} = A\dot{\mathbf{h}}_\mathbf{r} = -\mathbf{M_c} - \boldsymbol{\omega} \wedge A\mathbf{h_r} \tag{6.4}$$

$$\dot{\mathbf{h}}_\mathbf{r} = -A^*(\mathbf{M_c} + \boldsymbol{\omega} \wedge A\mathbf{h_r}) \tag{6.5}$$

where A^* is the pseudo-inverse of A, the matrix which provides the disposition of the wheels. For the pyramid configuration:

$$A = \begin{bmatrix} -a & a & a & -a \\ -a & -a & a & a \\ a & a & a & a \end{bmatrix} \quad a = \frac{1}{\sqrt{3}} \tag{6.6}$$

$$A^* = \begin{bmatrix} -b & -b & b \\ b & -b & b \\ -b & b & b \end{bmatrix} \quad b = \frac{\sqrt{3}}{4} \tag{6.7}$$

Once the angular momentum h around the rotation axis is obtained for each wheel, the rpm can be measured applying:

$$rpm = round\left(\frac{h}{I}\frac{60}{2\pi}\right) \tag{6.8}$$

both the rpm and h are controlled to not exceed the maximum allowable value for the wheels employed.

Thrusters The thrusters, differently from the reaction wheels, are not always active. In particular, they are powered just for a very short duration. These imposes two problems: the first is that a control logic must determine whether or not to activate them, second, the simulation must be performed with a very short fixed step size, to allow Simulink being able to simulate such a short dynamics. This last fact determines noticeable slow downs in the simulation, however this will be commented in Chap. 8. As for the logic behind their operation, this is implemented in the apposite block depicted in Fig. 6.11.

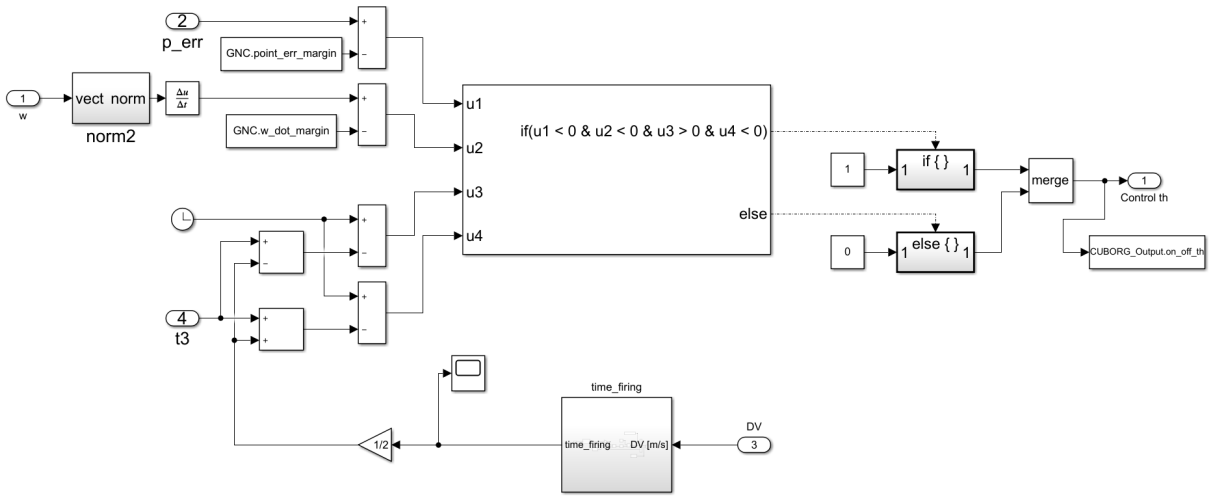


Figure 6.11: *Thrusters control logic* model in Simulink.

In few words, the block determines if three conditions are verified to activate or not the thrusters. First, the time slot has to be correct. This consists in a simple check of the simulation time t to be within the manoeuvre time window:

$$t_3 - \frac{\Delta t_f}{2} < t < t_3 + \frac{\Delta t_f}{2} \tag{6.9}$$

where Δt_f is the firing duration, computed following the rule for non-impulsive manoeuvres [34]:

$$\Delta t_f = \frac{c_e \Delta m}{T_{max}} \tag{6.10}$$

$$\Delta m = |m_f - m_i| = |m_i \left(e^{-||\Delta \mathbf{v_{SK}}||/c_e} - 1\right)| \tag{6.11}$$

in particular, $c_e = I_{sp} g_0$ is the exhaust velocity of the thrusters, m_i is the initial spacecraft mass, m_f the final and $\Delta \mathbf{v_{SK}}$ is expressed in m/s. $T_{max} = 0.245\ N$, for each thruster, and $I_{sp} = 204\ s$ are obtained from the data sheet in Table 4.5. Therefore, for each $\Delta \mathbf{v_{SK}}$, the

model computes the time interval Δt_f which establishes the manoeuvre duration, then the thrusters are activated in that time slot. From Eq. (5.7), it is reminded that the minimum Δt_f is equal to 0.1429 s, accordingly to the thrusters MIB, which also determines the maximum step size to use in the simulation.

The second and third criteria for the thrusters activation are related to the attitude error and angular velocity:

$$p_{err} < 1\ deg \qquad ||\dot{\boldsymbol{\omega}}|| < 10^{-5}\ rad/s \tag{6.12}$$

which are thresholds chosen with a moderate large margin, especially for p_{err} which is usually 10 times smaller. This ensure the thrusters to be activated only if the firing direction is sufficiently accurate and the spacecraft does not oscillate over the $||\dot{\boldsymbol{\omega}}||$ threshold.

If all these three conditions are verified then the thrusters are activated. In all the other cases the output thrust is zero.

The thrusters are modelled as in Fig. 6.12.

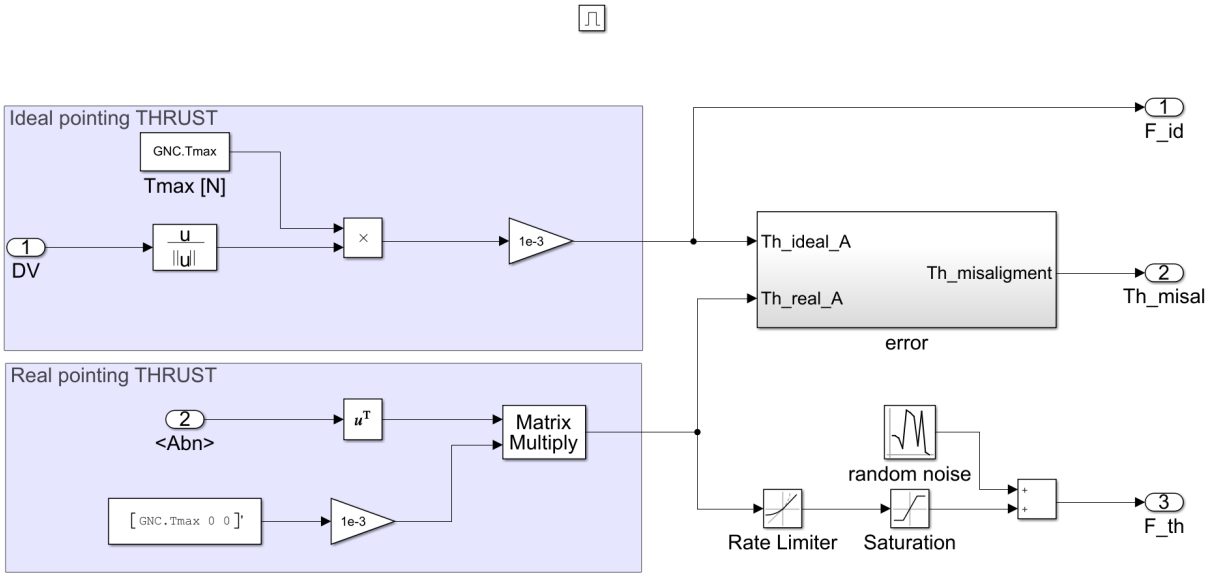


Figure 6.12: *Thrusters* model in Simulink.

It can be noticed the presence of two sub-models in the thrusters architecture:

- **Ideal pointing**: it simulates the firing accordingly to the ideal direction of the manoeuvre, i.e. that provided by the direction of the $\Delta\mathbf{v_{SK}}$, $\Delta\hat{\mathbf{v}}_{\mathbf{SK}}$. Hence, a constant thrust $T_{max} = 1\ N$ is applied, previously converted in kN to be compliant

with the forces unit of measure employed in the tool:

$$\mathbf{T_{ideal}} = T_{max}\Delta\hat{\mathbf{v}}_{\mathbf{SK}} \tag{6.13}$$

- **Real pointing**: it simulates the firing accordingly to the actual attitude of the spacecraft, A_{BN}. The constant thrust is applied in the x direction of the LUMIO frame, where the thrusters are placed, and then converted in the inertial frame:

$$\mathbf{T_{real}} = A_{BN}^{T}[T_{max}\ 0\ 0] \tag{6.14}$$

The computation of both $\mathbf{T_{ideal}}$ and $\mathbf{T_{real}}$ has been performed to retrieve the information about the thrusters pointing error during the manoeuvres, which will be commented in Chap. 7. The output thrust is $\mathbf{T_{real}}$, in order to simulate what really happens during the firing.

The thrust, after being converted in kN, is modelled introducing realistic behaviours as the presence of noise, saturation and rate limiter, to account for a not immediate response of the thrusters. The settings used are reported in Table 6.2.

Parameter	Value
Noise mean	$0\ [kN]$
Noise variance	$10^{-9}\ [kN]$
Saturation limits	$\pm 0.6 \times 10^{-3}\ [kN]$
Rising/Falling rate	$\pm 0.1\ [kN/s]$

Table 6.2: Settings for the thrusters model in Simulink.

6.4. Output saving

The output section, on the right-hand side of the ADCS-GNC block in Fig. 6.2, is in charge in saving the simulation data related with the AOCS system. Moreover it connects the control torque and force to the *WORLD* section, closing the loop.

A simple trick employed to speed up the simulation, consists in the automatic disabling of the output saves in case of simulations longer than 51840 s, which is enough time to see the first S/K manoeuvre from the start of the simulation. This allows to save just the trajectory originated from the closed-loop and reduce the allocation memory for the saved data, speeding the simulation time up.

7 | Results

"Quality without results is pointless.
Results without quality is boring."

- Johan Cruyff

The results obtained are here reported and critically analysed, highlighting their pros and cons. The presented solutions are derived with the models and settings described in the previous chapters, especially Chap. 5 and Chap. 6.

The overall results are commented in two separate sections: the first is used to present the results of the guidance, hence the computation of the $\Delta\mathbf{v_{SK}}$, while the second is dedicated to the attitude control and GNC operations in Simulink.

7.1. Manoeuvres determination

Running the station keeping algorithm, see Alg. 2, for the entire duration of the mission, considering the settings in Chap. 5, the sample set of $\Delta\mathbf{v_{SK}}$ in Table 7.1 is obtained. This sample of manoeuvres has been chosen for its common characteristics to the average simulations outcomes. Together with manoeuvre magnitudes, position and velocity deviations in correspondence of the cut-off time[1] are saved for each orbit. They are particularly useful to comprehend the effective utility of the station keeping, and to get an idea of the oscillations the orbit undergoes, before eventually diverging. $\delta\mathbf{r_c}$ are reported in Table 7.1 next to $\Delta\mathbf{v_{SK}}$. It can be noticed that the 1st and 17th manoeuvres are null; this is due to the constraint imposed to the minimum $\Delta\mathbf{v_{SK}}$, see Eq. (5.8).

To provide an affordable and more representative result, however, a Monte Carlo simulation of 100 runs has been performed to determine the mean total $\Delta\mathbf{v_{SK}}$ during the 1 year operative mission duration, the standard deviation, the 3σ value. In particular, the Monte Carlo is performed with 7 variables quantities, being the standard deviations of Table 5.3; at each run, the guidance will be determined with a different set of errors, to

[1] Defined as $\delta\mathbf{r_c}$ and $\delta\mathbf{v_c}$ in the algorithm notation in Sec. 4.1.1.

evaluate its effective functioning. In the cases in which the simulation does not provide a suitable solution, as a too large position error, that particular results is discarded, not contributing to the statistics. In the 100 simulation considered in this particular case, a total of 59 orbits presented a failed station keeping, where, with "failed", it is intended that the simulation has been terminated because the spacecraft departed more than 1000 km from the reference path, see Alg. 2.

N°_{SK}	$\|\|\Delta\mathbf{v_{SK}}\|\|$ $[m/s]$	$\|\|\delta\mathbf{r_c}\|\|$ $[km]$	N°_{SK}	$\|\|\Delta\mathbf{v_{SK}}\|\|$ $[m/s]$	$\|\|\delta\mathbf{r_c}\|\|$ $[km]$
1	0	1.2	**22**	0.0154	23.5380
2	0.1360	12.0708	**23**	0.0054	20.8714
3	0.0066	7.4417	**24**	0.0426	6.2559
4	0.0151	6.6617	**25**	0.0146	6.5414
5	0.0046	8.9217	**26**	0.0054	3.2917
6	0.0388	12.1295	**27**	0.1137	23.3459
7	0.0279	5.9545	**28**	0.0835	15.1866
8	0.0088	3.7950	**29**	0.0184	21.1927
9	0.0073	2.0888	**30**	0.1127	47.0519
10	0.0066	3.2174	**31**	0.0914	30.9591
11	0.0038	8.3727	**32**	0.0182	48.2707
12	0.0293	13.2653	**33**	0.0757	53.6968
13	0.0204	6.2165	**34**	0.0713	24.5083
14	0.0064	6.9845	**35**	0.0175	38.7332
15	0.0154	4.8798	**36**	0.0193	40.8360
16	0.0137	0.4758	**37**	0.0165	24.3918
17	0	3.3609	**38**	0.0050	35.4387
18	0.1592	23.8112	**39**	0.1397	51.6363
19	0.0790	5.6043	**40**	0.0407	57.1104
20	0.0345	20.2880	**41**	0.0281	52.5009
21	0.0505	16.0668			

Table 7.1: Magnitudes of $\Delta\mathbf{v_{SK}}$ and $\delta\mathbf{r_c}$ for a sample controlled orbit.

The $\Delta\mathbf{v_{SK}}$ characteristics, considering the Monte Carlo statistics, are reported in Table 7.2.

$\mu_{\Delta\mathbf{v_{SK}}}$ $[m/s]$	$\sigma_{\Delta\mathbf{v_{SK}}}$ $[m/s]$	$3\sigma_{\Delta\mathbf{v_{SK}}}$ $[m/s]$
3.0838	0.0606	3.2654

Table 7.2: 1 year station keeping results.

which are coherent with the values found in literature, and in the version presented in [1]. Their validity and applicability are discussed in the next pages.

The minimum and maximum $\Delta\mathbf{v_{SK}}$, keeping in mind the value of $\Delta\mathbf{v_{SK\ min}}$ in Eq. (5.8), oscillate between the values:

$$min_{\Delta\mathbf{v_{SK}}} = 0.0029\ m/s \qquad max_{\Delta\mathbf{v_{SK}}} = 2.5309\ m/s \tag{7.1}$$

The algorithm is set-up to stop the simulation in the case in which $\delta\mathbf{r_c} > 1000\ km$. Considering this, the maximum position deviation reached during the 100 runs is:

$$max_{\delta\mathbf{r_c}} = 866.6376\ km \tag{7.2}$$

with a mean deviation of 38.4858 km.

Fig. 7.1 shows the 1 year station keeping on the halo, associated to the $\Delta\mathbf{v_{SK}}$ in Table 7.1, provided impulsively. In yellow, it is reported the first orbit of insertion stabilisation. Nav&Eng and Science cycles are represented in red and green, respectively.

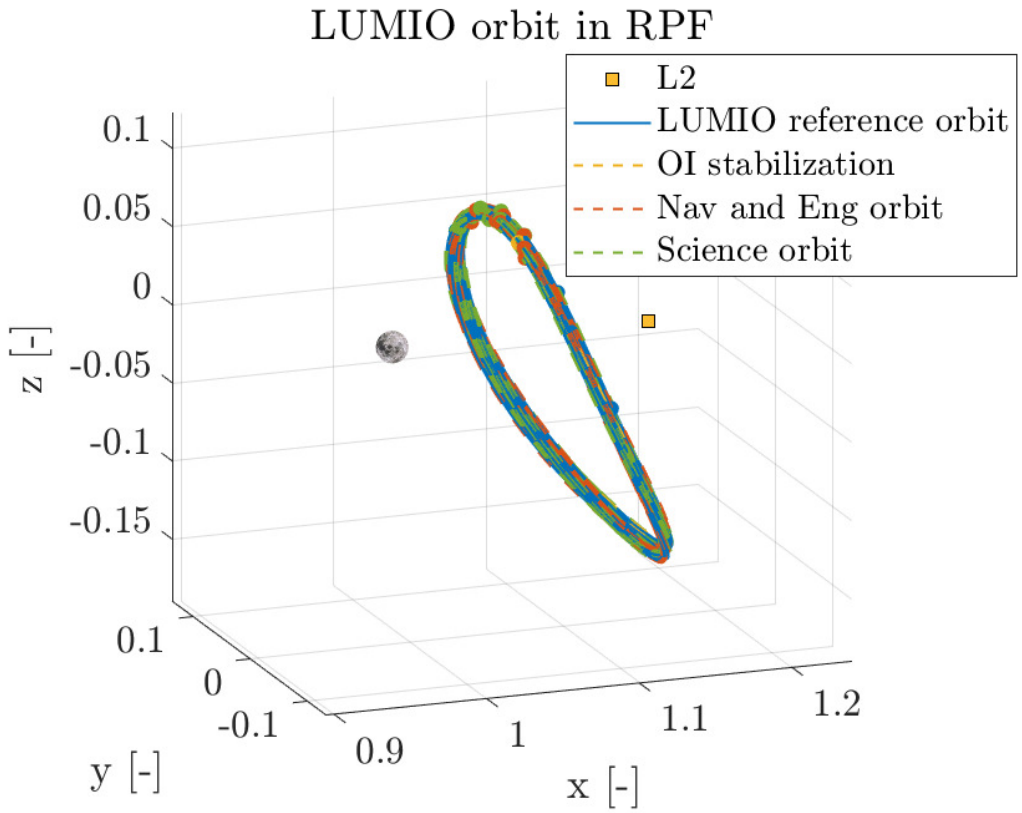


Figure 7.1: 1 year controlled orbit in RPF.

The same is depicted in Fig. 7.2 and Fig. 7.3, but in the Earth and Moon centred frames.

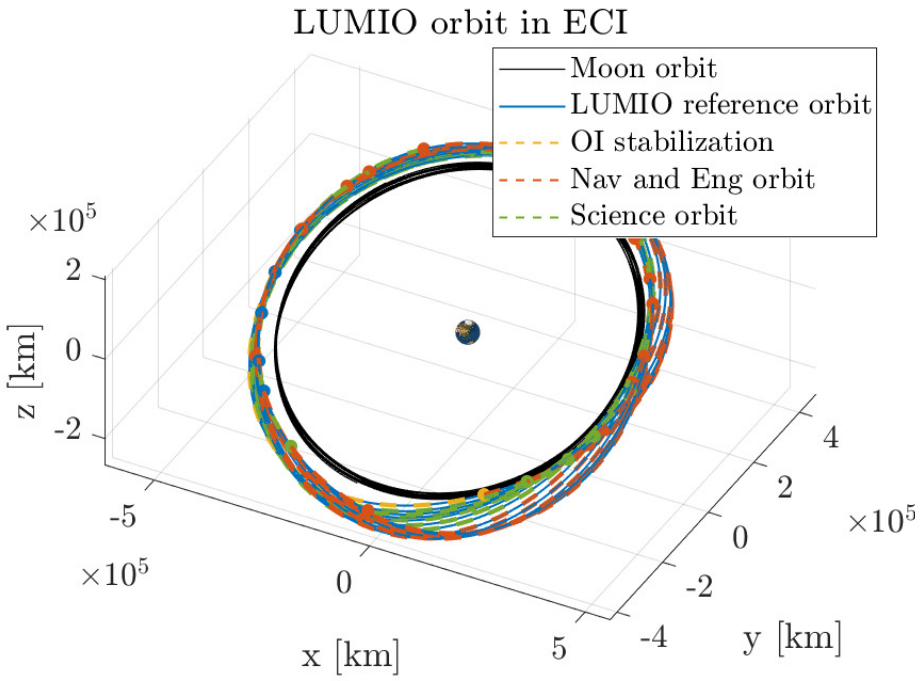


Figure 7.2: 1 year controlled orbit in ECI.

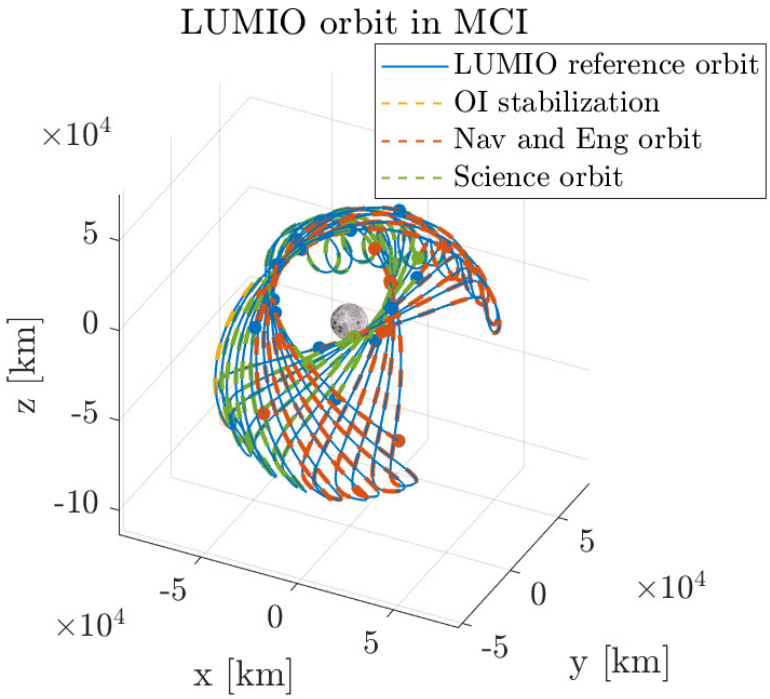


Figure 7.3: 1 year controlled orbit in MCI.

The previous results can be compared with Fig. 7.4, which is the simulated orbit without control, with the same initial offset from the reference path as the controlled one. It is evident the fundamental role of the manoeuvres, which prevent a divergence of the solution already after the first couple of months.

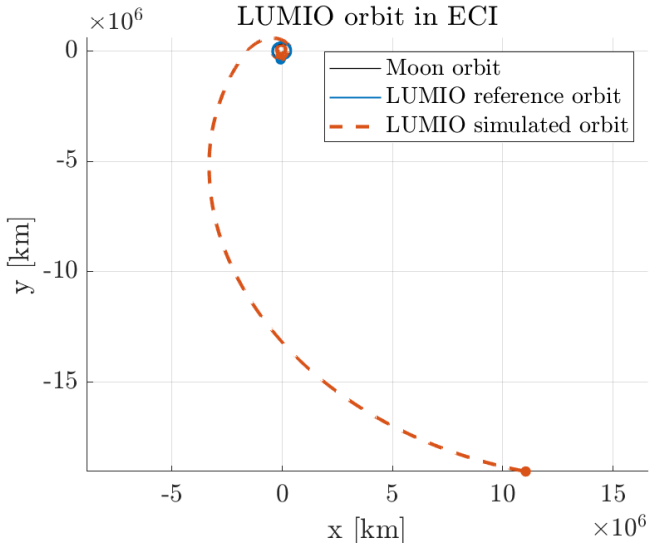


Figure 7.4: 1 year uncontrolled orbit in ECI.

In order to easily visualise the state deviation between the nominal and simulated orbit, Fig. 7.5 and Fig. 7.6 show the error evolution, with the same colour legend as before, indicating the manoeuvres execution with the dotted vertical lines.

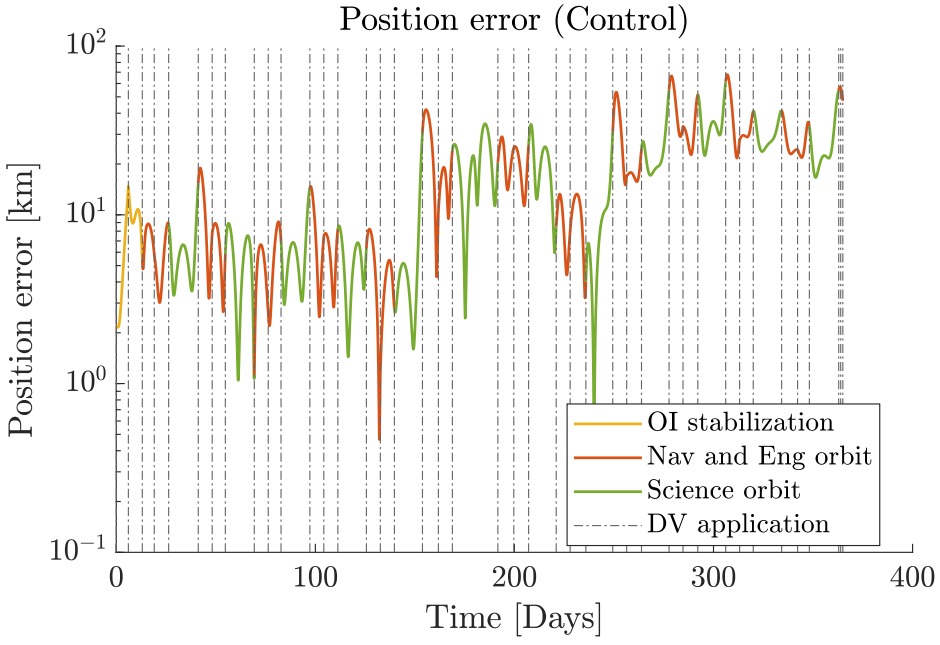


Figure 7.5: Position error in the 1 year controlled orbit.

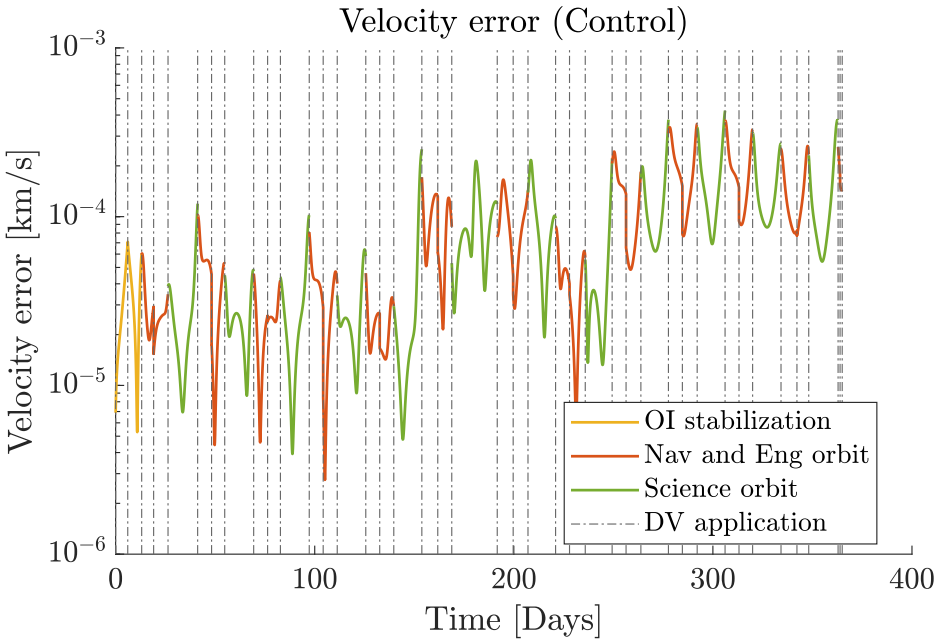


Figure 7.6: Velocity error in the 1 year controlled orbit.

It can be appreciated from the plots a remarkable quality of the station keeping applied, which is generally the same for all the simulations, with a positive outcome, executed in the Monte Carlo. The errors, both in position and velocity, exhibit a really low increment during the mission, demonstrating an effective orbit control. This can be appreciated in Fig. 7.8 and Fig. 7.7, where the mean values of position and velocity errors, $\delta\mathbf{r_c}$ and $\delta\mathbf{v_c}$, with the corresponding standard deviations, are reported.

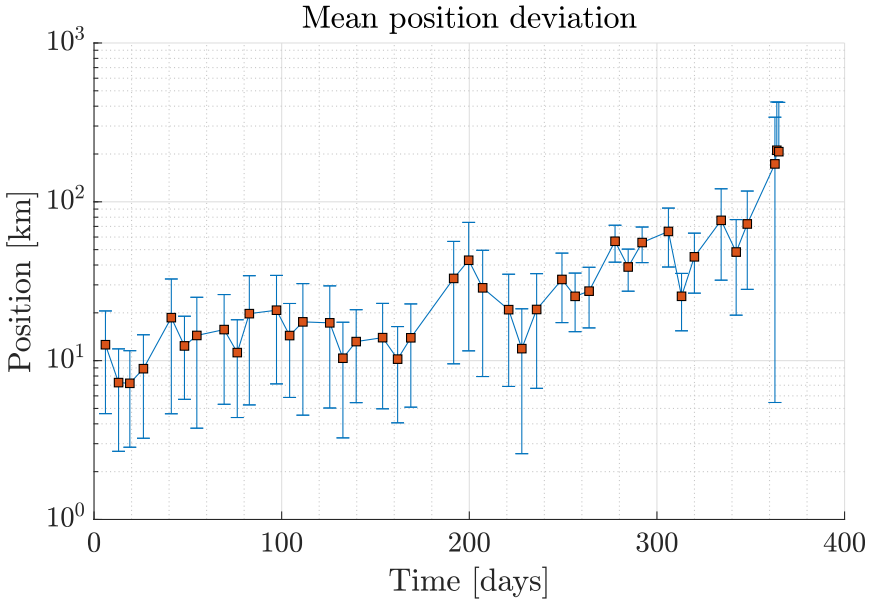


Figure 7.7: Position error from Monte Carlo.

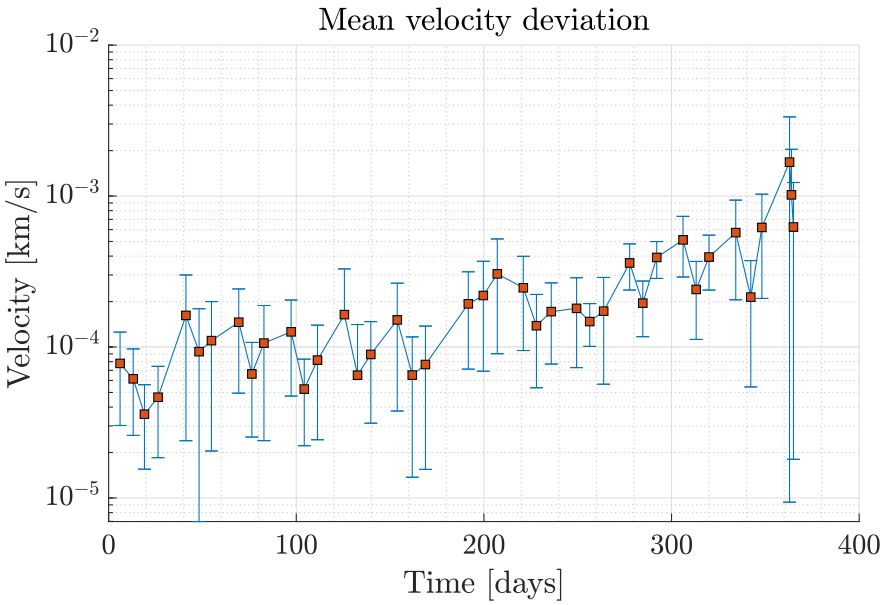


Figure 7.8: Velocity error from Monte Carlo.

The red squares represent the mean values, while the blue segments the standard deviations. As expected, the errors for both position and velocity increase as much the days pass, due the accumulation of errors. Moreover, the final orbits feature a larger standard deviation, caused by the increment of uncertainties accumulated up to that date. From an accurate analysis, and displaying Fig. 7.8 in a linear y scale, it is possible to recognise a typical behaviour, magnified in Fig. 7.9, which verifies for most of the manoeuvres.

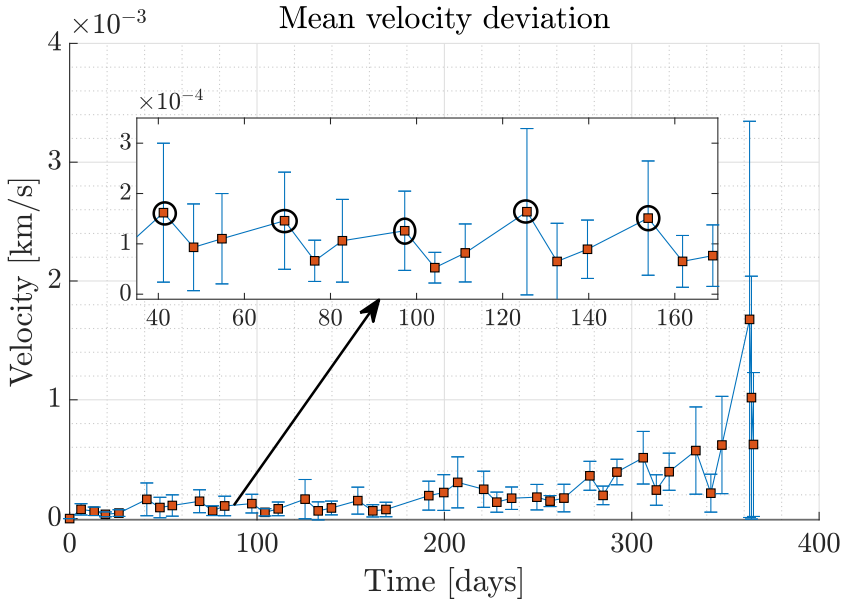


Figure 7.9: Velocity error magnification from Monte Carlo.

Fig. 7.9 shows in fact that, out of the 3 S/K manoeuvres happening during a Nav&Eng cycle, the first, marked with a black circle, is usually the one corresponding to a higher error, and uncertainty. This is coherent with the GNC operations; indeed, before that manoeuvre, a Science cycle with no control is performed. The second and third manoeuvres of the Nav&Eng cycles feature instead a lower velocity error, thanks to the previous corrections.

Unfortunately, a reduced $\Delta\mathbf{v_{SK}}$ cost and low trajectory dispersion solutions, are achieved only in the cases in which the navigation errors are not much bigger than those reported in Table 5.3, especially for ϵ_{OD}. This latter indeed, appearing at each manoeuvre, seems to heavily interfere with the convergence of the algorithm to a valid result for the whole 1 year. In particular, after some tests, it seems that the function which computes the $\Delta\mathbf{v_{SK}}$ is not sufficiently sensible to errors in the updated state: a little variation of this causes the computation of a too drastic manoeuvre by the algorithm. Generally speaking, modifying some of the simulation parameters, for the sensitivity problem mentioned, tend to determine a quick and uncontrollable change in the results, remarking the need to conduct further analyses on the guidance algorithm. A sensitive analysis can be performed keeping in mind the following factors:

- The cut-off time t_c, i.e. the time before the manoeuvre execution, can be increased up to 2 days to ensure a larger time window to perform GNC operations and meet the requirement formulated in [4]. However, this is certainly going to increase the accumulation of errors, hence the $\Delta\mathbf{v_{SK}}$ and the risk of a too large departure from the nominal path.
- The target points epoch t_i can be reduced to decrease the accumulated error, however, that time should not be lower than the duration of the Science cycles, which feature no orbit control.
- The weighting matrices can be tuned to reach more suitable values in manoeuvre cost or state dispersion. In particular the growth of Q determines a reduction of the $\Delta\mathbf{v_{SK}}$, while the growth of R_i a reduction in the state error. In the equation of the station keeping, being involved high order of magnitudes, particular care on ill-conditioning must be kept.
- The number of manoeuvres to execute at each Nav&Eng cycle could be increased by 1 if this could bring benefits to the overall costs and other performance parameters.
- As mentioned, the principal source of malfunction of the algorithm seems to be the OD error, appearing at each manoeuvre. Indeed, the OI error, or others, can be

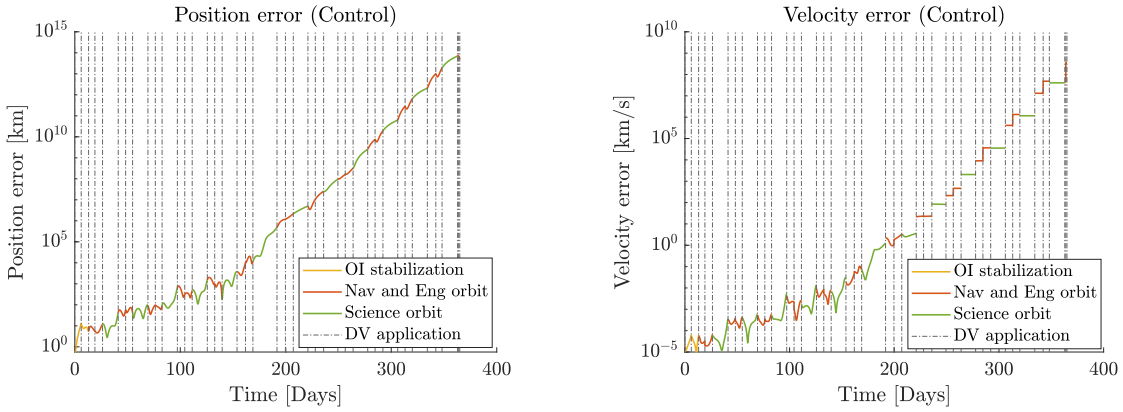
increased with no particular consequences, other than a corresponding increment of the $\Delta\mathbf{v_{SK}}$. A more refined analysis must be conducted to validate the algorithm, being sure that no parameter or source of error can be improved, before passing to modify the station keeping requirements and operations of LUMIO.

Further details are provided in Chap. 8.

Just to give an idea, the values of the parameters in Table 7.3 have been modified, to simulate a more arduous scenario in which the navigation errors are bigger. The orbit simulation errors are reported in Fig. 7.27a and Fig. 7.27b.

	ICs	**OI**	**OD**	**EX**
$\sigma_{x,y,z}$ $[km]$	1	2	1	-
$\sigma_{u,v,w}$ $[m/s]$	0.01	0.02	0.01	3%

Table 7.3: Larger σ values of the errors employed in the algorithm.



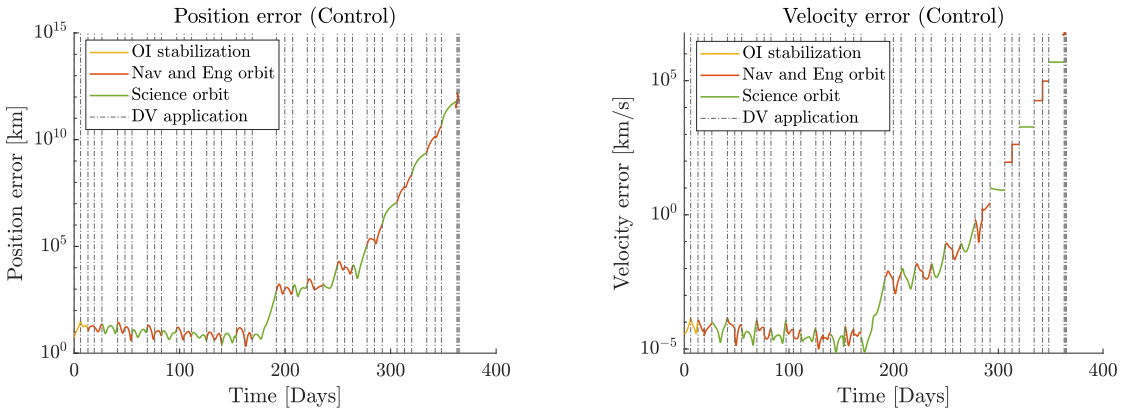
(a) Position error.

(b) Velocity error.

Figure 7.10: Trajectory dispersion with larger navigation errors.

It can be noticed that the error is still kept low for about 150 days, then the station keeping is no more working. Anyway, even for those days during which the control works fine, the total $\Delta\mathbf{v_{SK}}$ is quite larger to that obtained with lower navigation errors.

Considering the case of $\Delta t_c = 2$ days, selected in [4], a feasibility analysis has been conducted during 1 year of mission. The increment of the cut-off time allows to calmly prepare the GNC operations, although increasing the errors deriving from the computation of the $\Delta\mathbf{v_{SK}}$. In Fig. 7.11 the orbit errors obtained with a $\Delta t_c = 2$ days are reported. As in Fig. 7.10, the correct path results to be followed for a bit more than 150 days before the error grows dramatically.



(a) Position error. (b) Velocity error.

Figure 7.11: Trajectory dispersion with $\Delta t_c = 2$ days.

7.2. Attitude control and firing

The manoeuvres magnitude and direction serve as input to determine the attitude control action to apply to perform the station keeping during the mission. In the following, all the results related to the GNC operations are reported, subdivided to visualise the details of a single manoeuvre or a series of those.

7.2.1. Single manoeuvre detail

In order to clearly display what really happens during each manoeuvre, in the following, the plots related to the detail of a single manoeuvre are reported. Fig. 7.12 displays the pointing error, defined as the alignment error in the x-axis (see Eq. (6.1)), from the start of the simulation to the first S/K manoeuvre. The pattern of colour used, and the operations happening, are the same as in Fig. 6.4. Hence, after the OI phase, in yellow, when the nominal attitude is acquired, LUMIO maintains the Moon pointing up to the the first S/K manoeuvre. This latter, considering that the start of the simulation happens half day before, is scheduled exactly at 0.5 days from the start. In Fig. 7.13 it is magnified what happens during the first S/K manoeuvre operations, in a neighbourhood of the firing epoch. The three phases characterising each manoeuvre are recognised: the correction to get the manoeuvre attitude, the firing of the thrusters, and the re-alignment to recover the nominal attitude condition. In particular, the first blue peak verifies ΔT_1 s before the firing, while the second one ΔT_2 s after the firing, as described in Sec. 6.1. The error stabilises at about 0.008 deg half cone, which is considered to be enough accurate at this stage, also considering [6]. The settling time after each attitude change is less than 15 minutes, however it can be reduced if it will be needed by future requirements.

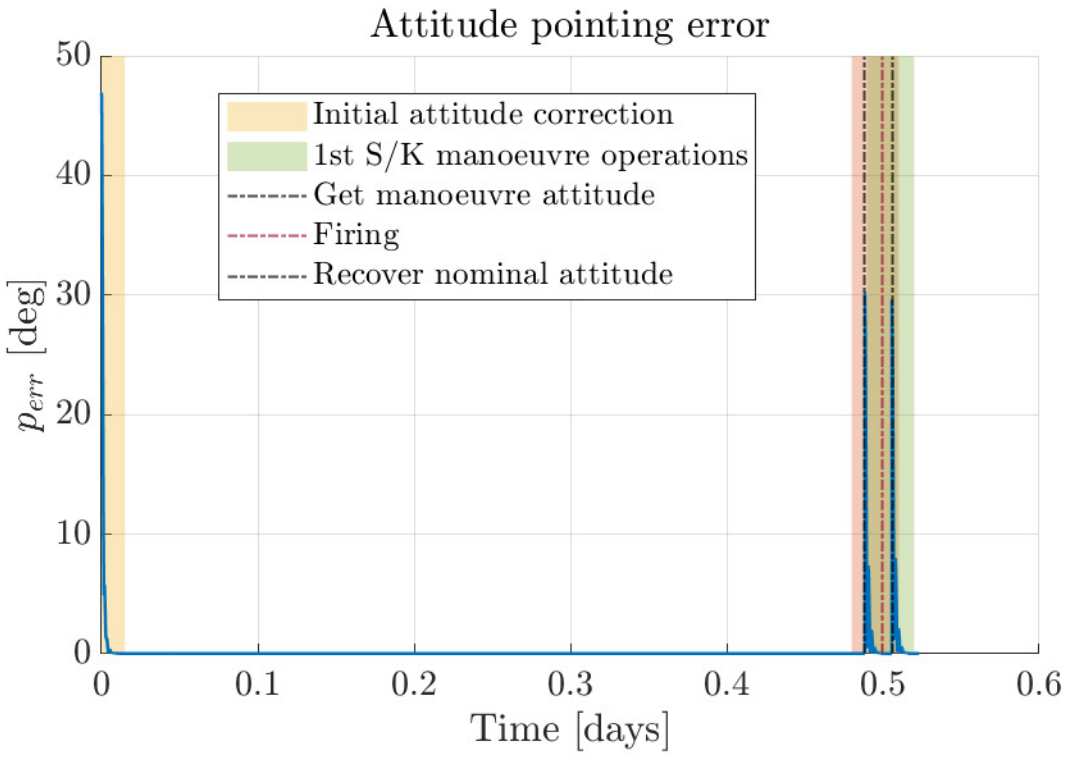


Figure 7.12: Attitude corrections up to the first manoeuvre.

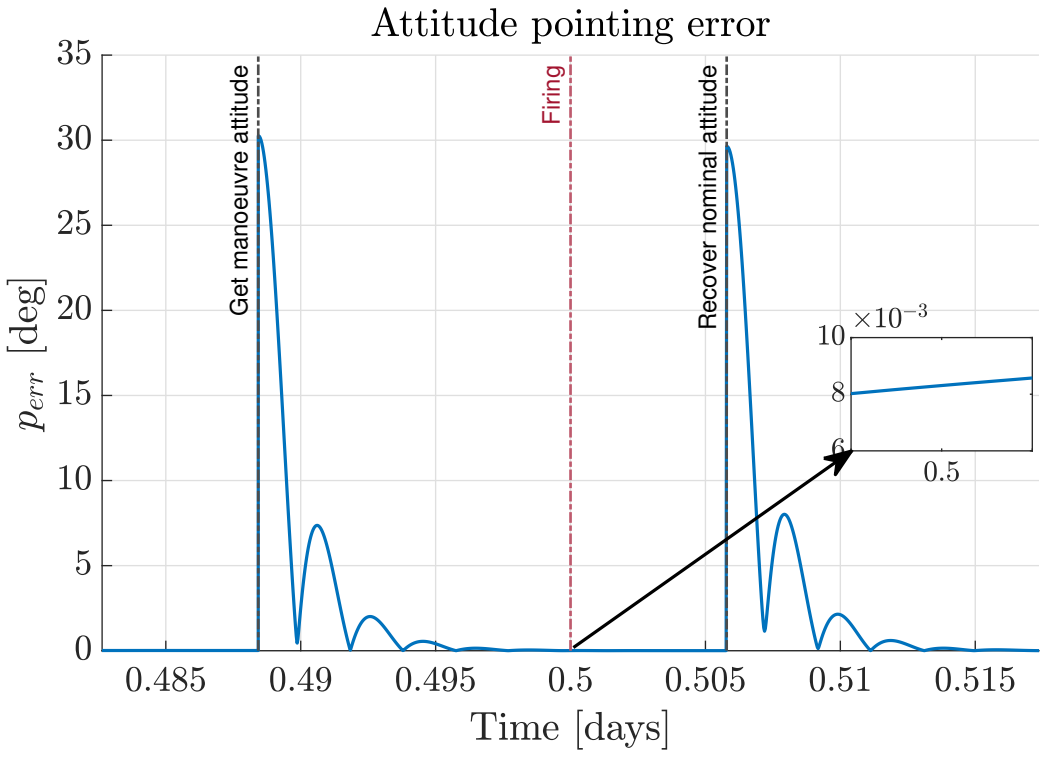


Figure 7.13: First S/K manoeuvre magnification.

Fig. 7.14 is obtained with different initial conditions, to show the attitude control action in a different situation. In particular, a larger initial error has to be corrected in this case.

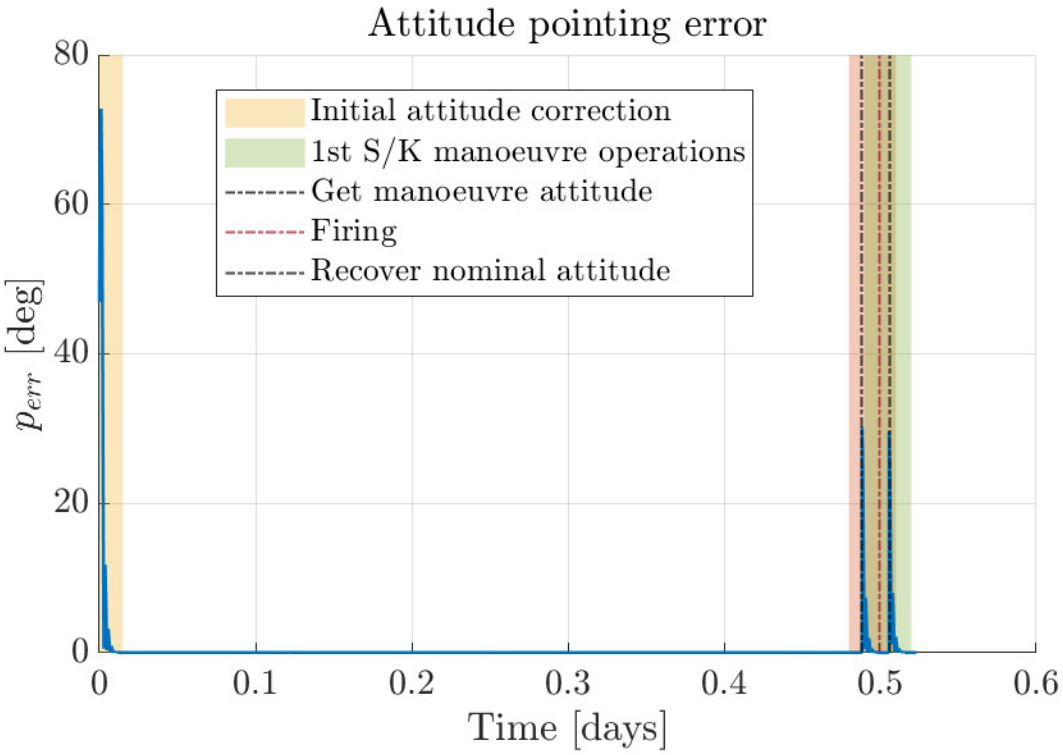


Figure 7.14: Attitude corrections up to the first manoeuvre, with different ICs.

From here to the end of the section, all the plots will be focused on the details of the S/K manoeuvre operations; therefore they are magnified in correspondence of the firing epoch, at 0.5 days from the start of the simulation.

Fig. 7.15 shows the components of the angular velocity of the spacecraft in the 2 slews happening before and after the first S/K manoeuvre.

In Fig. 7.16 it is reported the ideal torque required to perform the manoeuvre, derived by the control law, while in Fig. 7.17 the actual one delivered by the reaction wheels, incorporating the noise. Their norm is compared in Fig. 7.18.

Fig. 7.19 and Fig. 7.20 show the angular momentum h of the reaction wheels and the rpm reached during the first orbit correction. h in particular is a vector with four elements, as the number of actuators, and represents the angular momentum of each wheel around its spin axis. It can be noticed that the saturation levels of the reaction wheels, in Table 4.4, are not reached, hence a desaturation is not required in this case.

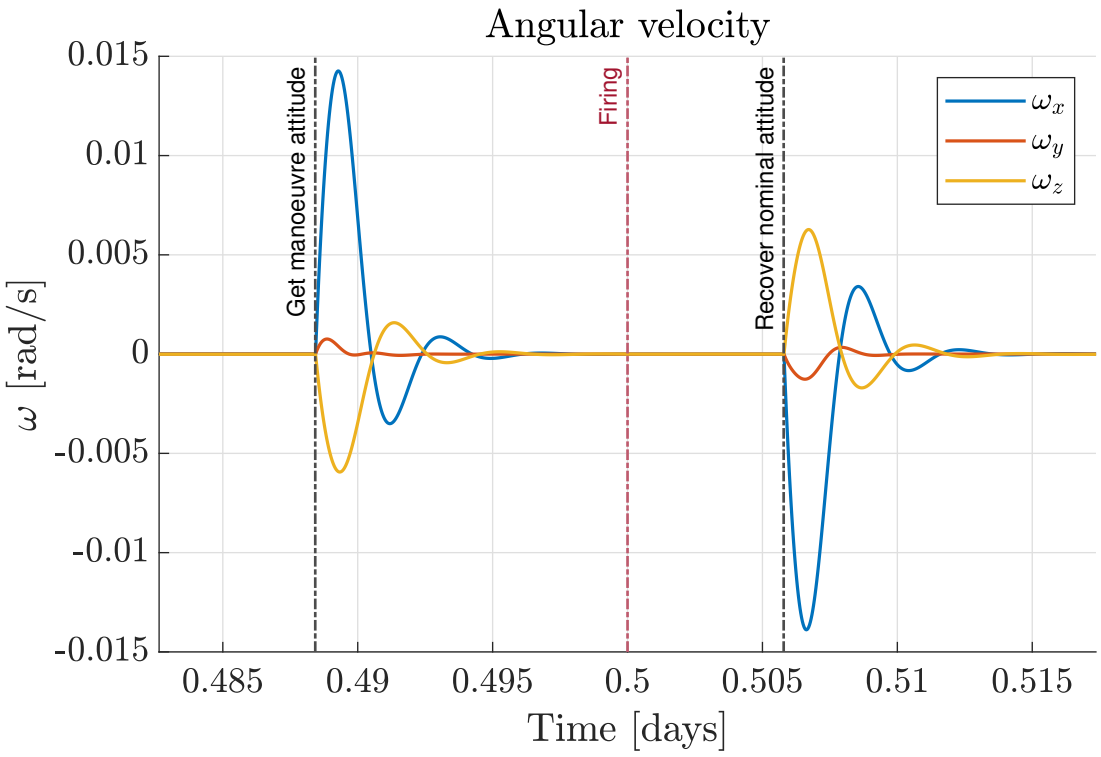


Figure 7.15: Angular velocity components during the first S/K manoeuvre.

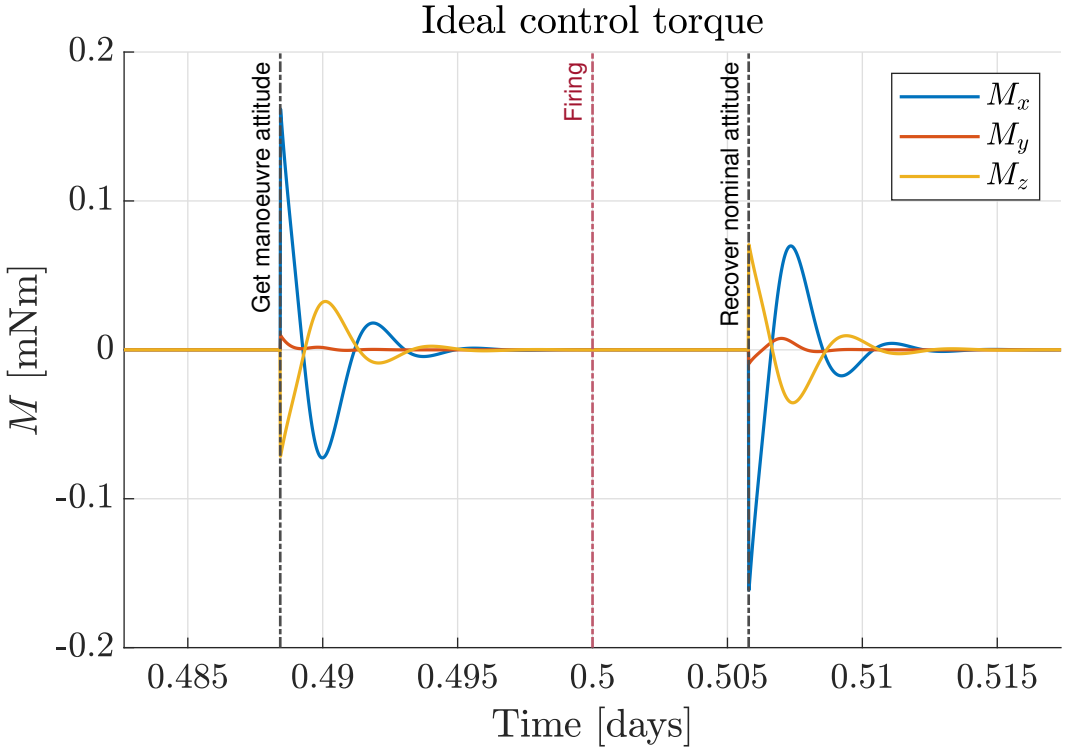


Figure 7.16: Ideal control torque during the first S/K manoeuvre.

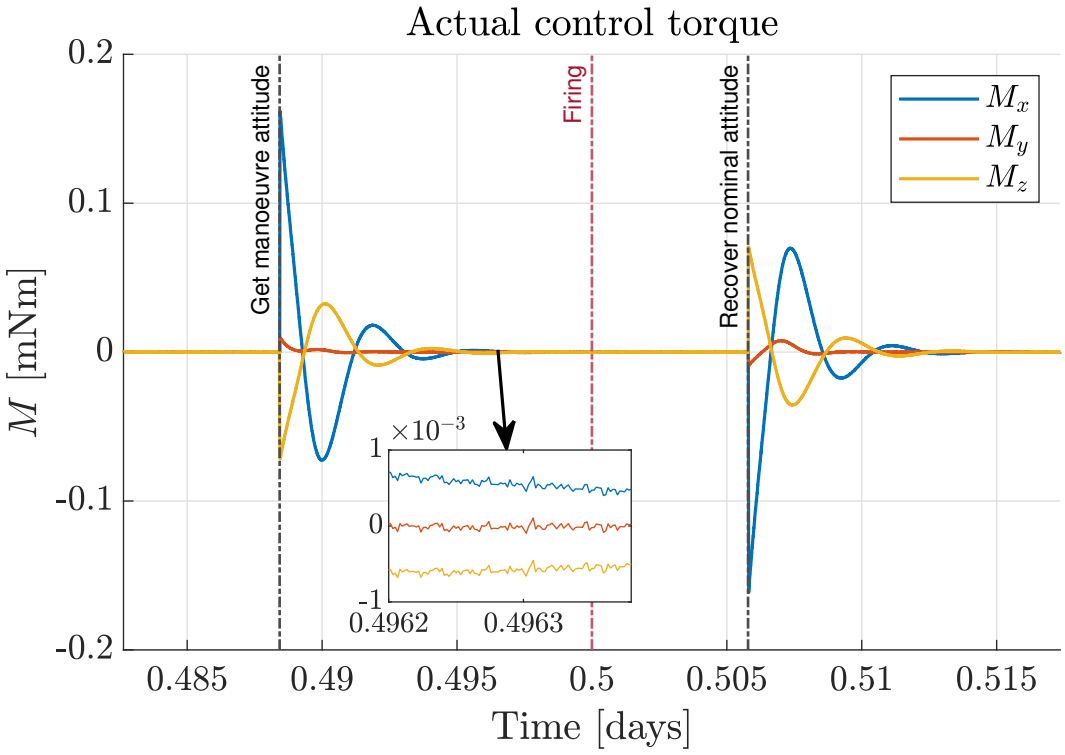


Figure 7.17: Actual control torque during the first S/K manoeuvre.

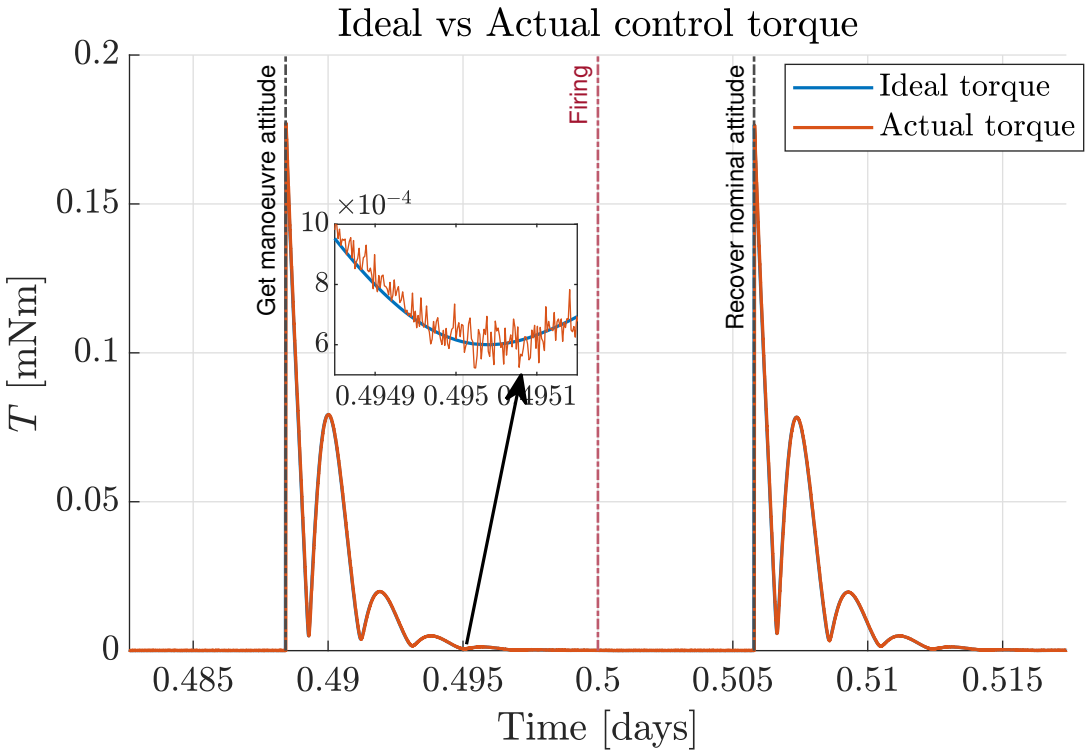


Figure 7.18: Ideal vs actual torque during the first S/K manoeuvre.

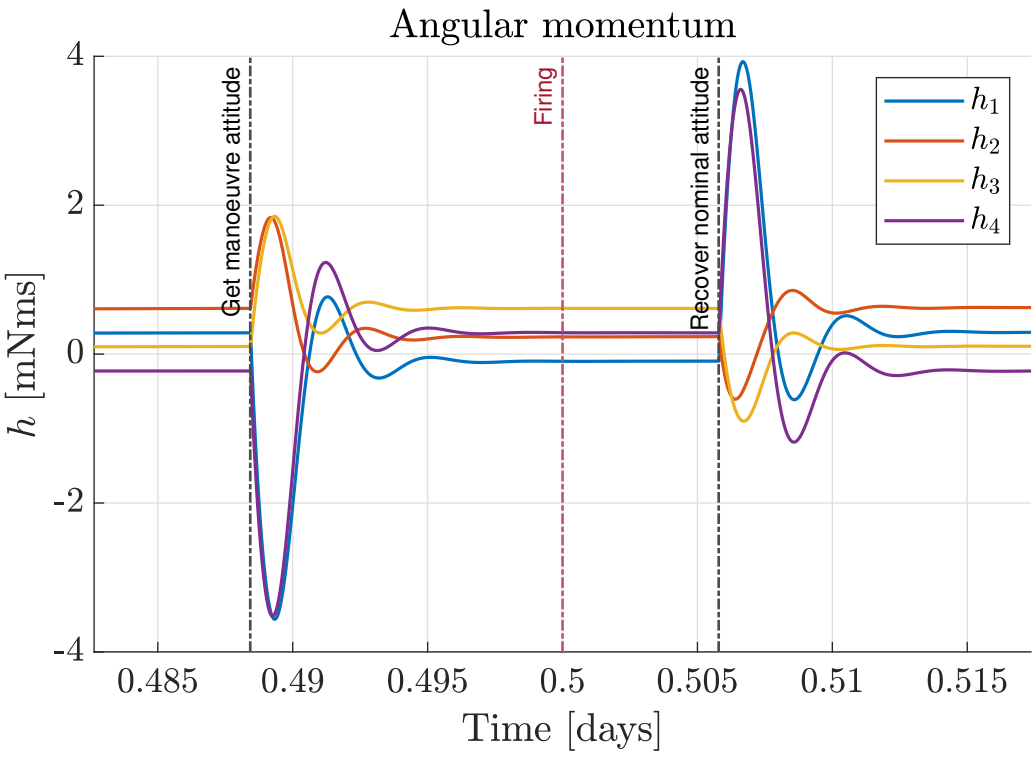


Figure 7.19: Wheels angular momentum during the first S/K manoeuvre.

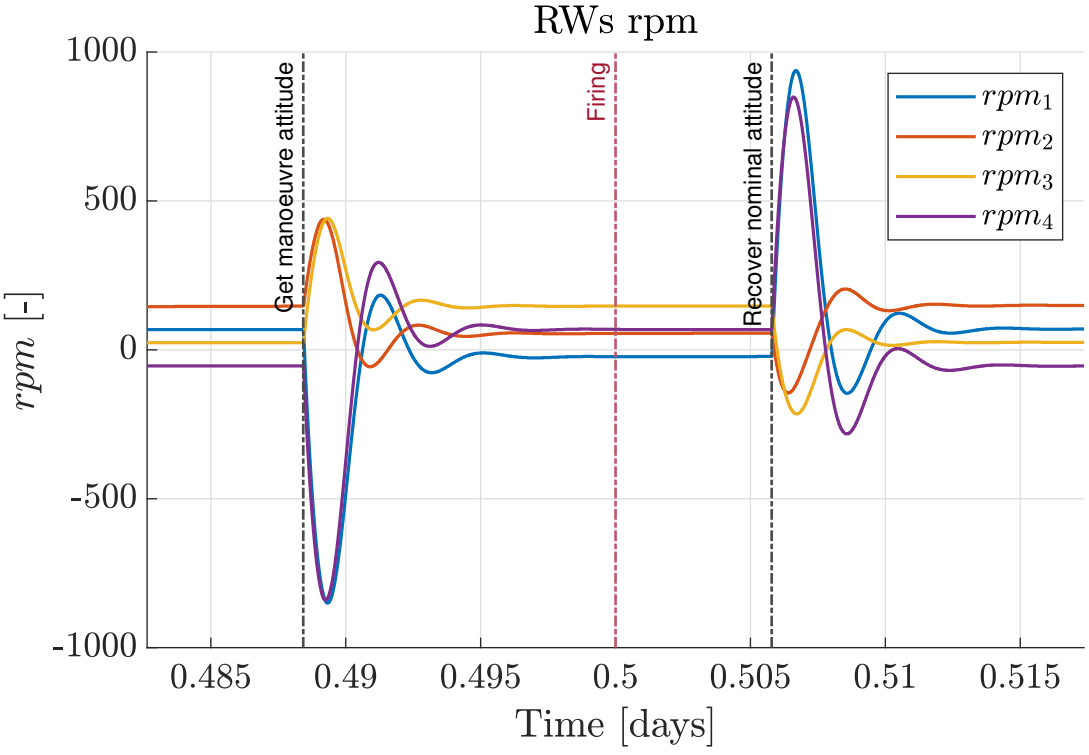


Figure 7.20: *rpm* of the wheels during the first S/K manoeuvre.

As for the thrusters simulation, it is executed with a step size of 0.15 s (see Table 3.2), in order to catch each firing event, considering the minimum duration in Eq. (5.7). This considered, the following plots are usually affected by a coarse profile, being the simulation step size almost as long as the duration of the dynamics itself. Reducing the step size allows to simulate each manoeuvre more accurately, although considerably increasing the simulation time.

The thrusters pointing error during a sample manoeuvre, with respect to the ideal thrust direction, is represented in Fig. 7.21.

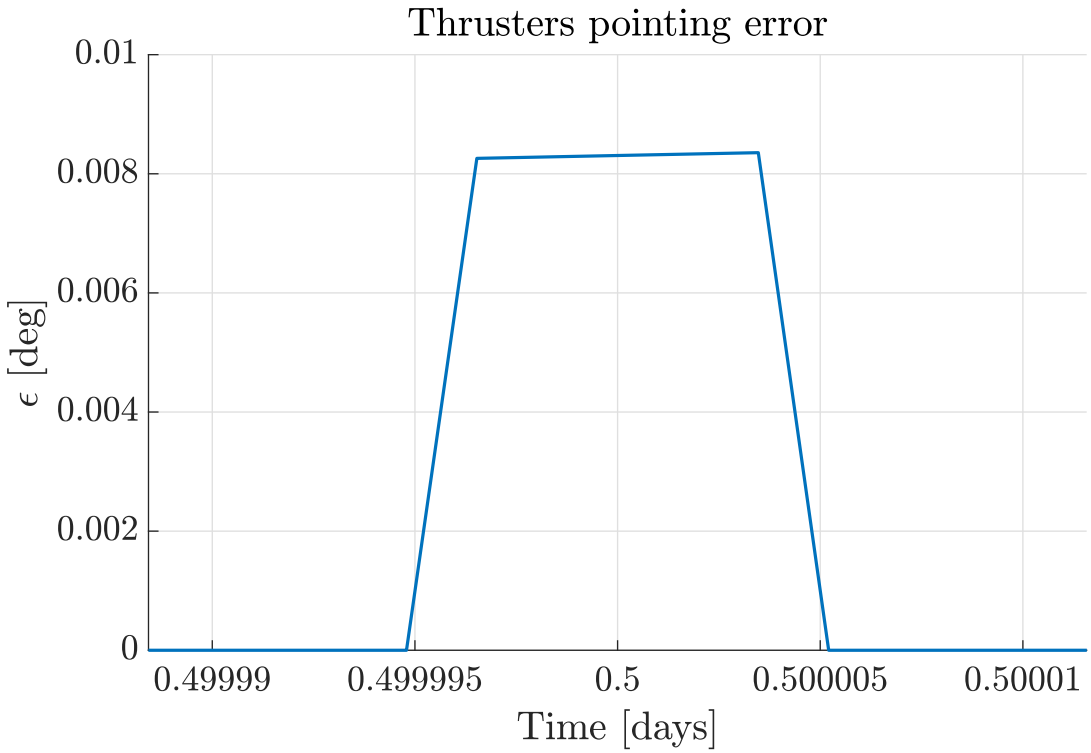


Figure 7.21: Thrusters pointing error during the first S/K manoeuvre.

The thrust direction error hovers around thousandths of a degree, ensuring a precise pointing. In particular, the illustrated manoeuvre lasts for 0.6886 s.

The comparison between ideal and actual thrust in the inertial frame is reported in Fig. 7.22. Fig. 7.23 displays only the components of the output thrust delivered by the thrusters, in the inertial frame. Remind that each manoeuvre features a total thrust with magnitude

$$T_{tot} = 2T_{max} = 2 \times 0.245 = 0.490 \ N$$

due to the presence of two thrusters.

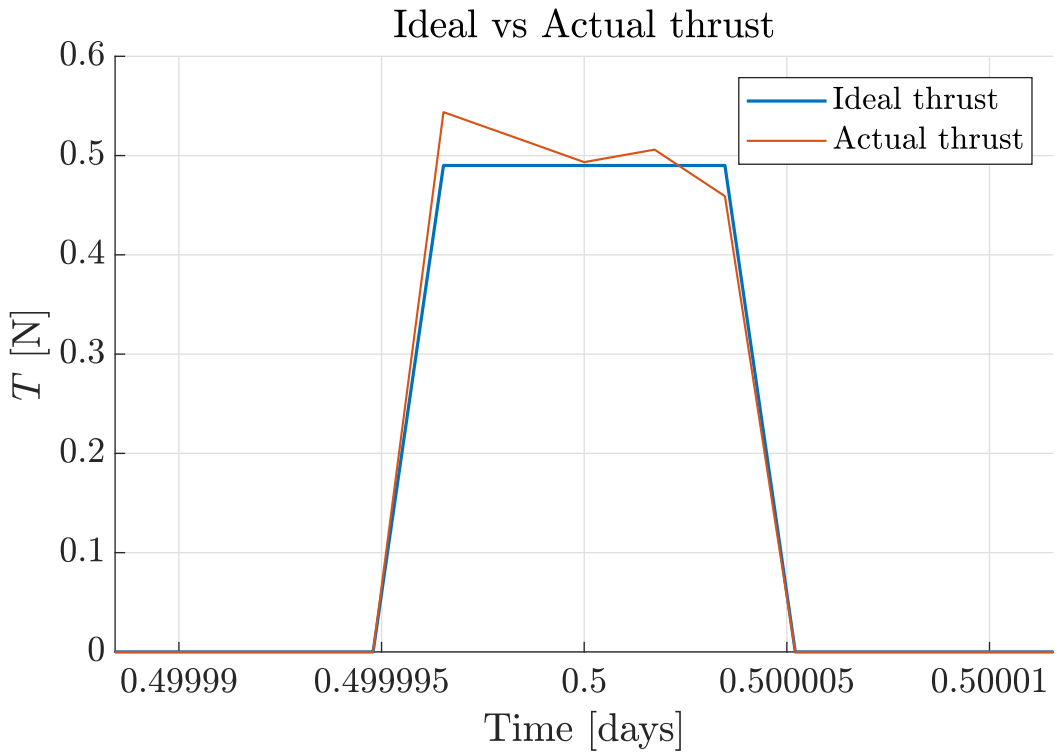


Figure 7.22: Ideal vs actual thrust during the first S/K manoeuvre.

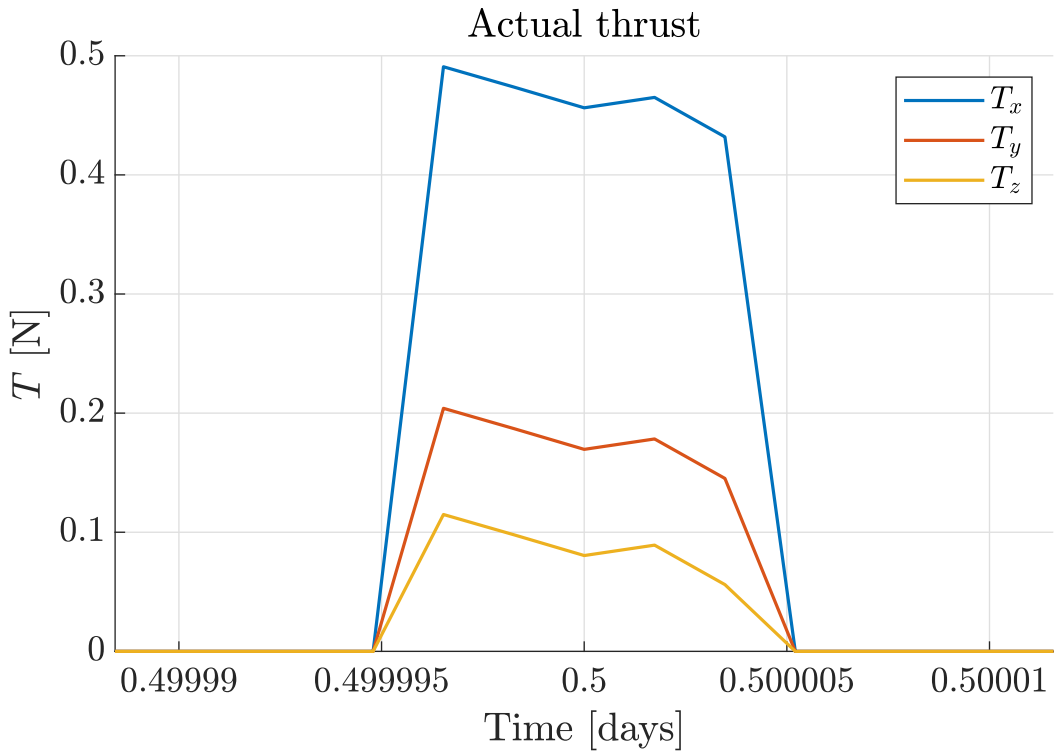


Figure 7.23: Thrust provided during the first S/K manoeuvre.

7.2.2. Series of manoeuvres

In the case in which the simulation is performed for a long period, CUBORG automatically simulates the manoeuvres execution according to the their date and the corresponding $\Delta\mathbf{v_{SK}}$. The simulation time can be theoretically elongated for the 1 year operative duration of the mission, to evaluate the effect of the overall station keeping. In practice, this requires long computational time and other problems related to integration errors, discussed at the end of this chapter, and more in detail in Chap. 8.

In Fig. 7.24 the first 5 manoeuvres, for a sample 1 year station keeping, are reported, as perceived by the activation command of the thrusters, i.e. 0 for no action, 1 for firing. Their date corresponds to the first five in Table 5.4, from the OI stabilisation orbit and the first Nav&Eng one; it must be recalled that the simulation starts at 2024-MAR-21-00:00:00, hence half day before the first manoeuvre (see Table 3.3).

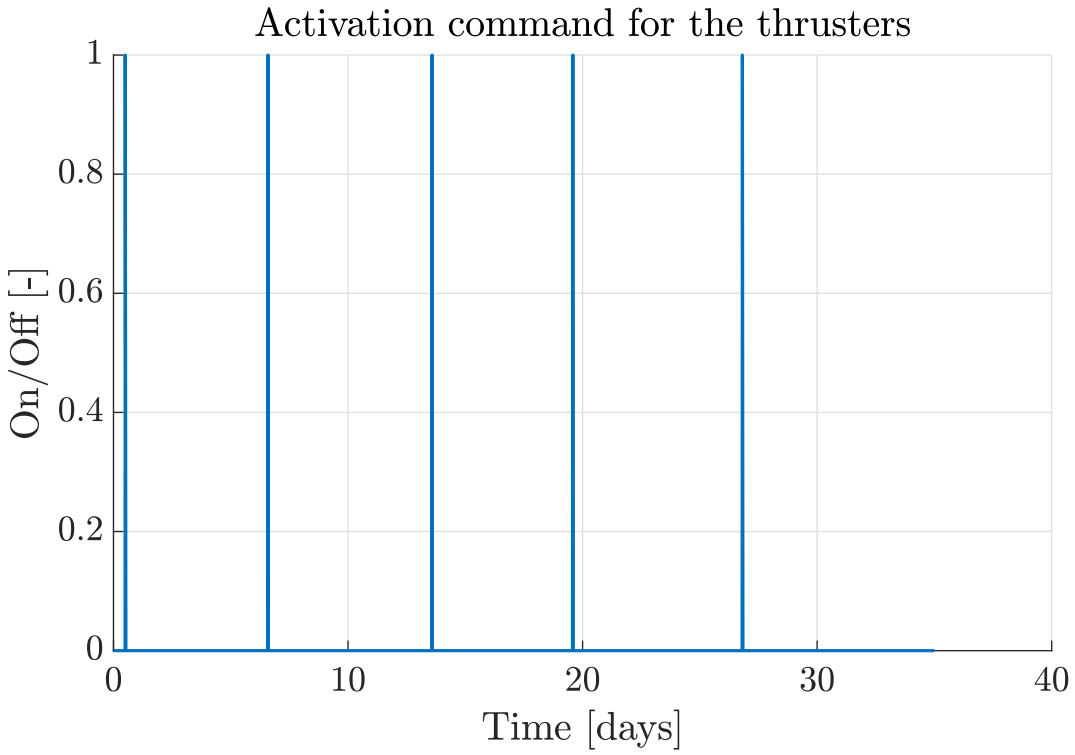


Figure 7.24: Thrusters command during the first 5 S/K manoeuvres.

Fig. 7.25 shows the thrust profile during these 35 days, while Table 7.4 the duration of the 5 firings. It can be noticed that the duration of the second manoeuvre is quite long, due the higher $\Delta\mathbf{v_{SK}}$ the thrusters have to provide.

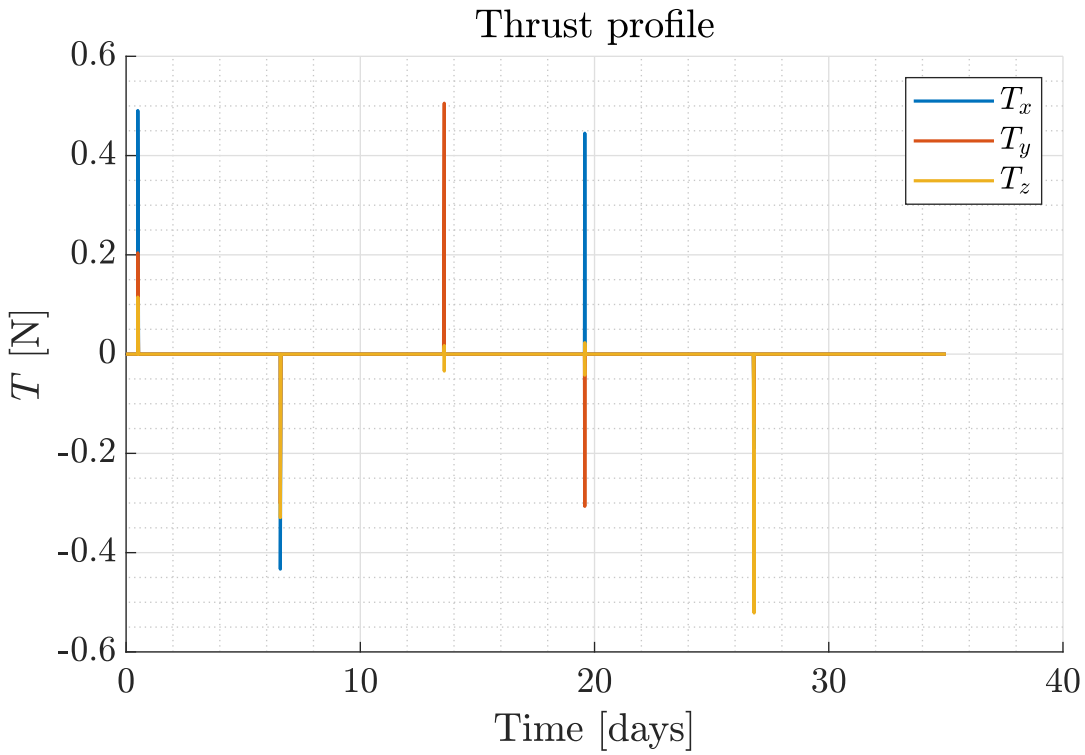


Figure 7.25: Thrust during the first 5 S/K manoeuvres.

	S/K $n°1$	**S/K** $n°2$	**S/K** $n°3$	**S/K** $n°4$	**S/K** $n°5$
Duration $[s]$	0.6886	9.0211	0.4739	0.9819	0.2181

Table 7.4: Firing duration of the first 5 S/K manoeuvres.

The effect of these 5 manoeuvres on the orbital path is reported in Fig. 7.26, the respective position and velocity errors in Fig. 7.27. It is clear that the station keeping manoeuvres, applied in this case by CUBORG, do not provide the desired result; indeed, the trajectory error, with respect to the nominal path, keeps increasing, and it is no more acceptable after 20 days.

This behaviour is mainly due to a problematic: the set of $\Delta\mathbf{v_{SK}}$, obtained in MATLAB, is derived with a different dynamics with respect to that propagated by CUBORG. First of all, the integration settings are different, due to CUBORG necessities; then the effect of the manoeuvres on the orbit is impulsive, in MATLAB, whereas of finite duration in Simulink. These, and other differences summarised in Table 7.5, contribute to the accumulation of errors in the propagation, which leads to an inevitable departure from the reference path after the first orbits. This, as mentioned many times, is of course amplified by the unstable nature of the halo, extremely sensitive to any source of error.

	MATLAB	Simulink
Ephemeris discretization	A max step size has been set for the integration. This determines the length of the intervals, different at each step, by which the ephemeris are retrieved from SPICE.	The ephemeris are discretized with the fixed, constant, step size in Table 3.2. In order to get an high accuracy of the integration, this step size must be kept short.
Solver type	A variable-step solver is employed, to have a faster simulation.	Although a variable step would remarkably improve the computational efficiency, a fixed-step is utilised to catch the firing events.
Solver	*ode113-Adams* is selected to perform the integration: it guarantees accurate results with low computational effort.	*ode2-Heun* is used as baseline solver in Simulink, due to its higher efficiency in terms of computational speed and accuracy, in the case of a fixed-step integration.
Dynamics equations	They are characterised by an instantaneous change of the velocity vector, due to the assumption of impulsive S/K manoeuvres.	The manoeuvres are executed in a finite time interval, during which the thrusters deliver a thrust, which sums up to the total force entering in the dynamics.

Table 7.5: Differences between MATLAB and Simulink models.

Those reported represent the main sources of the error which verifies when trying to insert the S/K MATLAB solution in Simulink. In order to make the station keeping working also in Simulink, over the 1 year duration, two main approaches can be selected: reduce the gap between MATLAB and Simulink propagation, considering what is reported in Table 7.5, or implement the computation of the manoeuvres in the CUBORG closed-

loop. In this last case, the corrections will be determined accordingly to the CUBORG trajectory errors, hence their effect should be fine to conduct the station keeping. The details of the future developments related to this topic are presented in Chap. 8.

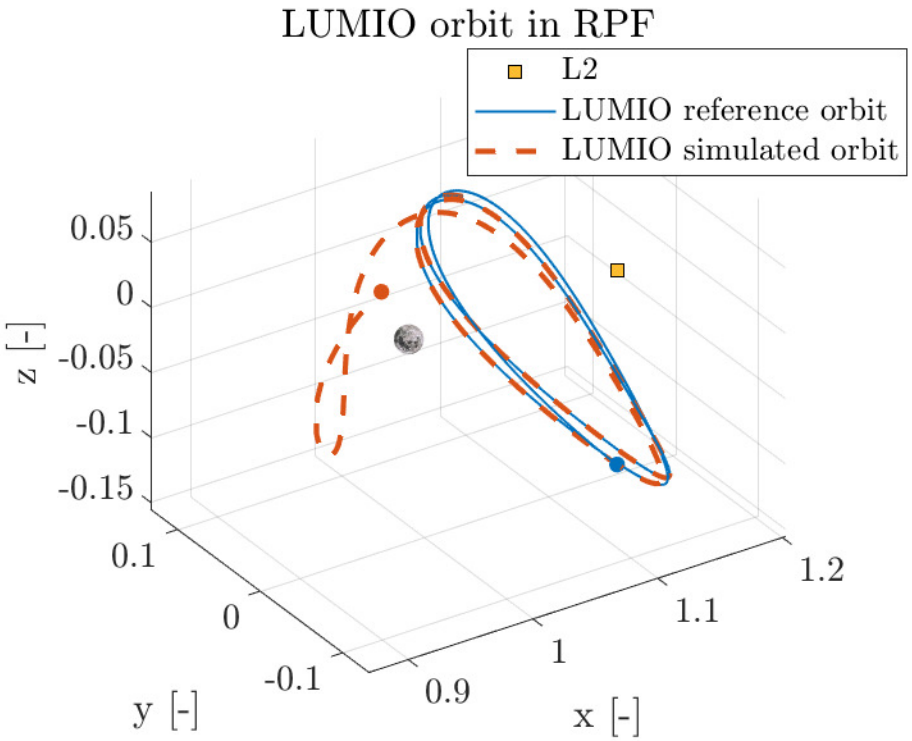
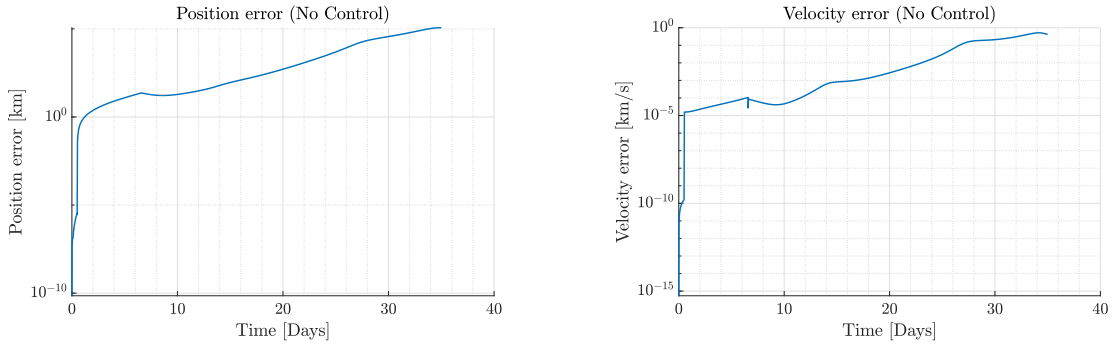


Figure 7.26: Orbital path during the first 5 S/K manoeuvres.



(a) Position error. (b) Velocity error.

Figure 7.27: Trajectory errors during the first 5 S/K manoeuvres.

It can be noticed, although being quite irrelevant to the overall result, a positive effect of the first manoeuvres, especially in the velocity error plot, having the effect of reducing the error when applied. However, the station keeping algorithm relies on an optimal control,

in which the solution is determined by minimisation of both the $\Delta\mathbf{v_{SK}}$ and the state error. This means that it is not necessarily meaning full to notice a reduction of the state error to determine the effectiveness of the S/K: the combined result of state error and $\Delta\mathbf{v_{SK}}$ shall be considered. For this reason, the only possibility for the S/K validation relies on the effective orbit control over a longer period, providing acceptable trajectory dispersion and moderate $\Delta\mathbf{v_{SK}}$ cost. This constitutes the main goal for a future development.

8 | Conclusions and Future work

"It was tough,
but I got through it."

- Myself

The work developed in the thesis has been divided in two main parts, with different purposes: the computation of the station keeping manoeuvres to be executed during the mission, and the simulation of GNC operations happening before and after each manoeuvre. The results have demonstrated the effectiveness of the station keeping computed and applied, for the whole 1 year operative phase of the mission. The $\Delta\mathbf{v_{SK}}$ computed are employed to run a 6 DOF simulation in CUBORG, with the purpose of validating the GNC operations happening for each manoeuvre, from the thrusters orientation, to the firing itself.

The proposed results represent a solid baseline for the future work in the models refinement. Considering the large complexity of the problem, and all the related details which affect the solution, the work presented in this thesis aims at creating an initial version of the GNC model for LUMIO which can be easily modified to be compliant with future requirements. Indeed, the codes created are designed to work in the wider spectrum of cases possible, just requiring a change in the initial settings.

The station keeping results, analysed in Chap. 7, and there widely commented, show effective solutions in the majority of the low navigation errors simulations. As much these errors increase, as much the station keeping fails for the entire operative mission duration, resulting successful just for a portion of it. A similar situation holds for the cut-off time Δt_c: increasing its duration generally determines an accumulation of errors with consequent divergence of the orbit at a certain epoch. Of course, once a too large error is reached, the algorithm cannot resolve it, hence the situation just gets worse.

Other parameters can be modified in the algorithm to try reaching a better convergence,

as the weighting matrices or the target points epochs. However, both these have been modified to try to reach better solutions, although without good results. For this reason the values in Table 5.1 and Table 5.2 have been maintained as baseline.

Moreover, the algorithm implemented appears to be too sensitive to the navigation errors, but not enough for the determination of a moderate, progressive, $\Delta\mathbf{v_{SK}}$. This behaviour of the solution is, from a point of view, comprehensible, considering the intrinsic instability of the halo orbit, from the other unacceptable if applied to a realistic scenario in which the navigation errors are certainly higher.

The current work foresees the computation of the manoeuvres $\Delta\mathbf{v_{SK}}$ out of the loop. This strategy, still valid, allows to have a faster and more manageable computation for the solely $\Delta\mathbf{v_{SK}}$. The chance of implementing the manoeuvres magnitude and direction in the CUBORG closed-loop has revealed, during this work, to be quite useless, for the simple reason that the simulation time would be increased and no improvement would be made. Indeed, the set of $\Delta\mathbf{v_{SK}}$ to execute during the mission is non other than a constant matrix containing those vectors. That being said, although its computation out of the loop represents the most computationally efficient implementation strategy, it could be the case, in the future developments, for it to be no more suitable. This is linked to what expressed at the end of Chap. 7: to be sure that the manoeuvres computed provide an effective station keeping, they may require to be computed in the CUBORG closed-loop, according to its models and integration scheme.

As far as the GNC operations are concerned, i.e. the manoeuvre execution itself through the thrusters orientation towards the $\Delta\mathbf{v_{SK}}$, the results are encouraging and, for some cases, above expectations. Indeed, the Simulink model developed is able to automatically simulate all the manoeuvres of the mission in a single run of the software, with a high precision attitude control. As described in Chap. 6, the CUBORG tool embeds now a dedicated model for ADCS-GNC simulations, which handles both pre and post-manoeuvre operations. The only problematic is related to the issue, already discussed, of the manoeuvres effect on the orbit; indeed, instead of adjusting it, they are a cause of departure from the nominal path. Possible solutions and future developments are discussed in Sec. 8.3.

To conclude, the research questions are now answered, on the basis of the thesis results. The first point to answer was:

What are the requirements of the guidance, navigation, and control (GNC) system needed to control the trajectory of LUMIO?

The work presented has allowed to formulate a set of requirements, listed in Appendix C, for the GNC system, most of which have been verified or addressed with the simulations performed. In particular, their application to the algorithms employed in the simulations, has provided awareness on the feasibility of the station keeping, clarify what can represent a source of risk or what provides benefits. Although some further verification must be conducted, a solid basis is now established for the future progresses.

The second research question was:

> *To what extent can we model and simulate the coupling between GNC and ADCS systems of LUMIO in closed-loop during the operational phase of the mission?*

The goal, in this case, was to develop a tool able to simulate the whole set of AOCS operations which must be performed during the 1 year station keeping. The work done has, for the most part, answered the question: once the manoeuvres are determined by the apposite algorithm, the series of pre and post-manoeuvre operations are automatically simulated by CUBORG, according to the GNC schedule. The results are encouraging and feature highly accurate solutions. By refining the application of the manoeuvres itself, which constitutes the main actual limitation, the Simulink model dedicated to AOCS operations can be considered concluded.

8.1. GNC requirements verification

Those reported in Appendix C represent the final version of requirements reached in this work. This means that the initial list formulated, at the beginning of the thesis, collecting the firsts, general requirements for the GNC, has then been updated with the results obtained. In order to provide a brief explanation of this process, Appendix C presents the major changes made on the requirements during the work.

8.2. Computational efficiency

The computational time is one of the main factors considered to drive the models design. Many trials have been performed and many refinements carried out before arriving at a good efficiency.

The simulations and the data reported have been obtained with a 1.3 GHz Intel Core i7, 8th Gen, Windows machine.

As for the MATLAB part of the work, most of the codes exhibit an instantaneous exe-

cution. Those which require the longer computational time are related to the integration of the orbital quantities; however, although this time is still very low, a reduction of the integration tolerances can be done to decrease it even more, without affecting significantly the accuracy of the results. The factor which mostly determines a slow down, of course, is the reduction of the integration step-size; however, it has been proved that reducing it too much does not determine a considerable higher accuracy.

The highest problematic of computational time have been advised in the Simulink context, as expected. Most of them have already been fixed; they were related to different reasons:

- The allocation memory occupied by the outputs during the saving process. This issue has been solved saving the whole set of outputs, which are not useful in the long simulations, only if the simulation time is short[1], such to ensure to have not too heavy data saved. In the case in which the simulation time is higher than the threshold, the only variable saved is the state of the trajectory, which is useful to create a final plot of the orbit. Moreover, this could be saved with a low frequency, to reduce even more the output variable size, which risks to grow a lot for long simulations.
- Two versions of the SRP torque have been implemented, one completely in Simulink, but slower, and one with a MATLAB function, faster and currently employed by the model. This little change allowed to moderately speed up the simulation.
- The choice of the integration scheme and settings is of primary importance in the computational efficiency. The initial choice was a variable-step solver, automatically selected by the software. The speed reached by this set up was remarkable, but had to face with a problem. This problem is related to the thrusters action, which often verifies in less than a second. This imposes the constraint of choosing a fixed-step solver[2] with 0.15 s step size, to ensure a proper simulation of such a short dynamics. As it can be imagined this reduces the efficiency of the simulation, which slow downs drastically, and represents one of the main issues to solve in the next developments.

8.3. Recommendations for future work

As mentioned before, the work done presents two main areas which can be improved: the station keeping algorithm accuracy, and its validation within CUBORG, also associated

[1] In this work it has been selected minor than 51840 s, during which a manoeuvre can be simulated entirely.

[2] ode2 is selected in the current model, having verified its higher efficiency with respect to others.

to the problem of the computational time of the Simulink tool.

The refinement of the station keeping algorithm should be mainly focused on a further validation of the state transition matrix, to understand if it can be obtained in a more accurate version. Indeed, although it being already validated and tested in the simulation, it does not coincide with the original *STM*, associated to the nominal orbit.

Moreover, a detailed sensitive analysis can be conducted with each of the parameters involved in the algorithm, from the weighting matrices to the navigation errors, to the cut-off or target points epochs. In particular, as it has been demonstrated with some examples, it is useful to take the reference values in Chap. 4 and slightly modify them, verifying if a better solution is reached.

As for the validation of the station keeping within CUBORG, further analyses must be carried out to understand how to make it work in the closed-loop simulation. Indeed, in the current version of the work, the manoeuvres are computed in MATLAB and then inserted in CUBORG, in two different processes. This leads to the problems of orbit departure analysed in Sec. 7.2.2, due to the main reasons in Table 7.5. In order to solve this issue, it could be proceed either by reducing the gaps between MATLAB and Simulink integration, to make the $\Delta\mathbf{v_{SK}}$ compliant with the CUBORG propagation, or move the computation of the manoeuvres from MATLAB to Simulink. In the first case, problems related to step size and dynamics modelling are likely to be faced, and probably cannot be solved. Following the second approach, instead, could allow to have new correct $\Delta\mathbf{v_{SK}}$, although at the price of slowing down even more the simulation. Moreover, in the case in which the accuracy of CUBORG is not sufficiently high, the station keeping cost could increase out of the tolerable limits. In all cases, the computational time represents one of the main obstacles to face with. It can be slightly improved by accurately tuning the integration parameters of the simulation, even though no amazing result is expected, being the simulation time bounded by the duration of the thrusters dynamics. Instead, a better condition could be reached by implementing a new system for the thrusters simulation, in which the step size is manually reduced for the solely duration of the firing. In this way, it would be possible to use a variable-step solver when the manoeuvres are not performed, remarkably increasing the simulation speed and, probably, the integration accuracy.

A | Appendix: Mission objectives

The mission objectives of LUMIO are herein presented. The list of top-level objectives is shown in Table A.1. Then, the list of the mission objectives follows in Table A.2. Finally, the tech-demo objectives are presented in Table A.3 [1].

Table A.1: Top-level objectives.

Objective ID	Objective	Stakeholder
TLO.01	To perform remote sensing of the lunar surface and measurement of astronomical observations not achievable by past, current, or planned lunar missions.	ESA
TLO.02	To demonstrate deployment and autonomous operation of CubeSats in lunar environment, including localisation and navigation aspects.	ESA
TLO.03	To demonstrate miniaturisation of optical instrumentation and associate technology in lunar environment.	ESA
TLO.04	To perform inter-satellite link to a larger Lunar Orbiter for relay of data and for TMTC.	ESA

Table A.2: Mission objectives.

Objective ID	Objective	Stakeholder
MO.01	To conduct observations of the lunar surface in order to detect meteoroid impacts and characterise their flux, magnitudes, luminous energies, and sizes.	ESA
MO.02	To complement observations achievable via ground-based assets in space, time, and quality in order to provide a better understanding of the meteoroid environment.	ESA

Table A.3: Tech-demo objectives.

Objective ID	Objective	Stakeholder
TDO.01	To perform autonomous navigation experiments by using images of the Moon.	ESA, Polimi
TDO.02	To demonstrate CubeSat trajectory control capabilities into lunar environment.	Polimi
TDO.03	To demonstrate the use of miniaturised optical payload in lunar environment.	Leonardo
TDO.04	To demonstrate the use of miniaturised technologies into lunar environment.	ISIS
TDO.05	To demonstrate the use of miniaturised propulsion systems in lunar environment.	TU Delft
TDO.06	To perform autonomous, high-performance on-board payload data processing.	S&T

B | Appendix: SRP data

Here are reported the parameters used to compute the solar radiation pressure torque, for the deployed and packed configurations of LUMIO, from [5]. α denotes the inclination angle of the solar panels provided by the SADA system.

Deployed

$$n_d = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & -1 & sin(\alpha) & -sin(\alpha) & sin(\alpha) & -sin(\alpha) \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & cos(\alpha) & -cos(\alpha) & cos(\alpha) & -cos(\alpha) \end{bmatrix}$$

$$r_d = \begin{bmatrix} 0 & 0 & 0 & 0 & -1.7 & 1.7 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1.13 & 1.13 & 0 & 0 & -3.68 & -3.68 & 3.68 & 3.68 \\ -1.23 & 1.23 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} 10^{-1} - \begin{bmatrix} 0.017 \\ 0.0123 \\ 0.0123 \end{bmatrix} \ [m]$$

$$A_d = [0.0768\ 0.0768\ 0.0836\ 0.0836\ 0.0556\ 0.0556\ 0.0899\ 0.0899\ 0.0899\ 0.0899]\ [m^2]$$

$$J_d = \begin{bmatrix} 0.4029 & 0 & 0 \\ 0 & 0.2204 & 0 \\ 0 & 0 & 0.4516 \end{bmatrix} \ [kg\ m^2]$$

Packed

$$n_p = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$r_p = \begin{bmatrix} 0 & 0 & 0 & 0 & -0.17 & 0.17 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.123 & 0.123 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.123 & 0.123 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0.017 \\ 0.0123 \\ 0.0123 \end{bmatrix} \ [m]$$

$$A_p = [0.0836\ 0.0836\ 0.0836\ 0.0836\ 0.0556\ 0.0556\ 0\ 0\ 0\ 0]\ [m^2]$$

$$J_p = \begin{bmatrix} 0.1701 & 0 & 0 \\ 0 & 0.2200 & 0 \\ 0 & 0 & 0.2310 \end{bmatrix} \ [kg\ m^2]$$

C | Appendix: GNC requirements

The requirements of the GNC subsystem listed for the case of LUMIO are reported in Table C.1. They represent the final version, according to the results obtained from the simulations. In the following, the requirements which have been modified with the values obtained, and those which shall be verified by the future works are reported.

- **GNC.020**: it has to be defined in a future phase of the mission.
- **GNC.022**: it has to be defined by a dedicated future analysis.
- **GNC.040**: various trials have been performed using higher cut-off times; as reported in the previous chapters, the 12 hours cut-off time is the most $\Delta\mathbf{v_{SK}}$ saving choice. Moreover it reduces the error accumulation, sometimes critical.
- **GNC.060**: the values reported have been retrieved from the preliminary simulations. A refined analysis shall confirm them.
- **GNC.070**: the values reported come from the Phase A of LUMIO. However, from the simulations, an higher accuracy seems to be required.
- **GNC.080**: the value reported has been obtained from the Monte Carlo analysis, accounting for GNC.081 and rounding up.
- **GNC.090**: average value obtained from the Monte Carlo, to be confirmed.
- **GNC.091**: minimum manoeuvre magnitude determined from the MIB of the thrusters selected for LUMIO.
- **GNC.092**: value obtained from the simulations of the pointing error during the firing.

Table C.1: GNC requirements.

Req. ID	Type	Identifier	Text	Rationale	Tracker	Verif.
GNC.010	CON	Ground-based GNC	The GNC system of LUMIO shall be primarily based on radiometric data coming from ground for orbit determination.			I
GNC.020	CON	Imaging-based GNC	The GNC system of LUMIO shall perform the experiment of autonomous optical navigation at the date **TBD** during the mission.		ONX.010	I
GNC.021	FUN	LUMIO-cam pointing	The ADCS system shall provide LUMIO-cam pointing precision of at least 0.2 *deg* before the orbit determination campaign occurs.	To point the LUMIO-cam in the correct direction before images acquisition.	GNC.020	A
GNC.022	FUN	LUMIO-cam precision	LUMIO-cam and the image processing shall be able to determine the Moon angular diameter and centre with a precision no less than **TBD**.	To retrieve a precise state from the LUMIO-cam acquisition.	GNC.020	A

GNC.030	CON	GNC operation	The GNC subsystem shall perform station keeping manoeuvres during each Nav&Eng cycle.	To ensure correct attitude for remote sensing during the Science cycles.	I
GNC.040	OPS	Cut-off time	The orbit determination campaign shall be executed at least 12 hours (**TBC**) before each S/K manoeuvre.	The 12 hours cut-off duration is sufficiently short to prevent spacecraft state to change excessively and long enough to schedule manoeuvres executions onboard LUMIO.	GNC.030	I
GNC.050	OPS	S/K manoeuvres number	A number of 3 S/K manoeuvres shall be performed at each Nav&Eng cycle: first at the beginning, second at the middle, third at the end.	To reduce the overall propellant required and increase the overall GNC precision with limited orbit determination and control effort.	GNC.030	I
GNC.060	SCI	Trajectory dispersion	Trajectory dispersion in position and velocity (3σ) shall remain below 1000 km (**TBC**) and 1 m/s (**TBC**) respectively.	In the case in which these upper bounds are not respected the S/K cannot be performed any more considering $\Delta\mathbf{v_{SK}}$ budget and thrusters capabilities.	GNC.030	A

GNC.070	SCI	S/K navigation error	LUMIO navigation position and velocity vectors (3σ) shall be determined with an accuracy of 30 km and 30 cm/s respectively during S/K operations of the Nav&Eng cycles.	To raise the precision of each manoeuvre and decrease the overall $\Delta\mathbf{v_{SK}}$ budget.	GNC.030	A
GNC.071	SCI	Data frequency	An high frequency of an acquisition per minute (HF = 16.7 mHz) shall be used for orbit determination during S/K operations of the Nav&Eng cycles.		GNC.070	I
GNC.072	CON	S/K manoeuvres failure	A S/K manoeuvre shall not be executed in the case in which position and velocity navigation errors exceed the minimum requirement for orbit determination.	To avoid really expensive manoeuvres which could injure the overall $\Delta\mathbf{v_{SK}}$ budget of the mission.	GNC.070	I
GNC.080	CON	S/K manoeuvre	The thrusters shall be able to provide a 3σ total $\Delta\mathbf{v_{SK}}$ = 4 m/s (**TBC**) just for S/K purposes during the 1-year mission lifetime.		GNC.030	A

GNC.081	CON	S/K maneuvers failure	The 3σ total $\Delta\mathbf{v_{SK}}$ for S/K shall account for missed burns, considering a total of 5% S/K manoeuvres failure during the 1-year mission lifetime.	Missed burns could happen for GNC.072 or for whatever error or damage in the control actuators.	GNC.080	R
GNC.090	CON	S/K thrusters firing	The thrusters shall be able to provide at least a 3σ $\Delta\mathbf{v_{SK}} = 0.2456\ m/s$ (**TBC**) for each S/K manoeuvre.		GNC.030	A
GNC.091	CON	S/K thrusters MIB	The $\Delta\mathbf{v_{SK}}$ for each single S/K manoeuvre shall be higher than $0.0029\ m/s$ (**TBC**), accordingly to the minimum impulse bit of the thrusters.		GNC.090	R
GNC.092	CON	S/K thrusters error	Each manoeuvre error shall feature a misalignment lower than $0.001\ deg$ (**TBC**) to reduce the overall $\Delta\mathbf{v_{SK}}$ budget for S/K.	Pointing accuracy of thrusters shall ensure the correct $\Delta\mathbf{v_{SK}}$ for each S/K manoeuvre.	GNC.090	A

GNC.100	SCI	Nominal navigation error	LUMIO navigation position and velocity vectors (3σ) shall be determined with an accuracy of 30 km and 50 cm/s respectively during nominal operations of the Nav&Eng cycles.	GNC.030	A
GNC.101	SCI	Data frequency Nav&Eng	A medium frequency of an acquisition per 10 minutes (MF = 1.67 mHz) shall be used for orbit determination during nominal operations of the Nav&Eng cycles.	GNC.100	I
GNC.110	SCI	Data frequency Science	A low frequency of an acquisition per 60 minutes (LF = 0.277 mHz) shall be used for orbit determination during the Science cycles.	GNC.030	I

Bibliography

[1] G. Merisio, C. Giordano, V. Franzese, and F. Topputo, "LUMIO phase A - D2 - Mission Analysis Report - v1.6," 2021.

[2] G. Merisio, V. Franzese, C. Giordano, M. Massari, P. D. Lizia, F. Topputo, D. Labate, G. Pilato, A. Cervone, S. Speretta, A. Menicucci, E. Bertels, K. Woroniak, A. Kukharenka, A. Thorvaldsen, D. Koschny, J. Vennekens, and R. Walker, "LUMIO: a CubeSat to monitor the lunar farside," jul 2021. [Online]. Available: https://doi.org/10.5194%2Fepsc2021-282

[3] R. W. Farquhar and D. W. Dunham, "Use of libration-point orbits for space observatories," in *International Astronomical Union Colloquium*, vol. 123. Cambridge University Press, 1990, pp. 391–395. [Online]. Available: https://doi.org/10.1007%2F978-94-011-3454-5_52

[4] G. Merisio, C. Giordano, V. Franzese, and F. Topputo, "LUMIO phase A - D1 - Mission Requirements Document - v1.7," 2021.

[5] G. Merisio, C. Giordano, V. Franzese, F. Topputo *et al.*, "LUMIO phase A - D5 - System Design Report - v1.2," 2021.

[6] G. Merisio, C. Giordano, V. Franzese, K. Woroniak, E. Bertels, A. Cervone, and S. Speretta, "LUMIO phase A - D4 - System Requirements Document - v1.3," 2021.

[7] V. Franzese, P. Di Lizia, and F. Topputo, "Autonomous optical navigation for the lunar meteoroid impacts observer," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 7, pp. 1579–1586, 2019. [Online]. Available: https://doi.org/10.2514%2F1.g003999

[8] A. D. Briuno, *The restricted 3-body problem: plane periodic orbits*. Walter de Gruyter, 1994, no. 17. [Online]. Available: https://doi.org/10.1515%2F9783110901733

[9] H. Zhang and S. Li, "A general method for the generation and extension of collinear libration point orbits," *Celestial Mechanics and Dynamical Astronomy*, vol.

126, no. 4, pp. 339–367, may 2016. [Online]. Available: https://doi.org/10.1007%2Fs10569-016-9698-8

[10] Y. Ulybyshev, "Station keeping strategy and possible lunar halo orbits for long-term space station," in *AIAA Guidance, Navigation, and Control Conference*, 2014, p. 0274. [Online]. Available: https://doi.org/10.2514%2F6.2014-0274

[11] M. Langer and J. Bouwmeester, "Reliability of CubeSats-statistical data, developers' beliefs and the way forward," 2016.

[12] P. C. Lai, D. C. Sternberg, R. J. Haw, E. D. Gustafson, P. C. Adell, and J. D. Baker, "Lunar Flashlight CubeSat GNC system development," *Acta Astronautica*, vol. 173, pp. 425–441, 2020. [Online]. Available: https://doi.org/10.1016%2Fj.actaastro.2020.01.022

[13] K. Oguri, K. Kakihara, S. Campagnola, N. Ozaki, K. Oshima, T. Yamaguchi, and R. Funase, "EQUULEUS mission analysis: Design of the science orbit phase," in *26th International Symposium on Space Flight Dynamics*, no. 72, 2017, pp. 1–7. [Online]. Available: https://doi.org/10.1007%2Fs40295-019-00206-y

[14] J. Hernando-Ayuso, S. Campagnola, T. Yamaguchi, Y. Ozawa, and T. Ikenaga, "OMOTENASHI trajectory analysis and design: Landing phase," *Acta Astronautica*, vol. 156, pp. 113–124, 2019. [Online]. Available: https://doi.org/10.1016/j.actaastro.2018.10.017

[15] H. Chen, J. Liu, L. Long, Z. Xu, Y. Meng, and H. Zhang, "Lunar far side positioning enabled by a CubeSat system deployed in an Earth-Moon halo orbit," *Advances in Space Research*, vol. 64, no. 1, pp. 28–41, 2019. [Online]. Available: https://doi.org/10.1016%2Fj.asr.2019.03.031

[16] C. Pirat, M. Richard-Noca, C. Paccolat, F. Belloni, R. Wiesendanger, D. Courtney, R. Walker, and V. Gass, "Mission design and GNC for in-orbit demonstration of active debris removal technologies with CubeSats," *Acta Astronautica*, vol. 130, pp. 114–127, 2017. [Online]. Available: https://doi.org/10.1016%2Fj.actaastro.2016.08.038

[17] B. Larbi, M. Grzesik, B. Radtke, T. CJ, and E. Stoll, "Active debris removal for mega constellations: CubeSat possible," in *Proceedings of the 9th International Workshop on Satellite Constellations and Formation Flying IWSCFF2017, Boulder, Colorado*, 2017.

[18] C. P. Newman, D. C. Davis, R. J. Whitley, J. R. Guinn, and M. S. Ryne, "Station

keeping, orbit determination, and attitude control for spacecraft in near rectilinear halo orbits," 2018.

[19] D. Guzzetti, E. M. Zimovan, K. C. Howell, and D. C. Davis, "Station keeping analysis for spacecraft in lunar near rectilinear halo orbits," in *27th AAS/AIAA Space Flight Mechanics Meeting*. American Astronautical Society San Antonio, Texas, 2017, pp. 1–20.

[20] U. Kalabic, A. Weiss, S. Di Cairano, and I. Kolmanovsky, "Station-keeping and momentum management on halo orbits around l2: Linear-quadratic feedback and model predictive control approaches," in *Proc. AAS Space Flight Mechanics Meeting*, 2015, pp. 15–307.

[21] L. Liu, C. Hu, M. Wang, and Y. Wang, "Maintenance of Earth–Moon halo orbit," in *2017 36th Chinese Control Conference (CCC)*. IEEE, 2017, pp. 5969–5974. [Online]. Available: https://doi.org/10.23919%2Fchicc.2017.8028305

[22] D. Chongrui, M. Fain, and O. Starinova, "Analysis and design of halo orbits in cislunar space," in *IOP Conference Series: Materials Science and Engineering*, vol. 984, no. 1. IOP Publishing, 2020, p. 012033.

[23] G. Gómez, J. Llibre, R. Martınez, and C. Simó, "Station keeping of a quasi-periodic halo orbit using invariant manifolds," in *Proceed. 2nd Internat. Symp. on spacecraft flight dynamics, Darmstadt*, 1986, pp. 65–70.

[24] D. Davis, S. Bhatt, K. Howell, J.-W. Jang, R. Whitley, F. Clark, D. Guzzetti, E. Zimovan, and G. Barton, "Orbit maintenance and navigation of human spacecraft at cislunar near rectilinear halo orbits," 2017.

[25] R. Burton, S. Weston, and E. Agasid, "State of the art in guidance navigation and control: A survey of small satellite GNC components," *Proc. Adv. Astron. Sci*, vol. 157, 2016.

[26] DAER-DART, "An N-Body, ephemeris based, trajectory and GNC simulator-Software documentation," in *CUBORG: CUBesat Orbit and GNC tool*, vol. 1. PoliMi, 2020, pp. 1–49.

[27] J. Voight, *Quaternion algebras*. Springer Nature, 2021.

[28] E. Pellegrini and R. P. Russell, "On the computation and accuracy of trajectory state transition matrices," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 11, pp. 2485–2499, 2016. [Online]. Available: https://doi.org/10.2514/1.G001920

[29] G. J. Der, "An elegant state transition matrix," *The Journal of the astronautical sciences*, vol. 45, no. 4, pp. 371–390, 1997. [Online]. Available: https://doi.org/10.2514%2F6.1996-3660

[30] N. Dwivedi, "Deterministic optimal maneuver strategy for multi-target missions," *Journal of Optimization Theory and Applications*, vol. 17, no. 1-2, pp. 133–153, 1975.

[31] K. C. Howell and H. J. Pernicka, "Station-keeping method for libration point trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 1, pp. 151–159, 1993. [Online]. Available: https://doi.org/10.2514%2F3.11440

[32] K. C. Howell and T. M. Keeter, "Station-keeping strategies for libration point orbits-target point and floquet mode approaches," *Spaceflight mechanics 1995*, pp. 1377–1396, 1995.

[33] S. Di Gennaro, "Passive attitude control of flexible spacecraft from quaternion measurements," *Journal of Optimization Theory and Applications*, vol. 116, no. 1, pp. 41–60, 2003. [Online]. Available: https://doi.org/10.1023%2Fa%3A1022106118182

[34] D. A. Dei Tos, M. Rasotto, F. Renk, and F. Topputo, "Lisa pathfinder mission extension: A feasibility analysis," *Advances in Space Research*, vol. 63, no. 12, pp. 3863–3883, 2019. [Online]. Available: https://doi.org/10.1016%2Fj.asr.2019.02.035