Debris Collision Avoidance Manoeuvres

Autonomous Guidance and Control System to Safeguard Satellites from Space Debris

Chloé Carer

TU Delft

MSc Thesis

Autonomous Guidance and Control System to Safeguard Satellites from Space Debris

Chloé Carer

Student number:	4795679	
Project duration:	September, 2023 – May, 2024	
Defence date:	August 26, 2024	
Thesis committee:	Dr.ir. E. Mooij	TU Delft, Supervisor
	Ir. M.C. Naeije	TU Delft, Committee Chair
	Dr. S. Gehly	TU Delft, External Examiner

In partial fulfilment to obtain the MSc degree at the Delft University of Technology.

Cover: Artist illustration debris orbiting Earth. Shutterstock illustration ID: 233084350

TU Delft

To Infinity and Beyond!
– Buzz Lightyear

Preface

This MSc thesis marks the end of an almost 6-year journey full of ups and downs, of numerous doubts, but mostly full of knowledge and growth. I am grateful for everything I have learned, the people I have met and the wonderful friends I have made along the way. Today I can say that I am proud of the hard work and sacrifices made in the past years, fully aware that this would not have been possible on my own. Many people, from close and far contributed to this journey, some of them from day one and others joined the adventure along the way.

To start with, I would like to thank my supervisor, Erwin, who supported my wish to pursue my MSc thesis in space situational awareness and who equipped me with all the tools to successfully do so. Working together on this project has been a great learning experience. This thesis is the perfect mix of a lot of work, dedication, some doubts, but most importantly a lot of fun. Our meetings were always sparked with laughs, gossips and excitement. I will truly miss those, I really enjoyed working with you and soaking up all your knowledge, it was very motivating. I hope, and am sure, we will work again together in the future. Thank you, for being so invested in this research.

I would like to thank both Vincent Conings as well as Marco Sagliano for the precious help you have provided throughout this research.

My dear Delft friends, Harald, Neil, Crina, Serena, Elena, Andrada, and Nico, none of this would have been possible without you by my side, I am so thankful to have you, you inspire me every day.

Thank you to my high school best friends; Juliette, Marie, Clémence, Chloé, Jean, Jojo, Maria, and Charlotte who have also been there to support me, even if sometimes more than thousands of kilometres away. Thinking about you brought me so much joy.

Gwen, thank you for the unconditional love and motivation you have given me throughout my studies. I am sure it was not the easiest to see a working Grinch in action but you have always supported me and would always have the right words.

Nonno, it probably all started thanks to you, I know you are proud, I miss you every day.

Thank you to Grand-mi, Grand-père, Nadia, Papa, Leslie, Simon and Basile for the support and for allowing me to take part in this journey.

Last but not least, I would like to thank (and the word is not strong enough) Maman, Mamie and Luca for the love and support you have given me every day, but especially for the past 6 years. You are my number one supporter, have always believed in me and knew how to bring me comfort when I was so down. Thank you for always taking the time to listen to me ranting about work, exams and coding. Along the way, you also learned quite a few vocabulary words, which will never serve you; rivets, wing box, dynamics and many more.

I am so lucky to have such loving surroundings that cherish me every day and never stop believing in myself.

Merci.

Now that we all have teary eyes, it is time to dry them out, sit back and enjoy the reading!

Chloé Carer
Delft, May 2024

Abstract

Ensuring the sustainability of future space missions and maintaining continuous access to space necessitates a serious and proactive approach to addressing the space debris dilemma. Failure to mitigate the proliferation of space debris poses a significant threat to future space endeavours. One crucial measure that space agencies can undertake is actively protecting against space debris by implementing collision avoidance manoeuvres (CAMs). This non-invasive technique allows manoeuvring satellites to deviate from their orbits, thereby avoiding potential collisions, by increasing the miss distance between themselves and other objects. This thesis focuses on researching methods to perform such manoeuvres, with a particular emphasis on the guidance and control systems of manoeuvrable vehicles. The Starlink constellation is used in this research as a reference scenario, considering its popularity and extensive development, although the developed system is applicable to any satellite. The core of achieving successful manoeuvres lies in the creation of a robust guidance system. This system integrates convex optimisation algorithms with reference tracking mechanisms, enabling the autonomous execution of CAMs to be both robust and fuel-efficient. By using a convex guidance approach, optimal fuel trajectories are determined, while ensuring adherence to predefined reference paths and constraints such as keep-out sphere and available thrust. Rigorous testing across diverse conditions showcases the system's suitability for satellites of varying characteristics (trajectory, propulsion type, size and more). For both state and attitude correction, linear quadratic regulator (LQR) controllers are employed. The first LQR controller is designed for trajectory tracking, with adaptable penalty weights tailored to different orbit sequences. The second LQR controller corrects attitude in the control sequence. Different scenarios are analysed in terms of trajectory profile, propulsion types and more, revealing that low-thrust manoeuvres with constant capabilities yield the most optimal manoeuvring scenarios (minimal fuel usage and desired miss distance). Furthermore, it has been discovered that the conduction of constant tracking was not a viable option, due to its significant fuel consumption and the limitations in terms of propulsion characteristics it introduced. The fuel usage was deemed excessive for manoeuvres that may occur frequently, a trend anticipated to intensify in the future. Although the outcomes of the research mark a significant advancement in space debris mitigation and contribute to improving space sustainability, further research and development are encouraged, to allow improvement in the field of space debris mitigation. For instance, this specific work could incorporate additional elements to better represent real-life scenarios, such as including more space debris or environmental perturbations in the analysis, and improving the manoeuvring sequences by including several manoeuvre scenarios.

Contents

Nomenclature

Abbreviations

ADM	Atmospheric Dynamics Misssion
AI	Artificial Intelligence
ASAT	Anti-satellite
ATV	Automated Transfer Vehicle
AUTOCA	Autonomous Collision Avoidance System
B	Body
BC	Boundary Condition
CAM	Collision Avoidance Manoeuvre
CDM	Conjunction Data Message
CF	Central Field
CHN	China
com	Centre of mass
COPUOS	Committee on the Peaceful Uses of Outer Space
CPU	Central Processing Unit
CSM	Conjunction Summary Messages
DA-GMM	Differential Algebra and Gaussian Mixture Model
DAM	Debris Avoidance Manoeuvre
DCM	Direction Cosine Matrix
DoD	Department of Defense
DOF	Degrees of Freedom
DUT	Delft University of Technology
ECI	Earth-Centred Inertial
EKF	Extended Kalman Filter
EOL	End-of-Life
EOM	Equation of Motion
ESA	European Space Agency
ESTEC	European Space Research and Technology Center
FCC	Federal Communications Commission
FRA	France
GEO	Geostationary orbit
GNC	Guidance, Navigation, and Control
GPS	Global Positioning System
GRADS	Generic Rendezvous and Docking Simulator

GRALS	GNC Rendezvous, Approach and Landing Simulator
GSSN	Global Space Surveillance Network
IADC	Inter-Agency Space Debris Coordination Committee
ID	Identification
IMU	Inertial Measurement Unit
INDI	Incremental Nonlinear Dynamic Inversion
IPOPT	Interior Point Optimiser
ISS	International Space Station
JSOC	Joint Space Operations Center
KOS	Keep-out Sphere
LEO	Low Earth Orbit
LQR	Linear Quadratic Regulator
LVLH	Local Vertical Local Horizontal
MJD	Modified Julian Date
ML	Machine Learning
MSc	Master of Science
NASA	National Aeronautics and Space Administration
NORAD	North American Aerospace Defense Command
NRLMSISE-00	US Naval Research Laboratory Mass Spectrometer Incoherent Scatter Radar to Exosphere (year 2000)
P	Propulsion
PID	Proportional Integral Derivative
POL	Poland
PM	Point Mass
RAAN	Right Ascension of the Ascending Node
RCS	Reaction Control System
RK4	Runge-Kutta 4^{th}
SATCAT	Satellite Catalogue
SCP	Splitting Conic Solver
SeDuMi	Self-Dual-Minimisation
SGP4	Simplified General Perturbations model 4
SH	Spherical Harmonics
SOCP	Second-Order Cone Programming
SRP	Solar Radiation Pressure
SSA	Space Situational Awareness
SSAP	Space Situational Awareness Programme
SSN	Space Surveillance Network
SST	Space Surveillance and Tracking
TLE	Two-Line Element
TOPO	Trajectory Operations and Planning Officer
UN	United Nations
US	United States
US62	US Standard atmosphere developed in 1962
US76	US Standard atmosphere developed in 1976
vs.	Versus
ZOH	Zero-order hold
1,2,3D	1,2,3-Dimensional

Symbols

a	Semi-major axis	[m]
$\boldsymbol{a}$	Acceleration vector	[m/s^2]
A	Cross-sectional area/ Reference surface	[m^2]
$\boldsymbol{A}$	State matrix	[variable]
$\boldsymbol{A}$	Control matrix	[variable]
$\boldsymbol{B}$	Magnetic flux	[Wb]
$\boldsymbol{C}$	Rotation matrix	[-]
C_D	Drag coefficient	[-]
C_r	Reflectivity coefficient	[-]
dm	Mass element	[kg]
e	Eccentricity	[-]
$\boldsymbol{e}$	Unit vector	[-]
E	Eccentric Anomaly	[rad]
$\boldsymbol{f}$	Gravitational acceleration from perturbing body	[m/s^2]
$\boldsymbol{F}$	Force	[N]
h	Altitude	[m]
$\boldsymbol{h}$	Angular momentum	[kg m^2/s]
i	Inclination	[rad]
$\boldsymbol{I}$	Inertia tensor	[km m^2]
I_{sp}	Specific impulse	[s]
$\boldsymbol{J}$	Cost function	[-]
$\boldsymbol{K}$	Feedback gain	[-]
m	Mass	[kg]
$\dot{m}$	Mass flow rate	[kg/s]
$\boldsymbol{m}$	Magnetic moment	[Nm]
M	Mean anomaly	[rad]
$\boldsymbol{M}$	Moment/ Torque	[Nm]
n	Mean motion	[rad/s]
P	Pressure	[Pa]
P_c	Collision probability	[-]
$P_{n,m}$	Legendre polynomials of degree = n and order = m	[-]
$\boldsymbol{q}$	Quaternion vector	[-]
$q_{1,2,3,4}$	Quaternion elements sequence	[-]
$\boldsymbol{Q}$	State cost matrix	[variable]
r	Distance	[m]
$\boldsymbol{r}$	Momentum arm	[m]
$\boldsymbol{r}$	Position vector	[m]
$\dot{\boldsymbol{r}}$	Velocity vector	[m/s]
$\ddot{\boldsymbol{r}}$	Acceleration vector	[m/s^2]
R_E	Earth radius	[m]
$\boldsymbol{R}$	Control cost matrix	[variable]
t	Time	[s]
t_p	Time of perigee passage	[s]
U	Earth gravitational field potential	[J/kg]
$\boldsymbol{u}$	Control vector	[N]
$\boldsymbol{V}$	Velocity vector	[m/s]

$\boldsymbol{x}$	State vector	[variable]
x, y, z	Cartesian position coordinates	[m]
$\dot{x}, \dot{y}, \dot{z}$	Cartesian velocity coordinates	[m/s]
α	Arbitrary angle	[rad]
Δ	Relative change	[-]
ϵ	Elevation angle	[rad]
η	Efficiency	[-]
γ	Flight-path angle	[rad]
Λ	Geocentric longitude	[rad]
μ	Gravitational parameter	[m^3/s^2]
ω	Argument of perigee	[rad]
$\boldsymbol{\omega}$	Angular momentum	[kg m^2/s]
$\boldsymbol{\omega}$	Rotation velocity vector	[rad/s]
Ω	Right ascension of the ascending node	[rad]
ϕ	Geocentric latitude	[rad]
ϕ, θ, ψ	Euler angles	[rad]
ψ	Azimuth angle	[rad]
ρ	Atmospheric density	[kg /m^2]
$\boldsymbol{\rho}$	Distance vector	[m]
θ	True anomaly	[rad]
θ_t	Argument of latitude	[rad]

CHAPTER

1 Introduction

From the first ever satellite successfully placed in orbit, Sputnik 1, in 1957, the number of objects in space never stopped increasing and is accumulating at a more rapid rate than it ever did. To provide context, in 2024, the number of tracked objects stands at approximately 45,500, whereas there were roughly 10,000 objects in 2004, just two decades ago. The accumulation of debris over the last two decades, in the low-Earth orbit (LEO) and Geostationary orbit (GEO) can be observed in Figure 1.1. The reason for such accumulation can be attributed primarily to two key factors: *catastrophic collisions* and *the expanding use of space*.

To start with, a series of catastrophic events, which took place in the past 20 years, drastically expanded the number of tracked objects in space. Events, such as the 2007 Fengyun-1C Chinese anti-satellite (ASAT) test alone, increased the number of trackable debris by 25%[2]. Additionally, the emerging NewSpace era is also directly increasing the occupation of LEO, which will in turn boost the number of debris in space in the near future. For example, SpaceX, through their Starlink mega-constellation, has the objective of placing more than 40,000 satellites in orbit around Earth, providing a large satellite Internet access coverage, largely contributing to space pollution.

As the number of objects in space goes up, so does the likelihood of collisions, resulting in more frequent collisions and the creation of additional debris. This phenomenon, first defined by Donald J. Kessler in 1978, indicates that once a specific critical threshold is reached, the accumulation of space debris will continue to grow in a self-sustaining manner

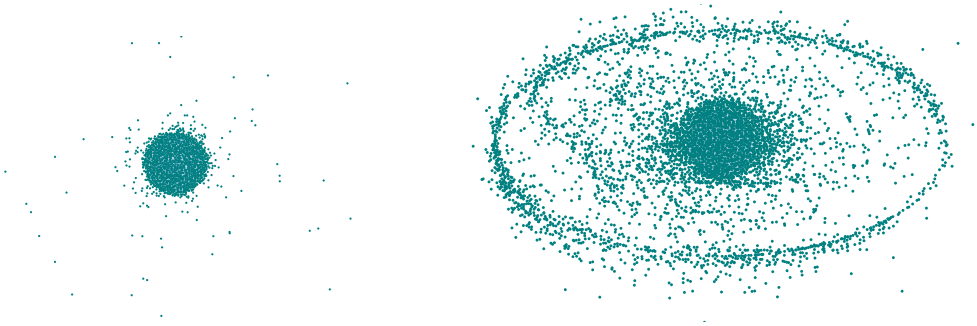

Figure 1.1: Evolution of the number of objects in LEO and GEO from 2004 (left) to 2024 (right). Constructed from two-line elements (TLEs) taken from Space-Track[1].

[1] Space-Track, *Satellite catalogue*, https://www.space-track.org/ [Last visited on 01/02/2024].

[2] European Space Agency, *About space debris*, https://www.esa.int/Space_Safety/Space_Debris/About_space_debris [Last visited on 23/10/2023].

(Kessler et al., 2010). Collisions not only produce more debris but also trigger subsequent collisions, setting off a chain reaction, which can pose a significant threat to the future of space exploration. This phenomenon, now referred to as the Kessler syndrome, highlights the urgency to improve research on the matter and put resources into solving such problems to guarantee the future of space exploration. For this reason, engineers are becoming more interested in studying conjunctions and developing ways to prevent them. This thesis will be part of such research and will have as its main objective to develop a technology capable of mitigating such conjunctions, allowing space to remain accessible.

To achieve the latter objective, conducting a comprehensive literature review is crucial for exploring this issue, bridging existing knowledge gaps, and comprehending strategies for CAM. Such a study has been conducted and enabled to find a gap in the literature, which could be filled to improve scientific research on the matter of collision avoidance. The specifics of this investigation will be compiled in Chapter 2, laying the foundation for this Master of Science (MSc) thesis research. Based on this analysis, it can be deduced that two primary technologies have the potential to address the problem of space debris proliferation: removing the debris through active debris removal and implementing CAM to prevent collisions. As the number of debris increases, so does the need for manoeuvres; on average, each of the European Space Agency's (ESA) Earth-orbiting satellites requires approximately two manoeuvres each year[3]. As human activity in space increases, manual planning of CAMs will become impractical. For this reason, developing a technology capable of automating these processes is crucial for safeguarding the future of space exploration. This study will focus on this objective, incorporating an analysis of the guidance and control of satellites. An extensive review of the literature suggests that further investigation into the potential of convex optimisation in the guidance sequences for collision avoidance could be highly beneficial. Given its widespread usage, convex optimisation appears to be an ideal candidate for this application. Consequently, this research will analyse its effects and development in conjunction with the development of an optimal attitude control sequence. The main research question of this thesis work can thus be formulated as follows:

To what extent would the use of convex optimisation-based guidance, increase the robustness, safety, and efficiency of an autonomous orbit control, for the space debris collision avoidance of mega-constellations?

To address this research question, several subquestions will be introduced and answered throughout the thesis work.

- *Which environmental models should be included and with which accuracy, to meet the operational and central processing unit (CPU) load requirements?*
- *How can a guidance system be modelled and simulated?*
- *How can a control system be modelled and simulated?*
- *Which type of guidance system would offer the best combination of robustness, safety, and fuel efficiency?*
- *What are the limitations of the developed guidance system on the satellite's propulsion capabilities?*

[3] European Space Agency, *FAQ: Frequently asked questions*, https://www.esa.int/Space_Safety/Space_Debris/FAQ_Frequently_asked_questions [Last visited on 01/10/2023].

- *Which objective function, state, and control constraints should be set for this specific mission?*
- *What are the necessary steps included in a conjunction analysis?*
- *Would the algorithm be capable of redirecting the satellite in case divergence from the optimal path is observed?*

Following this, the research objective for the MSc thesis can be outlined as follows:

Investigating the possible design and development of a robust guidance algorithm capable of efficiently executing collision avoidance manoeuvre, analysing its limitations, and potential integration with a control system.

The work report has been constructed in such a way that all these subquestions can be addressed throughout the different chapters. To start with, as already introduced earlier, the mission heritage, and thus most outcomes of the literature study will be introduced in Chapter 2. In this chapter, the emerging space debris problem and the specifics behind the space debris environment will be investigated. This chapter will lay the foundation of space situational awareness (SSA) and more importantly the specifics of debris populations and their growth. Additionally, the technologies used nowadays to prevent collisions will be touched upon (both active and passive), together with some examples of successful on-orbit executions. Then, the details behind the importance of mega-constellations will be described, together with some relevant work from Delft University of Technology (DUT). This chapter will then conclude with an overview of the reference scenario's specifics. From this first part of the study, the need for research on an onboard robust safety system for satellites with the main role of avoiding potential conjunctions can be perceived. This will be the focus of the remaining chapters of this report; each will introduce the essential theory, and development specifics and will present the relevant results obtained throughout the thesis work.
After the introduction of the mission heritage specifics, the next step in such research work is to lay the basics of flight dynamics. This will be performed in Chapter 3, in which the theory and working principles of flight dynamics will all be introduced. These include descriptions of the different reference frames and state variables that will be used throughout the work but also a thorough description of the space environment that will be used during simulations. Using this information, the necessary equations of motion (EOMs) dictating the satellite's state will be introduced. This chapter serves as a foundational resource for readers, providing an introduction to conventions and basic theory, thus establishing the groundwork for the remainder of the research.
Once the basis has been introduced, chapter specifics to the work at hand will be presented. To start with, a comprehensive introduction to conjunction analysis will be presented in Chapter 4. This analysis will cover all the essential aspects involved in conducting a conjunction analysis, encompassing procedures for retrieving requisite data, processing the data, conducting conjunction detection, and addressing mission-specific considerations.
Subsequent to this analysis, the fully developed guidance system, designed to facilitate adjustments to the satellite's state as required, will be presented in Chapter 5. This will include all the necessary theory linked with the algorithms used in this sequence as well as details concerning CAMs and the prescribed steps for achieving comprehensive, precise, and successful guidance. The output generated by the guidance system is crucially relevant and serves as an input to the control sequence.

Similarly, as for the guidance sequence, all details related to attitude control will be outlined in Chapter 6. These specifics include not only the used controller and its development, but also specifics behind the required actuators. Upon the introduction of both the guidance and control sequences, detailed information regarding the entire simulation development and execution will be provided in Chapter 7. This chapter will encompass comprehensive details of the simulation architecture, alongside the various steps executed to ensure thorough verification of the entire model. Then, in Chapter 8, the results of the entire research work will be presented together with their relevant evaluation and description. After highlighting the results, the research questions will be answered together with a comprehensive discussion on the entire project, and the presentation of recommendations will be conducted in Chapter 9. To finish with, Appendix A will present the theory behind the various numerical methods used throughout the simulation and Appendix B will give an overview of the different simulation files and libraries, how to access them, facilitating the reproduction and continuation of this research.

CHAPTER

2 Mission Heritage

This chapter marks the genesis of the research by introducing the basis required to define the mission objective and its characteristics. Prior to commencing this thesis project, an extensive and thorough literature review was undertaken, during which comprehensive knowledge was gathered, gaps in knowledge were identified, and the main objectives of the research were clarified. Fundamental information about the space debris environment, along with some data regarding the quantity of debris present in space and their evolution will be gathered in Section 2.1. From this overview, the need for debris mitigation guidelines will be highlighted. The guidelines that have been put into place to perform prevention and reduce space debris proliferation in the future, will be presented in Section 2.2. Subsequently, in Section 2.3, the different concepts employed to detect, track and catalogue debris will be introduced. Such methods play a pivotal role in enabling continuous space surveillance and detecting conjunctions. This will be followed by Section 2.4, in which an introduction to several collision risk assessment methods, and details on some past real-life CAM scenarios can be found. This section is dedicated to the collection of historical insights about CAMs, focusing on identifying areas of limited knowledge and potential improvements. The multiplication of space debris has emerged as a significant issue that has garnered increasing attention among scientists and engineers in recent years. With a shared objective of mitigating this problem, researchers and students at DUT have undertaken efforts to explore potential solutions and technological advancements related to this matter. In Section 2.5, several of these developments, particularly relevant to the thesis work, will be presented. To conclude this chapter, the definition of the reference mission and vehicle in Section 2.6 as well as the mission requirements in Section 2.7, will be introduced. An overview of the different topics that will be discussed in this section has been displayed in Figure 2.1.

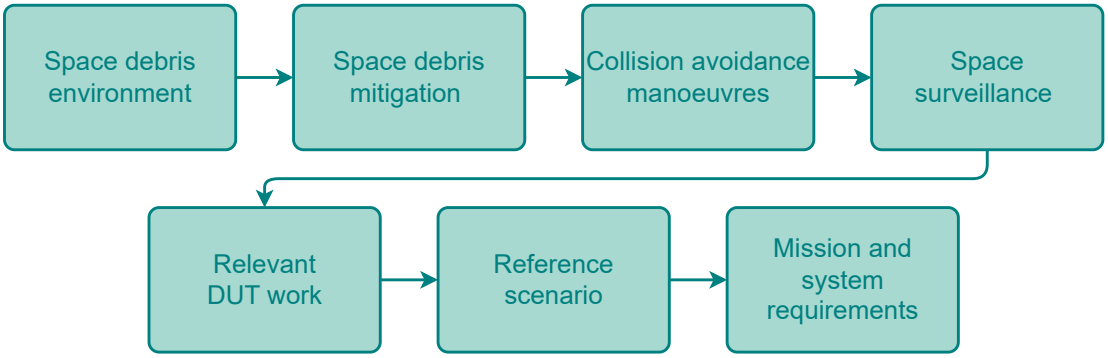

Figure 2.1: Overview of the mission heritage chapter's layout.

2.1 Space Debris Environment

Today, space is being polluted more than ever by a large number of debris. According to ESA Space Debris Office (2023), there would currently be 34,000 objects greater than 10 cm, 900,000 objects from 1 cm to 10 cm and finally, 128 million objects from 1 mm to 1 cm, orbiting Earth. Through collisions, these debris particles create additional fragments, increasing the debris population and subsequently, the probability of collision. To treat the problem at the source, it is important to gain insight into the origin of space debris (Section 2.1.1), but also identify where most debris originates from, by learning about the events responsible for their creation (Section 2.1.2), and their impact on space evolution (Section 2.1.3).

2.1.1 Space Debris Genesis

Space debris comes in all shapes and sizes. They are characterised as any artificial objects created by humans, being no longer operational. Some directly originate from large objects, such as parts of satellites, and abandoned or non-functioning vehicles, and others from small objects, which are often the direct result of fragmentation events or vehicle degradation. The ever-increasing number of space debris mainly results from two phenomena: the expanding number of space launches, which has doubled in the past 20 years (ESA Space Debris Office, 2023), and the chain reactions that occur due to collision events, explosions and space debris propagation. During these, additional debris is created through fragmentation events, which following a chain reaction, in turn, trigger further collisions, resulting in a renewed proliferation of debris. This phenomenon, first examined by Donald J. Kessler in 1978, is now referred to as the Kessler syndrome (Kessler et al., 2010). It is mainly described as a *"self-sustaining cascading collision of space debris in low-Earth orbit"*, which, if not addressed, can lead, in the long term, to limited access to space beyond LEO.

2.1.2 The Most Catastrophic Events

The majority of debris presently found in space originates from some catastrophic events, which occurred in the past years. Collisions between two satellites are often the most destructive, generating thousands of debris, subsequently creating more through fragmentations. As can be seen from Figure 2.2, in which the monthly number of objects around Earth by object type is displayed, the destructive events having the biggest contribution to the growth of the overall space debris population are the Fengyun-1C ASAT test (2007), the Iridium-Cosmos (2009) collisions and more recently the Russian ASAT test (2021). These collisions have devastating consequences for the space environment, both destructive accidents of 2007 and 2009 alone are responsible for a large majority of space debris found in space today, these two breakup events dramatically altered the LEO environment dynamics. Besides collision incidents, in-orbit explosions, such as the Ablestar explosion in 1961 or the recent detonation of an old Russian rocket engine in 2022, can also take place, significantly affecting the space environment. The visible result (in 2024) arising from four catastrophic events can be observed in Figure 2.3. In this figure, an overview of the debris (in orbit as of 01/01/2024) created from the four catastrophic events of 2022 (CZ-6A breakup), 2021 (Russian ASAT test), 2009 (Iridium-Cosmos collision), and 2007 (Fengyun-1C ASAT test) alone. From these four events, there are, as of January 2024, still around 8,500 pieces of debris in orbit, posing a constant threat to satellite operations. More information on these events can be found hereafter.

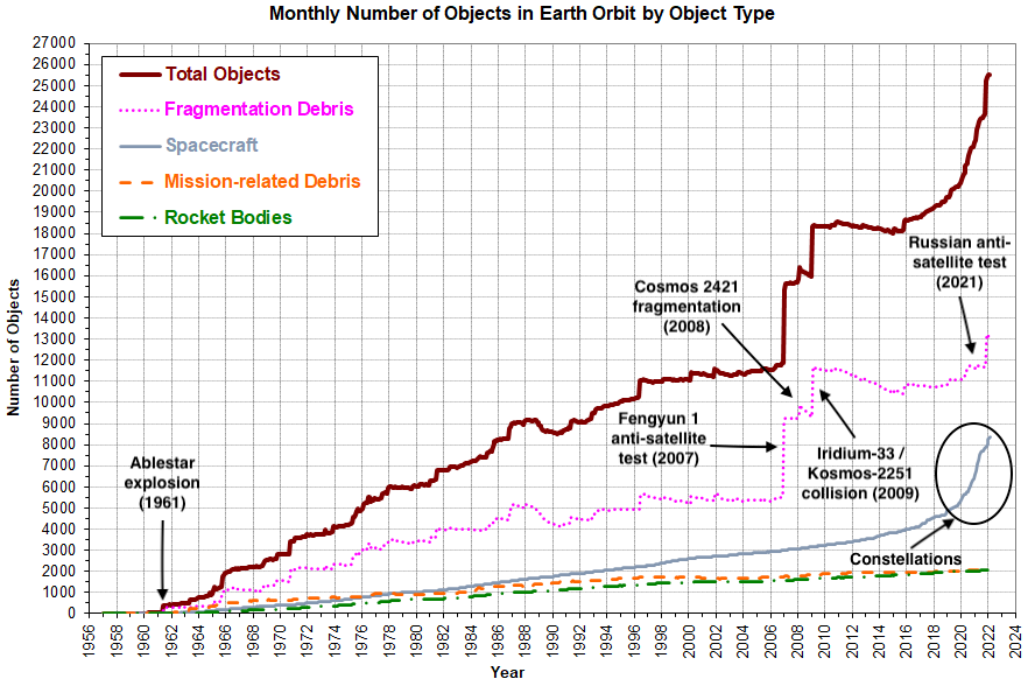

Figure 2.2: Evolution of the number of objects larger than 10 cm observed in LEO [Modified figure from NASA Orbital Debris Program Office (2023)].

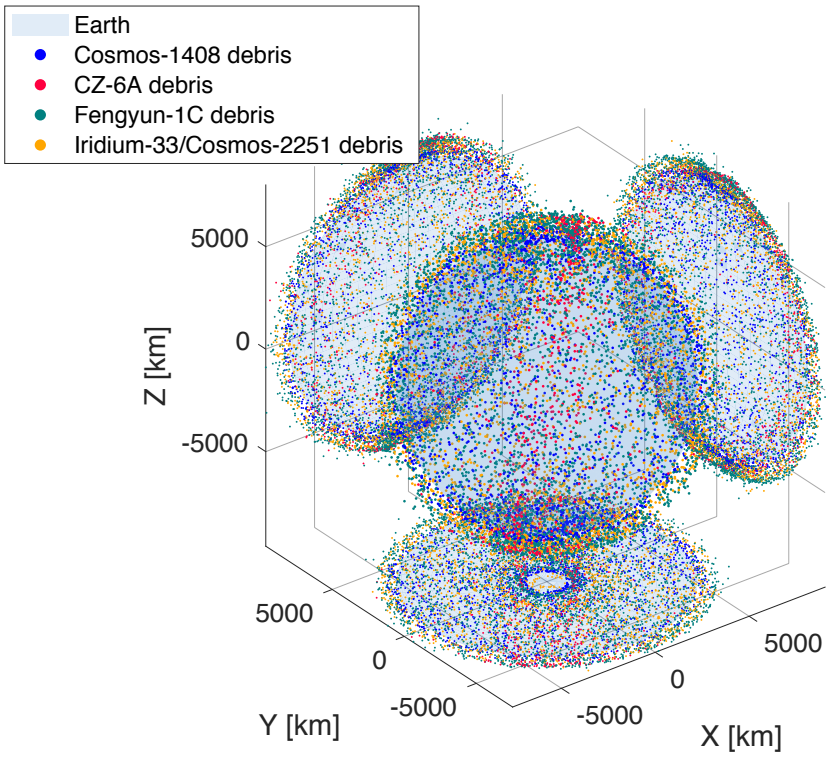

Figure 2.3: Overview of the debris, in orbit in 2024, created alone from four catastrophic events. Using the last TLE on 01/01/2024 taken from Space-Track[1].

Fengyun-1C – Chinese ASAT test: In January 2007, China performed an ASAT by launching a missile from the Xichang Space Launch Center, targeting the Fengyun 1-C meteorological satellite system. This auto-destructive event alone led to a 25% increase in the number of trackable space debris population[2].

- Debris assessment: 3,527 catalogued objects (as of 01/01/2024) and approximately 35,000 debris larger than 1 cm (Kelso, 2007),
- Orbit: 865 km altitude, 98.8° inclination,
- Fengyun-1C mass: 954 kg, and
- Fengyun-1C dimensions: 1.42 x 1.42 x 1.2 m (hexahedron) with a deployed length of 10.5 m[3].

Iridium-33 / Kosmos-2251 collision: In February 2009, another fatal incident occurred with the collision of the Iridium-33 – an active US communication satellite – and the Kosmos-2251 – a defective Russian military satellite. This accidental event occurred at a critical apogee altitude of 780 km; an orbital region widely used by LEO satellites.

- Debris assessment: 2,368 catalogued objects (as of 01/01/2024). (NASA Orbital Debris Program Office, 2023) and approximately 35,000 debris larger than 1 cm (Kelso, 2007),
- Orbit: 780 km altitude, 74.0° (Kosmos-2251) and 86.4° (Iridium-33) inclination,
- Masses: Kosmos-2251 – 954 kg, Iridium-33 – 560 kg (NASA Orbital Debris Program Office, 2009), and
- Dimensions: Kosmos-2251 – 3 x 2 x 5 m (cylindrical), Iridium-33 – 2 m (length) x 1 m ⌀ (triangular cylindrical shape), with two 3.5 x 0.6 m solar panels and three 1.2 x 0.6 m communication panels on its sides (Oliver et al., 2009).

Cosmos-1408 ASAT test: In November 2021, Russia conducted a direct-ascent ASAT test on the Cosmos-1408 satellite that had been in a 490 × 465 km altitude region actively used for space activities. The generation of additional debris at such altitude puts the International Space Station (ISS) at significant risk, given its proximity in orbit.

- Debris assessment: 1,806 catalogued objects (as of 01/01/2024),
- Orbit: 490 × 465 km altitude, 82.6° inclination,
- Cosmos-1408 mass: 1,750 kg (Pardini and Anselmo, 2023), and
- Cosmos-1408 dimensions: 4.5 × 1.2-1.4 m cylinder (Cerutti-Maori et al., 2023).

Explosion of CZ-6A upper stage: In April 2022, a zero-consumption rocket engine from a Russian rocket accidentally exploded in orbit. Such explosions are likely to happen again since they are most likely caused by residual propellant in the engine, which is present in most older rockets. In the past, these were not designed to go through the process in which the rocket expels any explosive before its end-of-life (EOL).

[1] Space-Track, *Satellite catalogue*, https://www.space-track.org/ [Last visited on 01/02/2024].

[2] European Space Agency, *About space debris*, https://www.esa.int/Space_Safety/Space_Debris/About_space_debris [Last visited on 27/05/2022].

[3] eoPortal ESA, *FY-1 (FengYun-1)*, https://www.eoportal.org/satellite-missions/fy-1#rf-communication [Last visited on 16/02/2024].

- Debris assessment: 795 catalogued objects (as of 01/01/2024),
- CZ-6A upper stage orbit: 847 x 813 km altitude, 98.8° inclination,
- CZ-6A upper stage dry mass: 5,800 kg, and
- CZ-6A upper stage dimensions: unknown.

2.1.3 Space Debris Population Evolution

As previously mentioned, the ever-growing use of the space sector has an increasing impact on its sustainability. Indeed, the space debris population continues to expand daily, due to its ever-increasing usage and to the numerous fatal events, introduced earlier. These debris manifest in various shapes, sizes and sources. It has been observed that multiple types of debris populate space; fragmentation debris, payload debris, mission-related debris, and rocket bodies. Furthermore, it can be observed, from the data presented in Table 2.1, that the majority of orbital debris can be found in LEO. This is primarily because most space vehicles are located in this region, which, through degradation or collisions, create large space debris clouds, that, in turn, contribute to the proliferation of debris in this location.

Low Earth orbit: LEO is a circular orbit with altitudes up to 2,000 km above Earth's surface as depicted in Figure 2.4 and described by ESA Space Debris Office (2023). This orbit is highly populated as can be observed in Table 2.1, highlighting the catalogued debris object number per region. It can be observed that LEO has a significant larger population than GEO, or other orbits, with a total of 21,659 catalogued objects as of 01/01/2024. This is due to several factors. To start with, LEO's popularity stems from its cost-effectiveness and convenience for satellite operations; its low altitude translates to less expensive launches compared to higher orbits. Moreover, its proximity to Earth allows for lower latency communications and more frequent passes over a specific location. This makes it specifically useful for imaging missions, enabling higher-quality pictures. Additionally, satellites coming from the commercial sector (constellations) are also located in LEO since they require a lower signal propagation delay. As discussed in Section 2.1.2, LEO is also densely populated due to the different collision events, which previously occurred in this region, creating thousands of debris.

Geostationary orbit: The GEO is a circular orbit with an altitude of 35,785 km ±200 km and latitude ranging from ±15 degrees as can be observed in Figure 2.4. Due to its high altitude, this orbit is widely used for telecommunication purposes, but also for weather forecasting and meteorology. The considerable distance it covers allows signals to be transmitted over extended ranges. While not as frequently used as LEO, due to the large distance required to reach it, there appears to be an increasing number of launches to GEO over the years (ESA Space Debris Office, 2023).

Table 2.1: Number of tracked objects orbiting Earth, these represent all types of objects: payloads, payload fragmented debris, payload debris, rocket bodies, rocket debris, etc. – ESA Space Debris Office (2023).

Catalogued objects orbiting Earth			
LEO	GEO	Other orbits	Total
21,659	915	12,689	35,263

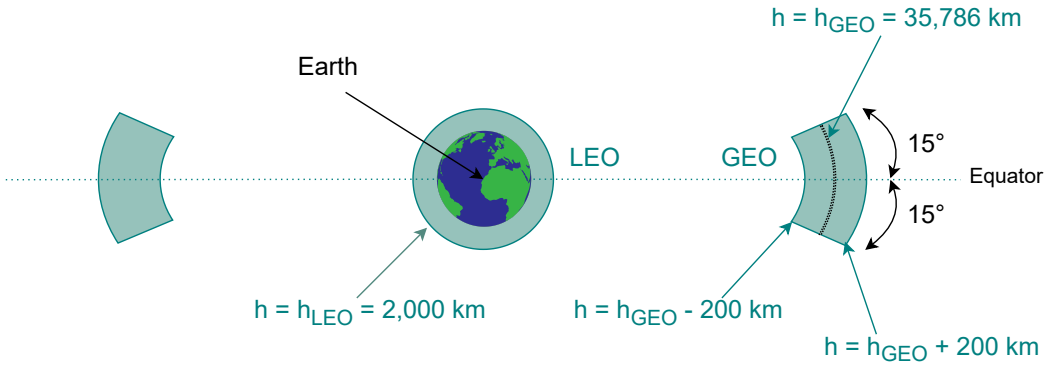

Figure 2.4: [Not to scale] Definition of LEO and GEO protected regions.

Future evolution: As mentioned earlier, the number of objects orbiting Earth has been steadily increasing in recent years, this is highlighted in ESA Space Debris Office (2023). In their environmental report, they conducted a simulation to predict the future evolution of the debris population, considering two scenarios: one based on the current launch rates and another with no future launches after 2023. Both scenarios are depicted in Figure 2.5. The simulations reveal two important findings. Firstly, even if the space industry halted all future launches today, the population of space debris would still increase due to collisions and explosions. Secondly, projecting current launch patterns indicates that the number of collisions among existing space debris objects will contribute to further growth in the space debris population. This extrapolation factors in ongoing explosions (at the 2023 rate, which may increase in the future), assumes constellations will maintain minimal post-mission disposal success rates, and relies on the continuation of currently estimated rates. From these simulations, it becomes evident that without intervention, space accessibility could become restricted. Consequently, efforts are underway to develop collision detection and avoidance methods, as well as mitigation techniques, which will be explored in the upcoming sections.

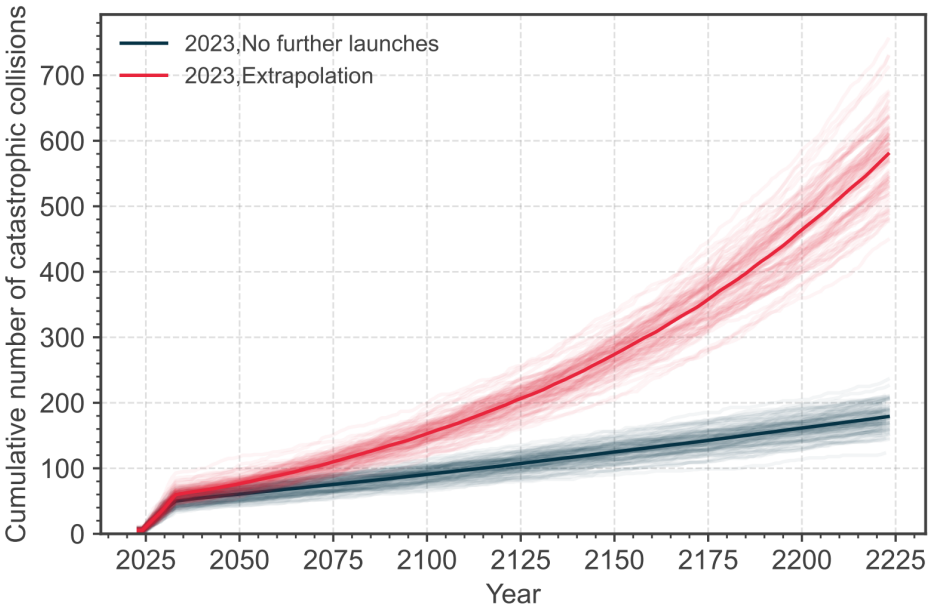

Figure 2.5: Long-term simulation of the cumulative number of objects in LEO. Environment evolution for extrapolation and no further launches scenarios – ESA Space Debris Office (2023).

Satellite constellations: In the previous sections, constellations were not explicitly included in the study of environmental evolution. These past years, the constellation sector has taken a major place in the space industry, contributing to the NewSpace era. Majorly located in LEO, these mega-constellations are systems of thousands of satellites working together towards a common objective. Operating at lower altitudes offers rapid and efficient communication with Earth, but it necessitates a large number of satellites to ensure comprehensive coverage, consequently further augmenting the number of objects in space. This increase has been visualised in Figure 2.2 in which an exponential curve is being observed starting in 2019; this marks the beginning of mega-constellation launches, such as Starlink[4], Eutelsat OneWeb[5] or E-Space[6]. While constellations offer numerous advantages, such as global Earth connectivity, they are also accompanied by certain drawbacks. These include issues such as light pollution, potential harm to astronomical observations, and, notably, the generation of space debris. The ESA Space Debris Office, as outlined in their environmental report from April 2023, has observed the repercussions on the number of objects in space resulting from the deployment of constellations. Their findings reveal that the use of constellations in 2021 led to a doubling of the rate of object proliferation in space. Unless there is a shift in behaviour, this impact is expected to worsen in the coming years. To gain a deeper understanding of their reach, an analysis of some of the larger and more significant constellations will be conducted, such an overview can be found in Table 2.2. Based on the collected data and observations, it becomes evident that constellations pose a significant threat to the space environment. For this reason, mitigation procedures ranging from mitigation methods, collision detection or CAMs should be applied.

Table 2.2: Some of the largest satellite constellations[7].

The largest satellite constellations					
Constellation	**Network**	**Altitude**	**1st Launch**	**Function(s)**	**Country**
Eutelsat OneWeb	648	LEO	2019	Internet	US
Starlink Gen 1	4,408	LEO	2019	Internet	US
SatRevolution	1,024	LEO	2019	Earth observation, Moon, Mars	POL
Galaxy Space	1,000	LEO	2020	Internet	CHN
E-Space	350,000	LEO	2022	Internet, Space debris removal	FRA
Astra	13,620	LEO	2022	Internet	US
Amazon (Kuiper)	3,236	LEO	2022	Internet	US

2.2 Space Debris Mitigation

Throughout the years, some non-biding rules, have been set up for the space sector to reduce the proliferation of space debris. Given the rising quantity of space debris, both government and private entities must adhere to these guidelines, and these recommendations should regularly be revised to safeguard the future of space. The following section will address several international guidelines that have been implemented over the years; the 2005 Inter-Agency Space Debris Coordination Committee (IADC) guidelines in Section 2.2.1, the 2007 United Nations (UN) guidelines in Section 2.2.2, and the new 5-year disposal rule established in 2022 by the Federal Communications Commission (FCC) in Section 2.2.3.

[4] Starlink, https://www.starlink.com/ [Last visited on 02/03/2024].
[5] Eutelsat OneWeb, https://oneweb.net/ [Last visited on 02/03/2024].
[6] E-Space, https://www.e-space.com/ [Last visited on 02/03/2024].
[7] Newspace, *NewSpace Constellations*, https://www.newspace.im/ [Last visited on 26/01/2023].

2.2.1 IADC Space Debris Mitigation Guidelines 2005 – Revised in 2007

These guidelines have first been formulated in 2002 by the IADC. And have been updated in the past years.

Mitigation measures (IADC Steering Group, 2020):

- Guideline 1: Limit debris released during normal operations.
- Guideline 2: Minimise the potential for on-orbit break-ups.
  - Minimise the potential for post-mission break-ups resulting from stored energy,
  - Minimise the potential for break-ups during operational phases, and
  - Avoidance of intentional destruction and other harmful activities.
- Guideline 3: Post-mission disposal.
  - Geosynchronous region to graveyard disposal,
  - Objects passing through the LEO region (“25 years disposal rule”), and
  - Other orbits.
- Guideline 4: Prevention of on-orbit collisions.

2.2.2 (UN) Space Debris Mitigation Guidelines – 2007

These guidelines were adopted in 2007 by the UN Committee on the Peaceful Uses of Outer Space (COPUOS). It has a membership of 95 Member States (from the UN Member States).

Mitigation measures (United Nations – Office for Outer Space Affairs, 2010):

- Guideline 1: Limit debris released during normal operations.
- Guideline 2: Minimise the potential for break-ups during operational phases.
- Guideline 3: Limit the probability of accidental collision in orbit.
- Guideline 4: Avoid intentional destruction and other harmful activities.
- Guideline 5: Minimise the potential for post-mission break-ups resulting from non-used and stored energy.
- Guideline 6: Limit the long-term presence of spacecraft and launch vehicle orbital stages in LEO after the end of their mission.
- Guideline 7: Limit the long-term interference of spacecraft and launch vehicle orbital stages with GEO after the end of their mission.

2.2.3 5-Year Disposal Mitigation Guideline – 2022

In response to the increasing debris issue in LEO, a new regulation has been introduced, overwriting one of IADC’s post-mission disposal guideline. According to this regulation,

adopted by the FCC in September 2022, newly launched spacecraft concluding their missions within orbits below 2,000 km must start de-orbiting as soon as feasible and no later than five years after mission completion. By working towards meeting these guidelines, the safety of satellites can be achieved (Federal Communications Commission, 2022).

2.3 Space Surveillance

Responding to the urgent need to prevent space debris collisions, several collision-detection techniques are being developed, enhancing tracking capabilities and enabling better protection against collisions. The first step in detecting space debris collisions is to catalogue as many space objects as possible and continuously monitor their trajectory to identify potential collisions and their associated risks, all while preparing response strategies.

Today, the largest space surveillance network is the Space Surveillance Network (SSN). It makes use of ground-based radars, commands and control systems, which detect, track, and catalogue man-made objects. To perform such tasks it uses several sensors; phased-array radars, conventional radars, and electro-optical sensors. Upon precise tracking of an object, the SSN has the capacity to provide advanced notice regarding potential collisions and the requirement for manoeuvres (Sgobba and Allahdadi, 2013). The current SSN has achieved comprehensive tracking, encompassing nearly all objects with diameters exceeding 1 m and 95% of objects larger than 30 cm in diameter. Some objects as small as 10 cm can also be tracked by them (National Research Council, 1997). Data on the tracked debris are open to the public through the satellite catalogue (SATCAT), which is updated daily and can be consulted on Space-Track[8]. In this catalogue, several satellite-related pieces of information are available, such as the type of satellite, their decay date (if applicable), but also information on their latest TLEs. This sequence of information, with details defined in Section 3.2, is commonly used to get information on the satellite state at a specific given epoch. Additionally, the US Department of Defence (DoD), through its Global Space Surveillance Network (GSSN), is actively keeping track of space debris, making use of ground-tracking sensors. For now, most of the catalogue's objects are 10 cm or larger (Buchs and Florin, 2021).

On the European side, ESA, through its Space Situational Awareness Programme (SSAP) and using a Space Surveillance and Tracking (SST) system, is also creating a space debris catalogue. They are able to detect potential collisions, fragmentation events and make reentry predictions. In addition to that, start-ups such as Aldoria are also tracking space objects, which are located in low, medium and Geostationary orbits. Images are received from their own telescopes from which they can extract astrometric and photometric data[9].

These various international programs employ a diverse array of telescopes, sensors, and radars, each possessing distinct capabilities for observing and tracking debris. These techniques vary not only in their ability to detect debris of varying sizes but also in their measurement precision, including uncertainties in position ($\boldsymbol{\Delta r}$) and velocity ($\boldsymbol{\Delta V}$). For example, the current US SSN is capable of observing a large range of altitudes using its surveillance radars. Space surveillance plays a pivotal role in advancing space sustainability. By providing data on the different objects located in space, objects positions and velocities can be propagated and conjunction analysis performed. From this analysis, conjunction data messages (CDM) and conjunction summary messages (CSM) can be generated. These signals contain precise information on the objects' orbits and comprehensive covariance of uncertainties, enabling operators to make informed decisions on when and how to take actions.

[8] Space-Track, *Satellite catalogue*, https://www.space-track.org/ [Last visited on 01-02-2024].
[9] Aldoria, *Unique Technology Breakthrough*, https://www.aldoria.com/technology [Last visited on 12/02/2024].

2.4 Collision Avoidance Methods

After cataloguing space debris, it becomes imperative to employ technologies that assess the likelihood of potential collisions and determine whether collision avoidance measures are necessary. This is particularly crucial for missions in LEO and GEO where the probability of collision is higher than elsewhere, being further accentuated during long-duration missions. These elements will be introduced in the following section. To start with, currently available methods to compute collision avoidance probabilities will be presented in Section 2.4.1. Subsequently, various techniques designed to safeguard space vehicles from debris will be detailed. These include passive protection in Section 2.4.2, principles and examples of active protection in Section 2.4.3 and, an examination of different types of thrusting manoeuvres will be given in Section 2.4.4.

2.4.1 Space Debris Collision Probability

The first step in assessing a risk of collision is performed by computing the probability of collision. From this information, it can be decided whether or not a CAM is required. When dealing with satellites located in LEO, short-term conjunctions are present, for which a number of assumptions on their motion, can be made:

- The bodies move along straight lines with a constant velocity, whose uncertainties can be neglected,
- The uncertainty in the position of the bodies can be described using a 3-dimensional (3D) Gaussian distribution,
- The position uncertainties of both objects are not correlated and are constant and thus are equal to the one at the time of closest approach, and
- Both objects are considered spherical.

To perform robust and accurate collision probability computations, several methods are currently available, which are all based on the principles of Gaussian distribution and covariance. The most used models are the ones developed by Foster et al. (1992), Patera (2001) which is known to be approximately 20 times faster than Foster's (Alfano, 2007), but also Alfano (2005)'s and Chan (1997)'s methods, being orders of magnitude faster than the other three methods. In addition to those four well-known methods, a DUT student, Leon Dasi (2022), has developed a new method, which enables the computation of collision probability between two space objects having any geometry. It does so by additionally decreasing the total computational time by still meeting high accuracy in the method. More information on this method will be introduced in Section 2.5.

As will be discussed at a later stage, no collision probability analysis will be performed and the need for a robust method is not necessary for this specific study. Indeed, for the mission scenario at hand, a conjunction analysis is solely required for a conjunction to be found and for the manoeuvring system to be triggered. The mission starts the moment a conjunction is found, but the process of finding and processing the data leading to the conclusion that a manoeuvre is necessary is beyond the scope of this MSc thesis. The reason for this is that it could be a MSc thesis work on its own, with other DUT students having already worked on the topic. For this reason, no additional details will be given on collision probability. Including such analysis in the study would introduce an additional dimension, considering extra external factors and constraints. This valuable addition will be addressed in Section 9.2.

2.4.2 Passive Protection

Passive protection can be defined as a way to shield spacecraft from debris. For human spaceflight, space vehicles should be shielded against orbital debris. Even if the size of these particles seems insignificant, they hit space vehicles with hyper-velocities in the order of 14 km/s, which can highly damage the vehicle through penetration or degradation. For this reason, it is important, especially for a manned mission, to make use of protective materials that shield against the impact of uncatalogued debris avoiding catastrophic events. In the case of the ISS, it is being shielded using a Whipple shield. This shielding method consists of three primary shielding configurations, which can protect it from space debris up to 3 mm in size[10]. To be protected from larger objects, the ISS should perform CAMs, which fall under the 'active protection' category.

2.4.3 Active Protection

Not every satellite possesses the capability or equipment to be safeguarded against debris, and being shielded against every type of debris is not feasible. As a result, active protection becomes crucial in scenarios where collisions are likely. Active protection involves taking proactive measures to avoid debris and, consequently, collisions. In the evolving space era, collision avoidance has become increasingly crucial, particularly in crowded orbits like LEO where prior collisions have left a substantial amount of inert debris, which accounts for roughly 90% of the debris in LEO. Presently, the ISS conducts a few CAMs each year with an annual average of approximately 1.2 CAMs. For example, between the year 1999 and August 2020, the ISS executed a total of 27 CAMs (NASA Orbital Debris Program Office, 2020). This figure is expected to rise as the debris population continues to grow. More details on different relevant past CAMs will be introduced hereafter .

ISS CAMs: The ISS has been more at risk in the past years, due to the increasing number of debris in LEO. In 2023 alone, it had to perform four CAMs, while before that, it had performed a total number of 33 collision avoidance manoeuvres since 1999, according to NASA Orbital Debris Program Office (2022). With its numerous manoeuvres, the ISS is an excellent candidate to study for data on CAM. To predict a potential collision between the ISS and debris, the Joint Space Operations Centre (JSOC) tracks the space debris object through radar tracking, keeping a catalogue of debris. Three times a day these objects are tracked against the ISS trajectory and the ISS Trajectory Operations and Planning Officer (TOPO) is notified if any object is susceptible of passing near a ±2 (local vertical) x 25 x 25 (local horizontal) km volume within the next 72 hours. If this is the case, TOPO computes the collision probability (P_c) and then notifies flight control teams in Houston and Moscow of a potential collision hazard. If this collision probability is larger than 1/100,000[11] (number specific to the ISS, since it is a manned vehicle), a debris avoidance manoeuvre (DAM) of the ISS will be conducted to avoid the debris (Cooney, 2016). Over the years, several avoidance manoeuvres have been performed by the ISS, some of which have been listed in Table 2.3.

[10] Aerospace, *Space Debris 101*, https://aerospace.org/article/space-debris-101 [Last visited on 31/05/2022].

[11] Science ABC, *How Do The ISS And Other Satellites Protect Against Space Debris?*, https://www.scienceabc.com/innovation/how-do-the-iss-and-other-satellites-protect-against-space-debris.html [Last visited on 03/06/2022].

[12] European Space Agency, *ESA ISS Science & System - Operations Status Report Increment 17*, https://www.esa.int/Science_Exploration/Human_and_Robotic_Exploration/Columbus/ESA_ISS_Science_System_-_Operations_Status_Report_br_Increment_17 [Last visited on 31/01/2023].

Table 2.3: Some of ISS collision avoidance events.

ISS collision avoidance events					
Debris	**Date**	**Predicted main cause**	**Propulsion system/ Manoeuvre event**	**ΔV**	**Prop. Δt**
Old debris	27/08/08	–	Retrograde manoeuvre. The ISS ATV's aft thrusters were relocated to the front of the station slowing it down (lower altitude)[12].	1 m/s	–
Pegasus rocket fragment	03/12/21	$\|\|\Delta r\|\|$ = 5.4 km	Russian Progress 79's thrusters. The propulsion compartment contains the main engine, a liquid-fuelled propulsion system and a low-thrust engine for orientation.	–	2 min 41 s
Fengyun-1C fragment	19/04/20	–	–	–	–
Iridium-33 fragment	26/07/15	$P_c >$ threshold	Progress 58P thrusters – NASA Orbital Debris Program Office (2015).	0.5 m/s	–
1) Pegasus debris 2) Fengyun-1C fragment	27/09/15	$\|\|\Delta r\|\|$ low	Progress resupply vehicle thrusters – NASA Orbital Debris Program Office (2015)	0.3 m/s	–
Unidentified object	22/09/20	$\|\|\Delta r\|\|$ = 1.39 km	ISS Progress 75 thrusters	–	2 min 30 s
Meteor 2-5 debris	23/04/15	–	Progress 58 thrusters – NASA Orbital Debris Program Office (2015)	0.3 m/s	2 min 20 s
Discarded Minotaur-I 4th stage	08/06/15	–	Progress 58 thrusters – NASA Orbital Debris Program Office (2015)	0.3 m/s	
Meteor 2-5 fragment	16/07/15	$P_c >$ threshold, but not enough time	No manoeuvre. "Shelter-in-place" procedure, crew emergency return with the Soyuz capsule– NASA Orbital Debris Program Office (2015)	N.A	N.A

Starlink CAMs: Several CAMs are performed recursively, between two active vehicles or between an active and inert vehicle. An example of collision avoidance between two active vehicles is the Aeolus/Starlink conjunction, which took place on September 2, 2019, at an altitude of 320 km (LEO). In this case, the collision avoidance manoeuvre had to be performed by ESA since SpaceX did not want to be involved. The CAM would thus increase Aeolus's altitude by 350 m, ensuring the remaining distance between both vehicles was high enough for the conjunction probability to be lowered[13]. To perform this, three consecutive thruster burns (Aeolus' thrusters) were performed half an orbit before the potential collision[14].

Space Shuttle CAMs: Similarly to the ISS or Starlink case, the SSN is responsible for warning the shuttle of potential conjunctions with debris. This debris is catalogued by them and closely tracked daily (National Research Council, 1997). Rather than using a probability-based approach, like in the case of the ISS, it makes use of the geometry of the shuttle by using an 'alert box' approach consisting of a 5 × 25 × 5 km box surrounding the shuttle, which no debris could ever penetrate. Through simulations, if it is observed that space debris could potentially penetrate this box then action should be taken. When a manoeuvre is necessary, then, on average, the required ΔV needed to manoeuvre will be 0.3 m/s, which, according to Loftus (1997), would require a propellant mass of about 11 to 14 kg.

[13] eoPortal, *Aeolus / formerly ADM (Atmospheric Dynamics Mission)*, https://www.eoportal.org/satellite-missions/aeolus#eop-quick-facts-section [Last visited on 27/01/2023].

[14] European Space Agency, *ESA Spacecraft Dodges Large Constellation*, https://www.esa.int/Space_Safety/ESA_spacecraft_dodges_large_constellation [Last visited on 03/06/2022].

Envisat CAMs: Envisat's first two CAMs were performed in 2004 (Klinkrad et al., 2005). The first one occurred on September 2, 2004, when the Cosmos-1269 satellite was predicted to pass by, with a collision probability of $2.186 \cdot 10^{-4}$ at a distance of 1.3 km of Envisat. To avoid such debris, a fuel-efficient manoeuvre was performed using a small along-track manoeuvre (oblique approach geometry) with a ΔV of 0.3 m/s using a total of 160 g of propellant for both the avoidance and restitution manoeuvres. The second potential collision took place between Envisat and a fragment of the Zenith-2 orbital stage, with a minimum approach distance of 81 m and maximum collision probability of $5.5 \cdot 10^{-4}$ (exceeding the accepted $P_c < 10^{-4}$). To manoeuvre, a ΔV of 0.4 m/s was required to move Envisat away and another 0.39 m/s was required to move Envisat back to a lower orbit after the event.

ESA's autonomous collision avoidance system (AUTOCA): An autonomous collision avoidance system is correctly being developed within the ARTES program. This system will make use of artificial intelligence (AI) and Machine Learning (ML) to predict how significant and critical the potential conjunction between two objects would be, as well as the risk of a collision happening. Depending on the data resulting from such analysis AUTOCA will then be able to say whether or not a manoeuvre is required, or possible[15].

The commonalities and discrepancies in past CAMs: Through the different described CAMs, which happened in the past, several commonalities have been found. Understanding those is crucial to, later on, define the specific mission concepts and requirements linked to them.

- **Tracking and detection:** To start with, the space debris is exclusively being tracked actively from the ground by the SSN, meaning no onboard equipment is used to track any of the debris to perform collision avoidance. Making use of a ground station significantly increases the time between collision perception and the performance of actual manoeuvres. According to National Research Council (1997), it has been estimated that once a collision detection warning has been received it would take approximately two hours to communicate instructions, prepare the ISS for the manoeuvre, boost the thrusters, and for the actual manoeuvre to be performed. This is extremely long when comparing it with the velocity at which both objects could be moving. For reference, a Starlink satellite ($h \approx 550$ km) has an orbital period of approximately 95 minutes. To reduce the detection time to a lower appropriate value, opting for an on-board detection mechanism could be a viable option. However, a downside of such technology is the field of view that is not substantial since both objects will be moving in a circular motion around Earth, meaning an entire 360° overview is not possible, unlike with ground station. A solution to this could be to propagate debris and satellites' states and perform an analysis of this propagated state, reducing accuracy.

- **Predicted cause for manoeuvres:** When analysing previous collision avoidance incidents, manoeuvres are executed when one of the prescribed criteria is not met. These criteria are frequently determined by either an imaginary safety zone or the likelihood of a collision. Furthermore, in these scenarios involving high velocities, timing is of utmost importance. Time must not be overlooked, and as a result, a manoeuvre can only be performed if there is sufficient time to carry out the procedure safely.

[15] European Space Agency, *Autonomous Collision Avoidance System For NGSO*, https://artes.esa.int/projects/autoca [Last visited on 27/01/2023].

- **Manoeuvring execution:** Looking at various examined CAMs, a common trend emerges in the manoeuvres executions. In many cases, depending on the objects involved (typically two or more), one of these objects, often the larger one, initiates the manoeuvre. This involves adjusting the spacecraft's orientation and utilising onboard thrusters to execute a prograde manoeuvre, effectively initiating a de-orbit process. Subsequently, this is counteracted by returning to its initial orbit. This time of events is frequent for the ISS, which most of the time uses the thrusters located on the Progress vehicle, producing a ΔV in the range 0.3 – 0.5 m/s. Two other, less common types of active avoidance were carried-out by the ISS. According to NASA Orbital Debris Program Office (2015), a 'shelter-in-place' type of manoeuvre was executed in 2015, where the ISS crew went into the Soyuz capsule in case of emergency return. A retrograde manoeuvre was performed in 2008, in which the aft thrusters were relocated at the front of the station to slow down the ISS, lowering its altitude. From the required ΔV budget of 1 m/s, this sequence costs more than twice the prograde manoeuvre.

From this analysis it can be concluded that the most common and less costly CAMs are prograde manoeuvres, those in which the orbital velocity will be increased, raising the altitude of the orbit. The ΔV required to move the ISS is in the range of 0.3 – 0.5 m/s. In addition to that, it has been observed that collision avoidance is performed solely based on tracking data from the ground station and tracking is not performed autonomously onboard the spacecraft. The tracking database is almost exclusively done by the US SSN.

2.4.4 Low and High-Thrust Manoeuvres

Today, most satellites in space use high-thrust chemical propulsion, with a few exceptions like the Eutelsat 172B satellite and the recent Starlink constellation, which employ low-thrust electrical propulsion. Chemical propulsion has long been favoured for its simplicity, affordability, and high thrust output. On the other hand, electrical propulsion is gaining prominence due to its environmental benefits and alignment with the trend toward smaller satellites. It reduces satellite mass, cutting launch costs and catering to the smallsat market. It is eco-friendly, safer, and more redundant than chemical propulsion. However, it has lower thrust, leading to longer transfer times and making last-minute avoidance manoeuvres challenging. As a nascent, evolving technology, it currently comes at a higher cost and offers lower efficiency compared to chemical propulsion. Two examples of propulsions will be introduced hereafter.

Starlink's low-thrust propulsion: The Starlink project uses an ion propulsion system (powered by Krypton) to perform orbit raise and de-orbit at the end of its lifetime. It is equipped with an autonomous collision avoidance subsystem, which makes use of its low-thrust propulsion (hall-effected thrusters) to perform guidance and avoid debris when required[16].

SwarmB high-thrust propulsion CAM: On January 15, 2017, the SwarmB satellite had to make use of its cold gas propulsion subsystem to avoid a high-probability collision with a debris generated from Cosmos-375. To avoid such piece of debris the satellite's cold-gas thrusters had to work for about 44 seconds, giving an impulse of approximately 7 – 8 mm/s[17].

[16] Starlink, *World's most Advanced Broadband Satellite Internet*, https://www.starlink.com/satellites [Last visited on 01/02/2023].

[17] Spaceflight101, *ESA Satellite faces close Orbital Conjunction with Soviet Space Debris, Avoidance Maneuver Canceled*, https://spaceflight101.com/swarm/swarm-close-orbital-conjunction/ [Last visited on 01/02/2023].

2.5 Relevant DUT Work

Presently, there is a notable surge in scientific research related to collision avoidance in various domains, including ongoing studies at DUT. Several students have conducted research on topics directly associated with space debris and, more significantly, collision avoidance phenomena. The examination of these studies holds direct relevance for this study, as it provides an opportunity to identify commonalities and derive recommendations for enhancing previous research. This section will present some pertinent studies and their outcomes.

Debris conjunction analysis: Leloux (2012) analysed the different filtering techniques used during conjunction analysis. Some of the filters detailed in this work will be directly relevant to the work at hand when performing the conjunction analysis and finding a conjunction.

Satellites impulsive collision avoidance manoeuvres: Wolfhagen (2020) has analysed what would be the most attractive approach to impulsive collision avoidance (in LEO) by taking into account aspects such as the optimisation of constellation performance, the collision probability after manoeuvre and the benefits linked to those. From their research, they could conclude that using a multi-objective optimisation procedure was beneficial to offer a set of feasible solutions and thus enable the optimisation of collision probability, propellant consumption as well as constellation performance loss.

Spacecraft GNC: Alforja Ruiz (2020) examined the robust de-orbiting of a large spacecraft having uncertain mass parameters. They went into detail about the guidance, navigation and control (GNC) of this spacecraft. In their guidance analysis, they examined strategies for complete orbit reduction and final de-boost. The spacecraft's autonomous attitude commands were executed using the pseudo-mean anomaly as a control variable, chosen for its ability to manage both time and satellite position. Additionally, they focused on the navigation system, emphasising the necessity of accurately modelling navigation sensor outputs and data errors. Their design aimed to closely replicate the e.Deorbit mission by incorporating an array of sensors, including an Inertial Measurement Unit (IMU), Global Positioning System (GPS), magnetometer, star tracker, and sun sensor. To mitigate signal errors from the sensors, they used a modified Extended Kalman Filter (EKF). Lastly, within the control subsystem, they employed an Incremental Nonlinear Dynamic Inversion (INDI) approach. Some of these design choices may be applicable to the current mission.

Collision probability: Leon Dasi (2022) developed a new method capable of computing the collision probability between space objects by significantly reducing the computational time required to perform high-accuracy propagation of uncertainty. Additionally, this method can be applied to a large number of space vehicle conjunctions, since it can take as an input any vehicle 2D geometry. Their method can also be extended to 3D shapes; thus enabling the collision probability computation to be accurately applied to any satellite. This was performed using a hybrid Differential Algebra and Gaussian Mixture Model (DA-GMM) method.

Software and hardware simulations for CAM: Cattani (2022) looked into how the GNC subsystems used to improve CAM could be supported and tested using the GRALS (Generic Rendezvous and Docking Simulator) laboratory of the European Space Research and Technology Centre (ESTEC). In their GNC sequence, they used an EKF, which they developed and integrated into Simulink, a GPS, an IMU, a laser rangefinder, and an INDI controller.

GNC for active debris removal: Conings (2022) looked into a GNC system to be used for active debris removal. Their guidance sequence was meticulously designed to achieve a fuel-efficient and time-effective trajectory while minimising thruster plume impingement. Convex guidance was used and proved to be optimal in terms of robustness, autonomy, and computational efficiency, especially in the context of close-range rendezvous phases. It iteratively converges to an optimal solution within a predetermined number of iterations, provided that a feasible solution exists within the optimisation space. Given its relevance to CAMs, their guidance system will also be a focal point of examination in the current research.

2.6 Reference Scenario

One of the primary goals of the research was to employ a scenario closely resembling reality. This aspect was meticulously examined to encompass a wide range of potential future mission scenarios and provide a comprehensive depiction of the present space environment. Therefore, it was determined that using a reference mission and vehicle accurately reflecting the current state of space exploration, would serve as the baseline scenario. Both the reference mission and vehicle will be described in Section 2.6.1 and Section 2.6.2, respectively.

2.6.1 Reference Mission

With 70% of space debris located in LEO[18], as shown in Table 2.1, the primary threat is evident for space vehicles in this region. Addressing the root of the problem and developing methods to significantly reduce debris generation in these dense areas is of paramount importance. As a result, the decision has been made to conduct the mission in LEO, where the density of space objects is higher compared to other locations, primarily due to its cost-effectiveness for access and its suitability for high-speed, low-latency communications owing to its proximity to Earth. More specifically, Figure 2.6 and Figure 2.7, created using the current (01/01/2024) TLEs from Space-Track[19], reveal a pronounced concentration of catalogued objects within altitudes ranging from 500 to 580 km with eccentricities close to 0°. Subsequently, it can be concluded that the conjunction probability in these altitude bands is much higher than in others, rendering this region a smart choice for further analysis. Furthermore, there is a significant growth observed in the constellation sector, marked by a rapid increase in the number of satellites (Figure 2.2), with certain constellations even aiming to deploy no fewer than 350,000[20] (Table 2.2). This growing market is the future of space exploration, making it crucial to understand and analyse its impact. For these reasons, it has been decided to use as a reference vehicle SpaceX's Starlink constellation, which is located in the critical band of 550 km altitude, 53° inclination and close to 0° eccentricities[21], meeting both criteria described earlier. With its planned-to-be-launched 42,000 satellites (Figure 2.8), the Starlink constellation has the aim of providing global coverage Internet access to the entire globe, currently based in 37 countries[22], it already has deployed a little over 5,500 satellites as of 2024 as displayed in Figure 2.9, where all satellites part of the Starlink constellation can be observed. The studied conjunction, detailed in Chapter 4, will involve both a Starlink satellite and trackable debris. Although the exact dimensions of this debris

[18] European Space Agency, *Debris objects in low-Earth orbit*, https://www.esa.int/ESA_Multimedia/Images/2008/03/Debris_objects_in_low-Earth_orbit_LEO2 [Last visited on 09/06/2022].

[19] Space-Track, *Satellite catalogue*, https://www.space-track.org/ [Last visited on 01/02/2024].

[20] E-space, *Bridging Earth and Space*, https://www.e-space.com [Last visited on 28/06/2022].

[21] Starlink, *How Starlink works*, https://www.starlink.com/satellites [Last visited on 25/06/2022].

[22] Starlink, *Starlink Updates*, https://www.starlink.com/updates [Last visited on 13/10/2023].

are unknown, its trajectory is thoroughly documented. Similar to the spacecraft, the selected debris for the simulation is an appropriate choice because its position in space allows it to represent the majority of debris found in LEO on a global scale. More details on the specifics of this spacecraft and debris will be introduced in Section 2.6.2.

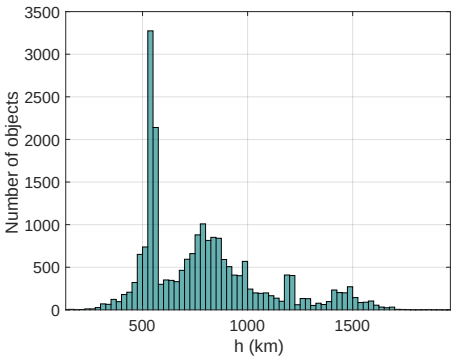

Figure 2.6: Distribution of all objects in LEO by mean altitude as of 01/01/2024.

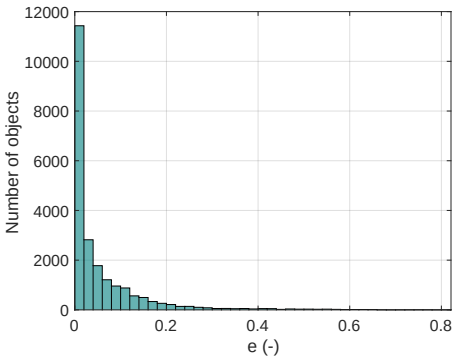

Figure 2.7: Distribution of all objects in LEO by eccentricities as of 01/01/2024.

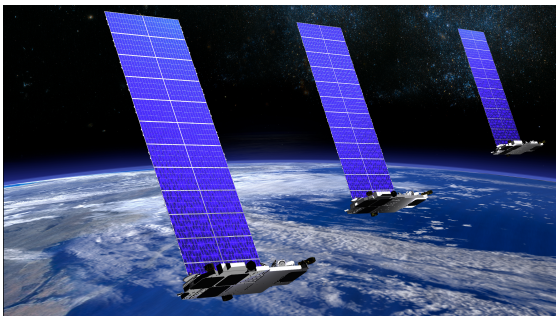

Figure 2.8: SpaceX's Starlink satellite illustration[23].

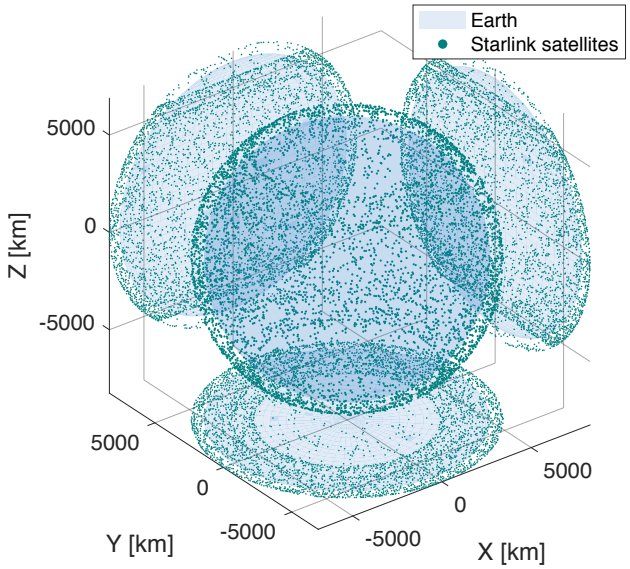

Figure 2.9: 3D representation of all Starlink satellites currently in orbit as of 01/01/2024.

[23] Shutterstock, *Starlink Satellite Illustration*, https://www.shutterstock.com/image-illustration/internet-broadcast-satellite-train-around-earth-2116913408 [Last visited on 11/03/2024].

2.6.2 Reference Vehicle and Debris

For the mission at hand, both a spacecraft and space debris will be of interest. Their respective characteristics, required for simulation purposes, will be introduced hereafter.

Reference vehicle (Starlink-4039): Due to the limited availability of information about Starlink, it has been determined that for the thesis project, in cases where essential data about the satellite is lacking, the characteristics of several operational satellites will be examined. From this analysis, a generalised model resembling the known attributes of Starlink will be developed, serving as the basis for all simulations. Although very little information on the Starlink satellites' geometry and mass distribution is available, deductions can be made using illustrations and other available data sources such as the Falcon 9 user guide[24] and pictures of how the satellites are stacked in the vehicle, before launch. From this, the combined height of a stack of 60 satellites can be approximated to be 6.7 m. Considering two satellites per unit of height, with spacers between them, the stowed height of each satellite can be estimated to be around 20 cm. The internal diameter of the fairing available to payloads is 4.6 m, although there is a slight allowance for space around the edges. Assuming a 5 cm gap between the satellite's edge and the fairing, the longer edge of the satellite can be approximated to be 3.2 m in length, with the shorter edge being half of that at 1.6 m (2:1 ratio).
In addition to its main body, Starlink has a large solar panel beam whose dimensions should also be deduced. Divided into 12 segments, each segment's longer edge matches the 3.2 m width of the satellite body. Each segment appears to feature 32 x 4 solar cells, with each cell maintaining a 2:1 aspect ratio. The dimensions of each segment can be approximated to be 3.2 m by 0.8 m (excluding the edges), leading to an approximate surface area of 30.72 m^2 for the solar panel[25]. The solar panel's thickness can be approximated from literature to be 7.5 mm. In Figure 2.10, some sketches of Starlink, with its deduced dimensions can be observed.
In addition to the dimensions, the satellite mass should also be estimated. While it is known that Starlink satellites weigh approximately 260 kg it is unknown how this mass is divided between the satellite's main body and its solar panel. In order to deduce the solar panel mass, it was decided to use solar panel specifications from Spectrolab, being a leader in solar panels for space applications. It was found that for space solar panels[26] the densities range between 1.76 and 2.08 kg/m^2. Taking the highest mass density leads to a solar panel mass of $\approx$ 64 kg and thus a satellite body mass of 196 kg. While these pieces of information await official confirmation from SpaceX, they have undergone comparison with other satellites and various known physical properties. As a result, they have been deemed acceptable values for utilisation. An overview of some of Starlink's characteristics[27] can be found in Table 2.4.
Additionally, taking the overall geometry of Starlink, the inertia of the satellite's main body and solar panels (with respect to their respective com) can be computed as detailed below, assuming a uniform mass distribution and approximating them to a box and plate respectively. By making use of the box approximation, with a volume of 50.18 m^3 the overall satellite inertia can also be computed (slightly over-estimated) using this kind of geometry.

[24] SpaceX, *Falcon User's Guide*, https://www.spacex.com/media/falcon-users-guide-2021-09.pdf [Last visited 22/02/2024].
[25] Blogger, *Starlink Satellite Dimensions Estimates*, https://lilibots.blogspot.com/2020/04/starlink-satellite-dimension-estimates.html [Last visited 22/02/2024].
[26] Spectrolab, *Space Solar Panels*, https://www.spectrolab.com/DataSheets/Panel/panels.pdf [Last visited on 04/03/2024].
[27] Starlink, *Technology*, https://www.starlink.com/technology [Last visited on 01/10/2023].

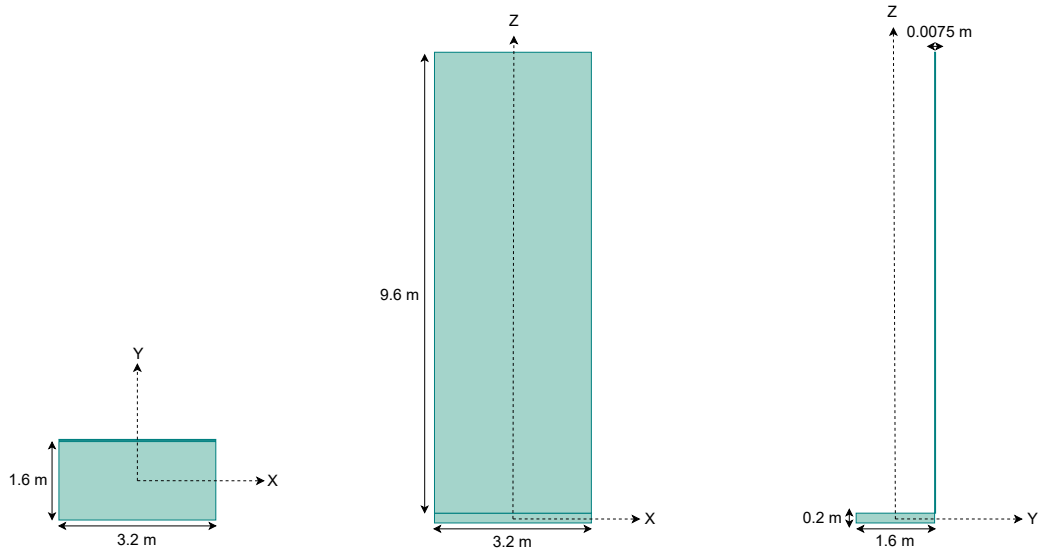

Figure 2.10: Starlink satellite sketches in 2D views.

$$J = \begin{bmatrix} \frac{1}{12}m(y^2+z^2) & 0 & 0 \\ 0 & \frac{1}{12}m(x^2+z^2) & 0 \\ 0 & 0 & \frac{1}{12}m(x^2+y^2) \end{bmatrix} \tag{2.1}$$

$$J_{body} = \begin{bmatrix} 42.47 & 0 & 0 \\ 0 & 167.91 & 0 \\ 0 & 0 & 209.07 \end{bmatrix} \tag{2.2}$$

$$J_{solarpanel} = \begin{bmatrix} 491.52 & 0 & 0 \\ 0 & 546.13 & 0 \\ 0 & 0 & 54.61 \end{bmatrix} \tag{2.3}$$

$$J_{Starlink(box)} = \begin{bmatrix} 2136.33 & 0 & 0 \\ 0 & 2302.27 & 0 \\ 0 & 0 & 277.33 \end{bmatrix} \tag{2.4}$$

A Starlink satellite possesses low-thrusting capabilities. Given that the majority of Earth-orbiting satellites currently rely on chemical propulsion, the current analysis will primarily use high-thrust propulsion for testing purposes. Nonetheless, an examination of the system's limitations and its ability to accommodate low-thrust and large ranges of different high-thrust propulsion will also be conducted. This evaluation is crucial as the ultimate aim of this autonomous system is its widespread usage and suitability. The primary propulsion system chosen for analysis will use one of Ariane Group's 10 N bi-propellant thrusters[28]. This thruster model is specifically engineered for precise attitude, trajectory, and orbit control of satellites and is already extensively deployed. With a nominal thrust of 10 N and a specific impulse of 292 s, it offers reliable performances.

[28] Ariane Group, *Chemical Bi-propellant Thruster Family*, https://www.space-propulsion.com/brochures/bipropellant-thrusters/bipropellant-thrusters.pdf [Last visited on 16/04/2024].

[29] Seradata, *Satellite Catalogue*, https://www.seradata.com/ [Last visited on 01/10/2023].

Table 2.4: Some of the known characteristics of SpaceX's Starlink satellites.

Starlink-4039 satellite known characteristics		
Orbit	**On-board subsystems**	**Characteristics**
• Altitude ≈ 546 km circular • Inclination ≈ 53.2194° • Eccentricity ≈ 0.000152 • Period ≈ 95.5 min	• Autonomous CAM • Star-tracker (navigation) • Optical space laser (autonomous data transmission) • 4 powerful phased antennas • 2 parabolic antennas • Ion thrusters (Krypton propelled, hall-effected thrusters) • Singular solar array • One rectangular main body	• Total mass ≈ 260 kg (minisatellite) • Body dimensions ≈ 3.2 x 1.6 x 0.2 m • Body mass ≈ 196 kg • Solar arrays: 12 deployable panels (3.2 x 0.0075 x 9.6 m) • Solar panel surface area[29] ≈ 3.2 x 9.6 m ≈ 30.72 m^2 • Solar panel mass ≈ 64 kg • Cross-sectional area: 31.36 m^2 (approximated the larger side of the satellite by a rectangle with sides 3.2 x 9.8 m) • Volume dimensions (box approximation): 50.18 m^2 (rectangle with dimensions 3.2 x 1.6 x 9.8 m) • Propulsion: Electric (Krypton)

Table 2.5: Some of the characteristics of the analysed space debris.

Debris characteristics	
Orbit	**Characteristics**
• Altitude ≈ 546 km circular • Inclination ≈ 1.4805° • Eccentricity ≈ 0.000152	• Mass: 1 kg • Body dimensions ≥ ⌀ 0.1 m

Reference Debris: The analysis of this specific piece of debris, particularly its trajectory and the details of the analysed conjunction event, will be thoroughly described in Chapter 4 in which the full conjunction analysis will be described. Unlike Starlink, the space debris geometrical characteristics are completely unknown but a range of dimensions can be assumed taking into account the fact that the debris is trackable. The US SSN tracks debris and satellites, which are approximately larger than 10 cm for LEO objects and 70 cm for GEO objects (National Aeronautics and Space Administration, 2008). It can thus be concluded that this piece of debris is larger than 10 cm in diameter. An overview of some of the debris' assumed characteristics can be visualised in Table 2.5.

2.7 Mission and Systems Requirements

With both the mission heritage, the reference mission and the vehicle descriptions, several requirements can now be set, whose identification (ID) format can be visualised in Figure 2.11. On one hand, the mission [M] requirements will encompass all requirements directly related to the satellite [SAT] or debris [DEB] characteristics, the space environment [ENV] and the reference mission [MIS]. On the other hand, the systems [S] requirements will enclose guidance [GUI], control [CON] and simulation [SIM] elements. The systems and mission requirements will be gathered in Table 2.6 and Table 2.7, respectively.

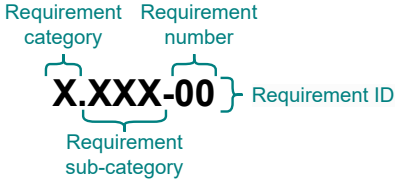

Figure 2.11: Overview of the requirements format.

Table 2.6: List of systems requirements.

Systems requirements
[S.GUI–01] The performed manoeuvre shall increase the miss distance between objects to an acceptable threshold of 10 km. **[S.GUI–02]** The main objective of the guidance system is to burn minimum fuel. **[S.GUI–03]** The manoeuvre sequence shall be performed in the most efficient manner. **[S.GUI–04]** The guidance system shall be on-board the spacecraft. **[S.GUI–05]** The guidance system shall be able to predict the optimum trajectory taking constraints on the thrust, keep-out volume and boundary conditions into account. **[S.GUI–06]** The guidance system shall be able to track the satellite for it to stay close to the nominal trajectory. Depending on the trajectory segment such distance varies between 1 and 10 km.
[S.CON–01] The mission vehicle shall be equipped with a main thruster engine. **[S.CON–02]** The main thruster engine shall be a chemical thruster. **[S.CON–03]** The main thruster engine shall have nominal thrust capabilities of 10 N in all three directions (X, Y, and Z). **[S.CON–04]** The main thruster engine shall have a specific impulse of 292 s. **[S.CON–05]** The mission vehicle shall be able to perform attitude control, assuming ideal attitude actuators. **[S.CON–06]** An LQR controller shall be used for attitude control.
[S.SIM–01] Constant trade-off between accuracy and CPU time shall be performed. **[S.SIM–02]** The simulation shall include two objects (satellite and debris). **[S.SIM–03]** All 3 DOF and 6 DOF shall be taken into account in the simulation.

Table 2.7: List of mission requirements.

Mission requirements
[M.SAT–01] The satellite shall be a satellite orbiting Earth. **[M.SAT–02]** The satellite shall have a mass of 260 kg. **[M.SAT–03]** The satellite shall have a body size of 3.2 x 1.6 x 0.2 m. **[M.SAT–04]** The satellite shall be positioned within the crucial LEO spectrum, at an altitude of 550 km. **[M.SAT–05]** The satellite shall move in a circular or near-circular orbit. **[M.SAT–06]** The mission vehicle choice shall be relevant for research. **[M.SAT–07]** The satellite shall have a solar panel size of 3.2 x 0.0075 x 9.6 m ± 10%.
[M.DEB–01] The reference debris shall orbit Earth. **[M.DEB–02]** The reference debris shall have a dimension larger than 0.1 m diameter (tracking). **[M.DEB–03]** The reference debris shall be positioned within the crucial LEO spectrum, between an altitude range of 450 to 650 km (conjunction analysis). **[M.DEB–04]** The reference debris shall come in conjunction with the reference satellite during the mission simulation time.
[M.ENV–01] All perturbations inducing an error as small as ≈ 1 m in space vehicle position over a time of approximately two orbital periods shall be included. **[M.ENV–02]** All perturbations inducing an error as small as ≈ 10^{-3} m/s in space vehicle velocity over a time of approximately two orbital periods shall be included. **[M.ENV–03]** All perturbations inducing an error as small as ≈ 10^{-6} m/s^2 in space vehicle acceleration over a time of approximately two orbital periods shall be included. **[M.ENV–04]** A trade-off between CPU time and the effect of including a complete environment model shall be performed.
[M.MIS–01] The reference mission shall cover the avoidance phase of the mission. **[M.MIS–02]** The reference mission process shall start as soon as a conjunction is flagged. **[M.MIS–03]** The miss distance between the satellite and debris shall be above the 10 km threshold after manoeuvre for it to be successful.

In this chapter, all relevant aspects of the mission heritage essential for the mission at hand have been thoroughly introduced. From this analysis, the key challenges and opportunities have been clearly identified, and the reference mission and vehicle have been outlined. Building on this foundation, the next chapter will provide a comprehensive analysis of orbital dynamics. This analysis will offer a deeper understanding of how the satellite navigates through space and the impact of various environmental factors on its trajectory and overall behavior. These insights will be detailed in Chapter 3.

CHAPTER

3 Orbital Dynamics

To accurately represent and analyse the motion, position and orientation of a space object, providing an accurate representation of its surrounding environment is crucial. The first step in doing so is to understand the different elements of such a study. These are the different reference frames used to represent the object and the diverse state variables employed to define the object's attitude. These concepts will be elaborated in Section 3.1 and Section 3.2, respectively. Subsequently, Section 3.3 will present the relevant transformations required for the conversion between different frames and elements. Moreover, satellites and debris orbiting Earth experience a multitude of forces and moments altering their trajectories. Examining and considering these perturbations is essential in motion studies, these will be presented in Section 3.4 for external forces and moments and in Section 3.5 for internal forces and moments. Finally, the equations, representing both the translation and rotational motions of the objects will be introduced in Section 3.6. An overview of the different elements that will be elaborated upon in this chapter has been displayed in Figure 3.1.

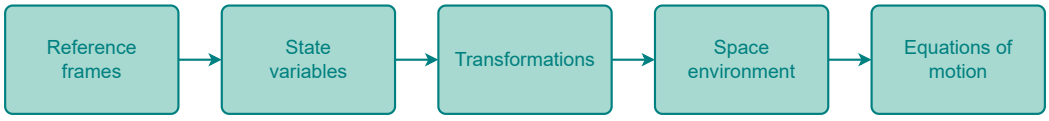

Figure 3.1: Overview of the different sections included in the flight dynamics chapter.

3.1 Reference Frames

Throughout the simulation, a combination of multiple frames will be employed to portray the state of objects and perform computations. In some cases, certain terms may be more clearly derived and communicated in a particular reference frame, while others may be better represented in a completely different one. It is imperative to possess a comprehensive understanding of these diverse reference frames and the capability to switch between them as required. In this scenario, four distinct reference frames will be used: Earth-Centered Inertial (ECI), Local-Vertical-Local-Horizontal ($LVLH$), Body-fixed (B), and Propulsion (P) reference frames. The purpose and geometry of each one will be introduced in the upcoming sections. In Figure 3.2, a Starlink satellite orbiting Earth, together with the different relevant reference frames are represented for visualisation purposes.

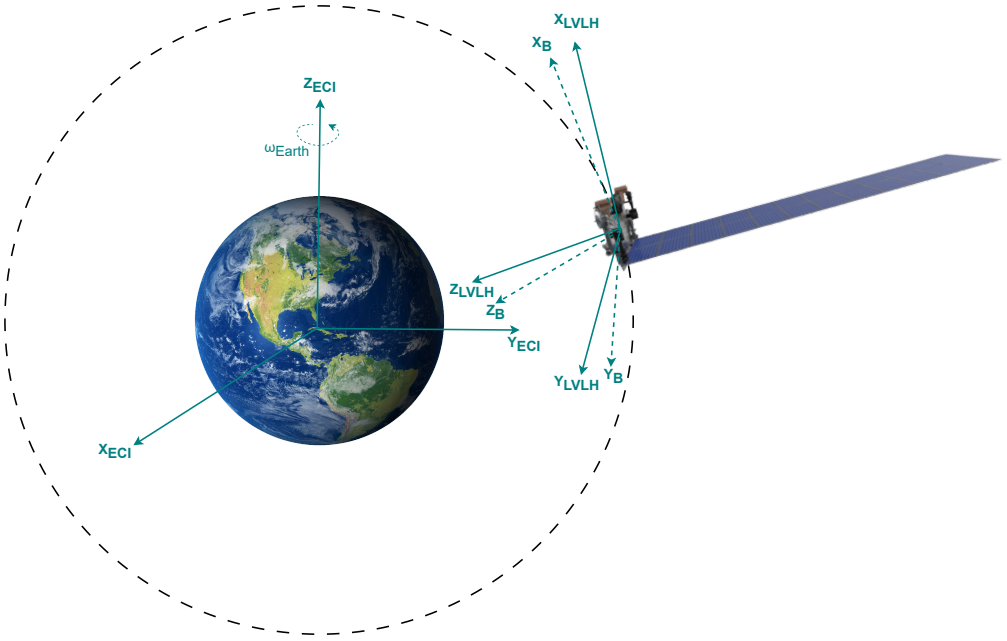

Figure 3.2: [Not to scale] Representation of the ECI, $LVLH$ and B reference frames.

Earth-Centered Inertial Frame

The ECI frame establishes its origin at Earth's centre of mass (com) and remains stationary, providing an excellent inertial reference frame for describing the propagated state of objects orbiting Earth. The frame's orientation is defined as follows: the Z-axis points towards the north pole, the X-axis is orthogonal to the Z-axis aligning with the vernal equinox, and the Y-axis completes the right-handed coordinate system. Throughout all simulations, the J2000 frame will be employed, with the timestamp set at 01/01/2000 at 12:00 terrestrial time. This reference frame simplifies object motion propagation over time, as it is independent of Earth's rotation.

Local-Vertical-Local-Horizontal Frame

The $LVLH$ frame has its origin at the com of the space object with its X-axis pointing along the radial direction, its Z-axis points in the radial direction of orbit, towards the centre of mass of Earth in this case. The remaining axis completes the right-handed coordinate system. This frame is used in this research for verification purposes, to describe the relative distance between two objects or the motion of an object with respect to another.

Body-Fixed Frame

The (B) frame is particularly useful when describing the space vehicle's dynamics for attitude representation, with respect to another reference frame. This rotational reference frame has its origin coinciding with the com of the body. Its X-axis lies in the plane of longitudinal symmetry and is positive in the forward direction, while the Z-axis, also lying in the plane of longitudinal symmetry, is positive in the downward direction. Finally, the Y-axis completes

the right-handed reference frame. In this research, this reference frame is applied to derive the main elements constituting the satellite, including its primary body, solar panel, and instruments. The Primary-Body-Fixed frame will be centred at the com of Starlink or the debris, the Solar-Panel-Fixed frame at the com. of Starlink's solar panel and the Instruments-Fixed frame at the com of the relevant instruments included in the spacecraft.

Propulsion Frame

The P frame is mainly used by the actuators. The X-axis of this reference frame is collinear with the thrust force and is positive in the thrusting direction. This propulsion frame can be described in terms of the body frame using both the elevation angle (ϵ) and azimuth angle (ψ) as will be introduced in Section 3.3.2.

3.2 State Variables

A number of state variables are necessary to describe the state of the space objects. For this simulation, Kepler elements, Cartesian elements, attitude (Euler) angles, quaternions, as well as TLE sequences will be employed.

Kepler Elements

The Kepler elements $[a,\ e,\ i,\ \Omega,\ \omega,\ t_p]$, also known as orbital elements, are widely used to describe the current state of an orbiting vehicle around a central body. They are referred to as the semi-major axis (a), the eccentricity (e), the inclination (i), the longitude of the ascending node (Ω), the argument of perigee (ω) and the time of perigee passage (t_p). Most of these are found in TLE sets. The Kepler elements, are constants and keep the same value for every position in the Keplerian orbit (Wakker, 2015). Even though they are commonly employed, Kepler elements also have limitations in terms of accuracy, particularly when external perturbations are taken into account. Despite these limitations, Kepler elements remain a valuable tool for describing and predicting the behaviour of objects in space and will be used in this work for analysis purposes.

Cartesian Elements

The Cartesian elements $[x,\ y,\ z,\ \dot{x},\ \dot{y},\ \dot{z}]$ also known as rectangular elements, are a way of expressing the position and velocity of a space object, in 3D, with respect to the origin of an inertial rectangular coordinate system. These elements define the state of the spacecraft by expressing its position and velocity as components along the X, Y, and Z axes. This provides a straightforward representation of the equations of motion, making it easier to describe the movement of the spacecraft in space and their ephemeris. Using Cartesian elements offers the advantage of being intuitive and straightforward, particularly for representing linear trajectories. However, they become more complicated when dealing with rotating bodies and complex systems, where Keplerian elements are often preferred. Using Cartesian elements, a space object state can be represented by a state vector as follows:

$$\boldsymbol{x} = (\boldsymbol{r} \quad \dot{\boldsymbol{r}}) \tag{3.1}$$

Attitude (Euler) Angles

Euler angles [ϕ, θ, ψ] also known as attitude angles, are used to represent the orientation of the body reference frame in relation to either the inertial space or the local horizontal frame. Usually, depending on the convention used, rotation about the X-axis is referred to as roll, Y-axis as pitch and Z-axis as yaw. In this research, the attitude Euler angles are used throughout the development of the LQR controller, to express the attitude of the satellite.

Quaternions

Quaternions [q_1, q_2, q_3, q_4], are used to represent the rotation of an object in three dimensions, in an efficient and compact format. Quaternions offer superior data stability compared to Euler angles. By representing rotations in four dimensions, quaternions eliminate singularities, such as gimbal lock, and enable smooth and continuous rotational representations. This makes them highly reliable for applications requiring precise and stable rotational computations. However, quaternions also have some disadvantages. Their four-dimensional nature makes them less intuitive and harder to interpret compared to more familiar three-dimensional rotation representations like Euler angles. Understanding the results of quaternion operations requires a deeper mathematical insight, which can complicate their practical use and understanding in some contexts.

Two-Line Elements

TLEs are a standardised format used to define the trajectory of an Earth-orbiting object. As observed in Figure 3.3, the TLE encompass information on the object's state and its characteristics. These are derived from several tracking techniques such as ground-based stations or radar systems. It is important to note that to have an accurate and up-to-date satellite state, regular updates of the TLE are necessary since they are highly time-dependent. TLE are used in this research work to analyse the conjunction event.

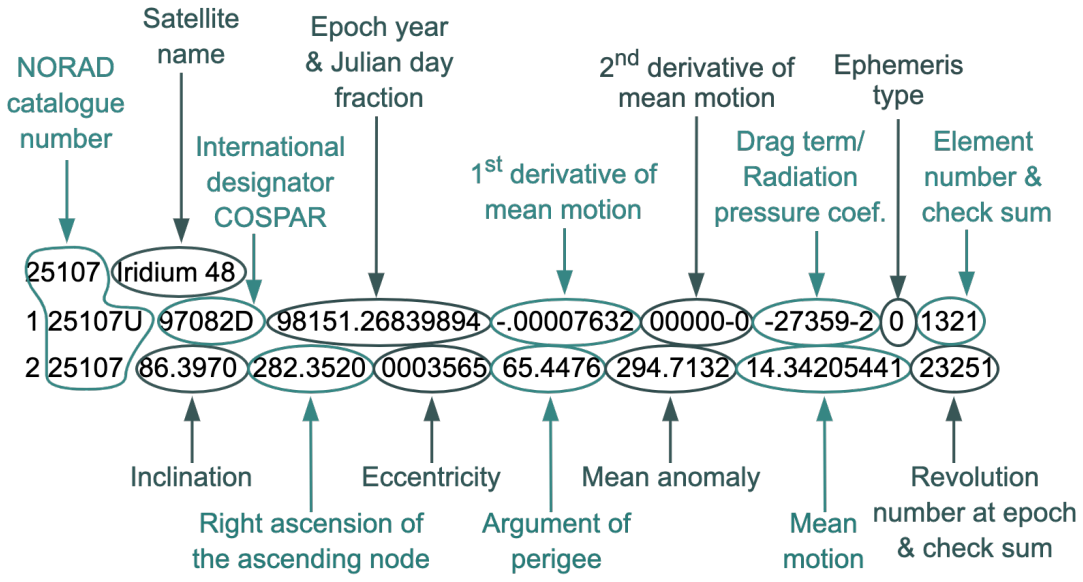

Figure 3.3: TLE sequence information overview.

3.3 Transformations

As observed in Section 3.1 and Section 3.2, some reference frames and state variables are used to describe the state and dynamics of space objects orbiting Earth. Being able to translate the motion of an object described in one reference frame to another frame is an essential skill, this is known as frame transformation. Similarly, the basis of coordinate transformations (Section 3.3.1), frame transformations (Section 3.3.2) and element transformations (Section 3.3.3) is important to master allowing the rotation from any initial frame/coordinate/element to another to be achieved either directly or by combining different transformations principles.

3.3.1 Coordinate Transformations

The transformations from one reference axis to another usually consist of both a translation and a rotation, meaning both a translation vector and a rotation matrix are required to express such transformation. According to Mooij (2020), such transformation can be expressed by means of unit-axis rotations. The unit-axis rotation is a transformation from one right-handed Cartesian frame to another using a positive rotation according to the right-hand rule. Such transformation from reference frame A to reference frame B can be derived as follows:

$$\boldsymbol{x_B} = \begin{pmatrix} x_B \\ y_B \\ z_B \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \begin{pmatrix} x_A \\ y_A \\ z_A \end{pmatrix} \tag{3.2}$$

From this expression, the unit rotation matrices around the X, Y, and Z-axis for a rotation about an arbitrary angle α can be derived.

$$\boldsymbol{C_X}(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \tag{3.3}$$

$$\boldsymbol{C_Y}(\alpha) = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \tag{3.4}$$

$$\boldsymbol{C_Z}(\alpha) = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

Any rotation sequence can be decomposed into a number of sequential axis rotation steps meaning the resulting matrix transformation can be determined as a combination of rotation matrices about a specific axis. The final transformation matrix from reference frame A to reference B contains a total number of three rotations about different axes [-X, -Z, +Y], leading to the following transformation matrix.

$$\boldsymbol{C^{B/A}} = \boldsymbol{C_Y}(\alpha_2)\,\boldsymbol{C_Z}(-\alpha_3)\,\boldsymbol{C_X}(-\alpha_1) \tag{3.6}$$

This expression can be inverted and applied to a transformation from reference B to A if necessary, leading to the following transformation matrix.

$$\boldsymbol{C^{A/B}} = \boldsymbol{C^{B/A}}^{-1} = \boldsymbol{C_X}(\alpha_1)\,\boldsymbol{C_Z}(\alpha_3)\,\boldsymbol{C_Y}(-\alpha_2) \tag{3.7}$$

In terms of Euler angles: The orientation of reference frame B relative to reference frame A can be described through three successive body-axis rotations and be expressed in terms of Euler angles. The sequence order matters and is described as follows, with ϕ, θ, ψ, the Euler angles and A' and A'', the two intermediate reference frames with basis vectors $(a_1{}', a_2{}', a_3{}')^T$ and $(a_1{}'', a_2{}'', a_3{}'')^T$.

$$\boldsymbol{C_Z}(\psi) : A' \leftarrow A \qquad \boldsymbol{C_Y}(\theta) : A'' \leftarrow A' \qquad \boldsymbol{C_X}(\phi) : B \leftarrow A'' \tag{3.8}$$

Using the elementary rotations matrices described in Equation (3.3) to Equation (3.5), combining them to see the effect of rotation in one expression, leads to the following expression of the direction cosine matrix (DCM) of B relative to A in terms of Euler angles $\boldsymbol{C^{B/A}}$.

$$\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \boldsymbol{C_X}(\phi) \begin{pmatrix} a_1{}'' \\ a_2{}'' \\ a_3{}'' \end{pmatrix} = \boldsymbol{C_X}(\phi)\boldsymbol{C_Y}(\theta) \begin{pmatrix} a_1{}' \\ a_2{}' \\ a_3{}' \end{pmatrix} = \boldsymbol{C_X}(\phi)\boldsymbol{C_Y}(\theta)\boldsymbol{C_Z}(\psi) \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \tag{3.9}$$

$$\begin{aligned} \boldsymbol{C^{B/A}} &= \boldsymbol{C_X}(\phi)\boldsymbol{C_Y}(\theta)\boldsymbol{C_Z}(\psi) \\ &= \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \end{aligned} \tag{3.10}$$

In terms of quaternions: Additionally, as stated above, the transformation from one reference frame to another can be expressed in terms of the four quaternion elements. This can be derived by first expressing the quaternions in terms of Euler angles using the following relations, with θ the rotation angle about the Euler axis and $\hat{\boldsymbol{e}} = (e_1, e_2, e_3)$ the unit vector along which the rotation is performed:

$$q_1 = e_1 \sin\left(\frac{\theta}{2}\right) \qquad q_2 = e_2 \sin\left(\frac{\theta}{2}\right) \qquad q_3 = e_3 \sin\left(\frac{\theta}{2}\right) \qquad q_4 = \cos\left(\frac{\theta}{2}\right) \tag{3.11}$$

These relations can then be used to derive an expression for the direction cosine matrix parameterised in terms of quaternions, which is described as follows:

$$\boldsymbol{C^{B/A}} = \boldsymbol{C}(\boldsymbol{q}, q_4) = \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2+q_3q_4) & 2(q_1q_3-q_2q_4) \\ 2(q_2q_1-q_3q_4) & 1-2(q_1^2+q_3^2) & 2(q_2q_3+q_1q_4) \\ 2(q_3q_1+q_2q_4) & 2(q_3q_2-q_1q_4) & 1-2(q_1^2+q_2^2) \end{bmatrix} \tag{3.12}$$

3.3.2 Standard Transformations

Three main frame transformations will be used during simulations; the transformation from $LVLH$ to ECI frame, the transformation from B to ECI and the transformation from P to B. These will be respectively introduced hereafter.

From *LVLH* to *ECI* frame: The transformation from the $LVLH$ to the ECI frame can be expressed using three rotations around the ECI frame axis, with i the orbit inclination, Ω

the right ascension of the ascending node (RAAN) and θ_t the argument of latitude (sum of argument of perigee ω and true anomaly θ):

$$\begin{aligned}\boldsymbol{C_{LVLH/ECI}} &= \boldsymbol{C_Z}(\theta_t)\,\boldsymbol{C_X}(i)\,\boldsymbol{C_Z}(\Omega)\\ &= \begin{bmatrix}\cos\theta_t\cos\Omega-\sin\Omega\sin\theta_t\cos i & \cos\theta_t\sin\Omega+\sin\Omega\cos i\cos\Omega & \sin\theta_t\sin i\\ -\sin\theta_t-\sin\Omega\cos\theta_t\cos i & -\sin\theta_t\sin\Omega+\cos\theta_t\cos i\cos\Omega & \cos\theta_t\sin i\\ \sin i\sin\Omega & -\sin i\cos\Omega & \cos i\end{bmatrix}\end{aligned} \tag{3.13}$$

From *B* to *ECI* frame: The transformation from the B to the ECI frame can be expressed with the following transformation matrix, with ϕ, θ, and ψ the rotation (Euler) angles or in terms of quaternions.

$$\begin{aligned}\boldsymbol{C_{B/ECI}} &= \boldsymbol{C_X}(\phi)\,\boldsymbol{C_Y}(\theta)\,\boldsymbol{C_Z}(\psi)\\ &= \begin{bmatrix}\cos\psi\cos\theta & -\sin\psi\cos\phi+\cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi+\cos_\psi\sin_\theta\cos\phi\\ \sin\psi\cos\theta & \cos\psi\cos\phi+\sin\psi\sin\theta\sin\phi & -\cos\psi\sin\phi+\sin\psi\sin\theta\cos\phi\\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi\end{bmatrix}\end{aligned} \tag{3.14}$$

$$\boldsymbol{C_{B/ECI}} = \boldsymbol{C}(\boldsymbol{q}, q_4) = \begin{bmatrix}1-2\left(q_2^2+q_3^2\right) & 2\left(q_1q_2+q_3q_4\right) & 2\left(q_1q_3-q_2q_4\right)\\ 2\left(q_2q_1-q_3q_4\right) & 1-2\left(q_1^2+q_3^2\right) & 2\left(q_2q_3+q_1q_4\right)\\ 2\left(q_3q_1+q_2q_4\right) & 2\left(q_3q_2-q_1q_4\right) & 1-2\left(q_1^2+q_2^2\right)\end{bmatrix} \tag{3.15}$$

From *P* to *B* frame: The satellite thrust force is always defined in the propulsion frame. To make use of such propulsion force ($\boldsymbol{F_T}$), it should be converted to the local body frame, leading to a thrust force modulus and two direction angles (ϵ and ψ). The transformation to the body frame is represented as follows:

$$\boldsymbol{C_{B/P}} = \boldsymbol{C_Z}(-\psi)\,\boldsymbol{C_Y}(-\epsilon) \quad = \begin{bmatrix}\cos\psi\cos\epsilon & -\sin\psi & \cos\psi\sin\epsilon\\ \sin\psi\cos\epsilon & \cos\psi & \sin\psi\sin\epsilon\\ -\sin\epsilon & 0 & \cos\epsilon\end{bmatrix} \tag{3.16}$$

3.3.3 State Variable Transformations

In addition, to coordinate and reference frame transformations, element transformation is also essential to master when two states are not defined with the same type of variables or for better results interpretation. Some relevant transformations will be introduced hereafter.

Transformation from Cartesian elements to Kepler elements: While Cartesian elements are extensively employed to calculate the equations of motion for the system, the use of Kepler elements is essential for a more meaningful interpretation of the obtained results. Such transformation will be introduced hereafter. To start with, the position (r) and velocity (V) of the space objects are respectively computed as follows:

$$r = \sqrt{x^2+y^2+z^2} \qquad V = \sqrt{\dot{x}^2+\dot{y}^2+\dot{z}^2} \tag{3.17}$$

Following this, the semi-major axis, a, and the flight path angle (γ) can be computed.

$$a = \frac{r}{2-rV^2/\mu} \qquad \sin\gamma = \frac{x\dot{x}+y\dot{y}+z\dot{z}}{rV} \tag{3.18}$$

Following this, the eccentricity (e) can be computed using, in which the eccentric anomaly (E) can be found as follows:

$$e\sin E = \frac{rV\sin\gamma}{\sqrt{\mu a}} \qquad E = \arctan\left[\sqrt{\frac{a}{\mu}}\frac{rV\sin\gamma}{a-r}\right] \tag{3.19}$$

Then, θ and i can be computed, in which $\boldsymbol{h}$, the angular momentum vector, is computed as follows.

$$\tan\frac{\theta}{2} = \sqrt{\frac{1+e}{1-e}}\tan\frac{E}{2} \qquad i = \arccos\left(\frac{h_z}{\|\boldsymbol{h}\|}\right) \qquad \boldsymbol{h} = \boldsymbol{r}\times\boldsymbol{V} \tag{3.20}$$

Following the same logic, the RAAN can be computed.

$$\Omega = \operatorname{atan2}(h_X, -h_Y) \tag{3.21}$$

Transformation from Kepler elements to Cartesian elements: In the event where only Kepler elements are known, one could want to transform them to Cartesian elements for ease of interpretation. As introduced in Section 3.2, Cartesian elements have the advantage of being intuitive and straightforward, particularly for representing linear trajectories. As suggested by Wakker (2015), the position of the celestial object needs to be defined in a Cartesian frame, centred on Earth, with one axis directed towards the perigee of the orbit (ξ), another axis (η) pointing to the semi-latus rectum, and the third axis perpendicular to the orbital plane of the Earth-orbiting body as described below. When θ is unknown, one can use Equation (3.23) to describe such frame, with the matrix elements described as follows:

$$\begin{pmatrix}\xi\\ \eta\end{pmatrix} = \begin{pmatrix}\mathrm{r}\cos\theta\\ \mathrm{r}\sin\theta\end{pmatrix} \tag{3.22}$$

$$\begin{pmatrix}x\\ y\\ z\end{pmatrix} = \begin{bmatrix}l_1 & l_2\\ m_1 & m_2\\ n_1 & n_2\end{bmatrix}\begin{pmatrix}\xi\\ \eta\end{pmatrix} \tag{3.23}$$

$$\begin{aligned} l_1 &= \cos\Omega\cos\omega - \sin\Omega\sin\omega\cos i\\ l_2 &= -\cos\Omega\sin\omega - \sin\Omega\cos\omega\cos i\\ m_1 &= \sin\Omega\cos\omega + \cos\Omega\sin\omega\cos i\\ m_2 &= -\sin\Omega\sin\omega + \cos\Omega\cos\omega\cos i\\ n_1 &= \sin\omega\sin i\\ n_2 &= \cos\omega\sin i \end{aligned} \tag{3.24}$$

The space object's velocity can then be computed, with the angular momentum ($\boldsymbol{h}$):

$$\begin{pmatrix}V_x\\ V_y\\ V_z\end{pmatrix} = \frac{\mu_E}{\boldsymbol{h}}\begin{bmatrix}l_1 & l_2\\ m_1 & m_2\\ n_1 & n_2\end{bmatrix}\begin{pmatrix}-\sin\theta\\ e+\cos\theta\end{pmatrix} \tag{3.25}$$

$$\boldsymbol{h} = \sqrt{\mu_E a\left(1-e^2\right)} \tag{3.26}$$

Transformation between Euler angles and quaternions: During control simulations, both Euler angles and quaternions will be used. The transformation form Euler angles to quaternions, and vice-versa, is an important and widely used one in simulations. On one hand, quaternions offers great numerical stability and introduces stability-free computations. On the other hand, Euler angles representations have the advantage of being intuitive and easy

to interpret. For these reasons and for the ease of results interpretation, it is essential to master conversions from one to the other.

From Euler angles to quaternions:

$$\begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} = \begin{bmatrix} \sin\frac{\phi}{2}\cos\frac{\theta}{2}\cos\frac{\Phi}{2} - \cos\frac{\phi}{2}\sin\frac{\theta}{2}\sin\frac{\Phi}{2} \\ \cos\frac{\phi}{2}\sin\frac{\theta}{2}\cos\frac{\Phi}{2} + \sin\frac{\phi}{2}\cos\frac{\theta}{2}\sin\frac{\Phi}{2} \\ \cos\frac{\phi}{2}\cos\frac{\theta}{2}\sin\frac{\Phi}{2} - \sin\frac{\phi}{2}\sin\frac{\theta}{2}\cos\frac{\Phi}{2} \\ \cos\frac{\phi}{2}\cos\frac{\theta}{2}\cos\frac{\Phi}{2} + \sin\frac{\phi}{2}\sin\frac{\theta}{2}\sin\frac{\Phi}{2} \end{bmatrix} \tag{3.27}$$

From quaternions to Euler angles:

$$\begin{pmatrix} \phi \\ \theta \\ \Phi \end{pmatrix} = \begin{bmatrix} \operatorname{atan2}\left(2\left(q_0 q_1 + q_2 q_3\right), 1 - 2\left(q_1^2 + q_2^2\right)\right) \\ \operatorname{asin}\left(2\left(q_0 q_2 - q_3 q_1\right)\right) \\ \operatorname{atan2}\left(2\left(q_0 q_3 + q_1 q_2\right), 1 - 2\left(q_2^2 + q_3^2\right)\right) \end{bmatrix} \tag{3.28}$$

Transformation from TLE to Kepler elements: When TLEs are accessible, it is frequently necessary to initially convert them to Kepler elements, which are, for most of them, already present as is, within the TLE sequence as observed in Figure 3.3. This conversion allows for direct retrieval of an object's attitude. Its Cartesian position and velocity, using the transformation from Kepler to Cartesian elements described earlier can also directly be computed. This practice is widely employed, particularly in conjunction analysis, for instance. While most Kepler elements are found as is in the TLE sequence, both a and t_p should nevertheless be computed using other elements found in the TLE sequence. The semi-major axis (a) is computed using the mean motion (n), taken from the TLE and converted to rad/s as observed. Then, the time of perigee passage (t_p) can be computed, with M, the mean anomaly found in the TLE sequence, and μ_E Earth's standard gravitational parameter (μ_E = 3.986004418·10^{14} m^3/s^2).

$$a = \left(\frac{\mu_E}{\left(\frac{2\pi}{86400}n\right)^2}\right)^3 \qquad t_p = \frac{M}{\sqrt{\frac{\mu_E}{a^3}}} \tag{3.29}$$

3.4 External Forces and Moments

When travelling in space, spacecraft and debris encounter diverse external forces which disrupt their trajectories and induce changes in the Kepler elements. Neglecting these forces leads to inaccurate orbit propagation, causing simulated trajectories to significantly diverge from actual ones. A simulation of the different forces acting on a Starlink satellite, located at approximately 550 km altitude, can be visualised in Figure 3.4. In the figure, the position difference between a perturbed and unperturbed orbit is displayed. The unperturbed orbit experiences only the primary gravitational field of Earth. To assess the influence of Earth's spherical harmonics (SH) on the trajectory, they are compared with a model having one order and degree lower (e.g., comparing Earth's SH 8/8 with Earth's SH 7/7). The same analysis has been performed on relative velocity and acceleration, with maximum deviations outlined in Table 3.1. Considering the information presented in the diagram, the table, along with the requirements outlined in **[M.ENV–01]**, **[M.ENV–02]**, and **[M.ENV–03]**, the significant perturbations resulting from this mission can be determined. Moreover, CPU

time must be considered **[M.ENV–04]**; the inclusion of numerous forces may substantially increase simulation time, posing challenges for simulations. Therefore, a trade-off between representation accuracy and simulation time needs to be addressed. In this context, the relevant forces and resulting moments to be included in the model are:

- Earth's gravity and spherical harmonics up to degree and order 32,
- Earth's induced gravitational and magnetic torques,
- Moon's point mass (PM),
- Sun's PM,
- Solar radiation pressure and moment, and
- Aerodynamic drag and moment.

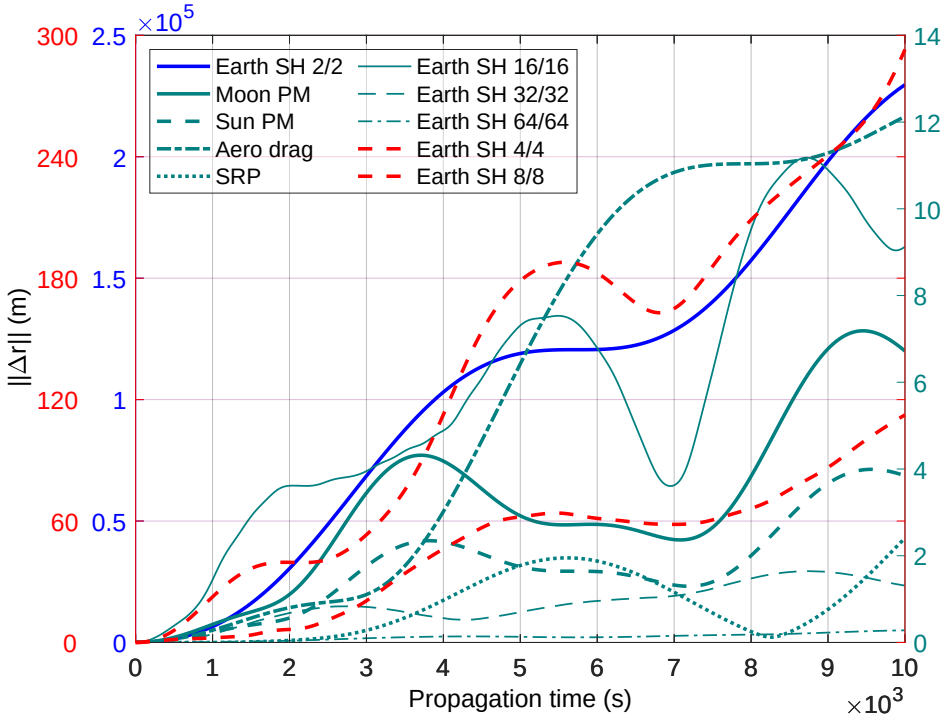

Figure 3.4: Effects of environmental disturbances on a Starlink satellite at an altitude of 550 km, over a period of almost 3 hours.

Table 3.1: Effect of the environment on Starlink's trajectory. Analyse of the relative velocity and acceleration between a perturbed and unperturbed orbit over a period of 3 hours (almost two orbital revolutions).

Perturbations effects on Starlink		
Perturbations	**max $\|\|\Delta V\|\|$ (m/s)**	**max $\|\|\Delta a\|\|$ (m/s^2)**
SH 2/2	$\approx 10^{2}$	$\approx 10^{-1}$
SH 4/4	$\approx 10^{-1}$	$\approx 10^{-4}$
SH 8/8	$\approx 10^{-1}$	$\approx 10^{-4}$
Aero drag	$\approx 10^{-2}$	$\approx 10^{-5}$
SH 16/16	$\approx 10^{-2}$	$\approx 10^{-5}$
Moon PM	$\approx 10^{-3}$	$\approx 10^{-5}$
Sun PM	$\approx 10^{-3}$	$\approx 10^{-6}$
SRP	$\approx 10^{-3}$	$\approx 10^{-6}$
SH 32/32	$\approx 10^{-3}$	$\approx 10^{-6}$
SH 64/64	$\approx 10^{-4}$	$\approx 10^{-7}$

It is important to highlight that a similar analysis has been conducted for the propagation of space debris. The outcome indicates that both objects will progress in identical environments, the same forces and moments will be applied to both scenarios, having different characteristics (geometry, material, etc.). The different relevant atmospheric models that have been used for simulation will be introduced in Section 3.4.1, Earth's environment will be detailed in Section 3.4.2 followed by the third body attractions in Section 3.4.3, the solar radiation pressure in Section 3.4.4 and the propulsion effects in Section 3.4.5.

3.4.1 Atmospheric Model

Earth's atmosphere is made out of a number of particles, which interact with objects orbiting it. Its density decreases the further away from Earth the object is. This means that the further away from Earth, the smaller the atmospheric drag is, and thus the smaller the atmospheric perturbation is on this space object. On the other hand, the closer to Earth, the more drag an object is experiencing. For the mission at hand, with the Starlink satellites being located at an altitude of approximately 550 km **[M.SAT–07]**, which is rather low, the atmospheric drag should be taken into account in the model. Its effect on the space object, as can be observed in Figure 3.4 and Table 3.1, is high. The atmospheric acceleration experienced by a space object can be computed using Equation (3.30) in which ρ is the atmospheric density, $\boldsymbol{v}$ the velocity of the satellite relative to Earth's atmosphere, m is the object's mass and C_D is the drag coefficient related to the vehicle's reference surface (A).

$$\boldsymbol{a_{drag}} = -C_D \frac{1}{2} \rho \frac{A}{m} |\boldsymbol{V}| \boldsymbol{V} \tag{3.30}$$

The resulting torque experienced by each element of the object can be defined using the cross product of momentum as observed in Equation (3.31):

$$\boldsymbol{M_{drag}} = \boldsymbol{r_{cm}} \times \boldsymbol{F_{drag}} \tag{3.31}$$

The air density (ρ) can be derived using a pertinent atmospheric model. Various atmospheric models have been simulated over the years, and one such model is the US standard atmosphere established in 1976. This revised version of US62 provides a framework describing the variations in pressure, temperature, density, and viscosity of Earth's atmosphere with altitude. For enhanced accuracy, the NRLMSISE-00 atmospheric model is employed. This model incorporates satellite accelerometer and mass spectrometer data, as detailed in Wakker (2015). Recognised as one of the most comprehensive models for precisely simulating Earth's complex atmosphere, NRLMSISE-00 surpasses the accuracy of US76, making it the preferred choice for simulations. Moreover, this model has already been developed, and tested through simulations, and is now readily available to use in MATLAB® Simulink.

3.4.2 Earth's Environment

As observed in Figure 3.4, when in LEO, the largest perturbation acting on a satellite is Earth's gravitational force. If it was assumed to be concentrated at its centre, it could be expressed using the following relation:

$$\boldsymbol{a_g} = -\frac{\mu_E}{\|\boldsymbol{r}\|^3} \boldsymbol{r} \tag{3.32}$$

Being not spherical, additional perturbations are introduced due to the changes in its internal mass distribution, resulting in changes in its gravitational potential. According to Wakker (2015), Earth's gravitational field potential, U, at a point outside Earth and the gravitational acceleration can be expressed as follows:

$$U = -\frac{\mu}{r}\left[1 - \sum_{n=2}^{\infty} J_n \left(\frac{R}{r}\right)^n P_n(\sin\phi) + \sum_{n=2}^{\infty}\sum_{m=1}^{n} J_{n,m}\left(\frac{R}{r}\right)^n P_{n,m}(\sin\phi)\left\{\cos m\left(\Lambda - \Lambda_{n,m}\right)\right\}\right] \tag{3.33}$$

$$\boldsymbol{a_{SH}} = \Delta U(\boldsymbol{r_{Earth-object}}) \tag{3.34}$$

This relation is expressed in terms of Earth's spherical harmonics, the body's spherical coordinates r (distance), geocentric latitude (ϕ), and geographic longitude (Λ) and the Legendre polynomials ($P_{n,}, P_{n,m}$) of degree n and order m whose expressions are the following:

$$P_{n,m}(x) = \left(1-x^2\right)^{m/2}\frac{d^m P_n(x)}{dx^m} \tag{3.35}$$

$$P_n(x) = \frac{1}{(-2)^n n!}\frac{d^n}{dx^n}\left(1-x^2\right)^n \tag{3.36}$$

This expression takes into account the perturbations exerted on the orbiting body due to Earth's irregularities (deep oceans, high mountain ranges, lakes), being the direct result of strength variation of the gravitational force on Earth's surface. These anomalies are described using spherical harmonics. As described earlier, and as can be observed in Figure 3.4, the spherical harmonics up to order and degree 32 ($J_{32,32}$) will be selected. The effects of higher terms spherical harmonics are not relevant enough for the selected altitude bands and accuracy requirements, if added they would lead to high computational time.

3.4.3 Third Body Attraction

Besides Earth, other celestial bodies (third bodies) can exert forces on orbiting objects, causing perturbations in their trajectories. As illustrated in Figure 3.4, the Moon and the Sun emerge as the two most influential third bodies in terms of perturbation. The acceleration experienced due to a third body, can be represented using Equation (3.37) from Wakker (2015). In this equation, $\boldsymbol{a_{body}}$ denotes the gravitational attraction acceleration from the perturbing body (Moon or Sun), μ_{body} represents the gravitational parameter of the body, and $\boldsymbol{r}$ represents the distances between the object or Earth and the perturbing body.

$$\boldsymbol{a_{body}} = -\mu_{\text{body}}\left[\frac{\boldsymbol{r_{body-object}}}{\|\boldsymbol{r_{body-object}}\|^3} - \frac{\boldsymbol{r_{Earth-body}}}{\|\boldsymbol{r_{Earth-body}}\|^3}\right] \tag{3.37}$$

3.4.4 Solar Radiation Pressure and Activity

Even if small, solar radiation pressure (SRP) is a relevant disturbance to be taken into account since it can disturb trajectories. In February 2022, a solar geomagnetic storm affected 38 of the 49 newly launched Starlink satellites. This event caused them to deorbit and burn up on their way down using atmosphere thickening and the drag increase (Kataoka, Ryuho et al., 2022). According to Wakker (2015), SRP is the force produced by the solar photons' impacts on the space objects. The SRP force is highly dependent on the object's orientation, shape, and material. A first approximation used to model such acceleration is using the cannonball model for the object. In this model, the object is assumed to be spherical and have constant surface properties. The SRP acceleration can thus be computed using Equation (3.38), with

C_r the reflectivity coefficient (from 0 for an ideal absorber, to 1 for an ideal reflector), A the area of the assumed sphere, m the object's mass, and P_{SRP} the solar radiation pressure at a distance $\boldsymbol{r}$ from the Sun.

$$\boldsymbol{a_{SRP}} = -\frac{P_{SRP} C_r A}{m} \frac{\boldsymbol{r}}{\|\boldsymbol{r}\|} \tag{3.38}$$

3.4.5 Propulsion Force and Moment

To effectively manoeuvre, satellites might make use of propulsion systems to thrust and adjust their position, especially when performing CAMs. These actions generate propulsion forces ($\boldsymbol{F_T}$) and moments ($\boldsymbol{M_T}$), which are termed as follows, with $\boldsymbol{r_T}$ the thruster's location.

$$\boldsymbol{M_T} = \boldsymbol{r_T} \times \boldsymbol{F_T} \tag{3.39}$$

It should be noted that Equation (3.39) is valid for one thruster, in the case where multiple are on board, the summation of each force and moment induced should be taken.

3.5 Internal Forces and Moments

In addition to external forces and moments acting on the satellite and debris, which have been described in Section 2.1, two internal forces and moments, introduced by the variable mass nature of the problem should also be analysed. These internal forces are the Coriolis and relative forces and moments, which are solely related to the mass-variation due to propulsion since a non-elastic vehicle is assumed. The effects due to variable mass and rotational rate are assumed to be small and are ignored. As described by Mooij (1997) the centre of mass flow of most satellites is located on the X-axis; leading to small rotations about the Y and Z-axis and negligible Coriolis forces.

3.6 Equations of Motion

The state of the space objects, which evolve with time, can be described through several EOMs. In this subsection, both the general translational EOM (Section 3.6.1) and rotational EOM (Section 3.6.2) will be introduced, which are respectively, the motion of the centre of mass (com) of the space object, and the motion around their com, introducing the 6 degrees of freedom (DOF) motion definition **[S.SIM–03]**.

3.6.1 Translational Motion

For most simplified simulations, Newton's second law of motion can be applied to the object's circular orbit to describe its motion. In this model, the influence of all other forces and moments is neglected, leading to a first approximation of the object's motion around the central body (Earth). Describing such motion between the object in question and the main body (Earth) in the ECI reference frame leads to Equation (3.40).

$$\ddot{\boldsymbol{r}} = -\frac{\mu_E}{\|\boldsymbol{r}\|^3} \boldsymbol{r} \tag{3.40}$$

To accurately define the total acceleration of the space objects in orbit, all the perturbing forces ($\boldsymbol{F_{pert}}$) that are relevant should be taken into account. Expanding Equation (3.40) for it to reflect on the complexity of the problem at hand, leads to the following expression.

$$\frac{d^2\boldsymbol{r}}{dt^2} = \ddot{\boldsymbol{r}} = \frac{\boldsymbol{F_{pert}}}{m} \tag{3.41}$$

As introduced in Section 3.4, the relevant external forces for the problem at hand are Earth's gravitational force ($\boldsymbol{F_g}$), the aerodynamic drag ($\boldsymbol{F_{drag}}$), the third body attraction of the Moon ($\boldsymbol{F_{Moon}}$) and of the Sun ($\boldsymbol{F_{Sun}}$), and the solar radiation pressure ($\boldsymbol{F_{SRP}}$). In addition to the environmental perturbations, the disturbances induced by the propulsion system ($\boldsymbol{F_T}$) should equally be taken into account.

$$m\frac{d^2\boldsymbol{r}}{dt^2} = \boldsymbol{F_g} + \boldsymbol{F_{drag}} + \boldsymbol{F_{Moon}} + \boldsymbol{F_{Sun}} + \boldsymbol{F_{SRP}} + \boldsymbol{F_T} \tag{3.42}$$

3.6.2 Rotational Motion

The rotational motion of a vehicle encompasses two types of motion: attitude kinematics and attitude dynamics. Both will be introduced hereafter.

Attitude Kinematics

The attitude kinematics represent the rotation of the satellite on itself, the change in body-fixed reference frame with respect to itself. The body attitude is popularly represented using either Euler angles or quaternions. As will be observed at a later stage during the research, although convenient for means of interpretation, the use of Euler angles introduces singularities in the model (at $\theta = \pm 90°$) while quaternions have the advantage of removing those. This comes with the expense of not being easy to interpret. It is for this particular reason that both representations are still used nowadays. Both will be introduced hereafter, with details on the derivation found in Mooij (1997).

On one hand, the definition of the object's attitude with respect to the inertial frame, in terms of Euler angles can be described as follows.

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} \tag{3.43}$$

On the other hand, the definition of the objects (body frame) with respect to the inertial planetocentric frame, in terms of quaternions, can be described as follows.

$$\begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} \tag{3.44}$$

Attitude Dynamics

The vehicle dynamics are influenced by several disturbing moments that have been introduced in Section 2.1. For the problem at hand, some assumptions have been made to allow the vehicle to be defined as a variable-mass, rigid body. It will not have a constant mass throughout the mission, but it will be assumed that the propellant is expelled in such a way that no sloshing or imbalances are experienced. In addition to this, it will be assumed to

be fully rigid, neglecting all induced vibrations. From this, the equation of rotational motion can be described using the so-called Euler equation, according to Mooij (2020) and Wie (1952), with all the external moments ($\boldsymbol{M_{ext}}$) acting on the vehicle expressed along the body axis, the inertia tensor $\boldsymbol{I}$ of the vehicle and $\boldsymbol{\omega}$ the angular velocity about the vehicle's com.

$$\dot{\boldsymbol{\omega}} = \boldsymbol{I}^{-1} \left(\boldsymbol{M_{ext}} - \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} \right) \tag{3.45}$$

With a comprehensive analysis of the surrounding environment now complete, and with the relevant reference frames, transformation principles, variables, and equations of motion thoroughly introduced, the foundation is set for the remainder of the study. The focus will now shift to guidance and control strategies aimed at collision avoidance. To initiate this phase of the research, the principles of conjunction analysis – essential for evaluating potential conjunctions – will be introduced in the forthcoming chapter, Chapter 4. This will provide the necessary framework to effectively address the complexities of collision avoidance in satellite operations.

CHAPTER

4 Conjunction Analysis

A conjunction occurs when two objects pass each other within a pre-determined threshold distance. To initiate a collision avoidance manoeuvre, it is necessary to ascertain the possibility and timing of a conjunction. To find such a potential conjunction, the orbits of the objects involved are extrapolated to assess if the minimum distance between objects falls below a predefined screening threshold. This miss distance between two objects also referred to as the distance of closest approach, is determined by taking the difference between both position vectors over time and finding the minimum distance as well as the moment when this happens.
Performing a comprehensive examination of all object pairs (known as all-vs-all) demands excessive computational resources and is not an efficient or practical choice, especially in time-sensitive situations where swift actions are imperative. This highlights the need for computational-efficient techniques. Various methods, ranging in complexity and efficiency, can be employed for this analysis. While the conjunction analysis is not the main focus of this thesis work, a comprehensive analysis has been conducted to address a real-life scenario and illustrate the complexities of defining such conjunctions in the first place.
The mission at hand actually starts when a potential conjunction has been found and with that, the guidance sequence is triggered to perform a CAM **[M.MIS–02]**. The subsequent sections will outline the different stages of this analysis. Firstly, in Section 4.1, the problem-specific conjunction will be defined together with a discussion on the appropriate miss distance in Section 4.2. Then, Section 4.3 will delve into the specifics of the first-order conjunctions finder, including how the space object catalogue was used, and the processes of retrieval, computation, filtering, and propagation of data. Finally, Section 4.4 will elaborate on how this conjunction information is used and integrated into the simulator for the relevant problem. An overview of the different steps included in a typical conjunction analysis can be found in Figure 4.1.

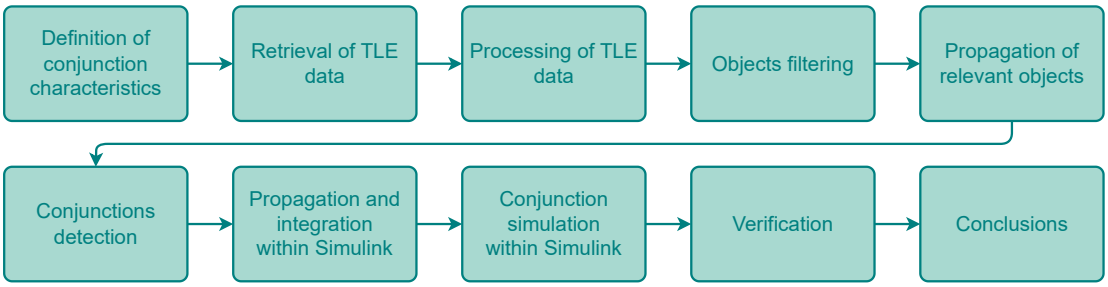

Figure 4.1: Overview of the different steps included in the conjunction assessment.

4.1 Satellite / Debris Conjunction Assessment Screenings

Conjunction occurs when objects approach each other within a certain distance. If the threshold distance is reached, an evasive manoeuvre should be planned. Understanding and identifying such conjunctions are crucial for further study and efficient manoeuvre planning. Initially, it is important to establish the specifics of the conjunction for the mission at hand. The process must be reliable to detect relevant conjunctions while avoiding irrelevant ones. Typically, the probability of conjunction is used to determine whether a manoeuvre is necessary, often in the order of 10^{-5} for manned missions and 10^{-4} for others. Defining what constitutes a 'dangerous' conjunction and the risk linked with a conjunction is beyond the scope of this thesis, for more information on different practices, readers are referred to the work of National Aeronautics and Space Administration (2020). What is important to master for the work at hand is to find any relevant conjunction that will trigger the guidance system and perform an evasive manoeuvre. Multiple criteria can be analysed to find the most accurate potential conjunction (that will be used for further analysis). From these, common used ones are the miss distance, the probability of collision and the covariance matrix (the uncertainty in position and velocity). Since finding the most accurate conjunction analysis is not at the heart of this thesis work, it has been decided to solely look at the miss distance between objects. A discussion on this will be established in the upcoming section.

4.2 Miss Distance Discussion

Establishing a miss distance threshold implies that space missions will only respond to potential conjunctions if the miss distance falls below this threshold. The selection of this threshold requires careful consideration; it must strike a balance between avoiding unnecessary manoeuvres due to excessively large thresholds and preventing catastrophic outcomes by not setting it too small and potentially missing genuine collision risks. In this work, it has been decided to solely look at the miss distance between debris and a Starlink satellite to assess the need for manoeuvre. Conjunction assessment screening involves constructing a volumetric region around the primary object's position and moving this volume along the predicted ephemeris. If any object penetrates this volume, a conjunction is flagged and the need for an evasive manoeuvre is assessed. To perform such analysis, a miss distance threshold has been selected based on the literature review and satellite characteristics.
An analysis conducted in National Aeronautics and Space Administration (2020) suggests that a simplified risk-tolerant conjunction assessment could use a threshold of 2 km. However, it is argued that implementing such a threshold would result in numerous unnecessary mitigation actions, as many events with miss distances below 2 km may not pose significant risks given their low probability of collision. For instance, the number of events with a miss distance of less than 1 km and a probability of collision (P_c) less than 10^{-3} (thus considered non-serious) exceeds those with P_c greater than 10^{-3} by six times. This analysis highlights the need for the threshold not be to excessively large to avoid overlooking potential conjunctions. Equally, it should not be set too low to ensure that no potentially catastrophic conjunctions are overlooked. When analysing the miss distance used for the ISS, the margin is orders of magnitude larger, with a of ±2 (local vertical) x 25 x 25 km (local horizontal) keep-out volume. In addition to this, according to Laporte (2014), for the emergency criteria, up to three days before TCA, the recommended overall miss distance criteria for LEO is 1 km and the radial is 200 m. Additionally, its 95% capture criteria, which comes with a seven days notice sets the overall miss distance to 40 km and radial to 800 m. Considering these stan-

dards and the characteristics of most Starlink satellites (altitude ≈ 550 km, cross-sectional area = 31.36 m^2, velocity ≈ 7.5 km/s, and unmanned), a miss distance threshold of 10 km in all directions has been established **[M.MIS-03]**. On the conservative side, this miss distance threshold has been established as the standard for this analysis. This threshold strikes a balance between the stringent precautions for the manned ISS and in comparison the less strict threshold for LEO satellites, ensuring that no very close conjunctions are overlooked. From the analysis performed by Lorda and Fremeaux (2008), among 75 conjunctions detected, only 23 had a miss distance below 10 km, showing that this threshold is not drastically large and does encompass a fair number of important conjunctions. In addition to this, it is stated in this research that all alerts showing a miss distance lower than 10 km were then further analysed by the flight dynamics team. The 10 km miss distance scenario is illustrated in Figure 4.2, which highlights the keep-out volume to be met. From this scenario, assuming the closest approach of a minimum of 10 km, with objects travelling in LEO at high velocities (≈ 7.5 km/s) leads to a resolution of 1.3 s, meaning the maximum integration step size should be in the same order for the simulation to be accurate. This would require a large CPU time if no data filtering is performed beforehand.

Overall, this analysis suggests that relying solely on simplistic miss distance thresholds derived from theoretical predictions for assessing conjunction risks has several limitations. It is recommended that decision-making should incorporate broader analytical data and factors beyond just miss distance and probability of collision. However, this broader approach lies beyond the scope of this thesis. Although a miss distance of 10 km has been used for this analysis, it has also been set as an input parameter in the simulation, allowing for easy modification at any time. This flexibility enables a more detailed and specific conjunction analysis to be integrated into this research work in the future.

Now that the conjunction assessment and its characteristics have been outlined, the methodology for identifying such conjunctions, including where to obtain the relevant data and the different processing procedures required to get a relevant and not CPU-intensive conjunction analysis will be described.

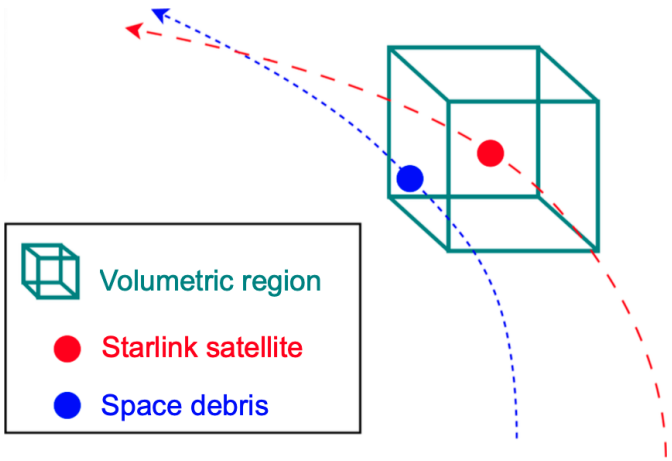

Figure 4.2: [Not to scale] Overview of the conjunction assessment screening in which a volumetric region is used to define potential conjunctions.

4.3 Statistical Conjunction Assessment

The conjunction assessment will purely be based on the miss distance between objects. To compute the relative state of each pair of objects, TLEs are required. These are first retrieved (Section 4.3.1), then analysed and processed (Section 4.3.2). After this, filtering steps are necessary to filter out irrelevant objects and reduce the CPU load (Section 4.3.3). The relevant objects are then propagated and conjunctions are detected (Section 4.3.4).

4.3.1 TLE Retrieval

To start with, the full catalogue of space objects' latest TLE will be downloaded using the ELSET search tool from Space-Track[1]. This will output a .txt file with the last TLEs of all space objects (North American Aerospace Defense Command (NORAD) from 1 to 58930) available on the day of retrieval (09/02/2024). The second step is to retrieve the data of all relevant objects for the analysis. A first filtering can already be done at this stage, to only include the objects that are in orbit (e.g., no decay date). These will be filtered and downloaded using the Query Builder tool from Space-Track[2]. A .csv file will be outputted, including data such as the objects NORAD, name, decay date, launch date, and more. An example of such a file has been displayed in Table 4.1.

INTLDES	NORAD_CAT_ID	OBJECT_TYPE	SATNAME	COUNTRY	LAUNCH	SITE	DECAY	PERIOD	INCLINATION	APOGEE	PERIGEE
2019-074AK	44746	PAYLOAD	STARLINK-1041	US	11/11/19	AFETR		95.59	53.06	548	546
2019-074AL	44747	PAYLOAD	STARLINK-1042	US	11/11/19	AFETR		95.41	53.05	540	537
2019-074AM	44748	PAYLOAD	STARLINK-1043	US	11/11/19	AFETR		95.59	53.05	548	546

COMMENT	COMMENTCODE	RCSVALUE	RCS_SIZE	FILE	LAUNCH_YEAR	LAUNCH_NUM	LAUNCH_PIECE	CURRENT	OBJECT_NAME	OBJECT_ID	OBJECT_NUMBER
			0 LARGE	8695	2019	74	AK	Y	STARLINK-1041	2019-074AK	44746
			0 LARGE	8696	2019	74	AL	Y	STARLINK-1042	2019-074AL	44747
			0 LARGE	8696	2019	74	AM	Y	STARLINK-1043	2019-074AM	44748

Table 4.1: Part of the .csv file retrieved with Space-Track's Query Builder tool.

4.3.2 TLE Processing

With both data files retrieved, the data processing can start. This processing step is crucial to get the information required to find a conjunction. In this specific case, a series of filtering on the type of objects, perigee and apogee will be performed as described in Section 4.3.3. For this reason, both files should be combined using the NORAD as the common information between each data. By doing this, a new .txt file combining the objects' information (non-decayed) and TLE has been created. Using the information in this file, an extra line has been added, one in which the Cartesian position and velocity of each object have been computed using the transformation equations found in Section 3.3.3. The output file of such processing is a .txt file having four lines per object; an example can be visualised in Table 4.2 (line 1 has the object information, lines 2 and 3 are the object's TLE and line 4 contains the object's Cartesian position and velocity). With both files grouped and new elements computed, all necessary data for filtering is now available.

```
2019-074AK|44746|PAYLOAD|STARLINK-1041|US|11/11/19|AFETR| |95.59|53.06|548|546| | |0|LARGE|8695|2019|74|AK|Y|STARLINK-1041|2019-074AK|44746
1 44746U 19074AK  24039.02660835  .00001986  00000-0  15225-3 0  9996
2 44746  53.0541  41.3715 0001754  80.4101 279.7086 15.06393605233710
4790.270251 4947.751179 727.208615 -3.736903 2.732722 6.010849
```

Table 4.2: Overview of the processed .txt file that will be used for further analysis in the conjunction analysis.

[1] Space-Track, *ELSET Search tool*, https://www.space-track.org/#gp [Last visited on 26/02/2024].

[2] Space-Track, *Query Builder tool*, https://www.space-track.org/#queryBuilder [Last visited on 26/02/2024].

4.3.3 Objects Filtering

To decrease processing time, satellite-conjunction prediction tools frequently use techniques aimed at minimising the number of objects involved in the analysis. In this specific case, at the date of retrieval (09/02/2024) the full catalogue of space objects' counted a total of 58,930 potential objects. If no filters would have been applied, the all-vs-all analysis would have lead to a total of 1,731,025,985 combinations to analyse, as suggested by Equation (4.1). This highlights the need for data filtering.

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{4.1}$$

As described above, some pre-filters have already been applied to the current data set, which only includes non-decayed tracked objects and Starlink satellites as one of the two objects involved. Diverse filters are applied to pinpoint orbital combinations that are unlikely to pose a threat within a specified time frame. These identified pairings are subsequently excluded from further analysis, reducing the overall computational process. For this specific case, a simple but efficient apogee-perigee filter, developed by Hoots et al. (1984), has been used to eliminate irrelevant pairings. This filter works on the basis of eliminating pairings that cannot result in conjunctions because there is no overlap in the range of radius values encountered by the two trajectories, the scenario is highlighted in Figure 4.3. In this particular case, there is prior knowledge that one of the two objects follows a circular trajectory at an altitude of approximately 550 km, making this technique straightforward to use. After the filtering process, two distinct datasets emerge: one containing all pertinent Starlink satellites (considered as primary objects), and another dataset consisting of all potential candidates for conjunction. These datasets will be used in the subsequent phase of the conjunction evaluation for further analysis and filtering.

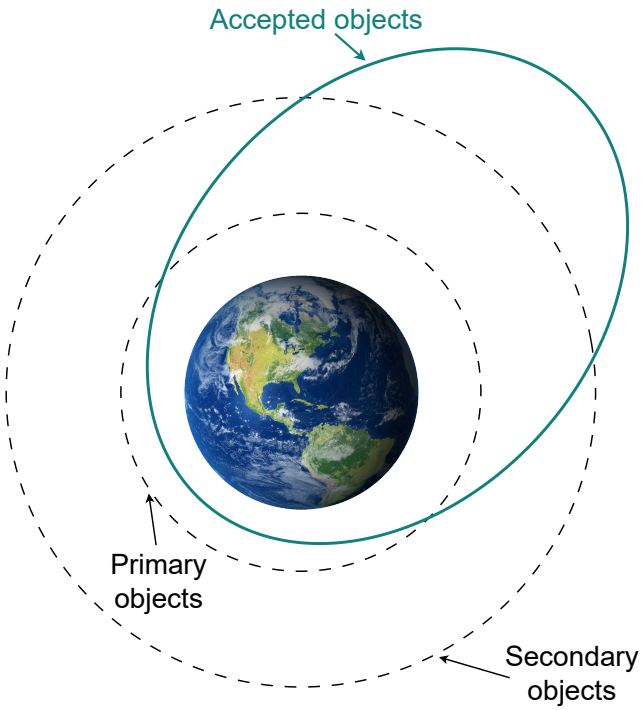

Figure 4.3: [Not to scale] Perigee-apogee filter visualisation.

4.3.4 Objects Propagation and Conjunction Detection

The MATLAB® code conducts object propagation to pinpoint potential close encounters between Starlink satellites and debris orbiting Earth. Initially, TLE data is imported from specified files for both sets of objects. Subsequently, the Simplified General Perturbations 4 model (SGP4) propagator is employed to predict the positions and velocities of each Starlink satellite and debris object at discrete intervals. SGP4, widely recognised for its simplicity and speed, efficiently computes orbital state vectors for near-Earth satellites (with orbital periods of less than 225 min). This model has been developed by Cranford and Lane (1969) and later improved by Hoots et al. (1979). It includes a large panel of orbital data since it is used by the NORAD to track all observable satellites daily. Additionally, it makes use of analytical gravitational and atmospheric models. While it is acknowledged for its efficiency, SGP4 exhibits an average error growth of 1 to 3 km per day according to Vallado et al. (2006). During simulation, each time step systematically assesses the proximity of all satellite-debris pairs. Should the calculated distance dip below a predefined threshold, indicative of a potential conjunction, the code logs pertinent details including timestamps, object names, and calculated distances. These details serve as inputs for subsequent, more detailed Simulink® simulations to accurately assess actual conjunction events. Despite its relatively low accuracy for larger propagation times, SGP4 is initially favoured due to its efficiency in CPU time, providing an initial estimation of potential conjunctions for further analysis steps.

4.4 Conjunction Simulation

The output data gathered from the method outlined in Section 4.3.4 have undergone verification using a more sophisticated simulation model developed within Simulink®. Further details on this simulation model can be found in Chapter 7. During this verification phase, discrepancies between the predictions made by the SGP4 model and the outcomes derived from the Simulink® simulation, which incorporates object geometries and more accurate perturbation models, were observed. Notably, when propagating pairs of objects anticipated to undergo conjunction, significant disparities in miss distance and conjunction timing were noted, sometimes differing by several orders of magnitude from the initial predictions. This disparity underscores the limitations of relying solely on the SGP4 model, as its inaccuracies could lead to unnecessary manoeuvres or, conversely, failure to execute necessary manoeuvres, potentially resulting in catastrophic scenarios. While the SGP4 propagator serves as a rapid means of identifying pairs with potential conjunction events, its lack of precision necessitates further investigation using a more robust system. To further work on the conjunction assessment, it has been decided to use one of the conjunction scenarios and adapt the non-linear simulation in such a way that an accurate conjunction would be found, triggering a manoeuvre. This approach allows for a manoeuvre performance based on a realistic scenario. This scenario has been described in Figure 4.4, outlining a conjunction event on February 9, 2024, at 4h 39min 39s, between the Starlink-4039 satellite (#52683). The conjunction involves the unchanged propagated trajectory, taking the last available TLE before the conjunction and a Falcon 9 piece of debris (#48609) with modified Kepler elements to simulate artificial conjunction on the basis of the conjunction initially found using the SGP4 propagator (Section 4.3.4). During this event, the computed miss distance between the satellite and debris was found to fall below the set 10 km threshold, registering a miss distance of 7,887.06 m, as depicted in Figure 4.5. The miss distance naturally increases with each subsequent orbit. This scenario will further be used in this study to illustrate the CAM using a real-life scenario.

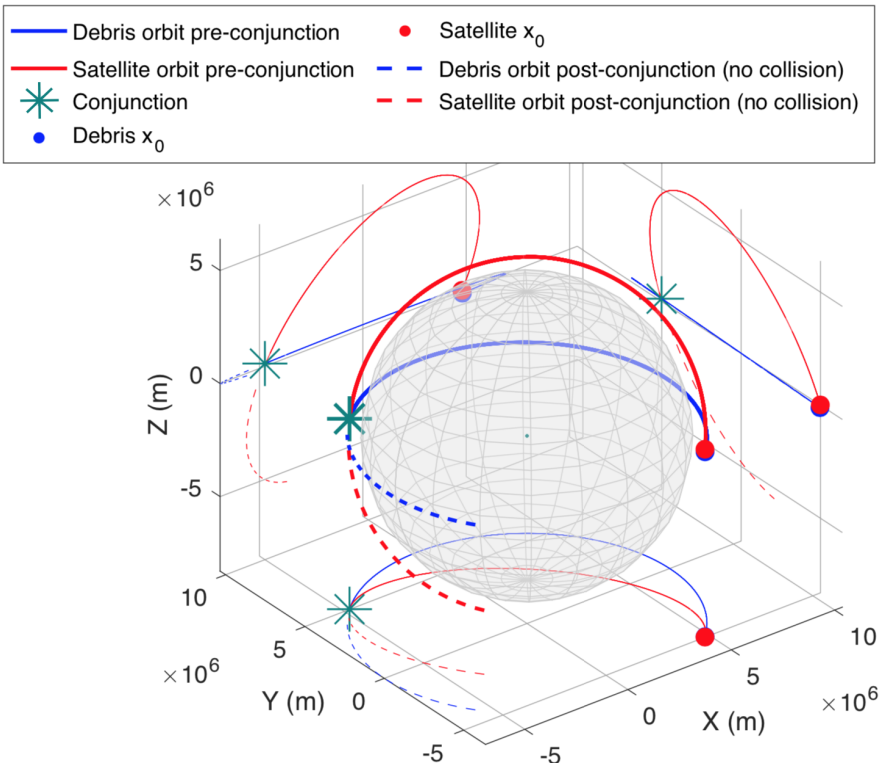

Figure 4.4: Overview of the conjunction scenario (no collision) between Starlink satellite (#52683) and debris. This has been constructed using the TLE of Starlink satellite on February 9, 2024, at 3h 52min 24s. TLE was taken from Seradata[3].

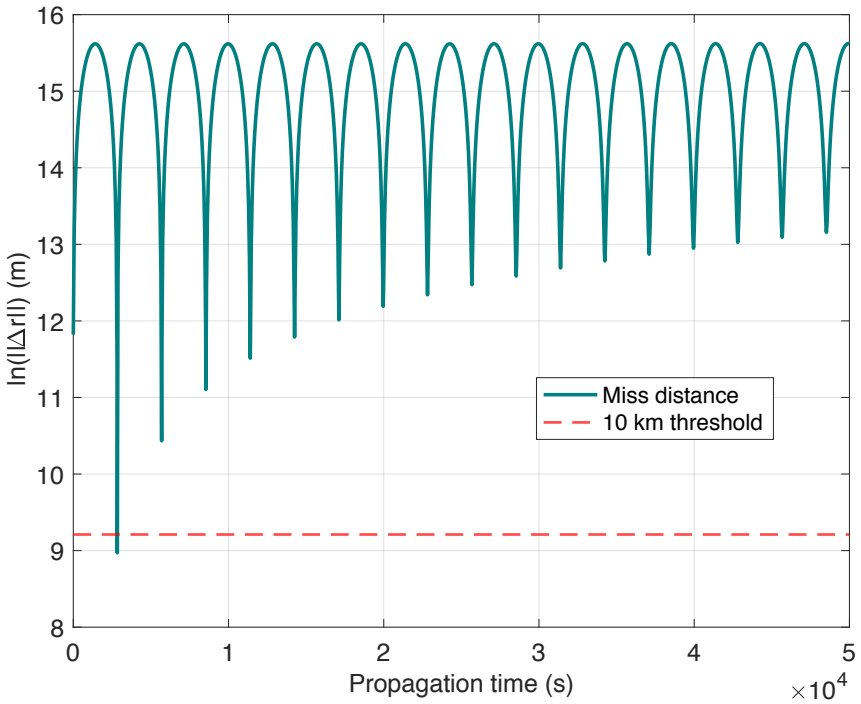

Figure 4.5: Miss distance overview from the outlined conjunction scenario between a Starlink satellite (#52683) and a debris on February 9, 2024. A conjunction was flagged since the miss distance between objects fell below the 10 km threshold.

[3]Space-Track, *ELSET Search*, https://www.space-track.org/#gp [Last visited on 20/03/2024].

Having detailed the key steps and components essential to conjunction analysis in this chapter, the specific conjunction scenario to be addressed is now clearly defined. With this understanding, the next phase of the study will concentrate on developing strategies to avoid the identified conjunction. The next phase of the research will begin by introducing the guidance sequence for CAM, which will be detailed in Chapter 5. This will serve as the foundation for developing effective control strategies to mitigate the risk of collision, ensuring the satellite's trajectory is safely adjusted.

CHAPTER

5 Guidance

To safely and robustly perform collision avoidance manoeuvres and efficiently move the vehicle, an onboard guidance system is required **[S.GUI–04]**. The guidance process refers to the combination of the required size and direction of the ΔV requirement to enable the spacecraft to perform its manoeuvre, meeting its requirements on flight time, precision, fuel consumption and risk casualty. For collision avoidance, a guidance sequence will be developed to manoeuvre in the most efficient manner **[S.GUI–03]**. The satellite should thus be guided to its desired state, following an optimal trajectory, meeting the requirement on miss distance **[S.GUI–01]** but also on propellant consumption **[S.GUI–02]**. A tool for swiftly generating optimal paths in real-time, considering a specified objective function along with dynamics, control, and state constraints, is the convex optimisation method **[S.GUI–05]**. This method has been instrumental in formulating the initial stage of the guidance process (known as nominal guidance), tasked with determining the optimal trajectory path for the given problem. The second part of the guidance is performed using continuous tracking, being one of the analysis objective of the study. For this, an LQR controller is used to ensure accurate execution of the nominal guidance by the satellite **[S.GUI–06]**. This chapter first outlines the objectives of guidance (Section 5.1), which serve as the foundation for the entire guidance system. Next, an overview of the guidance CAM procedure (Section 5.2) is presented, detailing the various guidance segments. Then, a discussion on convex optimisation (Section 5.3) introduces its applications and theory, providing the groundwork for this complex algorithm. Subsequently, the theory and specifics of the nominal guidance sequence are detailed (Section 5.4). Lastly, the tracking guidance sequence, essential for precise execution, is described (Section 5.5). This comprehensive chapter provides insights into the entire guidance sequence, including its theoretical framework and operational steps. An overview of the guidance sequence is depicted in Figure 5.1.

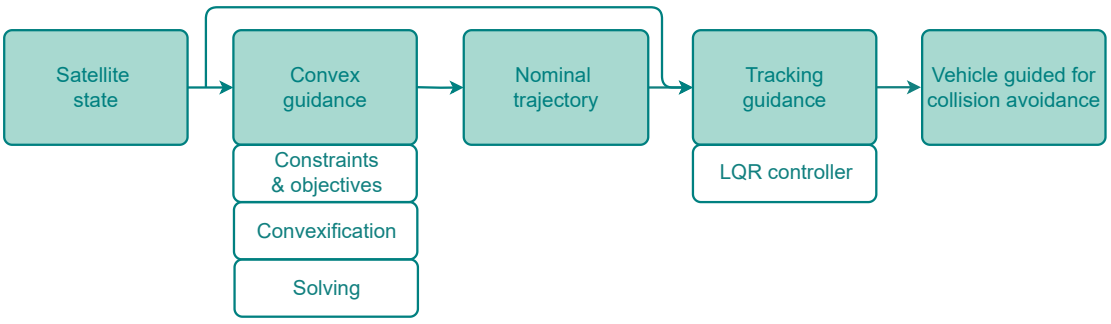

Figure 5.1: Overview of the different steps included in the guidance segment.

5.1 Guidance Objectives

The general aim of the guidance segment is to guide the satellite into performing an optimum manoeuvre. This can be translated into several sub-objectives, making this problem a multi-objective problem. The primary aim of the guidance segment, and the one guiding the entire guidance algorithm, is to minimise thrust throughout the duration of the flight **[S.GUI–02]**. This objective can also be translated into having the smallest Δm, being the most efficient in terms of manoeuvrability, the most time efficient, and the most performance-wise efficient (for operators, performing a manoeuvre is synonyms with losing operational time since during manoeuvre all operations are usually switched-off). From this main objective, a number of sub-objectives can flow, defining the guidance system:

- Fuel-optimal,
- Safety-optimal,
- Time-optimal, and
- Lowest environmental impact.

5.2 CAM Guidance Sequence

Once a potential conjunction is flagged, the satellite's guidance system is immediately turned on and a guidance procedure is to be performed to avoid collision and move the satellite to a lower or higher orbit in which collision is avoided (miss distance between satellite and debris has been increased to an acceptable level). Performing such a guidance manoeuvre is costly for the operating company; both in terms of fuel usage but also in operation resources, since during such manoeuvre, the satellite operation is put to a halt. For these reasons, the manoeuvre sequence needs to be as efficient as possible by moving the satellite to an optimum state without making use of unnecessary resources. It has been decided to use thrust as an indicator, leading to the following objective to be met by the guidance system: adjusting the satellite's trajectory with minimal energy expenditure, and optimising the use of available propellant resources. Commanded actions to manoeuvre the satellite will thus solely be obtained through thrusters available onboard Starlink. The guidance sequence will be divided into two parts; a nominal guidance and a tracking guidance.

Nominal guidance: The primary objective of the nominal guidance is to plan and execute CAMs solely based on predicted and pre-planned trajectories. This is achieved through convex optimisation, which ensures finding globally optimal solutions (optimal trajectory for CAM). This feature makes convex optimisation particularly advantageous for nominal guidance applications. This guidance sequence provides the overall manoeuvre.

Tracking guidance: The tracking guidance follows the nominal guidance and facilitates the continuous tracking of the satellite's position at specific time intervals. The satellite's state is evaluated with respect to the nominal trajectory. The satellite's actual state can deviate from the intended trajectory due to numerous disturbances that cannot all be anticipated in the nominal guidance (developed using simplified, linearised dynamics). Such guidance sequence will be performed using an LQR controller. Sensor feedback is used to detect deviations from the planned trajectory, which may occur due to manoeuvre deviations or disturbances. This sequential guidance process ensures the accurate execution of the nominal guidance by the satellite.

The concept behind using a combination of nominal and tracking guidance has been illustrated in Figure 5.2. The nominal trajectory describes a desired path that would allow the satellite to avoid debris by being moved into a new position outside the keep-out volume while meeting the different constraints imposed on the sequence. However, as previously explained and evident from the figure, the satellite's actual path diverges from this intended trajectory. Hence, continuous tracking guidance is indispensable to monitor such deviations and guide the satellite back onto its desired course.

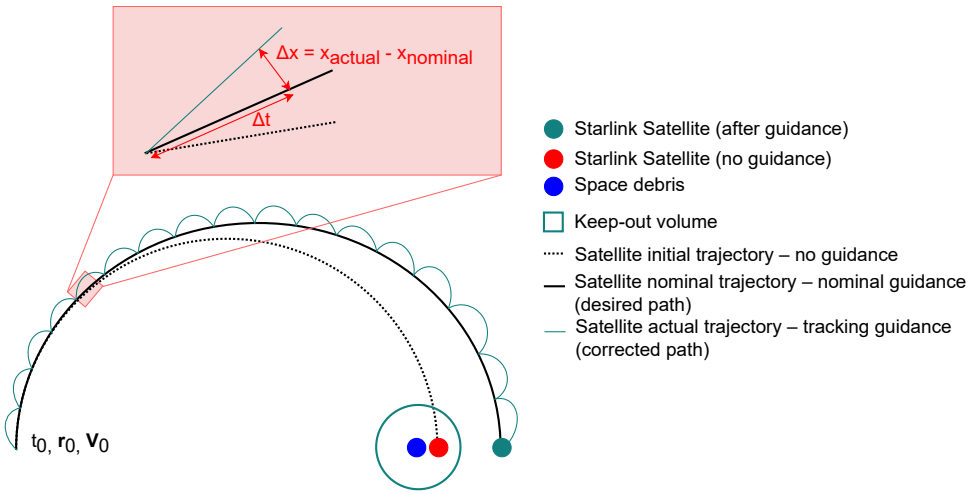

Figure 5.2: Overview of the guidance sequence, including the initial satellite trajectory, the nominal trajectory (convex guidance) and the corrected trajectory (tracking guidance).

Instead of relying on continuous tracking to monitor deviations in a satellite's behavior, an alternative approach involves executing multiple manoeuvres (ΔV) within the duration of half an orbit. Although this alternative has not been explored in the current research, it presents a promising area for future studies.

5.3 Convex Optimisation

Optimisation finds widespread application in the aerospace industry by enabling the discovery of optimal solutions. In guidance systems, optimisation theorems facilitate real-time trajectory optimisation. This is achievable through techniques such as convex optimisation. As highlighted by Malyuta et al. (2021), convex guidance stands out as a highly regarded spacecraft guidance approach due to its advantageous attributes, including robustness, autonomy, and computational efficiency. Moreover, as noted by Virgili-Llop et al. (2018), convex optimisation demonstrates notable convergence properties, particularly in scenarios involving complex path constraints. A significant advantage associated with convex optimisation lies in its deterministic nature: if a feasible solution exists for the problem, the algorithm will converge to the optimal solution within a fixed number of iterations. Notably, a convex function possesses precisely one optimal solution, which coincides with the global solution to the problem. Conversely, the absence of a global solution becomes apparent when no optimal solution other than the global one exists. For the problem at hand, convex optimisation could be used by the guidance system to compute the reference trajectory of the mission vehicle. In this section, the theory of convex optimisation will be elaborated upon. Initially, the formulation of a standard optimisation problem will be introduced in Section 5.3.1. Following this, Section 5.3.2 will provide an overview and general introduction to convex and non-

convex sets and functions. Finally, the convex optimisation methods will be introduced in Section 5.3.3 followed by a general convex optimisation road-map presented in Section 5.3.4. The latter will include the theory behind the solvers and software used to solve for convex guidance, along with an overview of the steps taken for the convex optimisation strategy.

5.3.1 General Optimisation Problem

A general optimisation problem is expressed in a standard format, which consists of an objective function to be minimised, subject to a set of constraints. The latter has been highlighted in Equation (5.1). This general problem definition will be tailored to the specific problem under consideration and will be discussed further in Section 5.4.1.

$$\begin{aligned} &\text{Minimise: } f_0(\boldsymbol{x}) \\ &\text{subject to: } f_i(\boldsymbol{x}) \leq b_i, \quad i = 1, \ldots, m \end{aligned} \tag{5.1}$$

in which f_0 is the objective function (to be minimised), f_i the constraint functions subject to constants b_i and $\boldsymbol{x}$ as the optimisation variable. Convex optimisation problems are characterised by the convexity of both their objective and constraint functions. This means that for any $\boldsymbol{x}$, $\boldsymbol{y} \in \mathbb{R}^n$ and any $\alpha, \beta \in \mathbb{R}$ such that $\alpha + \beta = 1$, $\alpha \geq 0$, and $\beta \geq 0$, the following holds:

$$f_i(\alpha \boldsymbol{x} + \beta \boldsymbol{y}) \leq \alpha f_i(\boldsymbol{x}) + \beta f_i(\boldsymbol{y}) \tag{5.2}$$

5.3.2 Convex and Non-Convex Functions

Every function involved in a convex optimisation problem should be convex. Therefore, it is essential to comprehend the concepts of convex and non-convex functions and understand the strategies used to determine their convexities. A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex if its domain is a convex set, and for all x, y in its domain, and all $\lambda \in [0, 1]$, this inequality holds:

$$f(\lambda x + (1 - \lambda) y) \leq \lambda f(x) + (1 - \lambda) f(y) \tag{5.3}$$

This means that for any two points x and y, the value of f at any convex combination of these points should be no greater than the same convex combination of $f(x)$ and $f(y)$. In geometric terms, as observed in Figure 5.3, the line segment connecting $(x, f(x))$ to $(y, f(y))$ must lie above the graph of f. Additionally, if f is continuous, ensuring convexity is achieved by checking the definition with $\lambda = \frac{1}{2}$ (or any other fixed $\lambda \in (0, 1)$). To finish with, a function f is termed concave if its negative, $-f$, is convex.

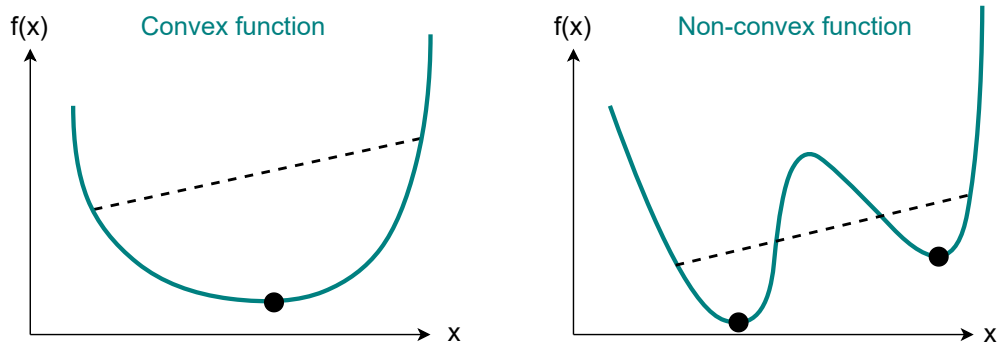

Figure 5.3: Illustration of convex and non-convex functions.

5.3.3 Convex Optimisation Methods

Convex optimisation problems offer a range of solution methods, each presenting its own set of strengths and weaknesses, with further elaboration on these methods addressed in Boyd and Vandenberghe (2004). For the convex optimisation problem at hand, after research in literature and analysis of different previously developed work, it has been decided to define this optimisation problem as a second-order cone programming (SOCP) problem. This has been performed following the work of Açıkmeşe and Carson (2007), who has, and is currently working intensively on similar matters. From previous applications, SOCP problems were found to be efficiently used in space optimisation problems, they have low complexity, are robust, efficient and can be solved in polynomial time. This demonstrates their capacity to address real-world optimisation problems demanding swift convergence.

5.3.4 Convex Optimisation Strategy

As will be observed in Section 5.4, most times, real-life scenarios are not convex and a convexification strategy should first be performed before the optimisation can start. Indeed, a prerequisite to using convex optimisation is ensuring the problem is fully convex. The forthcoming sections will elaborate on the steps involved in convexifying a non-convex problem, an overview of these steps can be visualised in the flowchart presented in Figure 5.4.

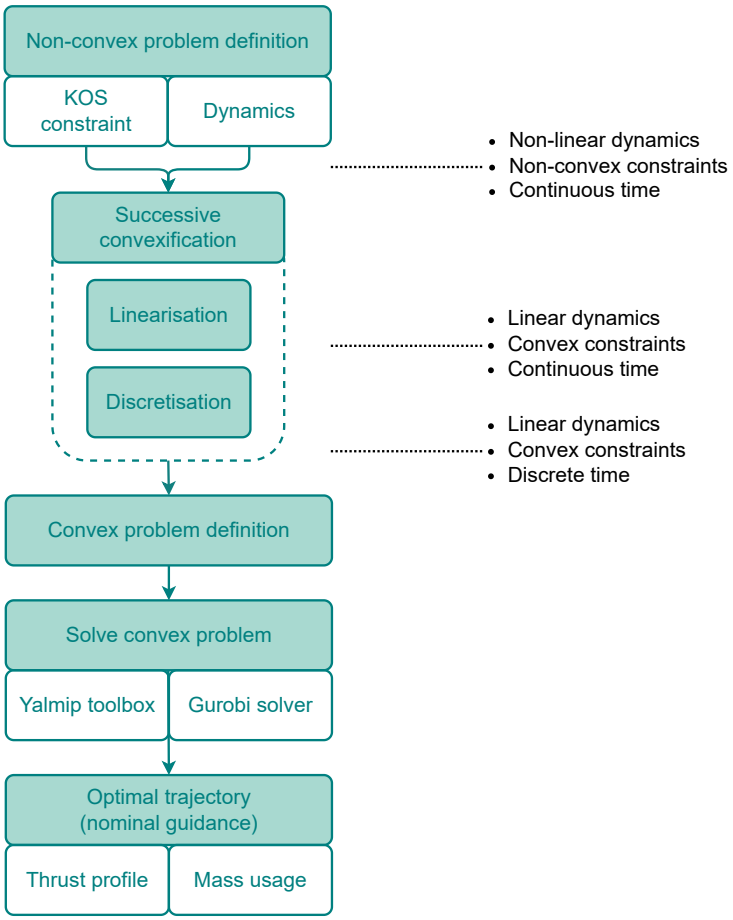

Figure 5.4: Flowchart showing an overview of the steps to be undertaken during the convex optimisation process.

5.4 Nominal Guidance

As outlined in Section 5.2, nominal guidance is employed to plan and execute manoeuvres using pre-determined trajectories and object states. Within this nominal convex guidance procedure, the optimal trajectory is constructed using convex optimisation techniques. Understanding the details of convex optimisation can be challenging, this section will primarily delve into the different steps to follow to create the convex problem. Subsequently, this problem will then be tested to determine if global optimality can be achieved for the current mission characteristics and evaluate whether convex optimisation is indeed a suitable candidate for the nominal guidance process. To start with, the nominal guidance problem definition will be outlined in Section 5.4.1, in which the objective function, satellite dynamics, constraints and boundary conditions (BC) will be introduced. From this definition, the need for convexification will be highlighted as several functions within the problem formulation exhibit non-convex behaviour. For this reason, problem convexification principles will be described in Section 5.4.2, yielding a convexified problem definition. Following this, the problem being fully convex, can finally be solved using a convex solver. An overview of the available solvers will be provided in Section 5.4.3.

5.4.1 Guidance Problem Definition

As discussed in Section 5.2, the primary aim of the guidance segment is to minimise thrust throughout the duration of the flight. Consequently, the optimisation's objective is to determine a trajectory that minimises thrust expenditure over the entire mission duration, while still meeting mission objectives, to avoid collision with debris and move the satellite to meet the miss distance threshold. It is crucial during the optimisation process to ensure adherence to the equations of motion (orbital dynamics). These equations account for external accelerations affecting the spacecraft, namely Earth's gravity and the thrust applied as a control input, as well as the relationship governing mass flow. Aside from the dynamics, as can be concluded from the mission objective, constraints are imposed on the thrust, and position vectors. Initially, limits must be established on the thrust force. This involves setting an upper and lower bound on the allowed nominal thrust, based on the thruster's capabilities to determine the maximum and minimum achievable thrust. Since the guidance sequence is divided into two segments; nominal and tracking, so should be the available thrust for each segment. A percentage of the nominal thrust will be dedicated to nominal guidance while the rest will be available in case of tracking, this percentage is represented using the η symbol. In this particular scenario, thrust is essential to execute responsive control commands, counter external perturbations, and thereby ensuring stability. Additionally, to achieve the CAM objective and adjust the satellite's trajectory to a safe state, it is essential to establish a keep-out sphere (KOS) constraint. This sphere, centered on the debris position, is designed to prevent collisions and ensure that the satellite maintains a safe distance from the debris throughout the simulation. A keep-in sphere has not been included in the problem definition because tests have shown it to be a redundant parameter that does not further constrain the problem. The current algorithm already ensures minimal fuel usage by keeping the satellite just outside the keep-out sphere, making the keep-in sphere unnecessary in this case. Furthermore, initial BC must be imposed on the satellite's position, velocity, and mass and a final BC on the satellite mass to account for the satellite's available propellant mass. This outlines the optimisation challenge to be tackled by the satellite's guidance sequence. An overview of this guidance problem is presented in Table 5.1.

Table 5.1: Definition of the non-convex problem, including the objective function, the constraints and BC it is subjected to.

Non-convex – Continuous – Minimum fuel consumption problem		
Minimise:	$\int_0^{t_f} \|\boldsymbol{F_T}(t)\| \, dt$	*Objective function*
subject to:	$\dot{\boldsymbol{r}}(t) = \boldsymbol{V}(t)$	*Orbital dynamics*
	$\ddot{\boldsymbol{r}}(t) = -\frac{\mu_E}{r^3(t)} \boldsymbol{r}(t) + \frac{\boldsymbol{F_T}(t)}{m(t)}$	
	$\dot{m}(t) = -\frac{\|\boldsymbol{F_T}(t)\|}{g_0 I_{\text{sp}}}$	
	$\eta \; F_{T_{\text{min}}} \leq F_{T_x}^{(-)}, \; F_{T_y}^{(-)}, \; F_{T_z}^{(-)} \leq 0$	*Thrust constraints*
	$0 \leq F_{T_x}^{(+)}, \; F_{T_y}^{(+)}, \; F_{T_z}^{(+)} \leq \eta \; F_{T_{\text{max}}}$	
	$\|\boldsymbol{r}(t) - \boldsymbol{r_{deb}}(t)\| \geq \rho_{\text{KOS}}$	*KOS constraint*
	$m(0) = m_0, \; \boldsymbol{r}(0) = \boldsymbol{r_0}, \; \dot{\boldsymbol{r}}(0) = \dot{\boldsymbol{r}}_{\boldsymbol{0}}$	*Initial BC*
	$m(t_f) \geq m_f$	*Final BC*

Upon examining the above-highlighted problem, it becomes apparent that the only convex functions are the objective function, due to its inclusion of the Euclidean norm of the thrust vector, which is strictly convex, the BC (initial and final), as they are affine constraints and finally the thrust constraints, since their limits include the zero value making them convex constraints. Conversely, the remaining equations are all non-convex. These non-convexities arise from the highly nonlinear dynamics of the satellite and of the concave formulation of the keep-out sphere. To address this challenge and enable the use of convex optimisation techniques, the problem must undergo a process known as convexification. The upcoming Section 5.4.2 will elaborate on the various steps and techniques involved in such a process.

5.4.2 Problem Convexification

The current challenge involves continuous-time nonlinear dynamics and non-convex control constraints. To address these complexities, two approaches are available to transform the problem into a convex optimisation problem: lossless convexification and successive convexification. Each method presents its own set of advantages and disadvantages and may be better suited for certain types of functions. Lossless convexification offers the benefit of introducing minimal or no approximation error, making it particularly advantageous. However, it is not suitable for problems characterised by highly nonlinear dynamics and path constraints. Consequently, successive approximation is employed in conjunction with lossless convexification to efficiently solve optimal control problems featuring nonlinear dynamics and non-convex constraints, which cannot be effectively addressed through lossless convexification alone. These methods will be elaborated upon next.

Lossless Convexification

According to Blackmore et al. (2011), lossless convexification provides a convex relaxation of the non-convex control constraints, along with guaranteeing that the optimal solution to the problem with the convex control constraints is the optimal solution to the problem with non-convex control constraints. This is performed by adding additional variables, known as slack variables. The main disadvantage of this method is that it cannot be applied to highly nonlinear dynamics and can thus not be applied to the satellite's dynamics. For this reason, lossless convexification will not be used for the convexification of this specific problem, instead, another method known as successive convexification will be introduced.

Successive Convexification

Successive convexification is designed primarily to efficiently tackle optimal control problems, which are characterised by nonlinear dynamics and non-convex constraints. In such scenarios, achieving complete convexification into a single convex problem without compromising accuracy is impractical (cannot be achieved through lossless convexification). This approach, first introduced by Mao et al. (2016), operates on the principle of iteratively solving the original problem to optimality by successively linearising non-convex dynamics and constraints around the solution obtained in the previous iteration.

Linearisation: By employing successive convexification and applying principles of linearisation, the nonlinear, non-convex dynamics and KOS constraints can be converted into affine functions, thereby rendering the previously formulated problem entirely convex. It involves iteratively linearising a nonlinear term around a known solution (reference trajectory) obtained in the previous iteration. The dynamics can be linearised around an operating point by approximating the non-linear functions with their first-order Taylor series expansions. This involves computing the Jacobians ($\boldsymbol{A_c}$ and $\boldsymbol{B_c}$) capturing the linearised dynamics and input control effects, respectively. The linearisation principles necessary for this convexification method are described in Appendix A.1.
The results will be highlighted in the linear time-invariant state-space form as defined in Equation (5.4). The matrix $\boldsymbol{A_c}$, known as the state or system matrix, depicts the dynamics of the system, describing the linear connections among its state variables as observed in Equation (5.5). $\boldsymbol{\Delta x}$ is the vector describing the state deviation from the reference point. The $\boldsymbol{B_c}$ matrix, known as the input or control matrix, specifies the influence of control inputs $\boldsymbol{\Delta u}$ on the system's dynamics. Additionally, the state and control vector is defined as follows:

$$\boldsymbol{\Delta \dot{x}} = \boldsymbol{A_c \Delta x} + \boldsymbol{B_c \Delta u} \tag{5.4}$$

$$\boldsymbol{A_c} = \begin{bmatrix} \frac{\partial \dot{x}}{\partial x} & \frac{\partial \dot{x}}{\partial y} & \frac{\partial \dot{x}}{\partial z} & \frac{\partial \dot{x}}{\partial V_x} & \frac{\partial \dot{x}}{\partial V_y} & \frac{\partial \dot{x}}{\partial V_z} & \frac{\partial \dot{x}}{\partial m} \\ \frac{\partial \dot{y}}{\partial x} & \frac{\partial \dot{y}}{\partial y} & \frac{\partial \dot{y}}{\partial z} & \frac{\partial \dot{y}}{\partial V_x} & \frac{\partial \dot{y}}{\partial V_y} & \frac{\partial \dot{y}}{\partial V_z} & \frac{\partial \dot{y}}{\partial m} \\ \frac{\partial \dot{z}}{\partial x} & \frac{\partial \dot{z}}{\partial y} & \frac{\partial \dot{z}}{\partial z} & \frac{\partial \dot{z}}{\partial V_x} & \frac{\partial \dot{z}}{\partial V_y} & \frac{\partial \dot{z}}{\partial V_z} & \frac{\partial \dot{z}}{\partial m} \\ \frac{\partial \dot{V}_x}{\partial x} & \frac{\partial \dot{V}_x}{\partial y} & \frac{\partial \dot{V}_x}{\partial z} & \frac{\partial \dot{V}_x}{\partial V_x} & \frac{\partial \dot{V}_x}{\partial V_y} & \frac{\partial \dot{V}_x}{\partial V_z} & \frac{\partial \dot{V}_x}{\partial m} \\ \frac{\partial \dot{V}_y}{\partial x} & \frac{\partial \dot{V}_y}{\partial y} & \frac{\partial \dot{V}_y}{\partial z} & \frac{\partial \dot{V}_y}{\partial V_x} & \frac{\partial \dot{V}_y}{\partial V_y} & \frac{\partial \dot{V}_y}{\partial V_z} & \frac{\partial \dot{V}_y}{\partial m} \\ \frac{\partial \dot{V}_z}{\partial x} & \frac{\partial \dot{V}_z}{\partial y} & \frac{\partial \dot{V}_z}{\partial z} & \frac{\partial \dot{V}_z}{\partial V_x} & \frac{\partial \dot{V}_z}{\partial V_y} & \frac{\partial \dot{V}_z}{\partial V_z} & \frac{\partial \dot{V}_z}{\partial m} \\ \frac{\partial \dot{m}}{\partial x} & \frac{\partial \dot{m}}{\partial y} & \frac{\partial \dot{m}}{\partial z} & \frac{\partial \dot{m}}{\partial V_x} & \frac{\partial \dot{m}}{\partial V_y} & \frac{\partial \dot{m}}{\partial V_z} & \frac{\partial \dot{m}}{\partial m} \end{bmatrix} \tag{5.5}$$

$$\boldsymbol{B_c} = \begin{bmatrix} \frac{\partial \dot{x}}{\partial F_{T_x}^{(-)}} & \frac{\partial \dot{x}}{\partial F_{T_y}^{(-)}} & \frac{\partial \dot{x}}{\partial F_{T_z}^{(-)}} & \frac{\partial \dot{x}}{\partial F_{T_x}^{(+)}} & \frac{\partial \dot{x}}{\partial F_{T_y}^{(+)}} & \frac{\partial \dot{x}}{\partial F_{T_z}^{(+)}} \\ \frac{\partial \dot{y}}{\partial F_{T_x}^{(-)}} & \frac{\partial \dot{y}}{\partial F_{T_y}^{(-)}} & \frac{\partial \dot{y}}{\partial F_{T_z}^{(-)}} & \frac{\partial \dot{y}}{\partial F_{T_x}^{(+)}} & \frac{\partial \dot{y}}{\partial F_{T_y}^{(+)}} & \frac{\partial \dot{y}}{\partial F_{T_z}^{(+)}} \\ \frac{\partial \dot{z}}{\partial F_{T_x}^{(-)}} & \frac{\partial \dot{z}}{\partial F_{T_y}^{(-)}} & \frac{\partial \dot{z}}{\partial F_{T_z}^{(-)}} & \frac{\partial \dot{z}}{\partial F_{T_x}^{(+)}} & \frac{\partial \dot{z}}{\partial F_{T_y}^{(+)}} & \frac{\partial \dot{z}}{\partial F_{T_z}^{(+)}} \\ \frac{\partial \dot{V}_x}{\partial F_{T_x}^{(-)}} & \frac{\partial \dot{V}_x}{\partial F_{T_y}^{(-)}} & \frac{\partial \dot{V}_x}{\partial F_{T_z}^{(-)}} & \frac{\partial \dot{V}_x}{\partial F_{T_x}^{(+)}} & \frac{\partial \dot{V}_x}{\partial F_{T_y}^{(+)}} & \frac{\partial \dot{V}_x}{\partial F_{T_z}^{(+)}} \\ \frac{\partial \dot{V}_y}{\partial F_{T_x}^{(-)}} & \frac{\partial \dot{V}_y}{\partial F_{T_y}^{(-)}} & \frac{\partial \dot{V}_y}{\partial F_{T_z}^{(-)}} & \frac{\partial \dot{V}_y}{\partial F_{T_x}^{(+)}} & \frac{\partial \dot{V}_y}{\partial F_{T_y}^{(+)}} & \frac{\partial \dot{V}_y}{\partial F_{T_z}^{(+)}} \\ \frac{\partial \dot{V}_z}{\partial F_{T_x}^{(-)}} & \frac{\partial \dot{V}_z}{\partial F_{T_y}^{(-)}} & \frac{\partial \dot{V}_z}{\partial F_{T_z}^{(-)}} & \frac{\partial \dot{V}_z}{\partial F_{T_x}^{(+)}} & \frac{\partial \dot{V}_z}{\partial F_{T_y}^{(+)}} & \frac{\partial \dot{V}_z}{\partial F_{T_z}^{(+)}} \\ \frac{\partial \dot{m}}{\partial F_{T_x}^{(-)}} & \frac{\partial \dot{m}}{\partial F_{T_y}^{(-)}} & \frac{\partial \dot{m}}{\partial F_{T_z}^{(-)}} & \frac{\partial \dot{m}}{\partial F_{T_x}^{(+)}} & \frac{\partial \dot{m}}{\partial F_{T_y}^{(+)}} & \frac{\partial \dot{m}}{\partial F_{T_z}^{(+)}} \end{bmatrix} \tag{5.6}$$

$$\boldsymbol{\Delta x} = \left(\boldsymbol{\Delta r}^T \boldsymbol{\Delta \dot{r}}^T \Delta m\right)^T \in \mathbb{R}^7 \qquad \boldsymbol{\Delta u} = \left(\boldsymbol{\Delta F_T}\right)^T \in \mathbb{R}^3 \tag{5.7}$$

in which these relations apply:

$$x, y, z = r_{x,y,z} \tag{5.8}$$

$$\dot{x}, \dot{y}, \dot{z} = \dot{r}_{x,y,z} = V_{x,y,z} \tag{5.9}$$

$$\ddot{x}, \ddot{y}, \ddot{z} = \ddot{r}_{x,y,z} = \dot{V}_{x,y,z} = -\frac{\mu_E}{\|r\|^3} r_{x,y,z} + \frac{F_{T_{x,y,z}}}{m} \tag{5.10}$$

$$\dot{m} = -\frac{\|\boldsymbol{F_T}\|}{g I_{\text{sp}}} \tag{5.11}$$

Linearisation of the satellite dynamics leads to the following $\boldsymbol{A_c}$ and $\boldsymbol{B_c}$ matrices. In this specific case, only Earth's gravity field and the control thrust are taken into account. Additionally, it is assumed that between every time step for which the matrices are evaluated, the central field gravity remains constant, so no gradient information is required.

$$\boldsymbol{A_c} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -\frac{\mu_E}{\|r\|^3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{\mu_E}{\|r\|^3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{\mu_E}{\|r\|^3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.12}$$

$$\boldsymbol{B_c} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 & 0 & \frac{1}{m} & 0 \\ 0 & 0 & \frac{1}{m} & 0 & 0 & \frac{1}{m} \\ \frac{1}{g_0 I_{\text{sp}}} & \frac{1}{g_0 I_{\text{sp}}} & \frac{1}{g_0 I_{\text{sp}}} & -\frac{1}{g_0 I_{\text{sp}}} & -\frac{1}{g_0 I_{\text{sp}}} & -\frac{1}{g_0 I_{\text{sp}}} \end{bmatrix} \tag{5.13}$$

Similarly, the KOS constraint should also be linearised about the reference trajectory point, as follows, with $\boldsymbol{r_{\text{ref}}}$ the reference trajectory position vector, $\boldsymbol{r_{\text{deb}}}$ the debris position vector and $\boldsymbol{r_{\text{opt}}}$ the optimal trajectory position vector defined by the optimiser.

$$\|\boldsymbol{r_{\text{ref}}} - \boldsymbol{r_{\text{deb}}}\| + \frac{(\boldsymbol{r_{\text{ref}}} - \boldsymbol{r_{\text{deb}}})^\mathsf{T}}{\|\boldsymbol{r_{\text{ref}}} - \boldsymbol{r_{\text{deb}}}\|} \cdot \left(\boldsymbol{r_{\text{opt}}} - \boldsymbol{r_{\text{ref}}}\right) \geq \rho_{\text{KOS}} \tag{5.14}$$

Discretisation: Discretising the problem transforms the convex, continuous free-final-time problem with linearised dynamics and convex constraints into a convex, discrete free-final-time problem. This involves dividing the continuous time interval into discrete time steps, during which the state is iteratively computed. Consequently, a simplified approximation of the solution to the differential equations is obtained at these discrete time points. Various discretisation methods can be used to approximate continuous-time problems into discrete-time ones. Each method has its advantages and drawbacks in terms of computational efficiency and accuracy. Among these methods, the zero-order hold (ZOH) method is commonly used in control theory to convert a continuous-time linear system into an equivalent discrete-time system. This method has the advantage of providing precise and finite solutions but some of the assumptions made to use this method can also introduce discrepancies between the continuous-time solution and the discretised solution. This is because, during ZOH, it is assumed that the input signal remains constant during each sampling interval (Equation (A.3)). This is a good assumption to make, if careful attention is placed on having a small discretisation time step for the problem at hand. The specifics behind the discretisation process have been described in Appendix A.2 with the outcome of such process being the discretised state-space system definition as observed in Equation (A.5). The discretisation process will be carried out in MATLAB® using the continuous to discrete-time function *c2d*[1]. The performance of ZOH has been analysed and compared to the continuous-time system for different discretisation time steps, with output observed in Figure 5.5. It can indeed be observed that the discretisation error drastically reduces when smaller time steps are being used, making it more accurate.

$$\boldsymbol{x_{k+1}} = \boldsymbol{A_d x_k} + \boldsymbol{B_d u_k} \tag{5.15}$$

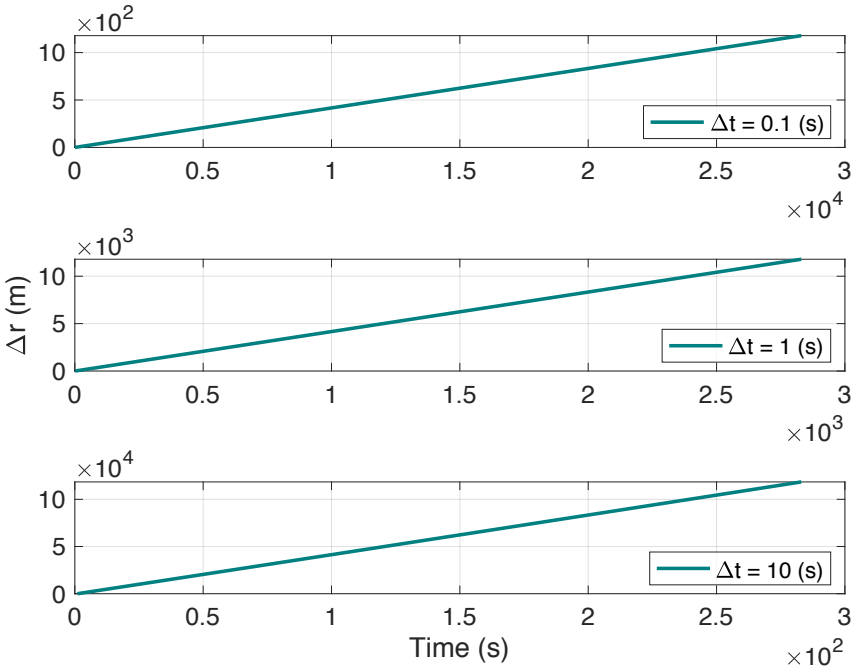

Figure 5.5: Overview of the error introduced by the ZOH discretisation using different discretisation time steps. Comparison of the linearised and discretised satellite reference trajectory (central field only).

[1] Continuous to discrete-time conversion, MATLAB®, https://nl.mathworks.com/help/ident/ref/dynamicsystem.c2d.html [Last visited on 09/04/2024].

Table 5.2: Definition of the convex discrete problem, including the objective function, the constraints and boundary conditions it is subjected to.

Convex – Discrete – Minimum fuel consumption problem		
Minimise:	$\int_0^{t_f} \|\boldsymbol{F_T}(t)\|\, dt$	*Objective function*
subject to:	$\boldsymbol{x_{k+1}} = \boldsymbol{A_d}\,\boldsymbol{x_k} + \boldsymbol{B_d}\,\boldsymbol{u_k}$	*Orbital dynamics*
	$\eta\ F_{T_{\min}} \ \leq\ F_{T_x}^{(-)},\ F_{T_y}^{(-)},\ F_{T_z}^{(-)}\ \leq\ 0$	*Thrust constraints*
	$0\ \leq\ F_{T_x}^{(+)},\ F_{T_y}^{(+)},\ F_{T_z}^{(+)}\ \leq\ \eta\ F_{T_{\max}}$	
	$\left\|\boldsymbol{r_{ref}} - \boldsymbol{r_{deb}}\right\| + \dfrac{\left(\boldsymbol{r_{ref}} - \boldsymbol{r_{deb}}\right)^{\mathsf{T}}}{\left\|\boldsymbol{r_{ref}} - \boldsymbol{r_{deb}}\right\|} \cdot \left(\boldsymbol{r_{opt}} - \boldsymbol{r_{ref}}\right) \geq \rho_{\mathrm{KOS}}$	*KOS constraint*
	$m(0) = m_0, \quad \boldsymbol{r}(0) = \boldsymbol{r_0}, \quad \dot{\boldsymbol{r}}(0) = \dot{\boldsymbol{r}}_{\boldsymbol{0}}$	*Initial BC*
	$m(t_f) \geq m_f$	*Final BC*

Following this entire convexification process, the problem is fully convex and can be written as a proper SOCP problem and be solved. The definition of this SOCP problem is described in Table 5.2.

5.4.3 Convex Optimisation Solvers

Now that the problem is fully convex, it can be tackled using a variety of optimiser algorithms, often referred to as solvers, specifically designed for solving SOCP problems. The selection of a generic solver is confined to those available in academic or open-source domains, which already offer a wide array of options. Notable optimisers applicable in MATLAB include *SDPT3*[2], *Gurobi*[3], *MOSEK*[4], *ECOS*[5], *SCS*[6], *SeDuMi*[7], *IPOPT*[8] and *fmincon*[9].
Although conducting exhaustive testing to determine the optimal solver falls beyond the scope of this thesis, three different solvers will be used and tested against each other for the research work. These solvers are *Gurobi*, *IPOPT*, and *fmincon*. The specifics behind each of these have been introduced in Appendix A.6. Furthermore, in addition to selecting an appropriate solver, it has been decided to make use of a toolbox capable of interfacing with a SOCP solver in MATLAB®. One such tool is YALMIP[10], renowned for its user-friendly interface and extensive support for various solvers. This wrapper facilitates seamless integration with a multitude of SOCP solvers and has been noted for its intuitiveness and ease of use.

[2] SDPT3 Solver, https://blog.nus.edu.sg/mattohkc/softwares/sdpt3/, [Last Visited on 15/03/2024].
[3] Gurobi Optimization, https://www.gurobi.com/, [Last Visited on 15/03/2024].
[4] Mosek ApS, https://www.mosek.com/, [Last Visited on 15/03/2024].
[5] ECOS, GitHub Library, https://github.com/embotech/ecos, [Last Visited on 15/03/2024].
[6] SCS, https://www.cvxgrp.org/scs/, [Last Visited on 15/03/2024].
[7] SeDuMi, GitHub Library, https://github.com/SQLP/SeDuMi, [Last Visited on 15/03/2024].
[8] IPOPT Documentation, GitHub Library, https://coin-or.github.io/IPOPT/, [Last Visited on 15/03/2024].
[9] Fmincon Documentation, MathWorkshttps://www.mathworks.com/help/optim/ug/fmincon.html, [Last Visited on 15/03/2024].
[10] YALMIP MATLAB® toolbox, https://yalmip.github.io/, [Last Visited on 15/03/2024].

5.5 Tracking Guidance

As mentioned in Section 5.2, the guidance sequence has been divided into two types of guidance. The optimal trajectory to be followed for CAMs has been computed using convex optimisation, creating the nominal guidance sequence. This nominal guidance sequence is constructed using an approximation of the actual dynamics that affect the spacecraft (only the central field). As has been highlighted in Section 3.4, other environmental events have great impact on the trajectory. Ignoring these effects in simulations can lead to significant deviations between the calculated ideal trajectory and the actual path of the satellite. Furthermore, discrepancies can also arise due to imperfections in the control mechanism and errors in the navigation system. Consequently, implementing a trajectory-tracking controller can be an option to counteract those. This controller would continuously recalculate the optimised trajectory, ensuring the satellite maintains accurate adherence to the intended course. The tracking guidance works on the principle of comparing the actual trajectory of the satellite (using sensors onboard the satellite) with the nominal guidance computed by the convex guidance algorithm. Any discrepancies between both trajectories exceeding the set threshold should be corrected for, using control actions. This is typically performed using a closed-loop control system in which the satellite trajectory is monitored and adjusted in real-time and at each time step. Trajectory tracking can be computed using several controllers, two of which are LQR or Proportional-Integral-Derivative (PID) controllers. While PID controllers might have a straightforward structure and require fewer computational resources, the required tuning can be cumbersome and challenging when dealing with complex systems. LQR controllers on the other hand show great performance when handling complex system dynamics as is the case for the problem, and is a straightforward controller to tune as will be demonstrated later. For these reasons, it has been decided to make use of an LQR controller to allow the satellite to be maintained in its desired orbit and with this, correct the error between the desired state and the estimated state. In the case of state feedback, for this tracking guidance, the LQR control law (optimal controller) can be defined as Equation (5.16), in which $\boldsymbol{K}$ is the feedback gain matrix, $\boldsymbol{\Delta u}$ the corrective control input and $\boldsymbol{\Delta x}$ the state deviation vector.

$$\boldsymbol{\Delta u} = -\boldsymbol{K}\boldsymbol{\Delta x} \tag{5.16}$$

The LQR control technique takes as its objective to minimise the cost function ($\boldsymbol{J}$) and with this, penalises the deviation from both the desired object's state and the control inputs (thrust magnitude). This objective is defined through the cost function, which is represented as follows, with $\boldsymbol{x^T Q x}$ represents the state deviation and $\boldsymbol{u^T R u}$ the control effort.

$$\boldsymbol{J} = \int_0^\infty (\boldsymbol{x^T Q x} + \boldsymbol{u^T R u})\, d\,t \tag{5.17}$$

$\boldsymbol{Q}$ and $\boldsymbol{R}$ can be given more or less weight depending on the desired performance. On one hand, $\boldsymbol{Q}$, the state cost matrix, will define how much importance is given to the system's specific state; some states should be controlled more tightly since more critical, this translates in a higher weight in the cost function. On the other hand, $\boldsymbol{R}$, the control cost matrix defines the control effort used by each control input. The closed-loop system response is thus dependent on the selection of those parameters. Both $\boldsymbol{Q}$ and $\boldsymbol{R}$ are often defined in an iterative approach but a great initial choice can be selected on the basis of 'Bryson Rule' (Bryson and Ho, 1975). This rule adjusts the variables present in the LQR formulation to ensure that each term's maximum acceptable value is one. Using this rule, both matrices can be represented using Bryson's rule, usually giving good results.

$$\boldsymbol{Q} = \begin{bmatrix} \frac{1}{\Delta x_{\max}^2} & 0 & 0 \\ 0 & \frac{1}{\Delta y_{\max}^2} & 0 \\ 0 & 0 & \frac{1}{\Delta z_{\max}^2} \end{bmatrix} \qquad \boldsymbol{R} = \begin{bmatrix} \frac{1}{\Delta u_x^2} & 0 & 0 \\ 0 & \frac{1}{\Delta u_y^2} & 0 \\ 0 & 0 & \frac{1}{\Delta u_z^2} \end{bmatrix} \tag{5.18}$$

Using the above-defined matrices, the gain matrix $\boldsymbol{K}$ can thus be defined as follows.

$$\boldsymbol{K} = \boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P} \tag{5.19}$$

with $\boldsymbol{P}$ being a positive definite matrix that can be computed by solving the matrix Riccati equation as follows:

$$\boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P} + \boldsymbol{Q} = \boldsymbol{0} \tag{5.20}$$

To solve these equations and get the optimal gains, available control tools in MATLAB® will be used. In this specific example, the *lqr.m*[11] command script will be used for continuous-time models and *dlqr.m*[12] for discrete-time models. Using the continuous-time state-space matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, this tool computes the optimal gain matrix $\boldsymbol{K}$, and the solution of the associated Riccati equation. To properly apply LQR, several assumptions have to be satisfied.

- The matrix $\boldsymbol{Q}$ must be positive semi-definite ($\boldsymbol{Q} \geq 0$),
- The matrix $\boldsymbol{R}$ must be positive definite ($\boldsymbol{R} > 0$),
- The solution $\boldsymbol{P}$ to the algebraic Riccati equation is always symmetric ($\boldsymbol{P}^T = \boldsymbol{P}$), and
- If $\boldsymbol{A}$ and $\boldsymbol{B}$ are stabilisable and detectable, then the correct solution of the algebraic Riccati equation is the unique solution $\boldsymbol{P}$ for which $\boldsymbol{P} \geq 0$.

Using the current controller for the entire trajectory has proven inefficient, since tracking is not constant. The necessity for tracking is highly dependent on the system's dynamics, which change throughout the trajectory, and specific problem characteristics, such as miss distance or available thrust levels. Control systems, while effective, may lack the physical insight into underlying system dynamics and may not always provide optimal control in every situation. To ensure the satellite avoids the keep-out sphere, it must closely follow the nominal trajectory with minimal deviation. However, maintaining such precision across the entire trajectory requires continuous thruster operation, leading to significant mass consumption. To address this issue, it has been proposed to partition the LQR controller into multiple sequences, each tailored to specific segments of the trajectory to better adapt to varying requirements. As shown in Figure 5.6, the satellite's trajectory deviation from the nominal path, when accounting for all perturbations, becomes increasingly significant. This illustrates why stringent deviation constraints are unnecessary from the beginning. Consequently, the tracking guidance will be divided into three segments, each with different maximum allowable position deviation amplitudes, as summarised in Table 5.3, while maintaining the same maximum allowed thrust equal to the defined nominal thrust. This method allows for more adaptable tracking guidance, taking into account the mission's specifics.

[11] Linear-Quadratic Regulator (LQR) design, MATLAB®, https://nl.mathworks.com/help/control/ref/lti.lqr.html [Last visited on 31/03/2024].

[12] Linear-quadratic (LQ) state-feedback regulator for the discrete-time state-space system, MATLAB®, https://nl.mathworks.com/help/control/ref/dlqr.html [Last visited on 31/03/2024].

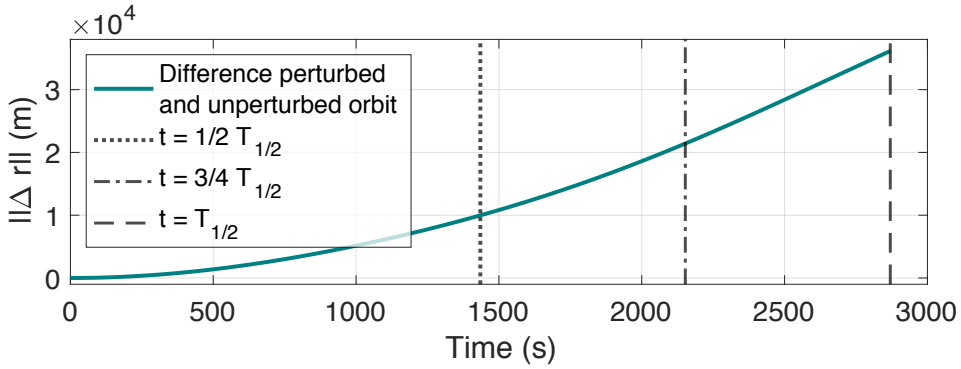

Figure 5.6: Difference in trajectory between an unperturbed nominal trajectory (10 km miss distance) and the actual trajectory of the satellite when all perturbations are turned on and no tracking is performed. [Simulation parameters: RK4, dt = 0.1 s, miss distance = 10 km].

Table 5.3: Minimisation parameters for tracking guidance.

Minimisation parameters per trajectory segment	
Segments (s)	$\Delta x_{max}, \Delta y_{max}, \Delta z_{max}$ **(m)**
$t = 0 - \frac{1}{2}T_{1/2}$	10,000
$t = \frac{1}{2}T_{1/2} - \frac{3}{4}T_{1/2}$	5,000
$t = \frac{3}{4}T_{1/2} - T_{1/2}$	1,000

In this chapter, the complete guidance sequence has been thoroughly introduced, establishing the nominal trajectory for optimal collision avoidance. Alongside this, the principles of convex optimisation and convexification techniques have been discussed, clearly outlining the objectives of the entire guidance sequence. Additionally, the tracking guidance sequence has been analysed and integrated, ensuring a comprehensive understanding of the guidance approach. With the guidance sequence fully introduced and developed, the focus now shifts to the next critical step: controlling the satellite to minimise the deviation between its actual and desired attitude. This aspect of the study will be explored in greater detail in the forthcoming chapter, Chapter 6, where strategies for maintaining accurate control will be discussed.

CHAPTER

6 Control

As introduced in Chapter 5, controlling the satellite's state is essential to minimise the difference between the desired and actual attitude **[S.CON–05]**. This is achieved through a control system that autonomously outputs commands to the actuators, which then respond accordingly. To perform such control analysis, a control system must be selected. Although there are many options available, since the control system is not the main focus of this thesis, a simple yet efficient controller, LQR, has been chosen **[S.CON–06]**. The principle behind LQR, previously introduced for orbit control in Section 5.5, will now be evaluated for attitude control. First, the control objectives will be presented in Section 6.1. Next, the actuators used for both attitude and state control will be discussed in Section 6.2. Finally, the LQR control algorithm that will manage the actuators will be introduced in Section 6.3. An overview of the control sequence can be observed in Figure 6.1.

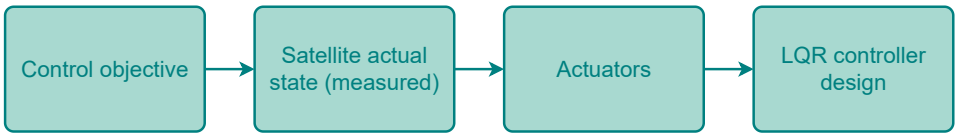

Figure 6.1: Overview of the different steps included in the attitude control sequence.

6.1 Attitude Control Sequence and Objectives

The attitude control sequence uses both the angular rate error and the attitude error as inputs. The attitude error is determined by the difference between the satellite's measured state and the commanded state derived from the guidance commands. Specifically, the guidance sequence provides a thrust force in the P frame, which, using Equation (3.16), can be converted into a thrust magnitude (guidance command) and two attitude angles (ϵ and ψ) in the B frame. These angles define the satellite's attitude and are converted into either Euler angles or quaternions for use as inputs to the attitude control sequence. The primary goal of the control segment is to compute the commands necessary to ensure the satellite's attitude aligns with the desired reference attitude. Specifically, the control system aims at:

- Minimising the error between the reference (desired) and actual (measured) attitude.

To achieve this, a closed-loop control sequence is used, where the controller responds to the reference attitude and incorporates feedback from the system, providing real-time information about the satellite's current attitude. Attaining precise attitude control requires

understanding the input needed to achieve the desired output. This can be accomplished by modelling the system using the state-space representation, as detailed in Section 5.5. The specifics of the control system's state-space representation will be further discussed in Section 6.3. Similar to the successful tracking guidance, the LQR controller will be used for attitude control. This control system is straightforward, widely used, and well-documented, making it an ideal choice for achieving optimal controllability.

6.2 Actuators

The full control system is divided into two types of commands: one for translational motion and the other for rotational motion. Translational motion, which involves executing the guidance commands, is detailed in Section 5.5. Rotational motion, which involves executing attitude control commands, will be discussed further in this chapter. Different actuators are used for each control sequence. The main engine thruster is used for translational motion control of the satellite **[S.CON–01]**(Section 6.2.1), while other attitude actuators are available and typically used for attitude control (Section 6.2.2).

6.2.1 Main Engine Thruster

The main engine thruster was developed optimally, using an ideal thruster model. This model operates on the principle of delivering error-free thrust, which can be adjusted within specified minimum and maximum levels. Upon receiving an on/off command, the generated thrust increases until it reaches saturation, depending on the set nominal thrust. The thruster's position and thrust direction are required for torque computation. Additionally, the thruster's location affects the torque moment, which can be calculated alongside mass flow, as fuel consumption occurs during combustion. This main engine, used for translational manoeuvres, activates when the nominal guidance sequence, initiates the manoeuvre. More information on the thruster's characteristics has been introduced in Section 3.4.5, with the main equation governing such a thruster defined by:

$$\boldsymbol{F_T} = \begin{cases} T_{nom} & \text{for } t_{\text{CAM, on}} \leq t \leq t_{\text{CAM, off}} \\ 0 & \text{otherwise} \end{cases} \tag{6.1}$$

This relation indicates that the thruster's engine operates in a non-throttled manner, working in a binary mode, it provides either full or no thrust at all ('bang-bang'). As will be shown later, this corresponds with the optimal thrust profile determined by the optimisation algorithm and aligns with real-life scenarios. With the sole objective of controlling translational motion, the translational and rotational controls are decoupled. Consequently, this ideal thruster can modify the satellite's state independently of its current attitude.

6.2.2 Attitude Actuators

Attitude control, on the other hand, is typically performed by specific actuators whose main objective is to control the satellite's orientation to match the desired one. Various attitude actuators exist for this purpose, including thrusters, reaction momentum wheels, gyros, and magnetic torques. These actuators primarily take as input a commanded moment required to counteract the desired satellite attitude. This moment is generated by the LQR system, which will be described in Section 6.3. In this study, since the control sequence was not the main focus, no specific attitude actuators have been developed; instead, an ideal actuator has been assumed.

6.3 LQR Controller

As introduced earlier, the need for a control system in such a mission is extremely important. Since the focus of the thesis work is not to create or make use of the most robust controller, it has been decided to make use, just like for the tracking guidance, of an LQR controller. The LQR controller has the advantage of being computationally efficient, making it a great algorithm when real-time response is required. This, however comes at the expense of being slightly less accurate than other methods, such as NDI or INDI controllers. Indeed, since LQR relies on linear approximations of the system dynamics, it may not handle nonlinearities effectively. In contrast, NDI and INDI are designed to address nonlinearities directly, resulting in more accurate and robust control performance for nonlinear systems.

Similarly as has been described in Section 5.5, the LQR controller works on the basis of finding the optimal control gain to solve Equation (5.17) and with this, correct for the error between the desired and actual attitude. To get such gain, several matrices should first be defined, based on the attitude kinematics defined in Section 3.6. These equations are nonlinear and should therefore be linearised to be input in the model for the LQR controller. The LQR controller can be constructed in terms of quaternions or Euler angles, since both versions will be used in a trade-off in the results analysis, their descriptions will be introduced hereafter. The $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices, and their respective state and control inputs, defined in terms of quaternions, are presented from Equation (6.2) to Equation (6.4), and in terms of Euler angles from Equation (6.5) to Equation (6.7), with c, s, t representing the cosine, sine, and tangent functions, respectively.

$$\boldsymbol{A_q} = \begin{bmatrix} 0 & \frac{I_{yy}-I_{zz}}{I_{xx}}\omega_{3_0} & \frac{I_{yy}-I_{zz}}{I_{xx}}\omega_{2_0} & 0 & 0 & 0 & 0 \\ \frac{I_{zz}-I_{xx}}{I_{yy}}\omega_{3_0} & 0 & \frac{I_{zz}-I_{xx}}{I_{yy}}\omega_{1_0} & 0 & 0 & 0 & 0 \\ \frac{I_{xx}-I_{yy}}{I_{zz}}\omega_{2_0} & \frac{I_{xx}-I_{yy}}{I_{zz}}\omega_{1_0} & 0 & 0 & 0 & 0 & 0 \\ \frac{q_{4_0}}{2} & \frac{-q_{3_0}}{2} & \frac{q_{2_0}}{2} & 0 & \frac{\omega_{3_0}}{2} & \frac{-\omega_{2_0}}{2} & \frac{\omega_{1_0}}{2} \\ \frac{q_{3_0}}{2} & \frac{q_{4_0}}{2} & \frac{-q_{1_0}}{2} & \frac{-\omega_{3_0}}{2} & 0 & \frac{\omega_{1_0}}{2} & \frac{\omega_{2_0}}{2} \\ \frac{-q_{2_0}}{2} & \frac{q_{1_0}}{2} & \frac{q_{4_0}}{2} & \frac{\omega_{2_0}}{2} & \frac{-\omega_{1_0}}{2} & 0 & \frac{\omega_{3_0}}{2} \\ \frac{-q_{1_0}}{2} & \frac{-q_{2_0}}{2} & \frac{-q_{3_0}}{2} & \frac{-\omega_{1_0}}{2} & \frac{-\omega_{2_0}}{2} & \frac{-\omega_{3_0}}{2} & 0 \end{bmatrix} \tag{6.2}$$

$$\boldsymbol{B_q} = \begin{bmatrix} \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{6.3}$$

$$\Delta \boldsymbol{x} = \begin{pmatrix} \Delta\boldsymbol{\omega} & \Delta\boldsymbol{q} \end{pmatrix}^T \in \mathbb{R}^7 \qquad \Delta \boldsymbol{u} = \begin{pmatrix} \Delta \boldsymbol{M_c} \end{pmatrix}^T \in \mathbb{R}^3 \tag{6.4}$$

Only three of the four quaternions can physically be controlled. For this reason, q_4 is removed from the controller design, leading to a 6x6 $\boldsymbol{A_q}$ matrix and 6x3 $\boldsymbol{B_q}$ matrix.

$$\boldsymbol{A_{Euler}} = \begin{bmatrix} 0 & \frac{I_{yy}-I_{zz}}{I_{xx}}\omega_{3_0} & \frac{I_{yy}-I_{zz}}{I_{xx}}\omega_{2_0} & 0 & 0 & 0 \\ \frac{I_{zz}-I_{xx}}{I_{yy}}\omega_{3_0} & 0 & \frac{I_{zz}-I_{xx}}{I_{yy}}\omega_{1_0} & 0 & 0 & 0 \\ \frac{I_{xx}-I_{yy}}{I_{zz}}\omega_{2_0} & \frac{I_{xx}-I_{yy}}{I_{zz}}\omega_{1_0} & 0 & 0 & 0 & 0 \\ 1 & s\phi t\theta & c\phi t\theta & c\phi t\theta\omega_{2_0} - s\phi t\theta\omega_{3_0} & \frac{s\phi\omega_{2_0}+s\phi\omega_{3_0}}{c\theta^2} & 0 \\ 0 & c\phi & -s\phi & -s\phi\omega_{2_0} - c\phi\omega_{3_0} & 0 & 0 \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} & \frac{c\phi\omega_{2_0}-s\phi\omega_{3_0}}{c\theta} & \frac{s\phi\omega_{2_0}+c\phi\omega_{3_0}}{c\theta} & 0 \end{bmatrix} \tag{6.5}$$

$$\boldsymbol{B_{Euler}} = \begin{bmatrix} \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{6.6}$$

$$\Delta\boldsymbol{x} = \begin{pmatrix} \Delta\boldsymbol{\omega} & \Delta\phi & \Delta\theta & \Delta\psi \end{pmatrix}^T \in \mathbb{R}^7 \qquad \Delta\boldsymbol{u} = \begin{pmatrix} \Delta\boldsymbol{M_c} \end{pmatrix}^T \in \mathbb{R}^3 \tag{6.7}$$

Following this, both the $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices should be defined depending on the mission specifics and the weights they should be given. As for the tracking guidance control, Bryson's rule will be used, giving a good first choice in defining both matrices. The $\boldsymbol{Q}$ matrix will differ for the quaternion Equation (6.8) and for the Euler angles Equation (6.9) representations, while the $\boldsymbol{R}$ is similar for both cases as observed in Equation (6.10).

$$\boldsymbol{Q_q} = \text{diag}\left(\frac{1}{\Delta\omega_{1,\max}^2}, \frac{1}{\Delta\omega_{2,\max}^2}, \frac{1}{\Delta\omega_{3,\max}^2}, \frac{1}{\Delta q_{1,\max}^2}, \frac{1}{\Delta q_{2,\max}^2}, \frac{1}{\Delta q_{3,\max}^2}\right) \tag{6.8}$$

$$\boldsymbol{Q_{Euler}} = \text{diag}\left(\frac{1}{\Delta\omega_{1,\max}^2}, \frac{1}{\Delta\omega_{2,\max}^2}, \frac{1}{\Delta\omega_{3,\max}^2}, \frac{1}{\Delta\phi_{\max}^2}, \frac{1}{\Delta\theta_{\max}^2}, \frac{1}{\Delta\psi_{\max}^2}\right) \tag{6.9}$$

$$\boldsymbol{R} = \text{diag}\left(\frac{1}{\Delta u_{\text{x}}^2}, \frac{1}{\Delta u_{\text{y}}^2}, \frac{1}{\Delta u_{\text{z}}^2}\right) \tag{6.10}$$

Having established the specifics of the control sequence and detailed the workings of the LQR controller, the theoretical foundation of this research is now complete. The focus will now shift to the practical implementation of these concepts, where the integration and development of all previously discussed elements will be showcased. This transition marks the move from theory to application, as the next chapter will delve into the design and development of the simulation environment, where these guidance and control strategies are brought to life. The comprehensive simulator design process will be explored in Chapter 7.

CHAPTER

7 Simulator Design

This chapter will consolidate all previously introduced elements, including the complete simulator and its constituent parts, as well as the comprehensive verification process for each component. The primary objective of the research is to design an on-board autonomous software capable of executing CAMs. The simulator integrates the environment, mathematical models, and all components required for a complete and autonomous system. Detailed descriptions of both the entire simulator and its individual components will be provided in Section 7.1. Additionally, to ensure the reliability and functionality of all developed elements, their verification will be discussed in Section 7.2, resulting in a fully verified CAM algorithm.

7.1 Simulator Architecture

This entire research work has been constructed using MATLAB/Simulink®, with which the entire simulation model has been developed. This programming environment encompasses a large number of sub-libraries, which have been regrouped into a large simulator. To better understand how the simulations work and how each part of the algorithm relates to each other, a simulator architecture has been constructed and will be introduced in this section. To start with, a visual overview of the architecture will be introduced in Section 7.1.1, enabling the reader to visualise the latter as well as the dependencies of each developed blocks. This will be followed in Section 7.1.2, by an explanation of what the Generic Rendezvous and Docking Simulator (GRADS) library is. This library has been an essential tool in the conception of the simulator. Following this, each part of the simulator will further be described, giving more context and information to the reader. The specifics behind the input elements will be introduced in Section 7.1.3, followed by the dynamics simulator characteristics in Section 7.1.4, the guidance sequence in Section 7.1.5, the control sequence in Section 7.1.6, the output elements' specifics in Section 7.1.7 and to finish with an explanation of the sampling time used for simulation in Section 7.1.8.

7.1.1 Architecture Overview

An overview of the entire simulator architecture that has been used for every simulation can be visualised in Figure 7.1. The relation between each element can be noted from such a graph, with more details on each element to come.

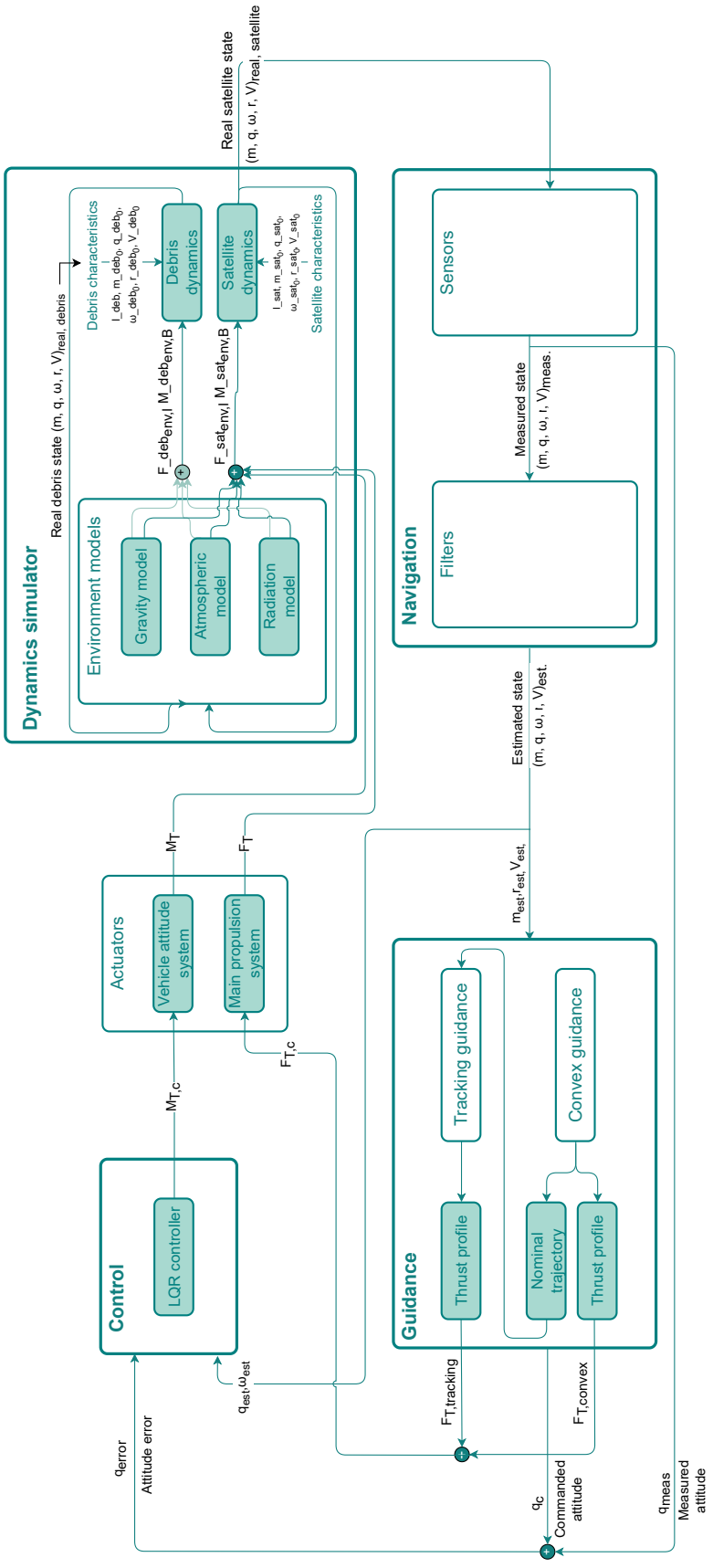

Figure 7.1: Simulator architecture overview.

7.1.2 GRADS Library

During the construction of the simulation environment, several pre-existing blocks from the GRADS library (Mooij, 2021) were integrated. Specifically, these blocks originated from the already established tool sets of ESA SpaceLAB and GAST, which were then refined based on their prior verification. It has been developed with the idea of creating a user-friendly interface having the minimum dependencies on other libraries, making it unique and easy to use. For the problem at hand, the GRADS library will be expanded to accommodate for the creation or expansion of other toolboxes, necessary for the current simulation.

7.1.3 Input Elements

Several elements must be input into the simulation algorithm for it to function correctly. These elements will be introduced in the following sections.

Objects characteristics: To properly define the initial value problem, the characteristics of various objects (satellites and debris) must be defined **[S.SIM–02]**. These characteristics include the object's initial mass, geometry, and state. From these entries, other elements such as the objects' inertial tensors can be derived. Additionally, the objects' drag and illumination coefficients are also required as part of the input data.

Time: The entire simulation is time-based. Time (t) will dictate the simulation and will be used throughout the study to evaluate changes in decision variables and Kepler elements, both of which are linked to time. Additionally, time plays a pivotal role in determining the positions of objects (extracted from their TLE, which is time-dependent) and influencing environmental factors. Various atmospheric, radiation, and gravitational effects depend on the time of day or the period during which the simulation occurs. Therefore, using time as a reference is indispensable in the simulation.

Constants: Certain constants are necessary for initialising the various models used in the simulation. These include the radii and masses of the Earth, Sun, and Moon, as well as the Earth's gravitational parameter and angular rate, among others.

7.1.4 Dynamics Simulator

The dynamics simulator includes both the environmental model and the on-board propagator. The environmental model accounts for the critical forces and moments essential for realistic propagation simulation. The detailed development of these models, which cover radiation, gravity, and atmospheric conditions, is extensively explained in Section 2.1. The on-board propagator, selected based on Cowell's method, describes motion in an inertial frame. This method handles the propagation of a variable-mass body in six DOF.

7.1.5 Guidance Sequence

The main purpose of the guidance sequence is to guide the motion of the vehicle. In this specific case, it does so through two different guidance sequences: nominal guidance and tracking guidance. In the nominal guidance sequence, a nominal trajectory to be followed is defined using mathematical tools known as convex optimisation. Through this optimisation sequence a nominal, ideal, path to be followed is created. On a second level, the tracking

guidance sequence is constantly tracking the vehicle making sure it follows the nominal trajectory to an acceptable level. The output (control force) of this sequence serves as direct input to the control sequence and actuators.

7.1.6 Control Sequence

The control sequence determines, in this case using an LQR controller, the control commands necessary to control the vehicle's attitude motion. The output of such control sequence is a control moment which is, together with the control force (the control commands), inputted into the actuator block. The actuators, in turn, convert those commands to actions (forces and moments), guiding the satellite to the required state and attitude.

7.1.7 Output Elements

From the simulation, a large number of elements can be output, many of which are used in both the verification and results sequences. These output parameters include the states of satellite and debris, mass and thrust profiles, nominal trajectories, commands for thrust forces and moments, simulation performance metrics, and numerous others.

7.1.8 Sample Time

The simulation model comprises multiple systems with varying sampling rates, necessitating specific considerations for time steps within the simulator. These subsystems are known as multi-rate models. In this scenario, the convex guidance optimisation requires a sample time step of 10 s, being excessively large for the control and tracking guidance sequence propagations. Therefore, it has been determined necessary to employ the ZOH function[1], enabling the use of different sample times throughout the simulation.

7.2 Verification

Verifying any developed and integrated elements is a crucial step in the creation of any robust algorithm. This process determines whether any system meets its requirements and all the pre-determined specifications. This is usually performed using several tests on the systems. In this section, the verification of the entire algorithm will be presented solely for the elements that have not yet been tested for. As has been introduced earlier, many blocks have been taken from the GRADS library, and already verified. The remaining verification process will be divided into different tests, for different parts of the system. To start with, the developed Simulink® model will be verified in Section 7.2.1, then the complex developed and integrated environment will undergo verification in Section 7.2.2. Following this, the testing of the guidance and control sequences will be introduced in Section 7.2.3 and Section 7.2.4, respectively. It must be noted that for the verification part of this work, unless stated otherwise, a circular and equatorial orbit has been used for every simulation and not the actual Starlink propagated orbit, which will itself be used for all results and remaining plots in this work. This choice has been made for the simplicity of this known orbit since it provides stable and predictable results, which is ideal for the verification and understanding of the system's behaviour.

[1] MathWorks, *Zero-Order Hold*, https://www.mathworks.com/help/simulink/slref/zeroorderhold.html, [Last visited on 17/05/2024].

7.2.1 Simulink® Model

Although the majority of models employed in building the Simulink® model are sourced from the GRADS library and have undergone prior verification, it remains crucial to test the integration of these blocks to confirm the model's behaviour. Moreover, since several inputs are taken from external sources, it is imperative to verify the integration of all these components. One efficient and complete test on a circular behaviour will be executed to assess the architecture and output behaviour of such a system. In scenarios where only the gravitational influence of the Earth's central field is taken into consideration, objects in circular orbits are anticipated to sustain their trajectories. To verify this prediction, it is customary to analyse the variations in the object's Kepler elements over time. Additionally, the velocity modulus is expected to remain constant throughout the propagation process. A verification test was executed for a satellite in LEO with initial orbital elements and velocity depicted in Table 7.1, for a propagation time of 10,000 s. The results obtained confirm the expected behaviour; the object exhibited the anticipated changes in its Kepler elements, and the velocity modulus remained stable as predicted. However, it is worth noting that some minor deviations were introduced; these are numerical errors.

Table 7.1: Overview of the satellite's behaviour over a 10,000 s period (central field).

Circular orbit Kepler elements evolution			
Kepler elements	**Initial**	**Final**	**Δ**
a (m)	$6{,}917{\cdot}10^{6}$	$6.917{\cdot}10^{6}$	$5.495{\cdot}10^{-11}$
e (-)	0	0	0
i (rad)	0.929	0.929	$1.388{\cdot}10^{-14}$
ω (rad)	0	0	0
Ω (rad)	0	$-5.859{\cdot}10^{-15}$	$-5.859{\cdot}10^{-15}$
V (m/s)	7,590	7,590	$1.091{\cdot}10^{-11}$

7.2.2 Complex Environment Verification

As has been observed in Section 2.1, the environment used throughout the simulations, is a complex one, encompassing a large number of forces and moments deemed essential for a comprehensive study. This approach ensures a close resemblance to the real environment, enhancing the fidelity of the simulations. Because of its elaborateness, it is essential to verify the model as a whole. While each individual model has already been tested for in previous studies and been taken from the GRADS library, the whole integration should now be tested and verified. The different tests performed for verification will be introduced hereafter.

TLE Propagation

One verification conducted to assess the accuracy of the Simulink® model and its environment involved examining the propagation of different satellites using their characteristics and TLE sequences at two distinct time steps. The TLE sequences, obtained from Space-track[2], for two consecutive time steps were selected. The first TLE sequence served as input data, with the propagation time determined from the epoch difference between the two TLE sequences. The resulting satellite state from the propagation was then compared to the second TLE sequence. Since the current Simulink® model includes an environment model specifically developed for satellites in LEO with altitudes close to 550 km (Section 3.4), a

[2] Space-Track, *ELSET Search tool*, https://www.space-track.org/#gp [Last visited on 15/04/2024].

satellite within this altitude range was used for this analysis. The outcomes of the latter are presented in Table 7.2. It is important to acknowledge that there may be slight discrepancies in the results. These discrepancies were inevitable due to the numerous elements considered in this analysis, which could lead to variations. Firstly, it is assumed that the observations provided by Space-Track are precise. While this assumption holds for short-term predictions, it may have limitations for longer propagation times, where adjustments become necessary. Furthermore, several conversions are employed (e.g., radians to degrees, TLE epoch conversion to date-time, and conversion from Kepler to Cartesian elements), which could introduce discrepancies in the model. Other sources of variation can be found in the unprecise satellite characteristics. In the example of Starlink, as outlined in Section 2.6.2, several input data had to be assumed for the simulation. Additionally, at the time of propagation, it is unknown how much propellant has been expended, directly impacting the current satellite mass and its overall behaviour. Another source of discrepancies being tested through this verification process is the definition of the environmental model and the inclusion of forces in the model to strike a balance between realism and computational efficiency. To finish with, it is assumed that the simulation models used are accurate (spherical harmonics, central field, aerodynamic drag but also Modified Julian Date (MJD) settings, conversions, constants used and more). Considering all these sources of discrepancies and examining the results, it can be concluded that the environmental model employed is acceptable and produces similar results to those observed and documented in Space-Track.

Table 7.2: Discrepancies between simulated and tracked orbital elements in real life for Starlink–4039 satellite from 05/05/2024 14h 30min 46s to 05/05/2024 16h 06min 09s.

Discrepancies between simulated and tracked trajectories for Starlink–4039					
Δa(m)	Δe(-)	Δi(°)	$\Delta\Omega$(°)	$\Delta\omega$(°)	Δt_p(s)
38.02	$3.62{\cdot}10^{-5}$	$7.72{\cdot}10^{-4}$	0.51	$5{\cdot}10^{-3}$	30.02

Perturbations

As described in Section 3.4, there exist multiple forces and moments relevant to include in the model, making sure the simulated trajectory resembles the closest to real-life scenarios. These have been thoroughly studied earlier. In order to verify the model used and their integration into the simulation, it has been decided to verify their order of magnitude with values that can be found in literature for the same orbital height. It has been decided to make use of the distance from the centre of Earth to the acceleration perturbation table that can be visualised in Montenbruck and Gill (2000) and compare these values with the ones from the simulation. The outcome of this analysis, which can be seen in Table 7.3, shows that the data from the simulation and the ones from literature are similar, verifying the developed perturbations model.

Table 7.3: Comparison between the environment model developed in this work and values found in literature.

Acceleration perturbations effects comparison		
Perturbations (m/s^2)	Montenbruck and Gill	Simulation
Aerodynamic drag	$\approx 10^{-5}$	$\approx 10^{-5}$
Moon PM	$\approx 10^{-5}$	$\approx 10^{-5}$
Sun PM	$\approx 10^{-6}$	$\approx 10^{-6}$
SRP	$\approx 10^{-7}$	$\approx 10^{-6}$

7.2.3 Guidance Verification

The guidance sequence constitutes a complex segment of the algorithm, incorporating multiple elements. Therefore, it is crucial to test each of these elements individually, as well as their integration within the guidance sequence. This section will introduce various tests aimed at ultimately verifying the entire guidance sequence.

Convex Optimisation Model Verification

The dynamics used for the convex optimisation, which have been linearised and discretised will be verified to make sure no mistakes have been made in the process. To do so, the open loop dynamics (no optimisation) for an equatorial orbit will be analysed against the reference trajectory (central field only), which serves as an initial value problem for the propagation of the dynamics. The result of such analysis is displayed in Figure 7.2. From this figure it can be observed that almost no discrepancies between both trajectories can be observed, verifying the model.

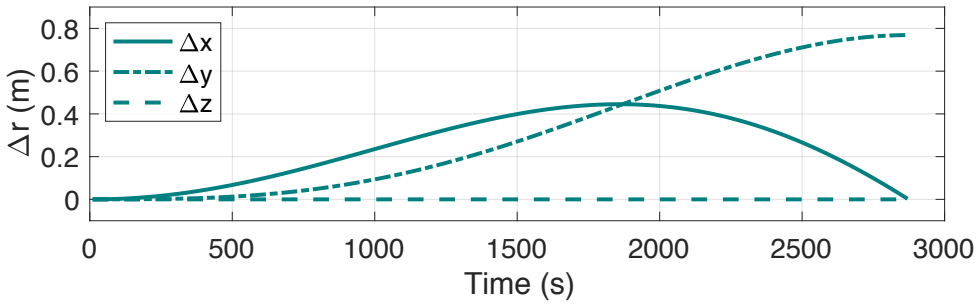

Figure 7.2: Difference between open-loop linearised and discretised dynamics and the simulated reference trajectory, in the case of an equatorial orbit.

Convex Guidance Verification

The entire developed guidance sequence will also undergo verification. The different tested elements and performed verification tests on the sequence will be introduced in this section.

Hohmann transfer comparison: A well-known trajectory transfer, due to its efficiency and overall global optimality, is the Hohmann transfer. This impulsive orbital manoeuvre consists of transferring a satellite from one circular orbit to another. This is the exact manoeuvre that is currently being investigated to move the satellite at a safe distance away from the debris, and thus on a new, higher, orbit. Due to the similarity between the optimised trajectory (convex) and the Hohmann transfer, but also because the Hohmann orbital manoeuvre is known for its global optimality (Prussing, 1992), it has been decided to compare both manoeuvres to verify the found convex guidance sequence. To make such verification, for simplification of the verification test, the convex optimisation has been performed on circular, equatorial and co-planar orbits. To start with, the theory behind a Hohmann transfer should be laid out. The ΔV required to perform a Hohmann transfer between two orbits around Earth can be described using the following sets of equations.

$$a_T = \frac{r_1 + r_2}{2} \tag{7.1}$$

$$\Delta V = V_2 - V_1 = \sqrt{\mu_{Earth}(\frac{2}{r_1} - \frac{1}{a_T})} - \sqrt{\frac{\mu_{Earth}}{r_1}} \tag{7.2}$$

With $r_1 = R_E + 550$ km and $r_2 = r_1 + 10$ km, the Hohmann transfer leads to a required ΔV consumption of 2.73 m/s. Using Tsiolkovsky equation (Equation (7.3)) the mass consumption can be computed using an I_{sp} of 292 s and a manoeuvre efficiency (η) assumed to be 1, leading to a Δm of 0.2476 kg for the first impulse.

$$\Delta m = m_0 \left(1 - e^{-\frac{\Delta V}{I_{sp} g_0 \eta}}\right) \tag{7.3}$$

Though highly efficient in optimal terms, a Hohmann transfer is a specific manoeuvre applicable to only a few scenarios. It is not a versatile method, as it necessitates execution at least half an orbit before conjunction, rendering it a slow process. Furthermore, Hohmann transfers lack flexibility in trajectory design, unlike convex optimisation, which tailors trajectories based on varied mission criteria. While it might not offer the utmost robustness and adaptability for the current mission's needs, the Hohmann transfer could serve for verification purposes. This is because the goal of achieving fuel optimisation for CAM aligns with the primary objective of Hohmann transfers. In terms of optimality, the Hohmann transfer is renowned for being the most efficient, making direct comparisons on this front impossible. However, by using data from Hohmann transfers, the current convex optimisation approach can be assessed in terms of performance, using the Hohmann transfer as a benchmark.

Both the thrust and mass profile of a non-impulsive Hohmann and of convex optimisation transfer will be compared. It has been chosen to simulate a non-impulsive Hohmann transfer, since in real life impulsive manoeuvres are not possible. This allows both mass and thrust profiles, using the same satellite characteristics to be compared with the nominal convex ones. The manoeuvre will be performed for both cases with a nominal available thrust of 10 N in all directions, an I_{sp} of 292 s for a central field circular orbit and a manoeuvre of 10 km at perigee. The results of such analysis can be observed in Figure 7.3 and Figure 7.4. Upon analysis, it becomes evident that the thrusting period does not coincide, though further details on this will be provided later. Examination of the mass and thrust profiles reveals that the Hohmann transfer indeed results in a more optimal profile, characterised by a shorter thrust period and reduced propellant mass consumption. Despite being larger and occurring at a different time, both the thrust and mass profiles exhibit behaviour akin to the Hohmann profile, displaying a 'bang-bang' control pattern. The required mass for convex guidance amounts to 1.73 kg, surpassing that of Hohmann, yet deemed satisfactory, given the nature of the Hohmann transfer.

Solver comparison: A final verification test will be performed on the use of different solvers. This will guarantee that the optimum solution found through convex optimisation is not dependent on the chosen solver and that any solver would output similar results. In order to perform such a verification step, it has been decided to compare the behaviour of *Gurobi*, *fmincon* and *IPOPT* solvers in the YALMIP environment. Based on this analysis, it can be inferred that all three simulations suggest that the most effective approach for executing such a manoeuvre involves initiating thrust at approximately the halfway point, exhibiting similar duration of maximum thrust behaviour. This observation ensures that trajectory is indeed optimised, with a thrust boost occurring around 1,400 s into the simulation. However, from the analysis, some discrepancies are worth noting. To start with, the simulation time is not comparable, while the *Gurobi* solver finds a solution to the problem in a few seconds, a few

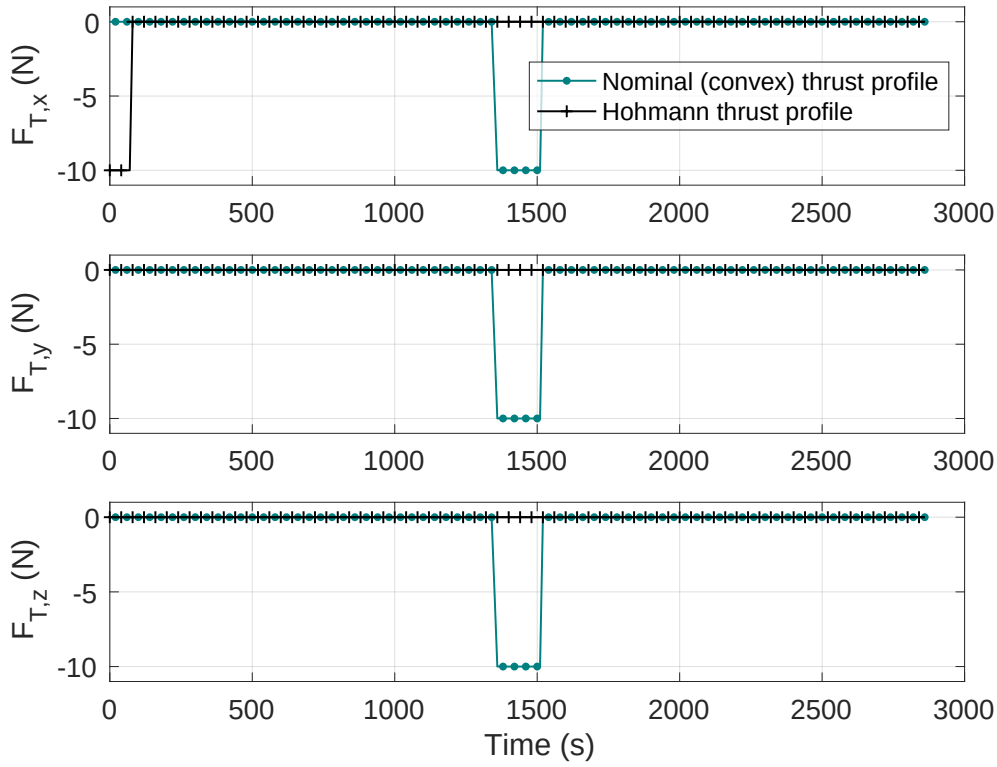

Figure 7.3: Thrust profile nominal (convex) vs. Hohmann manoeuvre.

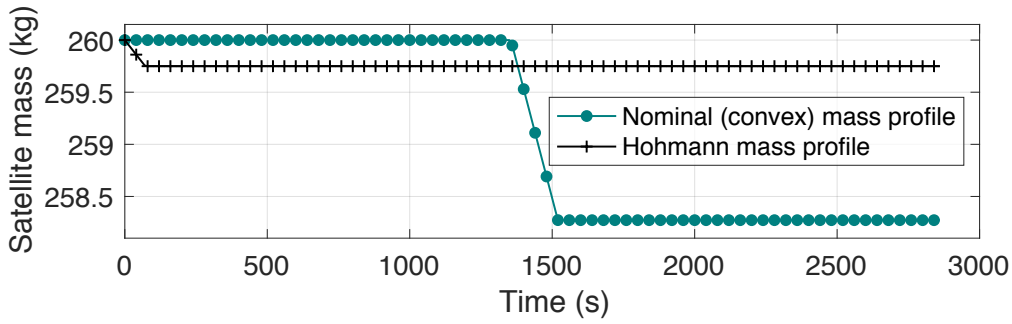

Figure 7.4: Propellant mass profile nominal (convex) vs. Hohmann manoeuvre.

minutes ($\approx$ 10 min) is necessary to get an output from *IPOPT*, with a fixed tolerance put to 10^{-6} to allow for faster solving. For *fmincon* the CPU is approximately 6 hours (tolerance put to 10^{-3}), making it a very slow solver that cannot be used on-board the satellite. In addition to this, by looking at Figure 7.5, it can be observed that the thrust profile from the *IPOPT* solver does not exhibit 'bang-bang' profiles, but as a result leads to a larger fuel consumption as observed in Figure 7.6. This smoother behaviour could be positive as input to the control system since it will avoid overshoots, but does not meet the main objective of the CAM sequence; to be fuel efficient. More analysis on the latter will be performed in Chapter 8. Overall, the mass profiles of the *fmincon* and *Gurobi* solvers are similar, with a difference in thrust usage of 8.2 g. The *IPOPT* solver mass consumption exceeds the other two solvers by approximately 0.99 kg.

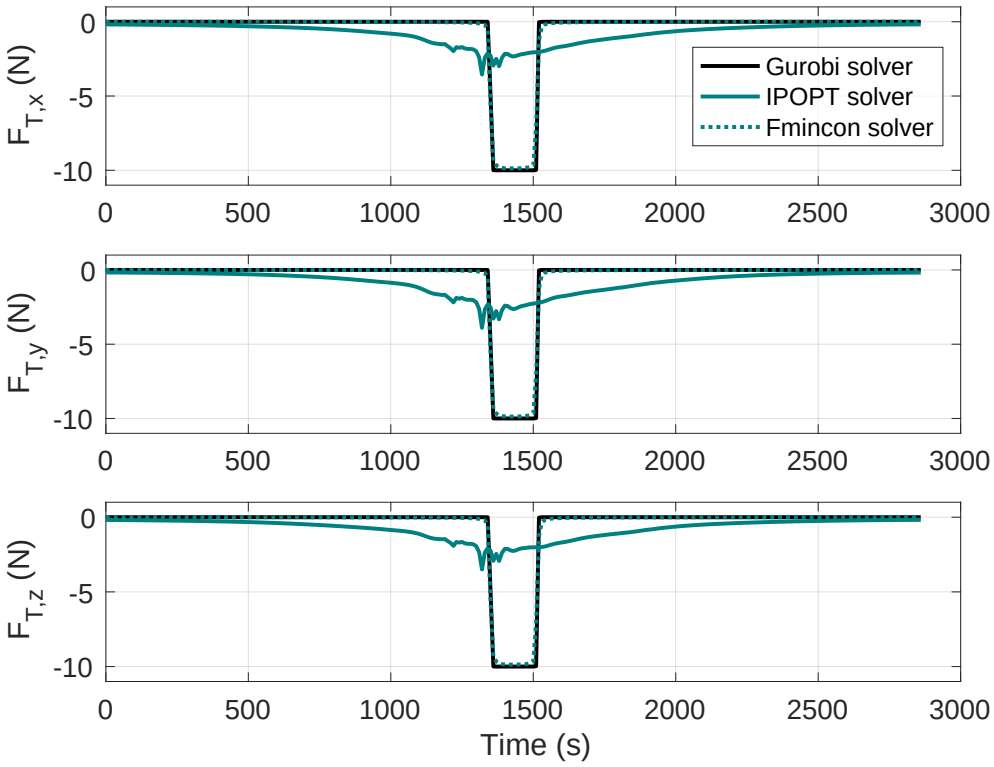

Figure 7.5: Variation in thrust behaviour between different solvers for an equatorial orbit.

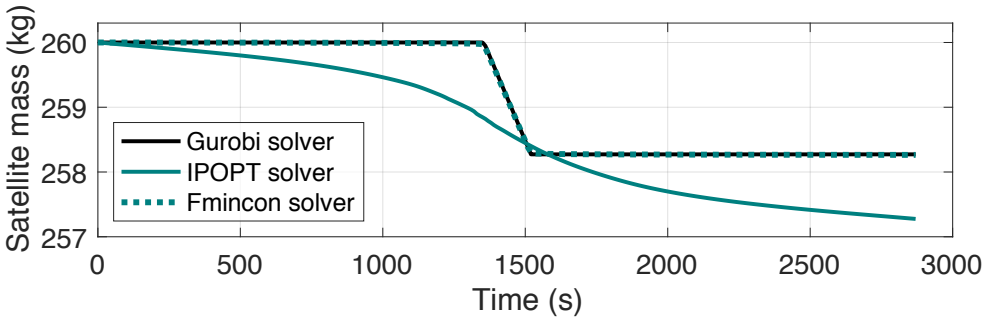

Figure 7.6: Variation in propellant usage between different solvers for an equatorial orbit.

In addition to verifying and testing the behaviour of different solvers, the tolerance difference using the *IPOPT* solver has also been analysed. Three different tolerances have been applied for the solver, 10^{-3}, 10^{-6} and 10^{-9}. The outcome of each of them has been displayed in Figure 7.7. From this analysis, it can be observed that no significant differences can be observed from changing the tolerances, and the effect on the mass consumption was also found to be small, in the order of a few grams. The only significant difference was in terms of computational time in which a smaller tolerance led to a larger CPU time, as expected.

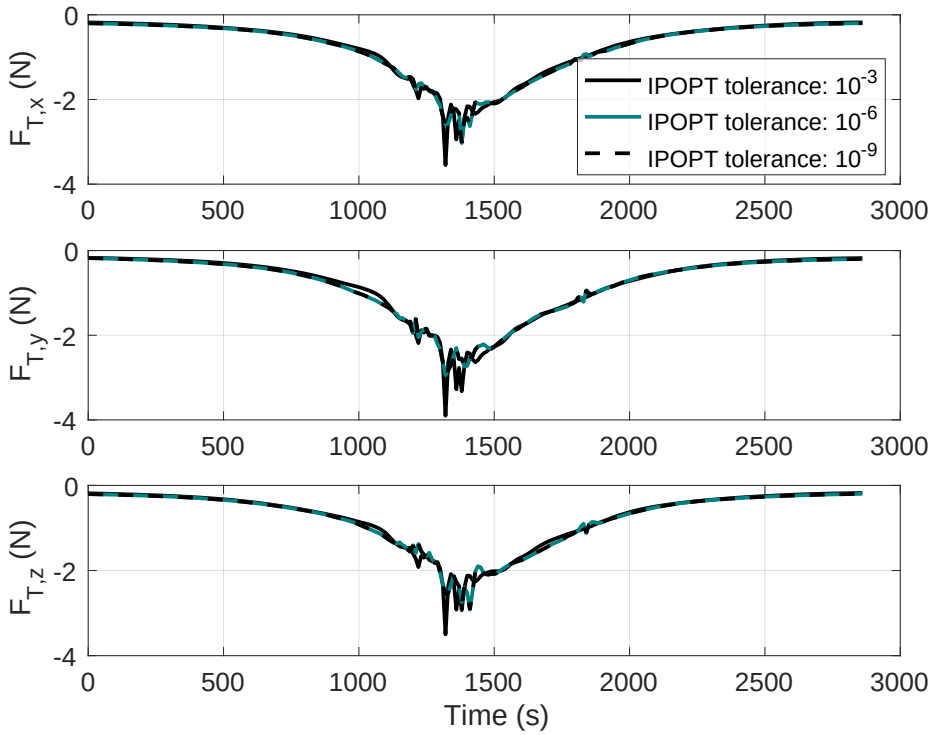

Figure 7.7: Overview of the thrust profiles for different tolerances set in the *IPOPT* solver settings for an equatorial orbit.

Tracking Guidance Verification

The tracking guidance system is mainly based on the LQR controller used for each guidance segment, to make sure the satellite follows the nominal trajectory adequately. The verification step to verify this system will thus be performed on the controller. The performance of the LQR control system will be verified by observing the transient response of the system to a step function on the state error. Since this step function is a known input, it can easily be tested for. This verification test is thus performed on a linear system, with the $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$, $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices needed to get the gain, constructed as described in Section 5.5. The gains should be defined such that the response to the control is damped before reaching the control steady state in the shortest amount of time possible while still meeting the system's characteristics. As suggested by Mooij (2020), when the response to any input can be analysed and understood, the reaction to any input can, in theory, be computed. To verify the performance of the presently developed LQR system, diverse inputs will be administered to the controller, and its efficacy will be assessed based on settling time, overshoot, steady-state error, and control effort. Subsequently, fine-tuning of the gains will be undertaken until the controller's input demonstrates satisfactory performance. The results of this analysis are depicted in Figure 7.8, where a step function representing the state and saturation on the commanded torque are used to produce a smoother and more realistic signal. Figure 7.8 illustrates that the isolated LQR controller's time response to a step function in state error is characterised by a smooth signal with no overshoot. The system's response is both smooth and prompt, which is advantageous for the control actuators. Thus, based on this behaviour, it can be inferred that the controller effectively alters the satellite's state while fulfilling the accuracy requirements.

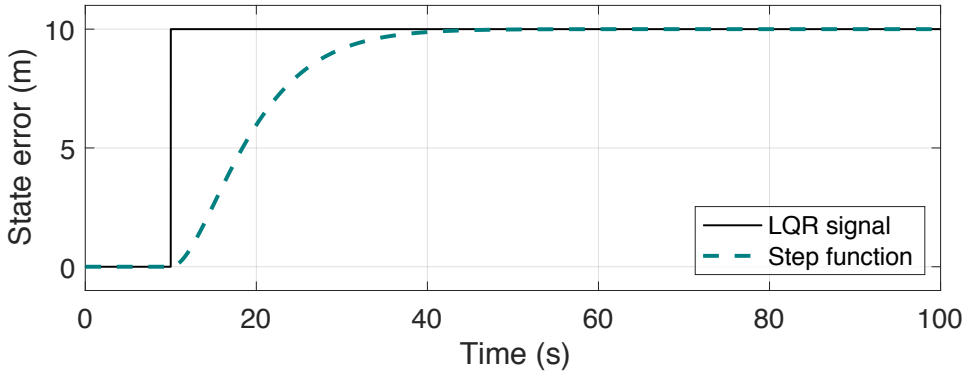

Figure 7.8: Time response of the LQR controller for tracking guidance of a state error step function.

7.2.4 Control Verification

Within the control system, several transformation matrices facilitate the translation of control elements into the appropriate reference frame. The rotation matrices introduced in Section 3.1 are considered already verified, since they have been sourced from the GRADS library. However, the integration of these matrices into the simulator underwent verification to ensure their functionality aligned with their intended purpose. This verification process involved using known analytical data and comparing the transformation output against the expected result. This verification process ensured the rotation matrices had been used and implemented properly in the simulator.

Another integral component of the control system involves the LQR controller used for attitude control. The effectiveness of this controller relies significantly on several input parameters as well as the meticulous tuning of its gains. Consequently, it is imperative for the controller to undergo fine-tuning to minimise quaternion control error in an efficient manner. This optimisation results in a highly responsive controller, facilitating the alignment of attitude control with the desired parameters within a brief duration. To verify such behaviour, the LQR control algorithm was implemented in an isolated system and executed using the tuning parameters highlighted in Table 7.4, to correct for a commanded attitude change of 4° for all three Euler angles (converted to quaternions). This verification process ensures that the controller performs optimally, as can be seen in the verification test output displayed in Figure 7.9. The figure reveals a rapid convergence of quaternion error to zero, indicative of the system's stability and efficiency. This observation underscores the robustness and proficiency of the implemented control system to meet the desired attitude with great accuracy and reliability. This control system will be used in the simulation to perform attitude control.

Table 7.4: Tuning parameters for the verification of the LQR system, for a commanded attitude change of 4° for all three Euler angles.

LQR tuning parameters						
$\Delta\omega_1(°/s)$	$\Delta\omega_2(°/s)$	$\Delta\omega_3(°/s)$	Δq_1	Δq_2	Δq_3	$M_{min/max}(N)$
15	15	15	0.005	0.005	0.005	± 10

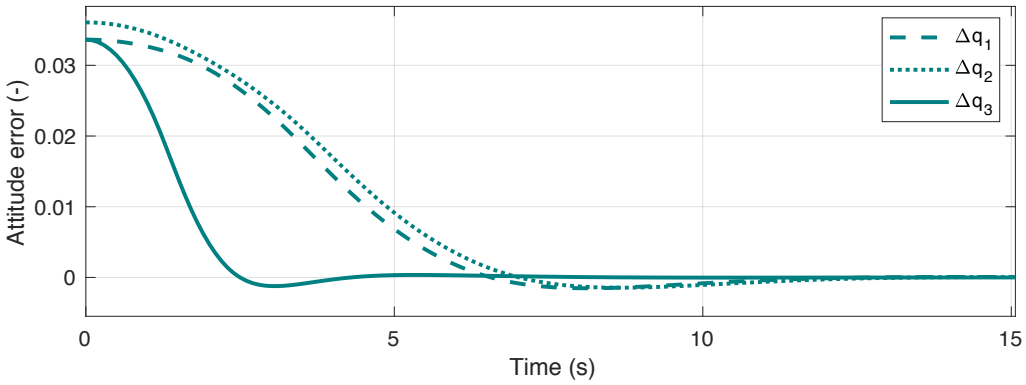

Figure 7.9: LQR controller response to an attitude error of the satellite for a commanded attitude change of 4° for all three Euler angles.

With the simulator's design fully detailed, along with a comprehensive review of the verification processes that ensure all requirements and pre-determined specifications are met across each developed system, the foundation is now set to present the simulation results. These will offer valuable insights into the effectiveness and accuracy of the implemented strategies, will be thoroughly examined in the forthcoming chapter.

CHAPTER

8 Results

The results of the various simulations and test runs have provided insights into the performance of the developed systems. Some of these results are presented in this chapter. In Section 8.1, the analysis of both nominal and tracking guidance will be highlighted. This will be followed by Section 8.2, in which the attitude control performance will be discussed. This chapter focuses on the results for the actual inclined Starlink trajectory, using the following standard settings, unless stated otherwise: $F_{T_{nom}}$ = 10 N, I_{sp} = 292 s, and the *Gurobi* solver.

8.1 Guidance System Results

All results related to the guidance system will be presented in this chapter. First, the results related to the nominal guidance will be discussed in Section 8.1.1. Then, the results of the tracking guidance will be introduced in Section 8.1.2.

8.1.1 Nominal Guidance Results

The nominal guidance results will include an analysis of the optimised thrust, mass, and trajectory profile to meet the miss distance requirement. Additionally, it will highlight the various trade-off analyses conducted on different solvers, thrusting capabilities, the effects of perturbations, and changes in the reference orbit scenario. The *fmincon* solver has been excluded from this analysis, as observed in Section 7.2.3, due to its slow computational time, making it unsuitable for this fast-speed CAM. Thus, its analysis will be omitted.

Optimised 3D Trajectory

The complete optimised trajectory of the Starlink satellite is depicted in the 3D plot available in Figure 8.1. Within this visualisation, one can observe the nominal, optimal trajectory determined by the convex guidance sequence, alongside the reference trajectory (restricted to the central field) used for reference during the convex optimisation process (as elaborated in Section 5.4). Due to the current scale of the trajectory, any distinctions between different solvers are indiscernible. Similarly, the difference between the reference and optimal trajectories appears negligible due to the plotting scale. A more detailed 2D analysis will be incorporated later to provide deeper insights into the trajectory's specifics.

Optimised Thrust Profile

In Section 7.2.3, it was evident that significant differences in CPU time were observed when comparing the performance of *fmincon*, *Gurobi*, and *IPOPT* solvers. Specifically, optimisa-

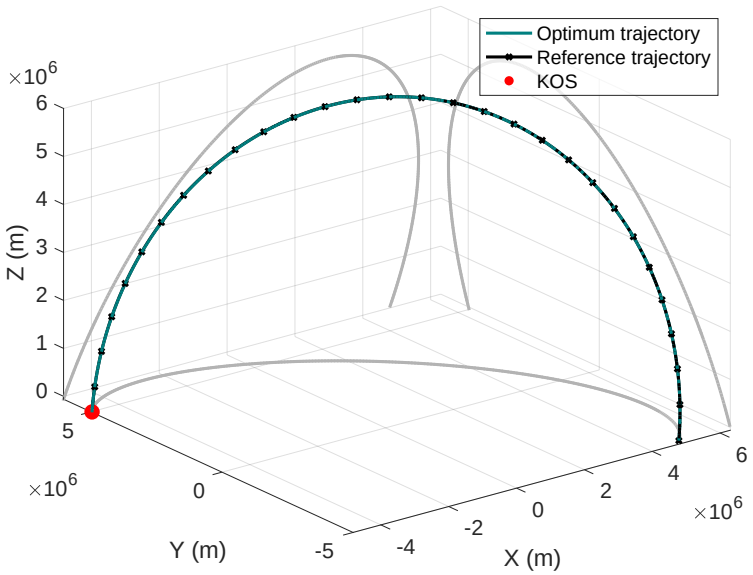

Figure 8.1: 3D trajectory overview of the reference and nominal (optimal) trajectory, with the KOS (size not to scale) position.

tion using *fmincon* took approximately 6 hours, rendering it impractical for on-board implementation. Consequently, the decision was made to discontinue the analysis of *fmincon* and focus on comparing the effects of *IPOPT* and *Gurobi* solvers on the control sequence. As illustrated in Figure 8.2, both solvers yield consistent thrust profiles but exhibit distinct behaviours. Notably, the *Gurobi* solver produces a 'bang-bang' profile characterised by rapid switching between two extreme control values, while the *IPOPT* solver generates a smoother trajectory. The 'bang-bang' control sequence is favoured for its optimality, as it minimises fuel consumption. However, this profile may introduce drawbacks, such as discontinuities in the control signal, potentially leading to over or undershoots in control behaviour. The smoother yet more energy-intensive profile produced by the *IPOPT* solver could mitigate these issues. These aspects will be further explored in Section 8.2.

Optimised Mass Profile

As previously discussed, and as evident from Figure 8.3, the 'bang-bang' profile demonstrates superior optimality in terms of energy usage. Examining the mass profiles for the trajectory of the Starlink satellite with thrusting capabilities set using the standard settings, it becomes apparent that the nominal convex guidance sequence necessitates a propellant consumption of 1.73 kg when using the *Gurobi* solver and 3.04 kg when employing *IPOPT*. This results in a discrepancy of 1.3 kg in fuel consumption between the two profiles, given these used parameters. The variance between solvers is notably significant.

Optimised 2D Trajectory and Constraints

The difference between the linearised reference and optimised trajectory has been analysed to observe how the satellite deviates from its initial trajectory to perform manoeuvre and stay outside the keep-out sphere at the end of the trajectory and by this, avoids the debris. In Figure 8.4, the deviation in trajectory in X, Y and Z direction can be observed, while in Figure 8.5, the difference in norm position for both *IPOPT* and *Gurobi* solver is shown. These

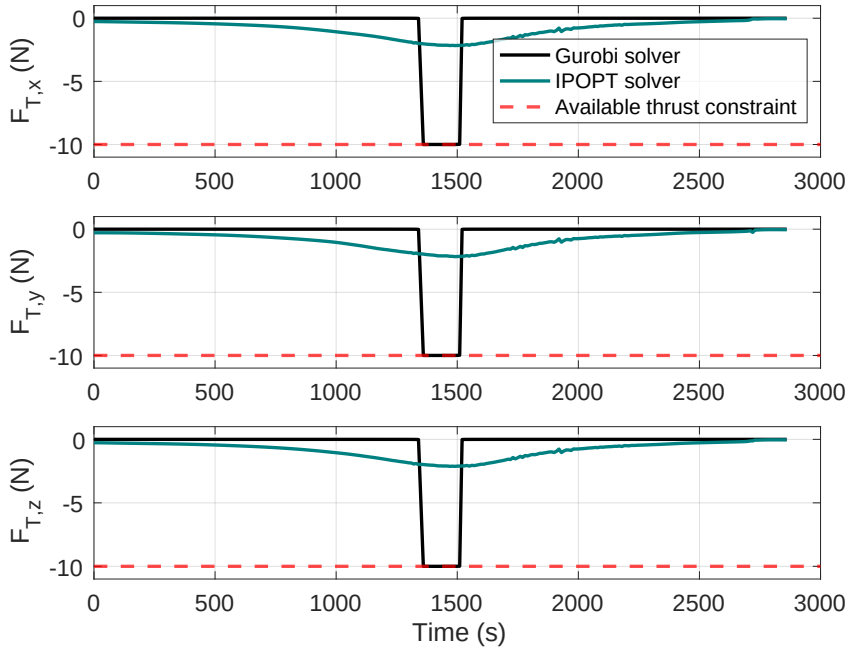

Figure 8.2: Thrust profile in all three directions for the following thruster's characteristics: $F_{T_{nom}}$ = 10 N, I_{sp} = 292 s.

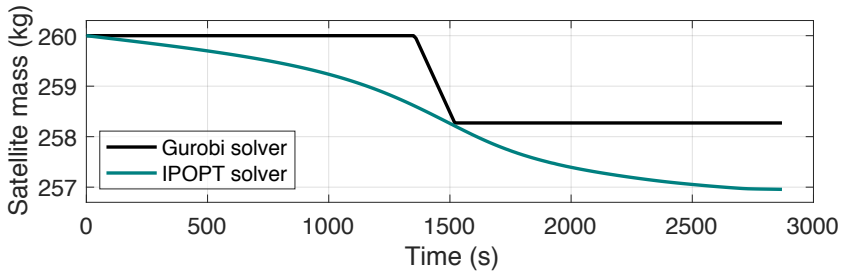

Figure 8.3: Satellite mass variation due to propellant usage necessary for nominal guidance, for the following thruster's characteristics: $F_{T_{nom}}$ = 10 N, I_{sp} = 292 s.

observations collectively affirm that the KOS constraint is adhered to and that the satellite successively avoids the debris. When simulating for the optimised thrust profiles introduced in Figure 8.2, it was observed that indeed a miss distance of 10.12 km was met at 2,867 s throughout the trajectory, as expected. From the simulation, the different Kepler elements can be plotted to better understand how the manoeuvre is performed. These are displayed in Figure 8.6. From this plot, it is evident that the manoeuvre has a direct impact on the Kepler elements of the satellite. Initially, a slight deviation in eccentricity can be observed, indicating a minor modification in the orbit's shape. The semi-major axis exhibits slight fluctuations, which implies that it remains relatively stable, preserving the orbital period. The inclination shows minimal change, suggesting that the satellite continues to follow a consistent orbital plane. Similarly, the variation in the RAAN is negligible, further confirming the stability of the orbital plane. However, the manoeuvre notably influences the satellite's true anomaly and argument of perigee, indicating a repositioning within the orbit and a reorientation of the elliptical orbit within its plane. This suggests that while the manoeuvre maintains the general characteristics of the orbit, it significantly adjusts the satellite's position and orientation within that orbit.

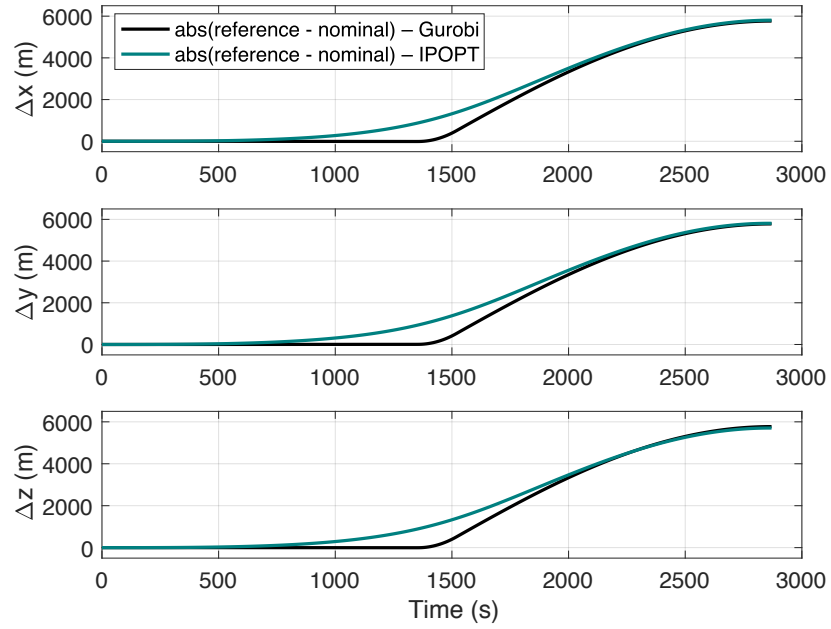

Figure 8.4: Difference in trajectory between the reference and nominal trajectory to avoid the debris and stay outside the KOS, using both *Gurobi* and *IPOPT* solvers and the standard thrust parameters – Profile from linearised and discretised model.

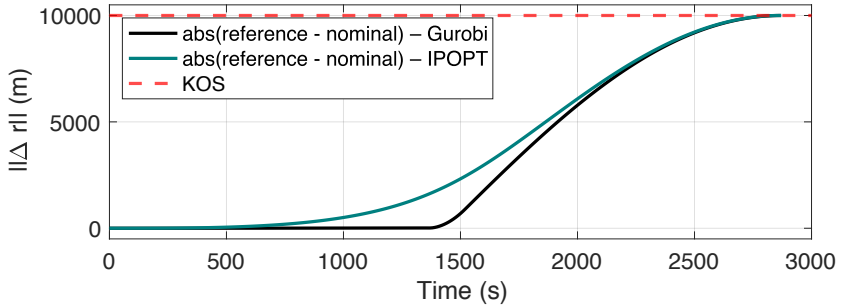

Figure 8.5: Overview of the position norm between the nominal and reference trajectory to evaluate how the KOS requirement is met for both *IPOPT* and *Gurobi* solvers, standard thrust parameters – Profile from the linearised and discretised model.

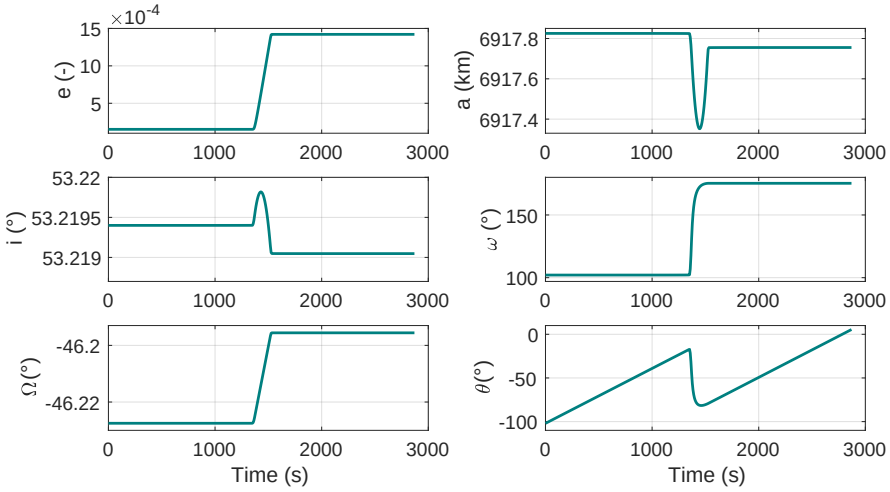

Figure 8.6: Relevant Kepler elements from the satellite trajectory during manoeuvre.

Manoeuvre Period Analysis

The analysis conducted examined the impact of reducing the warning time prior to manoeuvring and its effect on the optimised trajectory. The approach involved shortening the trajectory based on its initial true anomaly (θ_0). In the case of the initial half-orbit scenario, the initial true anomaly was 0°. The analysis explored various θ_0 within the ranges of [0°, 45°, 90°, 135°, 145°], resulting in each trajectory starting with a different true anomaly and having less time to manoeuvre before encountering debris (located at θ = 180°).

The findings of this analysis are presented in this section and will be examined sequentially. It is important to note key aspects of this analysis. The study used *Gurobi* as a solver, with a nominal thrust of 10 N available in all directions and I_{sp} of 292. It is worth emphasising that outcomes may vary with different profile characteristics, this factor will be analysed and included in subsequent stages of this chapter. Additionally, for visualisation purposes, a uniform time frame was maintained for all trajectories. Although each trajectory theoretically starts at t = 0 with varying trajectory times, the decision was made to align everything with the longest trajectory (beginning at θ = 0°) to aid visualisation and comprehension. However, it is essential to recognise that each simulation starts at a distinct time, as pointed out throughout the analysis. All five case characteristics that have been analysed for this study have been gathered in Table 8.1. Optimisation for a warning time smaller than that of case 5 results in failure under the specific constraints and objectives employed in this simulation (10 km miss distance, F_T = 10 N, I_{sp} = 292, and *Gurobi* solver – standard settings). This failure stems from the need for either greater thrust capabilities or tighter miss distance requirements to fulfil the criteria within such abbreviated time frames. This study thus exhibits the limitations linked with the problem's parameters. Further analysis of the matter will be introduced later in this chapter. For now, the results of all five cases will be highlighted hereafter.

The convex guidance sequence has been simulated for all five cases. To start with, the thrust profiles for each case have been analysed. It must be noted that only the thrust behaviour in the X direction has been displayed for conciseness, since, as has been observed earlier, the thrust profile is similar in all three directions. As shown in Figure 8.7, all thrust profiles demonstrate a consistent pattern. On one hand, when the manoeuvre begins earlier than the nominal starting time previously analysed (around 1400 seconds), the manoeuvring duration remains constant at approximately 200 seconds for the first three cases (θ_0 = 0°, θ_0 = 45°, and θ_0 = 90°). This consistency also leads to a similar mass consumption, as will be discussed later. On the other hand, as the warning time before the manoeuvre decreases, the duration of the manoeuvre tends to increase. This trend is observed in the case of θ_0 = 135°, where the manoeuvring time extends to approximately 250 seconds, and for θ_0 = 145°, where it reaches around 500 seconds. Consequently, this results in higher mass consumption. These trends can be explained by the increased necessity for a larger ΔV to

Table 8.1: Different characteristics for different manoeuvring periods.

Manoeuvring period cases characteristics			
Cases	**θ_0**	**Simulation period (s)**	**Simulation time (s)**
Case 1	0°	0 – 2880	2880
Case 2	45°	720 – 2880	2160
Case 3	90°	1440 – 2880	1440
Case 4	135°	2160 – 2880	720
Case 5	145°	2310 – 2880	470

achieve the desired miss distance sequence when the satellite is in closer proximity to the target. This characteristic is further reflected in the corresponding mass consumption profiles (Figure 8.8), where it is observable that propellant consumption rises as the warning time decreases, mirroring the thrust profile trend. Notably, the disparity between the two extreme cases, starting with a $\theta_0 = 0°$ and $\theta_0 = 145°$, results in a substantial difference in propellant mass usage of approximately 3.50 kg. This discrepancy is significant and cannot be neglected when searching for the optimum CAM strategy. Directly correlated with both the mass and thrust profiles, it becomes apparent that across all five cases, the 10 km miss distance requirement is achieved (Figure 8.9). Each trajectory profile is optimised according to the constraints specific to each case. Notably, for shorter warning times, the manoeuvres appear more abrupt and direct, requiring larger resources within a constrained time frame.

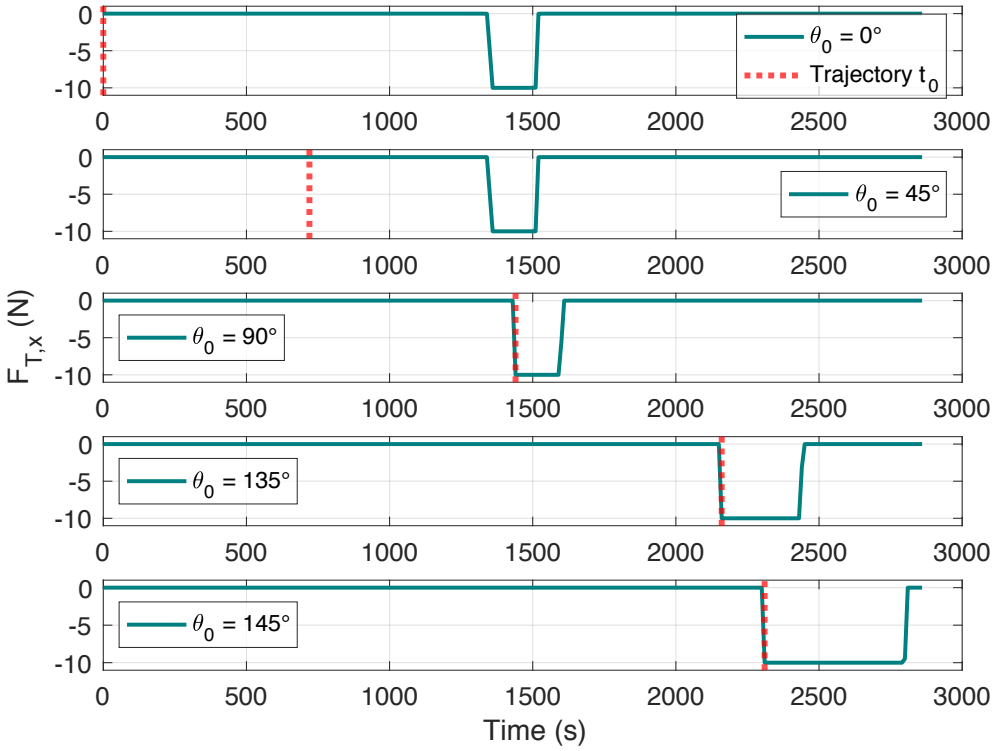

Figure 8.7: Thrust profile in the X direction for different manoeuvring starting periods.

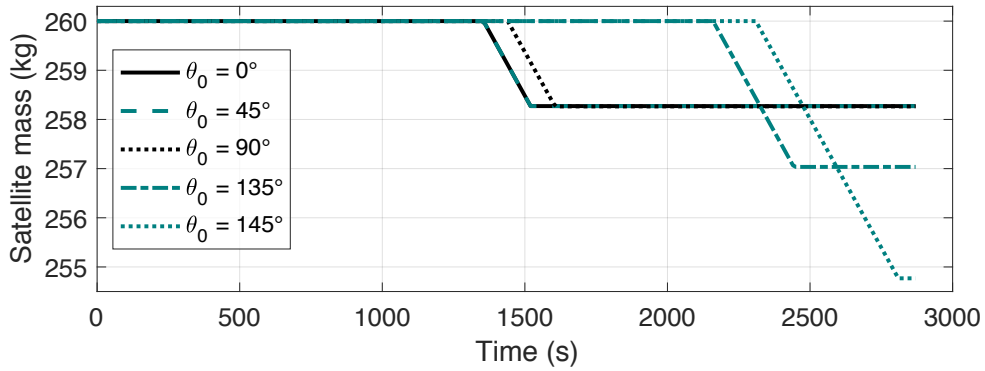

Figure 8.8: Satellite mass variation for different manoeuvring starting periods.

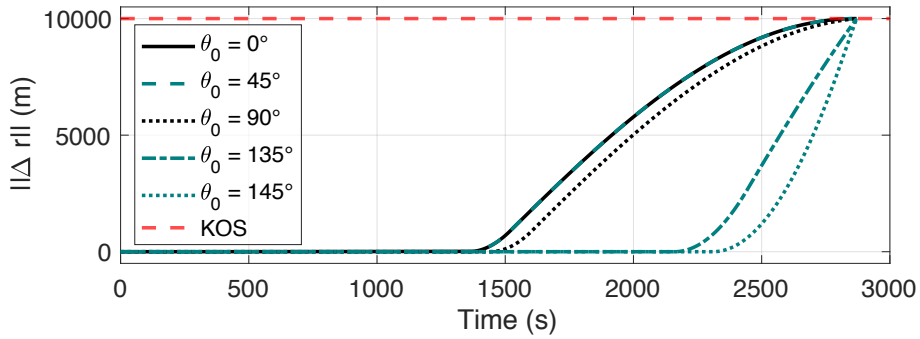

Figure 8.9: Position norm between the nominal and reference trajectory for different manoeuvring starting periods – Profile from the linearised and discretised model.

Thrusting Capabilities Limitations

To ensure the robustness of the current CAM algorithm, it must demonstrate effectiveness not only within the specific system it was initially designed for (described by the predetermined parameters) but across a spectrum of systems. This necessitates testing the algorithm's convex guidance capabilities across various propulsion types, including both high/medium-thrust (chemical propulsion) and low-thrust (electric propulsion) systems. Each propulsion system has its specifics and is often tailored for distinct mission sequences. To comprehensively assess the CAM algorithm's performance, testing will encompass high/medium-thrust propulsion systems, like cold gas, solid motor, mono-propellant, and bi-propellant. Additionally, evaluations will include low-thrust options like hall-thrusters. Furthermore, each system will undergo testing across various thrusting capabilities, ensuring a thorough examination of the algorithm's versatility. The different propulsion systems that will be tested, together with their characteristics have been highlighted in Table 8.2.

Table 8.2: Different propulsion systems and their characteristics

Propulsion System Characteristics			
Propulsion	**Vacuum I_{sp}(s)**	**Thrust range (N)**	**Propulsion type**
Cold gas	50 – 75 → 50	0.05 – 200 → [1, 200]	Medium-thrust (high pressure)
Solid motor	280 – 300 → 300	50 – 5×10^6 → [100, 10^6]	High-thrust (chemical)
Mono-propellant	150 – 225 → 200	0.05 – 5 → [0.1, 5]	Medium-thrust (exothermic)
Bi-propellant	300 – 450 → 400	5 – 5×10^6 → [10, 10^6]	High-thrust (chemical)
Hall thruster	1000 – 3000 → 2000	0.005 – 0.4 → [0.01, 0.1]	Low-thrust (electric)

For this analysis, due to its low CPU time, the *Gurobi* solver has been used. For each combination of thrust and specific impulse highlighted in Table 8.3, the miss distance is met and the optimisation is successful. From the plots displayed after, it can be observed that the exhibited profiles are very different from one propulsion type to the other. For conciseness, it has been decided not to include all thrust profiles since they can be deducted from the mass profile as they exhibit a 'bang-bang' scenario every time (*Gurobi* solver). For this reason, it has solely been decided to look at the mass consumption profile and from this, a number of conclusions can be reached. From Table 8.3 and Figure 8.10, it can be observed that cold gas

exhibits larger mass consumption, at least for the parameters tested for and for a 260 kg satellite and this specific type of manoeuvre with a warning time of half an orbit. In contrast, it can be observed from Figure 8.11 and Figure 8.12, that the low-thrust propulsion exhibits the smaller mass consumption of all, still meeting the miss distance and has a constant thrust for the entire trajectory. With these results, this profile makes it the most optimum, leading to the conclusion that low-thrust propulsion would be the future of low-Earth orbit manoeuvring propulsion. This analysis suggests that, despite the widespread use of chemical engines for their ability to rapidly alter a spacecraft's motion, they may not be the most optimal choice for the specific CAM under investigation. Although well-established and reliable, they necessitate significant propellant quantities and emit substantial fuel during bursts, contradicting current technological trends towards greener, more eco-friendly space systems. An alternative propulsion method involves continuous thrusting, using more mass-efficient low-thrust systems. These systems, characterised by lower thrust outputs compared to impulsive systems, offer higher exhaust velocities while minimising propellant consumption (Wakker, 2015). As was highlighted in Section 2.6, the Starlink constellation, which is the one used as a reference vehicle for this study, is actually making use of low-thrust propulsion, through its ion thrusters (Krypton-propelled, hall-effected thrusters). Electric propulsion is a current growing technology in the space market, making it a key revolutionary technology for the new generation of satellite[1], such as Starlink.

Table 8.3: Mass consumption for each propulsion type and studied characteristics.

Propulsion System CAM Mass Consumption			
Propulsion type	**Vacuum I_{sp}(s)**	**Thrust (N)**	**Δm(kg)**
Cold gas	50	1	12.6
Cold gas	50	200	10.08
Solid motor	300	50	1.68
Solid motor	300	10^6	1.68
Mono-propellant	200	0.1	0.4388
Mono-propellant	200	5	2.53
Bi-propellant	400	10	1.26
Bi-propellant	400	10^6	1.26
Hall-thruster	2000	0.01	0.0044
Hall-thruster	2000	0.1	0.042

Perturbations Effects

The impact of environmental perturbations should be analysed next, since their influence on a LEO satellite cannot be ignored, as has been highlighted in Section 3.4. As outlined in Chapter 5, the convex guidance, which issues commands to follow a nominal trajectory, has been developed solely using the central field model. Consequently, the effects of other perturbations have not been considered. An analysis has been performed by switching the relevant perturbations included in this research work one by one and looking at the effects each one had on the final miss distance. Looking at Table 8.4, it is evident that the impact of external perturbations is substantial, leading to a miss distance of 23.45 km. The necessity of tracking guidance to ensure the perturbed trajectory closely follows the nominal path, is indeed underscored. The tracking guidance specifics have been highlighted in Section 5.5 with the results of such analysis discussed later.

[1] ESA Electric Propulsion, *What is Electric propulsion?*, https://shorturl.at/mnoHQ [Last visited on 09/05/2024].

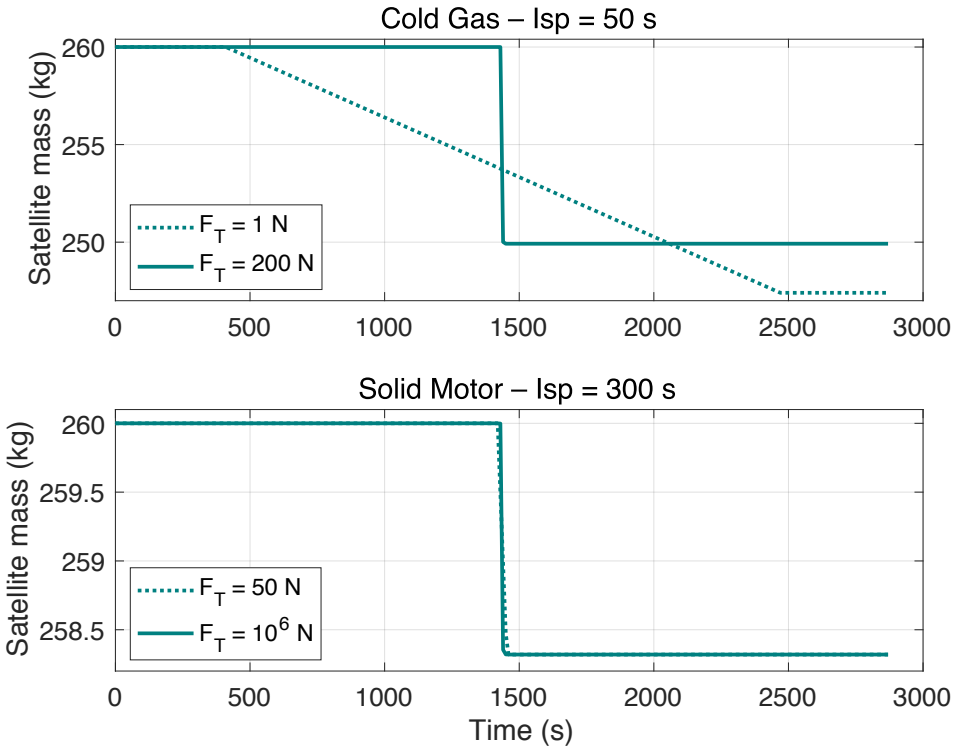

Figure 8.10: Mass consumption of both cold gas and solid motor propulsion systems for various available thrust and a specific impulse of 50 s (top) and 300 s (bottom), respectively.

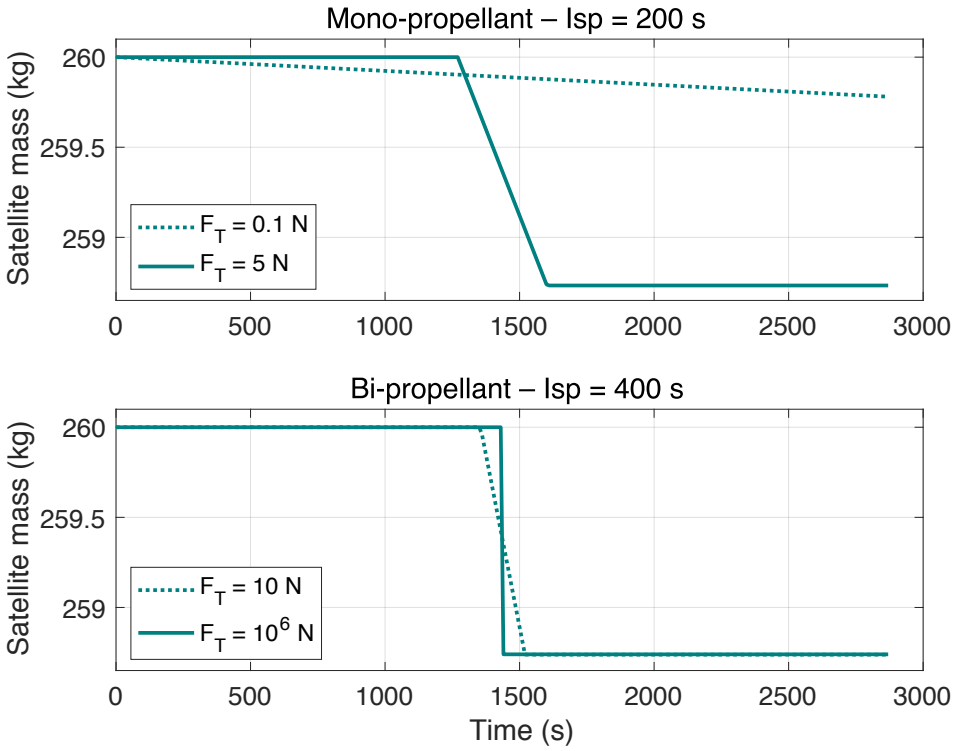

Figure 8.11: Mass consumption of both mono and bi-propellant propulsion systems for various available thrust and a specific impulse of 200 s (top) and 400 s (bottom), respectively.

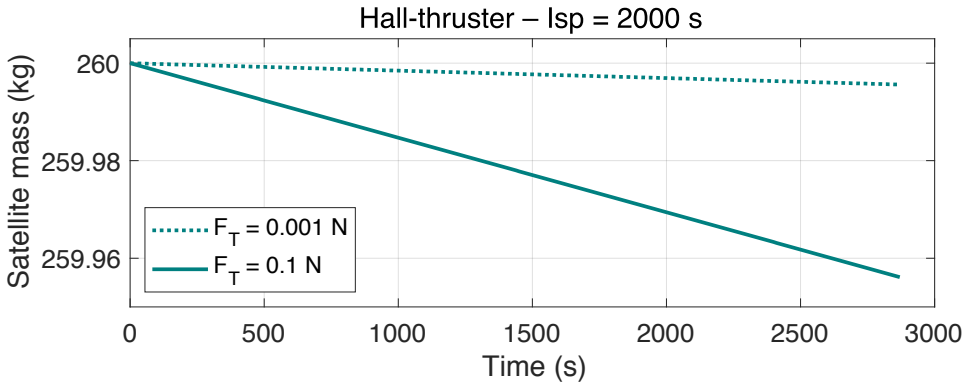

Figure 8.12: Mass consumption of hall-thruster (low-thrust) propulsion systems for various available thrust and I_{sp} = 2,000 s.

Table 8.4: Effect of the environmental perturbations on the satellite trajectory by looking at its deviation from the debris, which should be 10 km in the nominal case.

Perturbation effects on satellite – debris miss distance							
Perturbations	CF	SH 32/32	Aero. drag	SRP	Moon PM	Sun PM	All
Miss distance (m)	10,129	23,447	10,134	10,129	10,131	10,130	23,455

8.1.2 Tracking Guidance Results

As detailed in Section 5.5, the tracking guidance sequence is used to ensure that the satellite closely follows the nominal path defined by the convex guidance, taking all relevant perturbations into account. This tracking guidance is based on controlling the satellite's translational state using an adaptive LQR control, which aims at minimising mass consumption along the trajectory. As illustrated in Table 8.4, perturbations cause significant deviations in the satellite's state. The primary goal of the tracking guidance is to minimise these deviations as much as possible, according to the parameters set out in Table 5.3. Various thrusting capabilities were analysed to evaluate their impact on both deviation mitigation and mass consumption. Additionally, changes in LQR settings were also studied to understand their effect on the system. Several cases were examined, as highlighted in Table 8.5. From this analysis, only case 3 resulted in successful tracking, with a final state deviation of 275.5 m. All other cases led to unsuccessful tracking due to insufficient nominal thrust and the used settings, causing saturation in the LQR controller and an inability to redirect the satellite as needed. This issue is especially pronounced with low-thrust propulsion, where tracking was found to be unfeasible, as the satellite could barely counteract the deviations induced by environmental perturbations.

Table 8.5: Overview of different thrust profiles and LQR parameters on the tracking guidance sequence behaviour.

Analysis of tracking guidance performance for different cases.							
Cases	F_T(N)	I_{sp}(s)	$1/2T_{1/2}$	$3/4T_{1/2}$	$T_{1/2}$	$\|\Delta r_f\|$(m)	$m_{consumed}$(kg)
Case 1	10	292	5,000	2,500	1,000	3,539	8.3
Case 2	10	292	10,000	5,000	1,000	2,192	6.0
Case 3	100	300	10,000	5,000	1,000	275.5	10.1
Case 4	0.1	2,000	10,000	5,000	1,000	35,380	1.0

The detailed behaviour of the successful tracking has been illustrated through several plots to better understand the objectives and execution of the tracking guidance. Initially, the difference in state between the nominal commanded trajectory and the actual satellite trajectory, with constant tracking activated and all perturbations present, is shown in Figure 8.13. This demonstrates that the tracking successfully maintains the miss distance at an acceptable level. Given the initial deviation depicted in Figure 5.6, the tracking guidance proves effective in this specific case. The required thrust profile to achieve such attitude control is shown in Figure 8.14, resulting in a significant mass consumption of 10.1 kg, as highlighted in Figure 8.15.

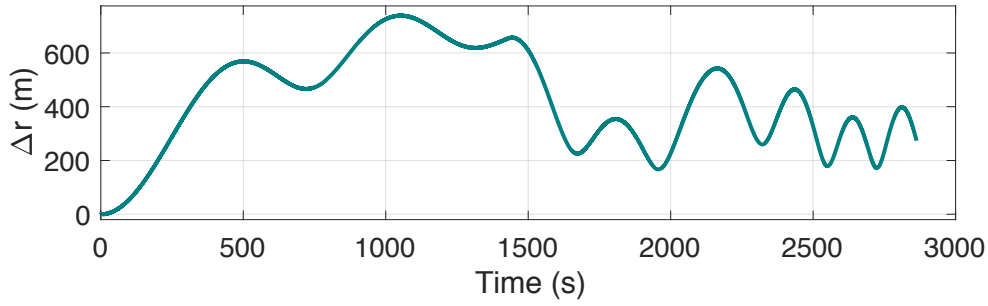

Figure 8.13: Difference in trajectory between the nominal commanded trajectory and the actual trajectory of the satellite with constant tracking activated and all perturbations on. For the following thruster's characteristics: $F_{T_{nom}}$ = 100 N, I_{sp} = 300 s.

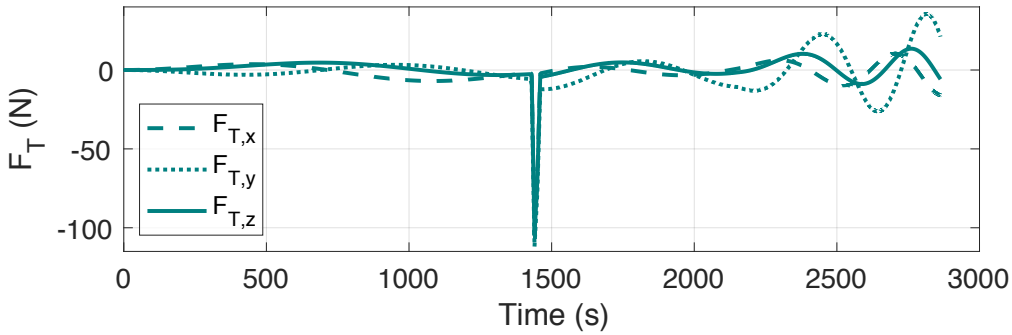

Figure 8.14: Thrust profile of the full guidance sequence (nominal + tracking). For the following thruster's characteristics: $F_{T_{nom}}$ = 100 N, I_{sp} = 300 s.

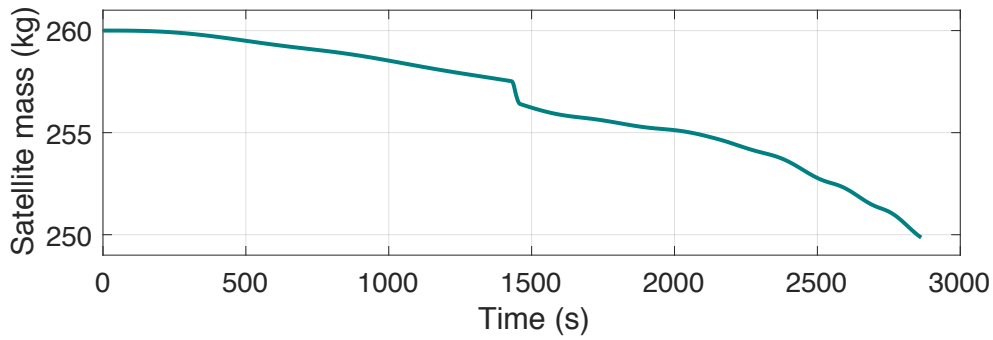

Figure 8.15: Mass consumption a the full guidance sequence (nominal + tracking). For the following thruster's characteristics: $F_{T_{nom}}$ = 100 N, I_{sp} = 300 s.

From this analysis, it can be concluded that maintaining constant tracking guidance, even with the developed adaptive system, is not viable due to excessive fuel mass consumption. Using over 10 kg of fuel per manoeuvre to constantly account for disturbances is impractical. Additionally, this approach is incompatible with most studied thrusting capabilities, as the deviation is too large to be effectively managed. Therefore, based on these findings, it has been decided to rely solely on the nominal profile for guidance. Further recommendations to improve the guidance system will be provided in Chapter 9.

8.2 Attitude Control System Results

The attitude of the satellite is controlled using an LQR controller, as described in details in Chapter 6. As highlighted in this section, the LQR controller can be implemented using quaternions, which offer the advantage of being singularity-free. Alternatively, it can be implemented using Euler angles, which, while potentially leading to singularities, provide a more stable and straightforward approach, since they do not exhibit coupling between the angles, unlike quaternions. This will be demonstrated in this section. The verification of this controller was conducted in Section 7.2.4 for an attitude error of 4° in all directions. In this real-case scenario, it was observed that during nominal thrusting, the larger observed direction angles (ψ and ϵ) could reach up to 90°, significantly exceeding the value for which the LQR was verified. Therefore, further gain tuning of the LQR settings is necessary to efficiently regulate the signal and ensure a stable and smooth control response. Four scenario cases have been outlined for the results of this attitude control sequence: the response to a commanded attitude of 45° for the Euler angles θ and ψ and the actual studied case scenario where these angles are 90° and 9.7°, respectively, for both the Euler and quaternion-based controllers. The results for the quaternion-based controller will be detailed in Section 8.2.1, and those for the Euler-based controller in Section 8.2.2, followed by a discussion on attitude control results and the potential integration of guidance and control in Section 8.2.3.

8.2.1 LQR Quaternion-Based

The quaternion-based LQR controller response to a commanded attitude of 45° for the Euler angles θ and ψ converted to quaternions can be observed in Figure 8.16, with the controller parameters highlighted in Table 8.6. This figure demonstrates that the attitude error for all three quaternion components decreases over time, eventually reaching a stable value of zero. This indicates that the controller effectively guides the satellite to the desired state. The stability of the response further confirms that the satellite's final orientation successfully aligns with the commanded attitude, with a settling time of approximately 70 seconds.

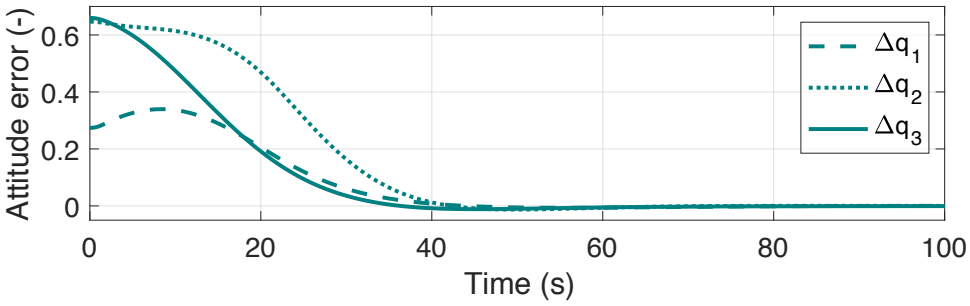

Figure 8.16: LQR controller response, in terms of quaternions, for a commanded attitude change of $\phi = 0°$, $\theta = 45°$, $\psi = 45°$. Settling time ≈ 70 s.

Table 8.6: Tuning parameters for the LQR system in terms of quaternions, for a commanded attitude change of $\phi = 0°$, $\theta = 45°$, $\psi = 45°$.

LQR tuning parameters						
$\Delta\omega_1(°/s)$	$\Delta\omega_2(°/s)$	$\Delta\omega_3(°/s)$	Δq_1	Δq_2	Δq_3	$M_{min/max}(N)$
5	5	0.5	0.2	0.1	0.1	± 10

The response to a commanded attitude of 90° and 9.7° for θ and ψ converted to quaternions is highlighted in Figure 8.17, with the controller parameters displayed in Table 8.7. In addition to the comments made for the previous figure, which also holds for this one, additional observations can be highlighted. From this figure, a rapid convergence can be observed, specifically, the majority of error reduction occurs within the first 20 to 30 seconds of simulation. In addition to this, no overshoot is observed, suggesting the controller to be well-tuned.

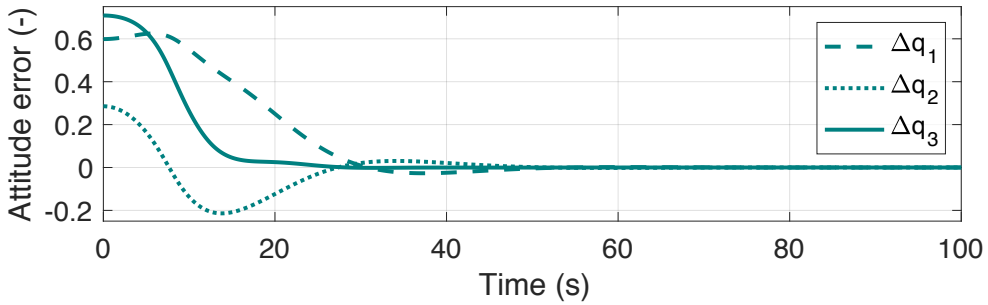

Figure 8.17: LQR controller response, in terms of quaternions, for a commanded attitude change of $\phi = 0°$, $\theta = 9.7°$, $\psi = 90°$. Settling time ≈ 60 s.

Table 8.7: Tuning parameters for the LQR system in terms of quaternions, for a commanded attitude change $\phi = 0°$, $\theta = 9.7°$, $\psi = 90°$.

LQR tuning parameters						
$\Delta\omega_1(°/s)$	$\Delta\omega_2(°/s)$	$\Delta\omega_3(°/s)$	Δq_1	Δq_2	Δq_3	$M_{min/max}(N)$
10	10	20	0.1	0.1	0.1	± 10

Overall, both profiles demonstrate the effectiveness of the quaternion-based LQR controller in achieving precise and stable satellite attitude control. The consistency across different scenarios suggests that the controller is robust, well-tuned, and capable of supporting a wide range of satellite missions.

8.2.2 LQR Euler Angles-Based

The Euler-based LQR controller response to a commanded attitude of 45° for the Euler angles θ and ψ can be observed in Figure 8.18 while the response to a commanded attitude of 90° and 9.7° for θ and ψ is highlighted in Figure 8.19. The controller parameters to obtain such a response have been highlighted in Table 8.8 and Table 8.9, respectively. From both figures, several important conclusions can be drawn. First and foremost, the response profiles demonstrate smooth transitions in the satellite's attitude, characterised by rapid convergence towards the desired orientation. Specifically, the settling time is approximately 50 seconds for Figure 8.18 and around 40 seconds for Figure 8.19. This quick settling time indicates the controller's efficiency in aligning the satellite with the commanded attitude

changes. Moreover, the absence of overshoot suggests that the LQR controller is not only effective but also highly stable, as it drives the system to the target state without introducing unnecessary deviations. Taken together, these observations imply that the LQR controller based on Euler angles is well-suited for precise attitude control.

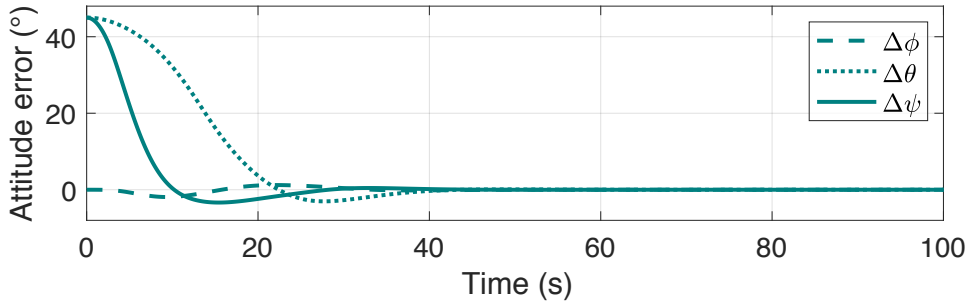

Figure 8.18: LQR controller response, in terms of Euler angles, for a commanded attitude change of $\phi = 0°$, $\theta = 45°$, $\psi = 45°$. Settling time ≈ 50 seconds.

Table 8.8: Tuning parameters for the LQR system in terms of Euler angles, for a commanded attitude change of $\phi = 0°$, $\theta = 45°$, $\psi = 45°$.

LQR tuning parameters						
$\Delta\omega_1(°/s)$	$\Delta\omega_2(°/s)$	$\Delta\omega_3(°/s)$	Δq_1	Δq_2	Δq_3	$M_{min/max}(N)$
5	5	5	0.1	0.2	0.2	± 10

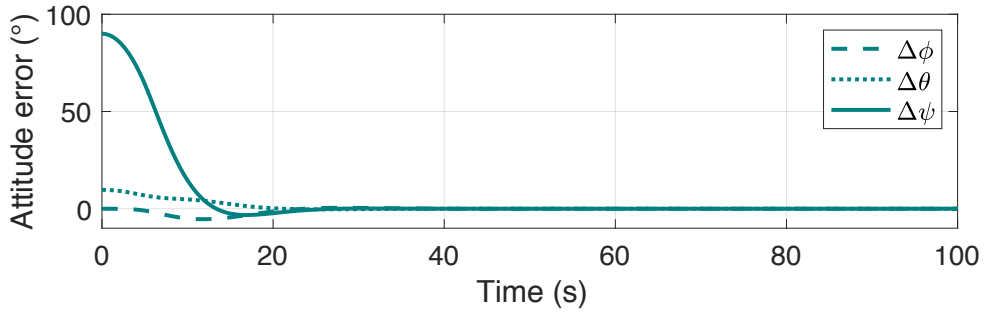

Figure 8.19: LQR controller response, in terms of Euler angles, for a commanded attitude change of $\phi = 0°$, $\theta = 9.7°$, $\psi = 90°$. Settling time ≈ 40 s.

Table 8.9: Tuning parameters for the LQR system in terms of Euler angles, for a commanded attitude change of $\phi = 0°$, $\theta = 9.7°$, $\psi = 90°$.

LQR tuning parameters						
$\Delta\omega_1(°/s)$	$\Delta\omega_2(°/s)$	$\Delta\omega_3(°/s)$	Δq_1	Δq_2	Δq_3	$M_{min/max}(N)$
5	5	20	5	5	20	± 10

8.2.3 Control Sequence Integration Discussion

From the results highlighted in Section 8.2.1 and Section 8.2.2, discrepancies between the profiles are evident. Notably, the Euler angle-based controller exhibits a slightly faster response to the commanded attitude. This could be due to the fine-tuning challenges inherent to quaternion-based control, where changes in a single Euler angle affect multiple

quaternions, leading to more complex behaviour and a less straightforward controllable system. While using an Euler-based LQR controller can lead to issues with singularities, a quaternion-based LQR controller addresses these issues. In this specific case, a controller for local Euler angles has been designed to rotate at the same rate as inertial Euler angles. This design allows the model to control inertial Euler angles effectively. When local Euler angles are used, and the satellite is assumed to be Earth-pointing or consistently oriented with respect to the orbital plane, singularities (which occur for slews above 90°) are avoided, because the local Euler angles remain at 0° as the satellite rotates. Additionally, it can be observed that reorientation takes approximately 70 s with the quaternion-based controller and around 50 s with the Euler angle-based controller. As described in Figure 8.2, the nominal guidance maneuver currently under study ($F_{T_{nom}}$ = 10 N, I_{sp} = 292, *Gurobi* solver) takes approximately 100 s. This highlights the need to decouple the control and guidance sequences for this scenario since a control sequence lasting 70 s out of a 100 s manoeuvre is not viable. Therefore, it has been decided that the satellite should reorient upstream of the nominal guidance sequence.

With the results thoroughly presented in this chapter and a comprehensive analysis provided throughout the report, it is now crucial to draw significant conclusions from the study. The final chapter will consolidate the findings, offering a clear understanding of the study's outcomes and their implications. In addition to summarising the key insights, the discussion will include recommendations for future work, focusing on improving the current approach and providing guidance for researchers who aim to further advance the field of collision avoidance.

CHAPTER

9 Conclusions and Recommendations

This chapter marks the conclusion of this research, summarising the findings and offering recommendations for future work. The overall conclusion of the research will be presented in Section 9.1, addressing each of the research subquestions introduced in Chapter 1. This section will synthesize the insights gained from the various analyses and simulations conducted throughout the study. Following this, several recommendations for future work will be proposed in Section 9.2, aimed at further developing and refining the current research. These recommendations will focus on potential improvements that could be applied to the current work, to further enhance the guidance and control systems explored in this research.

9.1 Conclusions

For the entire duration of the research work, the need for a robust guidance algorithm capable of efficiently executing collision avoidance manoeuvres has been highlighted and an analysis of its design and potential integration with a control system have been performed. With this, throughout the development and efforts performed in this work, the main research questions have been answered.

To what extent would the use of convex optimisation-based guidance increase the robustness, safety and efficiency of an autonomous orbit control for the space debris collision avoidance of mega-constellations?

To properly answer this question in details, it has been broken down into several subquestions, which will each be answered, serving as the main conclusion of this research work. Before delving into the various questions and concluding on the specific elements highlighted in this research, it is essential to present a general conclusion on the research's contributions to the scientific community and, more importantly, to the field of space situational awareness. One of the most significant outcomes of this thesis is the development and analysis of a robust guidance system capable of performing collision avoidance manoeuvres using convex optimisation. The development of this system establishes a foundation for future studies to build upon, enabling the incorporation of additional elements and the further application of convex optimisation in the collision avoidance domain. This research represents a critical step towards enhancing space sustainability and addressing the space debris problem that the space industry currently faces.

The sub-questions that will further be answered and developed are the following.

- *Which environmental models should be included and with which accuracy, to meet the operational and CPU load requirements?*

As demonstrated in Section 3.4, several forces and moments significantly influence the entire satellite's trajectory. Accurately developing and including these models is crucial for the simulation's accuracy, ensuring the satellite's propagation includes maximum accuracy and closely matches real-life scenarios. It has been observed that omitting certain forces from the model can lead to discrepancies in the satellite's trajectory of up to several kilometres, which is unacceptable for collision avoidance scenarios. However, incorporating numerous environmental models also impacts computational efficiency. After evaluating the trade-offs, it was decided to include the following forces and moments in the environment model: Earth's gravity with spherical harmonics up to degree and order 32, Earth's induced gravitational and magnetic torques, Moon's point mass, Sun's point mass, solar radiation pressure and moment, and atmospheric aerodynamic drag and moment.

- *How can a guidance system be modelled and simulated?*

As highlighted in Chapter 7, the entire research work was conducted using MATLAB Simulink®, which facilitated the development of the different sequences and their integration. Indeed, Simulink® is fully integrated with MATLAB®, allowing for easy interaction between simulations and developed scripts. Choosing the Simulink® library also allowed for the use of already developed blocks in GRADS as suggested in Appendix B.1. Additionally, MATLAB® scripts were used to develop the complex algorithms necessary for solving the CAM problem and creating the guidance sequence. As described in Chapter 5, the guidance sequence is divided into two segments: the nominal guidance segment and the tracking guidance segment. The nominal guidance segment was modelled using the principles of convex optimisation and simulated outside of Simulink®. Only the output of this algorithm (thrust commands) was used in the simulator. The second part of the guidance, the tracking segment, was partially simulated in Simulink®, using the estimated state of the satellite and generating a potential thrusting profile for execution. Additionally, MATLAB® scripts were employed to develop the LQR controller used for the tracking sequence.

- *How can a control system be modelled and simulated?*

Similarly, the control sequence was constructed using both MATLAB® and Simulink®. As detailed in Section 7.1, the integration of the control sequence, along with the use of actuators, was carried out within the Simulink® simulator environment. The development and testing of the LQR controller, however, were conducted using MATLAB® scripts. The outputs from these scripts were then fed into the simulator as inputs.

The dual-environment approach that has been used for the overall development of the collision avoidance manoeuvre sequence, allowed for precise modelling in the MATLAB® environment and at a later stage, integration of the models into the simulator. Using this approach allowed for smart testing, refinement, and the use of a more complex algorithm before applying it to the simulator.

- *Which type of guidance system would be the most optimum in terms of robustness, safety, and fuel efficiency?*

The guidance sequence is divided into two parts: a nominal part identifying the optimal trajectory for collision avoidance, and a tracking part ensuring the satellite adheres to this trajectory without deviation. For the nominal guidance, convex optimisation is employed to determine the optimal path given a range of constraints. Known for its robustness, autonomy, and computational efficiency, convex optimisation offers significant advantages for this application. Its deterministic nature ensures that if a feasible solution exists, the algorithm will converge to the optimal solution within a fixed number of iterations. The nominal guidance system has been completely developed and thoroughly tested and showed some convincing results across several key metrics. To start with, in terms of accuracy, the guidance sequence consistently calculated precise trajectories that closely matched the desired collision avoidance miss distance, ensuring the satellite efficiently moved to avoid potential collisions. Then, regarding CPU time, the algorithm demonstrated high computational efficiency, during which the convex optimisation sequence was solved within a reasonable time when selecting an appropriate solver. This makes the system suitable for real-time applications where quick decision-making is essential, ensuring the system meets the resources available onboard satellites. In addition to this, it is worth noting that the guidance system optimised the trajectories with as main objective to minimise fuel consumption. This ensured successful collision avoidance while meeting the main objective, and equally, enhanced the overall sustainability of the mission. Following this, by making use of convex optimisation, the reliability of the system was found to be excellent. As stated earlier, the deterministic nature of convex optimisation guarantees that the algorithm will find an optimal solution if one exists, ensuring consistent and dependable performance, which was found to be the case during the different testing performed. This is a crucial point to highlight in such a mission where unexpected deviations can happen. To finish with, it was found that the guidance sequence was a highly adaptable one, that could be tailored to a wide range of mission scenarios. This flexibility makes it a valuable tool for such applications in the industry where every mission is different. Overall, the developed guidance system represents a significant advancement in satellite collision avoidance technology, which could enhance space situational awareness and ensure the long-term sustainability of space operations.

- *What are the limitations of the developed guidance system on the satellite's propulsion capabilities?*

As highlighted in Chapter 8, several propulsion systems available on the market have been tested on the capability of the nominal guidance, to make sure such guidance sequence could be suitable for any manoeuvrable satellites. From this analysis, it was concluded that no real limitations in terms of the propulsion system were found. The most efficient nominal guidance sequence for this precise half-orbit CAM scenario was performed using low-thrust propulsion and more precisely hall-thrusters with I_{sp} = 2,000 s and $F_{T_{nom}}$ = 0.01 N, leading to mass consumption of only a few grams. However, it has been found that some limitations existed in the tracking guidance sequence, for which most propulsion means were efficient enough to counteract the deviation observed between the actual and nominal trajectories. For most analysed cases, the available thrust was not large enough and led to saturation and not meeting the set deviation threshold on state deviation.

- *Which objective function, state, and control constraints should be set for this specific mission?*

To successfully develop the guidance sequence using convex optimisation, various constraints and objective functions must be defined to ensure the algorithm converges to a robust and optimal solution. The primary objectives of the manoeuvre are to be fuel-optimal, safety-optimal, time-optimal, and have minimal environmental impact. To achieve these objectives, several mission constraints are imposed. The miss distance between the satellite and debris must remain above a 10 km threshold, which is translated into a keep-out sphere constraint. The thrust levels are limited by the satellite's available nominal thrust, and fuel consumption must be minimised.

- *What are the necessary steps included in a conjunction analysis?*

While not the primary focus of this research, a basic conjunction analysis was conducted to underscore the importance of such evaluations, to outline the necessary steps to identify a potential conjunction and address a real-life scenario in this research work. This study has shown that the key stages of conjunction analysis begin with an initial screening of the environment which enables the identification of potential threats. This is followed by a filtering process to exclude object pairs that are unlikely to result in a conjunction. Furthermore, defining critical parameters such as miss distance, collision probability, and other relevant factors is essential for a thorough analysis. Collectively, these steps establish the foundation of a robust conjunction analysis.

- *Would the algorithm be capable of redirecting the satellite in case divergence from the optimal path is observed?*

Such an algorithm, referred to as tracking guidance, has been developed and tested across a range of thrusting capabilities and settings. It operates on the principle of correcting the satellite's state whenever an unacceptable discrepancy between the estimated trajectory and the desired (nominal) trajectory is detected. An adaptive LQR controller is used for this state control sequence. It is termed 'adaptive' because the LQR parameters adjust based on the satellite's manoeuvring state to meet its specific control needs. While the algorithm was successfully developed, it was found to be impractical for several reasons. The constant tracking required by the satellite demands a significant amount of fuel, making it inefficient. Additionally, this approach is only viable for specific types of propulsion systems, limiting its general applicability.

To conclude, the work conducted in this thesis not only advances the theoretical framework for collision avoidance but also provides practical tools and methodologies that can be directly applied to real-world scenarios. By integrating convex optimisation techniques into the guidance system, the research offers an efficient and reliable approach to manoeuvre planning, significantly contributing to the ongoing efforts to maintain a sustainable and safe space environment. This progress is essential as the space industry continues to grow and the challenges of space debris become more pressing.

9.2 Recommendations

While the current research has advanced the research on collision avoidance and explored the potential benefits of convex optimisation in these systems, there remains room for further improvement and investigation. To guide future work and build upon these findings, several recommendations can be formulated. They will be presented below, classified into different categories to provide a structured approach for ongoing research efforts. To start with, some recommendations related to the overall mission set-up and characteristics will be introduced in Section 9.2.1. This will be followed by recommendations on the guidance, control, and potential navigation sequence, in Section 9.2.2. To finish with, additional recommendations not belonging to specific categories will be introduced on Section 9.2.3.

9.2.1 Mission Recommendations

The recommendations linked with the overall mission will be described in this section. These include the incorporation of additional debris in the analysis, the constellation performance but also elements linked with the overall manoeuvring sequence.

Debris Number

The current analysis focuses on a specific scenario involving the potential collision between a Starlink satellite and a single piece of debris. Although the developed algorithm allows for modifications of the satellite's characteristics, it has been designed based on the analysis of an event considering only one miss distance threshold. This limitation means that the algorithm does not account for scenarios where multiple pieces of debris might simultaneously pose a collision risk during the same segment of the satellite's trajectory. To more accurately represent the complexities of the space environment, future implementations should incorporate the capability to analyse and account for multiple debris conjunctions. Integrating this feature into the current algorithm would further enhance its robustness and applicability in real-world scenarios.

Constellation Performance

An important aspect of the mission that has yet to be explored is the potential impact of such manoeuvres on the overall constellation and more precisely on its performance. While manoeuvres are necessary for maintaining operational safety, they are not without consequences on mission performance. Companies typically aim to minimise these manoeuvres and execute them as efficiently as possible to avoid unnecessary disruptions. One recommendation for future research is to conduct a comprehensive study on how the constellation's performance is affected during and after a manoeuvre. This analysis could be crucial for refining the GNC sequence by integrating constellation performance as a key input parameter. Such an investigation would require examining a range of parameters that influence the constellation's functionality, potentially extending the scope of the current study into an entire thesis focused on optimising manoeuvre strategies for satellite constellations. This approach could lead to more informed decisions that balance safety and mission objectives, ultimately improving the overall effectiveness of the constellation.

Manoeuvring Sequence

The current mission scenario examines the manoeuvring of a satellite, one orbit before the conjunction, as well as the effects of initiating the manoeuvre at a later stage. However, the analysis does not consider the impact of starting the manoeuvre several orbits earlier. Initiating the manoeuvre earlier might result in lower fuel consumption but could also lead to less precise adjustments due to the increased propagation of uncertainties over a longer period. Investigating this aspect would provide valuable insights into the trade-offs between fuel efficiency and manoeuvre precision, making it a worthwhile addition to the research. Another option worth exploring is the possibility of performing multiple manoeuvres (multiple smaller ΔV) instead of just one. This scenario could be investigated further to determine its potential benefits and implications.

9.2.2 GNC Sequence Recommendations

This section presents recommendations related to the guidance, navigation, and control sequences. These recommendations include the potential incorporation of a navigation sequence, the integration of the later with the control and guidance sequence to get a fully integrated GNC system, the use of an improved controller and actuators, and the enhancement of the tracking guidance sequence, which demonstrated limitations in the analysis.

Guidance Sequence

The current guidance sequence includes an element that has been rigorously tested, yet it remains somewhat puzzling from a physical standpoint. During the manoeuvres, every solver, simulator, and algorithm tested consistently produced the same thrust profile: identical thrust levels in the X, Y, and Z directions. This uniformity is unexpected, as one would anticipate variations in thrust depending on the specifics of the manoeuvre. Despite extensive efforts to identify the cause of this behavior, no definitive explanation has been found. It is therefore recommended to conduct further investigation and additional testing to better understand this.

Navigation Sequence

This study did not delve into the navigation sequence or its inclusion. Consequently, the satellite's estimated state was assumed to match the actual state, making the navigation system ideal. However, for comprehensive GNC development, incorporating a navigation sequence would be a valuable recommendation for further enhancement. This addition would provide a more realistic representation of the system's operation and improve its overall performance and reliability.

Integrated GNC Sequence

With the proposed addition of a navigation sequence as mentioned earlier, another area of focus could be the development of a fully integrated GNC system. This could result in a fully autonomous system capable of avoiding debris with precision and efficiency.

LQR Controller

One conclusion drawn from the attitude control analysis was that integrating it with the nominal guidance sequence was not feasible due to the time required for the implemented

controller system to produce a stable signal. An improvement suggested in Chapter 6 is to use a more sophisticated controller, capable of greater robustness. Controllers such as NDI or INDI are designed to handle nonlinear systems with unknown dynamics or uncertainties, which aligns well with the requirements of this specific application. Exploring and implementing such a control system demands additional effort and time; however, it holds the potential to be a valuable addition, facilitating the successful integration of control with the guidance sequence.

Actuators

The attitude controllers used in this study were assumed to be ideal, without any noise or limitations. However, such perfect actuators do not exist in reality, and their imperfections must be taken into account when designing a comprehensive system. Therefore, one recommendation is to implement models of non-ideal actuators and include them in this analysis. This would provide a more accurate representation of real-world conditions, allowing for better assessment and enhancement of the system's performance.

Tracking Guidance Sequence

The current tracking guidance sequence has shown limitations in terms of propulsion requirements and overall feasibility, as the mass consumption for tracking was excessively high. Based on these findings, three recommendations are proposed to improve the system. To start with, the tracking guidance parameters could be adjusted to better suit the specific problem at hand. This would involve making the parameters problem-dependent, allowing for more tailored and efficient tracking. The second recommendation is to further develop the nominal guidance system. Currently, it uses a linearised state-space model that only includes the central field model. By incorporating additional effects, such as the J2 perturbation, the nominal guidance would more accurately reflect real-world conditions. This improvement would reduce the reliance on tracking guidance, as deviations from the real-life trajectory would be minimised. The last recommendation would be, similarly to the attitude control sequence, to use a more sophisticated controller system, such as the INDI controller. This could enhance the robustness of the controller, making it better equipped to handle the nonlinearities and uncertainties inherent in the system. By implementing these recommendations, the overall efficiency and effectiveness of the tracking guidance system could be significantly improved, leading to a more viable and practical solution.

9.2.3 Additional Recommendations

This section will present any additional recommendations, including those related to the post-manoeuvring phase and further development of the conjunction analysis by incorporating additional elements.

Post-Manoeuvre Phase

Currently, the mission scenario only analyses the collision avoidance manoeuvre and concludes once debris avoidance has been achieved. In a real-world scenario, most companies would want their vehicle to return to its initial orbit to continue the mission that was interrupted. Therefore, studying the post-manoeuvre phase is recommended. This study could

explore various scenarios, such as continuing on the new orbit, returning to the initial orbit, or moving to a different orbit. Each scenario's associated risks and benefits should be analysed. Additionally, the potential to use the developed algorithm twice – once for the avoidance manoeuvre and once for returning to the initial orbit – should be examined to understand its impact on the mission's overall characteristics, such as fuel consumption, mission duration, and risk of subsequent collisions.

Collision Probability

The current conjunction analysis is primarily focused on finding a conjunction event solely based on the miss distance. However, for a more comprehensive analysis, it is essential to consider additional factors, such as the collision probability, which significantly influences the outcome of such evaluations. In real-world scenarios, organisations like LeoLabs and Space-Track routinely incorporate collision probability in their tracking and analysis to identify and flag potential conjunctions. These entities continuously monitor and observe space objects to ensure accurate assessments of collision risks, demonstrating the importance of including collision probability in conjunction with analyses. Integrating this factor would align the research more closely with real-world practices and provide a more robust framework for evaluating potential collision scenarios.

APPENDIX

A Numerical Methods

Some numerical methods, used to approximate solutions to problems that cannot be solved analytically are used throughout the thesis. Some of the methods used are the principles of linearisation (Appendix A.1), discretisation (Appendix A.2), numerical integration (Appendix A.3), numerical propagation (Appendix A.4) as well as (Appendix A.5). The specifics behind each of these methods and how they are used in this context will be introduced in the following sections. In addition to these, an introduction of the different solvers used for optimisation will be given in Appendix A.6. This will include the theory behind the *Gurobi*, *fmincon* and *IPOPT* solvers, which have thoroughly been used in the research work.

A.1 Linearisation

Throughout the thesis work, the need for linearisation will be present. An example of such is during the optimisation process, where linearisation was used to construct the state-space equations for nonlinear dynamical systems. The concept behind linearisation and the steps to be performed to obtain linearised functions will be detailed in this section. A function can be evaluated near a specific point ($x = x_0$), using the Taylor series expansion of an infinitely differentiable function.

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{x}) &= \sum_{n=0}^{\infty} \frac{\boldsymbol{f}^{(n)}(\boldsymbol{x_0})}{n!}(\boldsymbol{x}-\boldsymbol{x_0})^n \\ &\approx \boldsymbol{f}(\boldsymbol{x_0}) + \left.\frac{d\boldsymbol{f}}{d\boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x_0}} (\boldsymbol{x}-\boldsymbol{x_0}) + \frac{1}{2!}\left.\frac{d^2\boldsymbol{f}}{d\boldsymbol{x}^2}\right|_{\boldsymbol{x}=\boldsymbol{x_0}} (\boldsymbol{x}-\boldsymbol{x_0})^2 + \frac{1}{3!}\left.\frac{d^3\boldsymbol{f}}{d\boldsymbol{x}^3}\right|_{\boldsymbol{x}=\boldsymbol{x_0}} (\boldsymbol{x}-\boldsymbol{x_0})^3 + \dots \end{aligned} \tag{A.1}$$

In this case, a first-order linear approximation is adequate for capturing the local behaviour of the system at this point. In this equation, $f(x_0)$ represents the function value at the point of study and the remaining depicts the contribution of the tangent line's slope.

$$\boldsymbol{f}(\boldsymbol{x}) \approx \boldsymbol{f}(\boldsymbol{x_0}) + \frac{1}{2!}\left.\frac{d^2\boldsymbol{f}}{d\boldsymbol{x}^2}\right|_{\boldsymbol{x}=\boldsymbol{x_0}} (\boldsymbol{x}-\boldsymbol{x_0}) \tag{A.2}$$

A.2 Discretisation

Discretisation of the dynamics is an important step on the convexification process described in Section 5.4.2. Different types of discretisation processes exist, each with advantages and disadvantages. In this specific work, the discretisation process is carried out using the *c2d*[1],

[1] Continuous to discrete-time conversion, MATLAB®, https://nl.mathworks.com/help/ident/ref/dynamicsystem.c2d.html [Last visited on 28/07/2024].

in MATLAB®. This process of converting from continuous to discrete time relies on methods such as ZOH, to be executed. According to Antsaklis and Michel (2006), the discretised $\boldsymbol{A_d}$ (non-singular) and $\boldsymbol{B_d}$ matrices can be described as follows, leading to the state-space system definition as a linear discrete system, following the ZOH discretisation.

$$\boldsymbol{u_k} = \boldsymbol{u}(t) = \begin{cases} 1 & \text{if } 0 \leq t < \Delta T \\ 0 & \text{otherwise} \end{cases} \tag{A.3}$$

$$\boldsymbol{A_d} = e^{\boldsymbol{A_c}\Delta t} \qquad \boldsymbol{B_d} = \int_0^{\Delta t} e^{\boldsymbol{A_c} s} \boldsymbol{B_c}\, ds \tag{A.4}$$

$$\boldsymbol{x_{k+1}} = \boldsymbol{A_d}\boldsymbol{x_k} + \boldsymbol{B_d}\boldsymbol{u_k} \tag{A.5}$$

In addition to the $\boldsymbol{A_d}$ and $\boldsymbol{B_d}$. The construction of the output equation Equation (A.6) is required, which is non-dynamical, with $\boldsymbol{C_d}$ the output matrix, and $\boldsymbol{D_d}$ the direct transition matrix remaining unchanged and defined by:

$$\boldsymbol{y_{k+1}} = \boldsymbol{C_d}\boldsymbol{x_k} + \boldsymbol{D_d}\boldsymbol{u_k} \tag{A.6}$$

$$\boldsymbol{C_c} = \boldsymbol{C_d} = \boldsymbol{I}_{7\times 7} \qquad \boldsymbol{D_c} = \boldsymbol{D_d} = \boldsymbol{0}_{7\times 6} \tag{A.7}$$

A.3 Numerical Integration

Integration methods are essential tools in the guidance and control of a space vehicle. The motion of the satellite is described by a set of differential equations (Section 3.6), numerical integration algorithms are used to solve these non-linear differential equations and thus predict the future state of the satellite and simulate its behaviour over time. Usually, the choice of an integrator is problem-dependent with variations in the integrator's stability, accuracy and speed. As was introduced earlier, there exist various categories of ordinary differential equations (ODEs) in the field of mathematics, which can be solved analytically resulting in a precise solution. For example, a first-order ODE is defined as observed in Equation (A.8), with $\boldsymbol{x}$ the differentiable function, $\dot{\boldsymbol{x}}$ the derivative function with respect to the independent variable, f the function, t, the independent variable (which can be different from time):

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}) \tag{A.8}$$

There exists an infinite number of distinct functions $\boldsymbol{f(x)}$ that can satisfy this equation. In reality, the majority of differential equations do not conform to a standard form and cannot be solved using analytic techniques. As a result, it is not possible to obtain a general solution. This applies to the current mission, which is described not by a single differential equation but rather by a set of equations. In order to find the solution to such a problem, numerical methods should be used. The solution is thus estimated by dividing the interval of interest into a discrete set of points and then using numerical algorithms to approximate the solution at each point, leading to an approximation of the solution, with its accuracy depending on the method used and the step size. The different relevant families of integrators used in this simulation, mainly Euler and Runge-Kutta 4, and their respective characteristics will be introduced in the upcoming subsections.

A.3.1 Euler Integration

The Euler integration method, sometimes referred to as the polygonal integration method, is a straightforward technique for approximating solutions to systems of differential equations. This method is considered to be one of the simplest numerical integration methods available. It is a first-order numerical technique, which means that the local error is approximately proportional to the square of the step size (Δt^2). One of the advantages of the Euler method is that it is an explicit integration method, meaning that the state of a system at a later time ($t+1$) can be calculated from the current state of the system (t) without the need to solve additional equations. However, it should be noted that the accuracy of the Euler method can decrease rapidly if the solutions change rapidly. The forward Euler method computes $\boldsymbol{x_{n+1}}$ as follows, with Δt the step size.

$$\boldsymbol{x_{n+1}} = \boldsymbol{x_n} + \Delta t \cdot \boldsymbol{f}(\boldsymbol{x_n}, t_n) \tag{A.9}$$

which is based on a truncated Taylor series expansion, expanding in the neighbourhood of $t = t_n$, as described by Zeltkevic (1998):

$$\boldsymbol{x}(\boldsymbol{t}_n + \Delta t) \equiv \boldsymbol{x}_{n+1} = \boldsymbol{x}(t_n) + \Delta t \frac{d\boldsymbol{x}}{d\boldsymbol{t}}\bigg|_{\boldsymbol{t}_n} + \mathcal{O}\left(\Delta t^2\right) = \boldsymbol{x}_n + \Delta t \cdot \boldsymbol{f}(\boldsymbol{x}_n, t_n) + \mathcal{O}\left(\Delta t^2\right) \tag{A.10}$$

Despite its limitations, the Euler method remains a useful tool for approximating solutions to a variety of problems. It is particularly effective when applied to systems that exhibit relatively smooth behaviour. In general, the Euler method can be used to provide a good approximation of the behaviour of a system over a short period of time, making it an attractive option for many practical applications.

A.3.2 Runge-Kutta 4 Integration

The family of Runge-Kutta integration method are a class of numerical techniques used for solving ODEs. They are related to Euler's method but they are more accurate and efficient. Indeed, they do not require excessive computational time and are relatively easy to implement. In the Runge-Kutta family, the most popular method is the Runge-Kutta 4 (RK4) (fourth order), which has an LTE of order Δt^5. This method looks at four different points in each step and uses this information to calculate a better guess for the next step. This process is then repeated over again.

$$\boldsymbol{x_{n+1}} = \boldsymbol{x_n} + \left(\frac{\boldsymbol{k_1}}{6} + \frac{\boldsymbol{k_2}}{3} + \frac{\boldsymbol{k_3}}{3} + \frac{\boldsymbol{k_4}}{6}\right)\Delta t + \mathcal{O}(\Delta t^5) \tag{A.11}$$

with Δt the time step, t_n and $\boldsymbol{x_n}$ the time and state at current time step (n). Additionally, $\boldsymbol{k_1}$, $\boldsymbol{k_2}$, $\boldsymbol{k_3}$ and $\boldsymbol{k_4}$ represent the time-derivative vectors at various points within the integration step (the slopes). Then, all estimates of the slope in a weighted average will be used to generate the final estimate. The midpoint slope estimates are typically given twice as much weight as the endpoint estimates as observed hereafter.

$$\begin{aligned}
\boldsymbol{k_1} &= \cdot \boldsymbol{f}(\boldsymbol{x_n}, t_n), \\
\boldsymbol{k_2} &= \cdot \boldsymbol{f}\left(\boldsymbol{x_n} + \frac{\boldsymbol{k_1}}{2}, t_n + \frac{\Delta t}{2}\right), \\
\boldsymbol{k_3} &= \cdot \boldsymbol{f}\left(\boldsymbol{x_n} + \frac{\boldsymbol{k_2}}{2}, t_n + \frac{\Delta t}{2}\right), \\
\boldsymbol{k_4} &= \cdot \boldsymbol{f}\left(\boldsymbol{x_n} + \boldsymbol{k_3}, t_n + \Delta t\right).
\end{aligned} \tag{A.12}$$

As stated earlier, RK4 is a very popular integration method since it is easy to implement and is already readily available in most software due to its popularity. RK4 may however not be suitable to solve complex systems that require higher-order methods for more accuracy. It can also be less accurate than higher-order methods like RK4(5), which can provide a higher level of accuracy with similar computational efficiency. For the propagation of dynamics in Simulink®, an RK4 integration with fixed step size has been used. This allows the propagation distance to be divided into equal time steps, having the advantage to simplify the post processing of the data. But also comes at the cost of efficiency and CPU time increase.

A.4 Numerical Propagation

For verification purposes or when dealing with simplified models, solving Kepler's equation is sufficient to allow the propagation of objects in time. However, when dealing with perturbations and when including all relevant perturbations introduced in Section 3.4, a more complex propagator is necessary to solve such an initial value problem. In the current work, two propagators have been used for distinct purposes, Cowell's method and the Simplified General Perturbations 4 (SGP4) model. Both methods will be introduced hereafter.

A.4.1 Cowell's Method

For the current research, when propagating the dynamics in time in the Simulink® model, a more sophisticated propagation method is required. A popular method, known as Cowell's formulation has been used for this purpose. Traditionally, it is applied in a Cartesian coordinate system, as in Equation (A.13). But it can also be written using spherical coordinates.

$$\frac{d^2\boldsymbol{r}}{dt^2} = -\frac{\mu}{r^3}\boldsymbol{r} + \boldsymbol{a_{tot}} \tag{A.13}$$

This approach offers significant benefits primarily due to its straightforward formulations, making it simple to program and solve. However, it also comes with some drawbacks, which mainly happen when significantly large forces are taken into account. This is for example the case when near a large perturbing body. The need for smaller time steps is then necessary, leading to larger computation times as well as steadily growing numerical integration errors. It has been found that Cowell's method, even if simpler, can be 10 times more efficient than Encke's method, another popular propagation method (Bate et al., 1971).

A.4.2 SGP4 Method

The SGP4 algorithm has been used, as described in Chapter 4, to perform the conjunction analysis. It has mainly been used to match with the environment and technology used by Space-Track. The SGP4 propagator has been directly downloaded from Celestrak[2] as an open source, fully compatible with MATLAB®. It was mainly employed to predict the positions and velocities of each Starlink satellite and debris object at discrete intervals. The currently used and updated SGP4 propagation is one programmed by David Vallado, which relies on TLE sequences provided by the U.S. Government for several decades, and which can be retrieved as open sources as has been described in Section 4.3.1. More information on the SGP4 can be found in Vallado and Crawford (2008).

[2]Celestrak, *Satellite Tracking Software* , https://celestrak.org/software/tskelso-sw.php,[Last visited on 10/05/2024].

A.5 Interpolation

The principle of interpolation was applied in this research, particularly in the development of the guidance system. For instance, during the testing phase, simulations were conducted to assess the feasibility of interpolating various true anomaly segments, the interpolation was performed using the "1D Lookup Table[3]", which works on the principle of generating an output signal by interpolating the table values. As the default interpolation method between break-point values, it makes of linear point-slope interpolation, whose principle will be introduced hereafter. A similar process is performed when using the "From Workspace[4]" block, which is used to generate output values for simulation times that fall between the sample times in the loaded workspace data. This block performs linear interpolation using two adjacent points from the workspace data to determine the output value. Linear interpolation is a numerical method used to approximate the values of a function between any two known discrete data points. The mathematical formulation to approximate an unknown value using two of the closest values and thus using 1D linear interpolation is as follows:

$$f(x) = f(x_1) + \frac{f(x_2) - f(x_1)}{x_2 - x_1}(x - x_1) \tag{A.14}$$

in which $f(x)$ is the unknown value we want to approximate, $f(x_1)$ and $f(x_2)$ are the known value surrounding $f(x)$ with corresponding values x_1 and x_2.

This method is the least accurate linearisation method but it has the advantage of being a fast method, which can sometimes be accurate enough when a large number of data are available or when the function is quasi-linear. Its main disadvantage comes from the fact that the first derivative is usually discontinuous at nodal points. It is a great numerical integration method to use when working with a database where a large number of tabulated data is available. However, when no such large amount of data is available and where a small discontinuity will lead to a large jump in the available data, another more accurate type of interpolation could be required. These types of interpolation methods are able to generate smooth functions, which allow for differentiation at the control points.

A.6 Convex Optimisation Solvers

As described in Section 5.4.3, three different solver algorithms are relevant and will be tested for in this research work. Each has different characteristics and works on different principles, their specifics will be introduced in this section. To start with, the details behind the *Gurobi* solver will be introduced in Appendix A.6.1. This will be followed by the specifics of the interior point optimiser (*IPOPT*) in Appendix A.6.2. To finish with, the characteristics of the *fmincon* solver will be introduced in Appendix A.6.3.

A.6.1 Gurobi

The *Gurobi* solver can mathematically solve a large range of optimisation problem types. It does so by employing different algorithm specifics to the problem at hand. For example,

[3] MathWorks, *1-D Lookup Table*, https://www.mathworks.com/help/simulink/slref/1dlookuptable.html, [Last visited on 17/04/2024].

[4] MathWorks, *From Workspace*, https://nl.mathworks.com/help/simulink/slref/fromworkspace.html, [Last visited on 17/04/2024].

it can deal with linear programming problems, quadratic programming as well as mixed-integer programming (linear and quadratic). The available algorithms used by *Gurobi* are the simplex algorithm, the interior-point algorithm or the branch-and-bound algorithm, their usage is dependent on the problem's nature.

Simplex algorithm: One of the pivotal breakthroughs in linear programming came with the development of the simplex method by George Dantzig. This algorithm offers a structured methodology for identifying the optimal solution to linear programming problems. It offers a strategy for addressing linear programming by incrementally improving enhancements of the objective function value. This is performed by converting the problem into standard form and subsequent expression in canonical form. From this, the basic feasible solution can be discerned and the objective function can be refined for optimisation. This method enables the facilitation of optimisation across diverse applications. More details on the theory of the simplex algorithm can be found in the work of Nash (2000).

Interior-point algorithm: The primary objective of the interior-point algorithm is to locate the optimal solution of the optimisation problem by traversing from one point along the objective function to another within the interior of the feasible region. This is typically done through using a two-phase approach. The initial phase is dedicated to establishing a feasible solution, while the subsequent phase focuses on refining this solution towards optimality. This dual-phase strategy ensures systematic and efficient optimisation, making this method an efficient one in optimisation strategy. More details on the theory of interior-point algorithm can be found in the work of Nemirovski and Todd (2008).

Branch-and-bound algorithm: This method involves dividing the main problem into smaller, manageable sub-problems. This process is known as branching. Each of these sub-problems is then solved, striving for the optimal solution. The key feature of this method lies in its use of bounds, which act as guiding constraints to eliminate the necessity of exploring sub-optimal solutions. By systematically navigating through the problem space, this method efficiently refines its search, ultimately converging towards the optimal solution. More details on the theory of branch-and-bound algorithm can be found in the work of Guida (2014).

A.6.2 Interior Point Optimiser (IPOPT)

IPOPT is a software package that can be used to solve general non-linear programming problems, defined as follows:

$$\begin{aligned} \text{Minimise:} \quad & \min_{x \in \mathbb{R}^n} f(x) \\ \text{subject to:} \quad & g^L \leq g(x) \leq g^U \\ & x^L \leq x \leq x^U \end{aligned} \tag{A.15}$$

It is a freely available open-source package, specifically crafted to discover local solutions for optimisation challenges. It operates on the principle of employing an interior-point line-search filter method, extensively detailed in the documentation by Wächter and Biegler (2006). The line-search filter method interprets the nonlinear function as a bi-objective optimisation problem, aiming to minimise both the objective function and constraint violations. According to this approach, a trial point is evaluated and accepted if it demonstrates sufficient progress towards either objective compared to the preceding iteration.

A.6.3 Fmincon

Fmincon is a non-linearly constrained multi-variable optimisation solver. This solver finds the minimum of a problem, specified as follows:

$$\min_x f(x) \text{ such that } \begin{cases} c(x) \leq 0 \\ c_{eq}(x) = 0 \\ A \cdot x \leq b \\ A_{eq} \cdot x = b_{eq} \\ lb \leq x \leq ub \end{cases} \tag{A.16}$$

in which, b and b_{eq} are vectors, A and A_{eq} are matrices, $c(x)$ and $c_{eq}(x)$ are functions that return vectors, and $f(x)$ is a function that returns a scalar. $f(x)$, $c(x)$, and $c_{eq}(x)$ can be non-linear functions. In MATLAB®, the *fmincon* algorithm is used. One that finds the minimum for the generated problem. The algorithm objective changes depending on the nature of the constraint. In this specific convex guidance case, there exists both linear inequality and equality constraints. With such nature of the problem, the *fmincon* algorithm will look for the minimum value of Rosenbrock's function:

$$f(x,y) = (a-x)^2 + b(y-x^2)^2 \tag{A.17}$$

This function usually serves as a benchmark for testing and comparing the performance of optimisation algorithms. This is particularly useful to solve for nonlinear and non-convex optimisation problems. This function is a standard test function being used in optimisation. It has a unique minimum value of 0 at the point [1,1]. Finding its minimum is the goal of most algorithms, but it can be a challenge for some since the function has a shallow minimum inside a deeply curved valley (Picheny et al., 2012).

APPENDIX

B Simulation User Guide

This chapter serves as a user guide for anyone wanting to access the programming files for further research. This user guide will allow the reader to further understand the MATLAB® GRADS Library (Appendix B.1) as well as how the YALMIP toolbox (Appendix B.2) is constructed. In addition to this, an overview of the different programming files and how they have been stored will also be introduced in Appendix B.3.

B.1 MATLAB® – GRADS Library

The development philosophy behind the GRADS library prioritises creating functional blocks that are not only used to build a simulator but are also easy to understand and verify. As a result, many of the new blocks are implemented as MATLAB® functions. The GRADS library is divided into a number of top-level blocks; *Utilities*, *Propagators*, *Environment*, *Sensors*, *Actuators*, *H/W Interfaces* and *GNC*. Each top-level block will be detailed in this section with the information provided directly taken from Mooij (2021), in which additional details can be found if required.

Utilities: This library contains a range of resources for working with transformations between matrices, time frames, and other related concepts. It is divided into several sub-blocks, including *Transformation Matrices*, *Time Utilities*, *Coordinate Transformations*, and *Mathematics*. Each sub-block delves deeper into the subject matter and includes its own set of sub-subgroups.

Propagators: This library contains a number of propagators to be used depending on the requirements and characteristics of the mission at hand. Some of the currently available propagators are the following: *Rigid-Body Cowell (3 DOF translational), Rigid-Body Cowell (3 DOF translational) External initialisation, Variable Mass Cowell (3 DOF translational), Orbital Propagator (Central Gravity Field), Orbital Propagator With Thrust (Central Gravity Field), Rotational Dynamics (Euler Equation)*, and more.

Environment: This library includes a number of forces and moments models to represent the current space environment. This library has been developed for space-centred missions and thus the environment models are all Earth-based, except for the Sun and Moon ephemerides models. Some of the currently available physics models, forces and moments are the following: *Central Magnetic Field, Sun (and Moon) Position in ECI (J2000) coordinates, Gravity-Field Force and Moment, Third-Body Force and Moment*, and more.

Sensors: A number of sensors are also simulated within the GRADS library. Some of them are the *Fine Sun Sensor*, *Star tracker*, *Three-channel gyroscope*, and more.

Actuators: A number of actuators are also simulated within the GRADS library. Some of them are a *Magnetorquer*, *Reaction Wheels Simple*, *Single Thruster*, and more.

B.2 MATLAB® – YALMIP Toolbox

YALMIP is a a toolbox used for modeling and optimisation in MATLAB®. Such toolbox can be easily installed through the following link: https://github.com/yalmip/yalmip/archive/master.zip. It has been used as a tool to define the guidance optimisation problem. The toolbox has the advantage of allowing the definition of different types of programs, including linear, quadratic, semi-definite, and nonlinear. The toolbox has been useful to use in the problem definition since it can handle various problem complexities, including non-convex ones, as was the case for the initial problem at hand. In addition to this, multiple constraints and objectives can be input making it a versatile tool to work with. In addition to the optimisation toolbox, solvers are required for optimisation purposes. These are not included into the YALMIP environment, should be carefully selected and installed. More information on these solvers has been introduced in Section 5.4.3.

B.3 Programming Files Overview

This section will introduced an overview of the different programming files, allowing for easier reproduction of the research work. Each folder comes with subfolders containing a *"Scripts"*, a *"Figures"* and where applicable a *"Data"* folder. To start with, the *"Scripts"* folder contains all the MATLAB® and Simulink® codes. The *"Data"* folder contains all the ".mat" or ".txt" files that are necessary for the correct execution of the programming files. And to finish with, the *"Figures"* folder contains all generated figures from the scripts. There are many scripts that have been created for different purposes. Only the main ones, encompassing the most important elements of the thesis work will be discussed hereafter.

B.3.1 Initialisation

Several initialisation files are required to run any of the script created in this thesis work. It has been decided to group them in one folder for ease of use and to easily add them to the programming path all at once. The *"1.0 | Initialisation Files"* folder contains all the necessary scripts for such purpose. As described in Appendix B.1 and Appendix B.2, both YALMIP and GRADS environments are necessary throughout the simulations, these can be found in the initialisation folder. Then, *"Run.m"* encompasses all initialisation files containing information on the objects and environment to be simulated. In addition to these, some other files with useful formulas or post-processing codes can also be found in this folder. It is one of the most used folder since necessary to run before any simulation.

B.3.2 Environment

The *"2.0 | Environment"* folder encompasses all files used to build and test the environment that will be used for every simulation throughout the study. This analysis has been performed to simulate and analyse the different forces and moments acting on a Starlink satel-

lite, located at approximately 550 km altitude. The main Simulink® file developed for that matter is *"EnvironmentTesting.slx"*. In the latter, the state of a satellite is inputted in the simulation, it is then subjected to different forces and moment and the outcome of the analysis is then saved in the *"DataEnvironmentTesting_10000.mat"* for further post processing, analysis, and plotting as performed in *"EnvironmentTestingScript.m"*. The outcome of the analysis are grouped in the *"Figures"* folder.

B.3.3 Conjunctions Analysis

The entire conjunction analysis is contained in the *"3.0|Conjunction Analysis"* and divided into three subfolders. To start with, a data processing sequence has been performed to understand the state of the space at a specific time. This was performed using the files gathered in the *"Space State Analysis"* folder. A large number of ".txt" files created with data from Space-Track[1] can be found in this folder. Additionally, all scripts used for post-processing and the figures created through this can also be found there. Then, the necessary scripts and relevant ".txt" files needed for conjunction simulation have all been gathered in the *"Simulation"* folder. To finish with, the Simulink model including the full conjunction event between the Starlink satellite and debris can be found in the *"Simulink"* folder. In this folder additional relevant scripts and constructed figures from the event can also be found.

B.3.4 Guidance Sequence

The whole guidance sequence algorithm and the different scripts it encompasses have all been gathered under *"4.0 | Convex Guidance Starlink"*.
The main convex optimisation and guidance sequence script, used for the entire guidance sequence simulation, is named under *"Convex_Optimisation_Script.m"*. Through this script, the entire optimisation process is ran together with the full Simulink model which is found under *"FullArchitectureTracked_Structured_Final.slx"*. The required data files that are necessary for the scripts can all be found under the *"Data"* folder. An analysis of the different means of propulsion has also been performed, with the relevant coding sequence found in *"Convex_Optimisation_Propulsion_Analysis.m"* and the relevant figures in the *"Figures"* folder. Additional scripts can also be found, used for plotting, for the tracking guidance analysis, the guidance LQR development, and more.

B.3.5 Control Sequence

The full control sequence analysis has been performed under *"5.0 | Control"*. To start with, the full architecture simulation, including the control sequence, can be found in the *"FullArchitectureTrackedControl_Euler.slx"* simulation script (for an LQR in terms of Euler angles) and in the *"FullArchitectureTrackedControl_quaternion.slx"* simulation script (for an LQR in terms of quaternions). Several additional tests for the LQR controllers have been performed, with the scripts named under *"LQR_q_control.m"* and *"LQR_euler_control.m"*, for a quaternions LQR and Euler angles LQR respectively.

B.3.6 Verification

A number of verification tests have been performed, as described in Section 7.2. All tests that have been performed through MATLAB® scripts can be found in the main *"6.0 | Verification Tests"* folder and more specifically in different subfolders depending on the test category.

[1] Space-Track, *Query Builder tool*, https://www.space-track.org/#queryBuilder [Last visited on 26/02/2024].

References

Açıkmeşe, B., & Carson, J. (2007). Convex programming approach to powered descent guidance for mars landing. *Journal of Guidance, Control, and Dynamics, Vol. 30*(No. 5). https://doi.org/10.2514/1.27553

Alfano, S. (2005). A numerical implementation of spherical object collision probability. *Journal of the Astronautical Sciences, Vol. 53*, pp. 103–109. https://doi.org/10.1007/BF03546397

Alfano, S. (2007). Review of Conjunction Probability Methods for Short-term Encounters. *Advances in the Astronautical Sciences, Vol. 127*, pp. 719–746.

Alforja Ruiz, I. (2020). *Guidance, Navigation and Control System Design of Envisat Active Debris Removal*. MSc Thesis, Delft University of Technology,

Antsaklis, P. J., & Michel, A. N. (2006). *Linear systems*. Birkhäuser.

Bate, R., Mueller, D. D., E., W. J., & Saylor, W. W. (1971). *Fundamental of Astrodynamics*. Dover Publications, Inc.

Blackmore, L., Açıkmeşe, B., & Carson, J. (2011). Lossless convexification of a class of optimal control problems with non-convex control constraint. *Automatica, Vol. 47*(No. 2), pp. 341–347.

Boyd, S., & Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Bryson, A., & Ho, Y. (1975). *Applied pptimal control: Optimization, estimation, and control*. Taylor; Francis Group.

Buchs, R., & Florin, M.-V. (2021). Collision Risk from Space Debris: Current Status, Challenges and Response Strategies, pp. 8. https://doi.org/https://doi.org/10.5075/epfl-irgc-285976

Cattani, B. (2022). *Enhancing Collision Avoidance Software and Hardware Techniques to Support and Validate GNC Developments*. MSc Thesis, Delft University of Technology,

Cerutti-Maori, D., Carloni, C., Rosebrock, J., & Siminski, J. (2023). Observation of Cosmos-1408 Debris Cloud with the Tracking and Imaging Radar System. *Proc. 2nd NEO and Debris Detection Conference, Vol. 2*(No. 1).

Chan, F. K. (1997). Collision Probability Analyses for Earth-Orbiting Satellites. *Applicationes Mathematicae*.

Conings, V. (2022). *GNC Design for Active Debris Removal*. MSc Thesis, Delft University of Technology.

Cooney, J. S. (2016). International Space Station (ISS) Orbital Debris Collision Avoidance Process. *Stinger Ghaffarian Technologies*.

Cranford, K., & Lane, M. (1969). An improved analytical drag theory for the artificial satellite problem. *AIAA Paper Number 69-925*.

ESA Space Debris Office. (2023). *ESA's Annual Space Environment Report* (7.1). The European Space Agency. Retrieved October 12, 2023, from https://www.sdo.esoc.esa.int/environment_report/Space_Environment_Report_latest.pdf

Federal Communications Commission. (2022). Space innovation mitigation of orbital debris in the new space age. (No. 2).

Foster, J. L., Estes, H. S., & Lyndon, B. (1992). *A Parametric Analysis of Orbital Debris Collision Probability and Maneuver Rate for Space Vehicles*. Lyndon B. Johnson Space Center. Navigation, Control, Space Center. Navigation, Control, & Aeronautics Division.

Guida, A. (2014). *A Branch and Bound Algorithm for the Global Optimization and its Improvements*. Universita degli Studi di Firenze.

Hoots, F. R., Crawford, L. L., & Roehrich, R. L. (1979). General Perturbations Theories Derived from the 1965 Lane Drag Theory. *Project Space Track Report*, (No. 2). https://doi.org/10.21236/ada081264

Hoots, F. R., Crawford, L. L., & Roehrich, R. L. (1984). An Analytical Method to Determine Future Close Approaches Between Satellites. *Celestial Mechanics*, *Vol. 33*, pp. 143–158. https://doi.org/10.1007/BF01234152

IADC Steering Group. (2020). *IADC Space Debris Mitigation Guidelines*. IADC. Retrieved February 21, 2023, from https://orbitaldebris.jsc.nasa.gov/library/iadc-space-debris-guidelines-revision-2.pdf

Kataoka, Ryuho, Shiota, Daikou, Fujiwara, Hitoshi, Jin, Hidekatsu, Tao, Chihiro, Shinagawa, Hiroyuki, & Miyoshi, Yasunobu. (2022). Unexpected space weather causing the reentry of 38 starlink satellites in february 2022. *J. Space Weather Space Clim.*, *Vol. 12*, pp. 41. https://doi.org/10.1051/swsc/2022034

Kelso, T. S. (2007). Analysis of the 2007 Chinese ASAT Test and the Impact of its Debris on the Space Environment. *Advanced Maui Optical and Space Surveillance Technologies Conference*, pp. 321–330.

Kessler, D., Johnson, N., Liou, J.-C., & Matney, M. (2010). The Kessler Syndrome: Implications to Future Space Operations. *33rd, Rocky Mountain Guidance and Control Conference*, 47–62.

Klinkrad, H., Alarcon, J. R., & Sanchez, N. (2005). Collision Avoidance for Operational ESA Satellites. *Fourth European Conference on Space Debris*, *Vol. 1*(No. 1).

Laporte, F. (2014). Jac software, solving conjunction assessment issues. *AMOS*.

Leloux, J. (2012). *Filtering Techniques for Orbital Debris Conjunction Analysis*. BSc Thesis, Delft University of Technology.

Leon Dasi, M. (2022). *Collision Probability through Orbital Uncertainty Propagation*. MSc Thesis, Delft University of Technology.

Loftus, J. P. (1997). *Letter to Committee on Space Shuttle Meteoroid/Debris Risk Management from Joseph Loftus, Jr., Assistant Director (Plans)*. NASA Johnson Space Center.

Lorda, L., & Fremeaux, C. (2008). Collision Avoidance Practice for GEO Satellites in an Operational Control Center. *Acta Numerica*, pp. 191–234. https://doi.org/10.1017/S0962492906370018

Malyuta, D., Reynolds, T., Szmuk, M., Lew, T., Bonalli, R., Pavone, M., & Açıkmeşe, B. (2021). Convex Optimization for Trajectory Generation.

Mao, Y., Szmuk, M., & Açıkmeşe, B. (2016). Successive convexification of non-convex optimal control problems and its convergence properties. *2016 IEEE 55th Conference on Decision and Control (CDC)*. https://doi.org/10.1109/cdc.2016.7798816

Montenbruck, O., & Gill, E. (2000). *Satellite Orbits: Models, Method and Applications*. Springer. https://doi.org/10.1007/978-3-642-58351-3

Mooij, E. (1997). *The motion of a vehicle in a planetary atmosphere*. Series 08: Astrodynamics; Satellite Systems. Delft University Press.

Mooij, E. (2020). *AE4870B Re-entry Systems*. Delft University of Technology.

Mooij, E. (2021). *On-Board Constrained Optimisation of Final Rendezvous: Technical Note 3*. Delft University of Technology. Delft, The Netherlands.

NASA Orbital Debris Program Office. (2009). Satellite Collision Leaves Significant Debris Cloud. *Orbital Debris – Quarterly News, Vol. 13*(No. 2), pp. 1–2. Retrieved February 16, 2024, from https://orbitaldebris.jsc.nasa.gov/quarterly-news/pdfs/odqnv13i2.pdf

NASA Orbital Debris Program Office. (2015a). International Space Station Performs Two Debris Avoidance Maneuvers and a Shelter-in-Place. *Orbital Debris – Quarterly News, Vol. 19*(No. 3), pp. 1. Retrieved February 21, 2023, from https://orbitaldebris.jsc.nasa.gov/quarterly-news/pdfs/odqnv19i3.pdf

NASA Orbital Debris Program Office. (2015b). Two More Collision Avoidance Maneuvers for the International Space Station. *Orbital Debris – Quarterly News, Vol. 19*(No. 4), pp. 1. Retrieved February 21, 2023, from https://orbitaldebris.jsc.nasa.gov/quarterly-news/pdfs/odqnv19i4.pdf

NASA Orbital Debris Program Office. (2020). International Space Station Maneuvers to Avoid Debris. *Orbital Debris – Quarterly News, Vol. 24*(No. 3), pp. 1. Retrieved July 20, 2024, from https://orbitaldebris.jsc.nasa.gov/quarterly-news/pdfs/ODQNv24i3.pdf

NASA Orbital Debris Program Office. (2022). International Space Station Maneuvers to Avoid Another Russian ASAT Fragment. *Orbital Debris – Quarterly News, Vol. 26*(No. 4), pp. 1. Retrieved October 12, 2023, from https://orbitaldebris.jsc.nasa.gov/quarterly-news/pdfs/odqnv26i4.pdf

NASA Orbital Debris Program Office. (2023). Two More On-orbit Fragmentations in 2022. *Orbital Debris – Quarterly News, Vol. 27*(No. 1), pp. 2. Retrieved November 10, 2023, from https://orbitaldebris.jsc.nasa.gov/quarterly-news/pdfs/ODQNv27i1.pdf

Nash, J. (2000). *The (Dantzig) Simplex Method for Linear Programming*. University of Ottawa.

National Aeronautics and Space Administration. (2008). *Handbook for Limiting Orbital Debris* [NASA-HANDBOOK 8719.14]. NASA.

National Aeronautics and Space Administration. (2020). *NASA Spacecraft Conjunction Assessment and Collision Avoidance Best Practices Handbook* [NASA/SP-20205011318]. NASA.

National Research Council. (1997). *Protecting the Space Shuttle from Meteoroids and Orbital Debris*. The National Academies Press. https://doi.org/10.17226/5958

Nemirovski, A., & Todd, M. (2008). Interior-point methods for optimization. *Acta Numerica*, pp. 191–234. https://doi.org/10.1017/S0962492906370018

Oliver, S., Cook, K., Fasenfest, B., Jefferson, D., Jiang, M., Leek, J., Levatin, J., Nikolaev, S., Pertica, A., Phillon, D., Springer, K., & De Vries, W. (2009). High-Performance Computer Modeling of the Cosmos-Iridium Collision. *Advanced Maui Optical and Space Surveillance Conference*.

Pardini, C., & Anselmo, L. (2023). The short-term effects of the cosmos 1408 fragmentation on neighboring inhabited space stations and large constellations. *Acta Astronautica, Vol. 210*, pp. 465–473. https://doi.org/https://doi.org/10.1016/j.actaastro.2023.02.043

Patera, R. P. (2001). General Method for Calculating Satellite Collision Probability. *Journal of Guidance, Control, and Dynamics, Vol. 24*(No. 4), pp. 716–722. https://doi.org/10.2514/2.4771

Picheny, V., Wagner, T., & Ginsbourger, D. (2012). A benchmark of kriging-based infill criteria for noisy optimization. *Hal Open Science*.

Prussing, J. (1992). Simple proof of the global optimality of the hohmann transfer. *Journal of Guidance Control and Dynamics*. https://doi.org/10.2514/3.20941

Sgobba, T., & Allahdadi, F. (2013). *Chapter 8 – Orbital Operations Safety*. Safety Design for Space Operations. https://doi.org/https://doi.org/10.1016/B978-0-08-096921-3.00008-8

United Nations – Office for Outer Space Affairs. (2010). *Space debris mitigation guidelines of the committee on the peaceful uses of outer space*. Vienna, Austria. Retrieved February 21, 2023, from https://www.unoosa.org/pdf/publications/st_space_49E.pdf

Vallado, D., Crawford, P., Hujsak, R., & Kelso, T. (2006). Revisiting spacetrack report #3: Rev. *AIAA 2006-6753*.

Vallado, D., & Crawford, P. (2008). Sgp4 orbit determination. *American Institute of Aeronautics and Astronautics*.

Virgili-Llop, J., Zagaris, C., Zappulla II, R., Bradstreet, A., & Romano, M. (2018). A convex-programmingbased guidance algorithm to capture a tumbling object on orbit using a spacecraft equipped with a robotic manipulator. *The International Journal of Robotics Research, Vol. 38*(No. 1). https://doi.org/10.1177/0278364918804660

Wächter, A., & Biegler, L. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming, Vol. 106*, pp. 25–57. https://api.semanticscholar.org/CorpusID:14183894

Wakker, K. F. (2015). *Fundamental of Astrodynamics* [ISBN: 9789461864192].

Wie, B. (1952). *Space Vehicle Dynamics and Control* (Vol. Vol. 2). American Institute of Aeronautics and Astronautics, Inc. https://doi.org/https://doi.org/10.2514/4.860119

Wolfhagen, J. (2020). *Collision Avoidance in the New Space Era*. MSc Thesis, Delft University of Technology.

Zeltkevic, M. (1998). *Runge-Kutta Methods*. https://web.mit.edu/10.001/Web/Course_Notes/Differential_Equations_Notes/node5.html