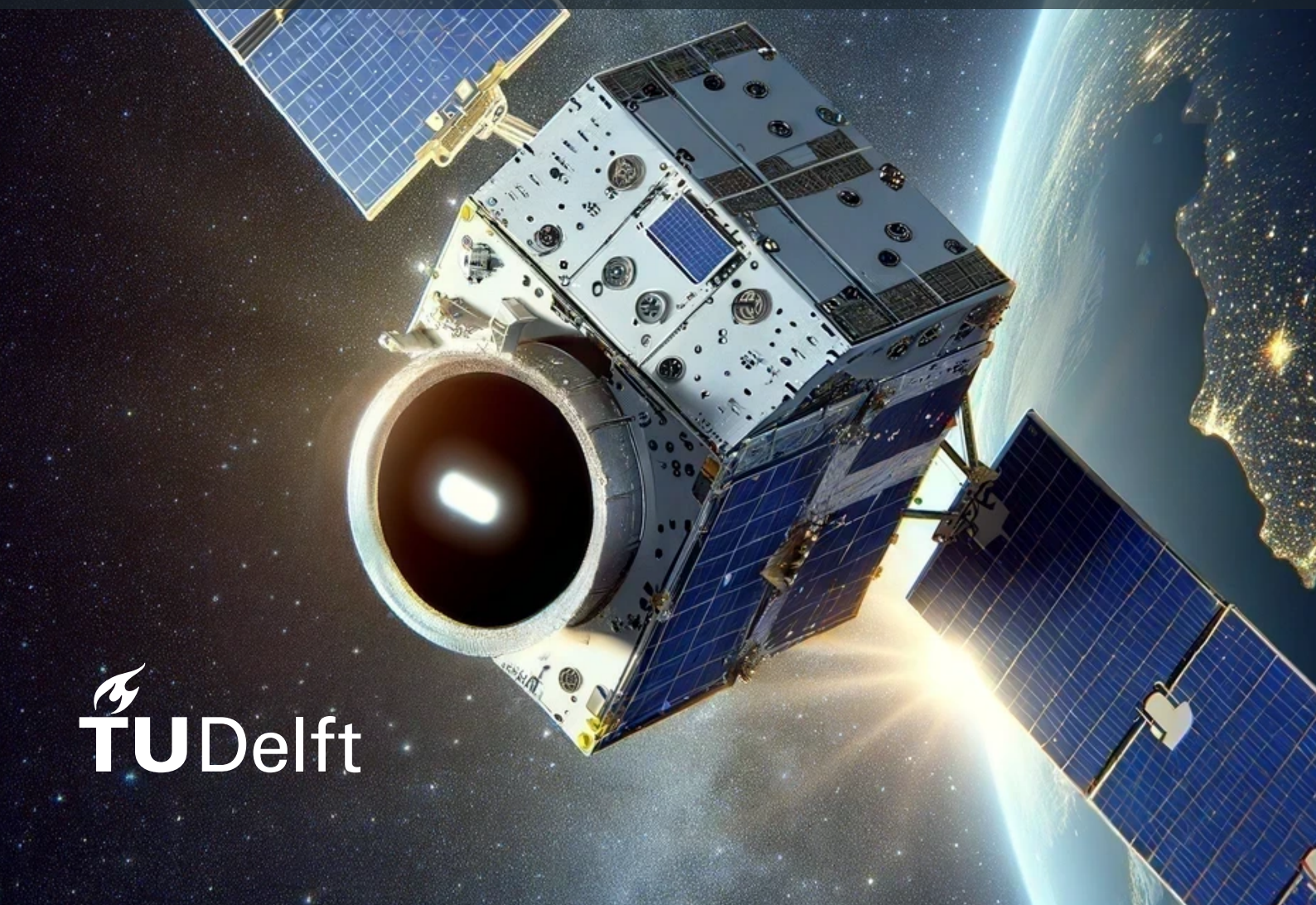
Agile Space-Observation Satellite Task Scheduling Optimisation

MSc Thesis

Pietro Silvagni



TU Delft

Agile Space-Observation Satellite Task Scheduling Optimisation

MSc Thesis

Thesis report

by

Pietro Silvagni

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on June 14, 2024 at 10:00

Thesis committee:	
Chair:	Dr. D. Dirkx
Supervisors:	Dr. S. Gehly
	Dr. L. Buinhas (external)
External examiner:	Dr. P. Piron
Place:	Faculty of Aerospace Engineering, Delft
Project Duration:	June, 2023 - June, 2024
Student number:	4652649

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Faculty of Aerospace Engineering · Delft University of Technology

Preface

This report summarises the work I have been doing over the past 12 months, in which I immersed myself into the field of satellite task scheduling, which was new to me. To the interested reader, I hope you find this report useful and insightful. And if you plan on continuing this work, I wish you luck!

I would like to thank my supervisor at Vyoma Dr. Luisa Buinhas for proposing this project, and for her dedication to spending time each week discussing concepts that were new to me. I also want to thank my co-supervisor at Vyoma, Ignacio Viñuela, for his support over the past year and a half and for being willing to dive into new topics with me. Additionally, I want to thank the rest of my colleagues at Vyoma. The questions asked when I presented intermediate results helped me better understand the bigger scope of the work I was doing, and steered the research into interesting new directions.

Special thanks are due to my daily supervisor, Assistant Professor Dr. Steve Gehly. His enthusiastic involvement in this new project and willingness to take me on as his first thesis student at TU Delft, despite it being outside his direct area of expertise, has been invaluable. Over the past twelve months, his instrumental support and guidance in maintaining a balanced workload have been greatly appreciated.

Additionally, I want to use this space to express gratitude to those people who played a crucial role in this academic journey, and made completing this project a reality.

Prima di tutto, voglio ringraziare i miei genitori. Grazie mamma e papà per avermi supportato in questi sette anni di università, senza farmi mancare nulla e consentendomi di dare tutto me stesso nelle varie fasi di questo percorso. Grazie per avermi dato gli strumenti per diventare il giovane ingegnere che sono oggi. Grazie Livi, per aver condiviso insieme questi ultimi cinque anni in Olanda. Grazie per riuscire a distrarmi quando mi impunto su dei problemi, e a ricordarmi che ci sono altre priorità nella vita. Grazie Andre, per darmi l'esempio a puntare in alto, e a credere in me stesso. Non sarei arrivato fino a qui senza la tua motivazione ed esempio costante.

Grazie Ciro, per essermi stato vicino per tutta la durata di questo progetto. Grazie per farmi vedere il lato positivo delle cose, anche in quei momenti in cui perdevo la motivazione. Grazie per aver condiviso con me questo viaggio, alternando momenti in cui ti spiegavo tutto pimpante come si muovono i satelliti attorno alla Terra, a momenti in cui non riuscivo a vedere la fine di questa tesi. Non ce l'avrei fatta senza di te.

A tutti i miei parenti e amici a Ravenna e in Italia, grazie per farmi sentire accolto e voluto bene ogni volta che torno a casa. Intraprendere progetti ambiziosi è più facile quando si hanno persone come voi alle spalle.

Finally, I want to thank all my friends in Delft. You have had such a profound impact on the person I am today, that I cannot imagine who I would be without your influence. I am extremely thankful for growing up and sharing every aspect of life together over the past seven years. You helped me so much both at a personal and an academic level, that I cannot find words to describe it. I would not be here writing this preface today if it were not for you. Thank you for being a constant source of inspiration; seeing the journeys each of you has gone through has been truly motivating. I am looking forward to seeing where life takes you.

Pietro Silvagni
Delft, June 2024

Abstract

The congestion of orbits around the Earth, particularly in Low Earth Orbit (LEO), necessitates the implementation of space-based Space Situational Awareness (SSA) missions. This increasing orbital congestion, driven by the proliferation of mega constellations, poses significant challenges to satellite operations and collision avoidance. In response, commercial entities are beginning to plan and launch space-based SSA missions. Despite this emerging interest, there is a notable gap in the literature regarding the scheduling of operations for these missions, which are characterised by the inclusion of multiple operational modes.

Additionally, existing literature on operational scheduling often lacks high-fidelity modelling of onboard resource dynamics. This research addresses this gap by focusing on energy-constrained scenarios, developing novel operational scheduling frameworks suitable for assigning data collection and data downlink tasks for optical space-based SSA missions under such constraints.

To enhance the fidelity of the scheduling process, this research employs both low-fidelity and high-fidelity simulators to model the dynamic behaviour of onboard resources. The developed frameworks are tested in various scenarios to validate their practicality and effectiveness. A significant contribution of this research is the open-source availability of part of the developed scheduling frameworks, utilising software accessible under the TU Delft license. This open-source approach ensures that the research can be extended and built upon by future researchers, fostering ongoing advancements in the field of space-based SSA operations.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

AEOS	Agile Earth-Observation Satellite
AEOSSP	Agile Earth-Observation Satellite Scheduling Problem
AoP	Argument of Periapsis
AU	Astronomical Unit
CA	Collision Avoidance
CEOS	Conventional Earth-Observation Satellites
CoM	Centre-of-Mass
COMM	COMMUNICATION (mode)
D/O	Degree and Order (of Spherical Harmonics coefficients)
DCM	Direct Cosine Matrix
DoD	Depth-of-Discharge
DoF	Degrees-of-Freedom
EBC	Equivalent Battery Cycles
ECEF	Earth-Centred, Earth-Fixed
ECI	Earth-Centred Inertial
EOS	Earth-Observation Satellite
EPS	Electrical Power System
GEO	Geostationary Earth Orbit
GSN	Ground Station Network
ISC	IDLE - SURVEILLANCE - COMMUNICATION (framework)
ITC	IDLE - TASKED TRACKING - COMMUNICATION (framework)
LEO	Low Earth Orbit
LLA	Latitude, Longitude, Altitude
LoS	Line-of-Sight
LQR	Linear-Quadratic-Regulator
LTAN	Local Time of the Ascending Node
LTI	Linear Time Invariant
MILP	Mixed-Integer Linear Programming
MRP	Modified Rodrigues Parameters

PL	Payload
RAAN	Right Ascension of the Ascending Node
RP	Rodrigues Parameters
RSO	Resident Space Object
RSSP	Range Satellite Scheduling Problem
S/C	Spacecraft
SC	SURVEILLANCE - COMMUNICATION (framework)
SNR	Signal-to-Noise Ratio
SoC	State-of-Charge
SRP	Solar Radiation Pressure
SSA	Space Situational Awareness
SSO	Sun-Synchronous Orbit
SURV	SURVEILLANCE (mode)
TT	TASKED TRACKING (mode)
tudat	TU Delft Astrodynamics Toolbox

1

Introduction

At the time of writing this report, there are about 7,560 active satellites in orbit, as reported by the Union of Concerned Scientists [55]. These satellites are integral to many facets of modern life, gathering vital data and providing essential services. Examples include significant advancements in weather monitoring and the use of GPS-based timing devices crucial for regulating financial markets, highlighting the deep interconnection between contemporary society and space-based technologies. The number of satellites has been steadily increasing and is expected to continue rising in the next decade, driven by the launch of mega constellations like those planned by Starlink, OneWeb, and Project Kuiper [51] [39] [4]. However, this expansion has not come without challenges. The exponential increase in satellite launches has resulted in a congested space environment, with inactive satellites and debris complicating orbital paths and posing collision risks. Flohrer describes space debris as human-made objects, mainly in Earth orbit, that no longer serve any functional purpose [18]. These debris pose significant risks to active satellites due to potential collisions. As of June 2024, the catalogue of objects in Earth-orbit is close to 28,000, which represents less than 4% of the estimated number of space objects in orbit larger than 1 cm, highlighting the lack of knowledge on the location of most of the debris objects [1] [37]. This uncertainty is largely due to the insufficient number of Space Situational Awareness (SSA) sensors available to track the debris.

Tracking space debris currently relies heavily on ground-based sensors like telescopes and radars, which have their limitations. Telescopes can only operate at night and are dependent on clear weather conditions, while radars, though capable of 24/7 operation, face power limitations that restrict their effective range beyond Low Earth Orbit (LEO) due to signal attenuation proportional to $1/r^4$. Moreover, the vast number of debris objects significantly outnumbers the available sensors, creating data gaps that can last from several hours to days, leading to inaccurate orbit predictions.

Observing space debris from space itself offers a promising solution to these limitations, enhancing the existing ground-based sensor network. In recent years, initiatives such as the Space-Based Space Surveillance (SBSS) satellite by the United States Space Force [49] and the Missile Defence Agency's Space Tracking and Surveillance System (STSS) [35] have been operational. Additionally, commercial ventures like the German startup Vyoma are preparing to launch space-based SSA sensors [14].

A distinctive aspect of this mission is its focus on space rather than Earth observation, marking a fundamental shift in satellite orientation. Unlike conventional Earth-observing satellites that are directed towards the planet's surface, space-based SSA satellites look outwards into space. The difference between space-based SSA missions and conventional Earth-observation missions is the consideration of two distinct observation modes, namely surveillance and tasked tracking, which means that the scheduling framework needs to account for multiple operational modes. While extensive literature covers the task scheduling for Earth-observation, little exists for space-observation satellites, highlighting a significant gap in spacecraft task scheduling knowledge. This research aims to bridge this gap by developing an operational scheduling framework for space-based SSA missions, particularly for satellites equipped with optical telescopes.

The research is guided by the following key questions:

Research Questions

RQ1: *"What is the best scheduling strategy for SURVEILLANCE mode, that maximises the amount of data downlinked while satisfying the energy and data requirements?"*

RQ2: *"To what spacecraft and operational constraints is the scheduling framework most sensitive?"*

RQ3: *"How do initial conditions and operational scenarios affect resource management along the (continuous) scheduling horizon?"*

RQ4: *"What are the impacts of including additional operational modes (i.e. IDLE and TASKED TRACKING) in the scheduling framework?"*

The structure of the report unfolds as follows: chapter 2 covers the relevance and justification for the research, providing a background on the escalating challenges posed by increased satellite and debris populations. This is followed by chapter 3, presenting a detailed examination of the theoretical foundation necessary to understand the dynamics of space-based SSA operations. Chapters 4 to 7 deal with the methodology employed for this research, explaining the approach followed and the two tools involved, namely a discrete-time scheduler presented in chapter 5 and a continuous-time attitude simulator presented in chapter 7. Chapter 6 focuses instead on the modifications implemented in the scheduling framework, that lead to the creation of two novel approaches. Subsequently, chapter 8 presents the results, offering insights into the framework's application to a case-study space-surveillance mission and discussing potential adjustments for broader application. The report concludes with chapter 9, which summarises the research findings and provides recommendations for future research directions.

2

Relevance and Justification for Research

In this chapter, the reasons that make this research work relevant are presented. Section 2.1 presents the trends that have been characterising the space environment in the last years. Subsequently, section 2.2 introduces space debris, discussing why they constitute a problem for active satellite and section 2.3 describes how Collision Avoidance (CA) manoeuvres are planned and performed to protect active satellites. SSA, a field dealing with improving our knowledge of the space environment and of objects residing therein, is discussed in section 2.4, while section 2.5 introduces the satellite scheduling problem. To conclude, section 2.6 clearly points out the knowledge gap that this research aims at filling, and relates it to the topics previously introduced in the chapter.

2.1. Space Object Population Trend

The first object inserted in the space environment was Sputnik 1, launched successfully on October 4th 1957. Since then, there have been more than 6500 successful launches, each leading to an increase in the number of objects occupying the orbits around our planet [36].

The last few years have witnessed a remarkable surge in satellite launches, a trend that is not only continuing but expected to accelerate further in the near future. This unprecedented increase is largely fuelled by the advent and development of mega-constellations, ambitious projects that aim to deploy vast networks of satellites primarily into LEO to provide global internet coverage, among other services. This resulted in commercial players significantly amplifying their presence in the space sector, particularly in LEO. This shift is underscored by the exponential growth in the number of active satellites, with commercial payloads becoming the dominant contributors to the LEO population post-2020, as can be seen in Figure 2.1. The implications of such growth are profound, not only in terms of technological advancements and global connectivity but also in the challenges it presents to space traffic management and the sustainability of space activities.

The influx of satellites into LEO is not without consequences. This region, characterised by its proximity to Earth, offers distinct advantages such as lower launch costs and higher payload efficiencies, making it an attractive orbit for a wide array of satellite applications. For this reason, it was defined as a protected orbital region by the Inter-Agency Space Debris Coordination Committee (IADC) in 2002 [21], together with the Geostationary orbit (GEO), which allows spacecrafts to be stationary with respect to an observer on the Earth's surface. These regions are depicted in Figure 2.2. However, this desirability comes at a cost. The increasing density of satellites in LEO aggravates the risk of space debris generation and collisions, posing a significant threat to both existing and future space assets.

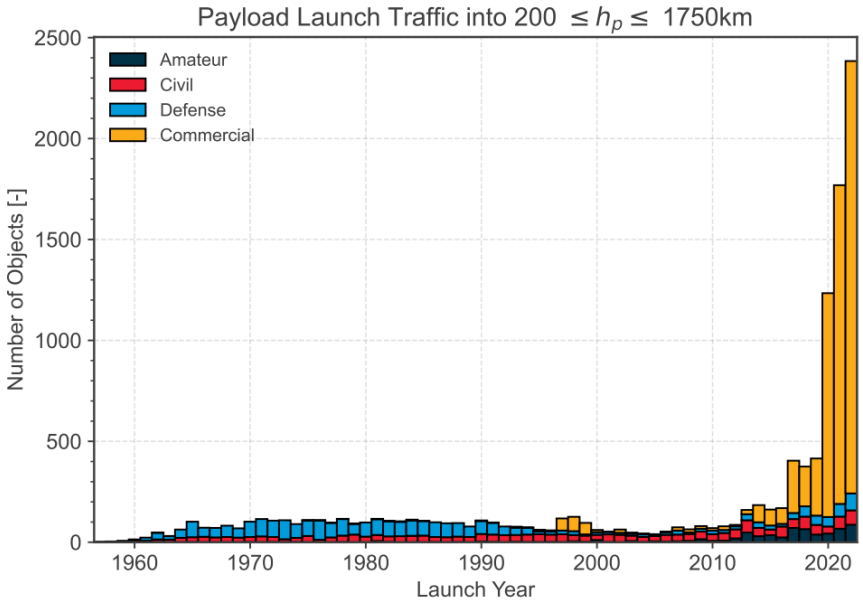


Figure 2.1: Object count evolution per mission funding [36]

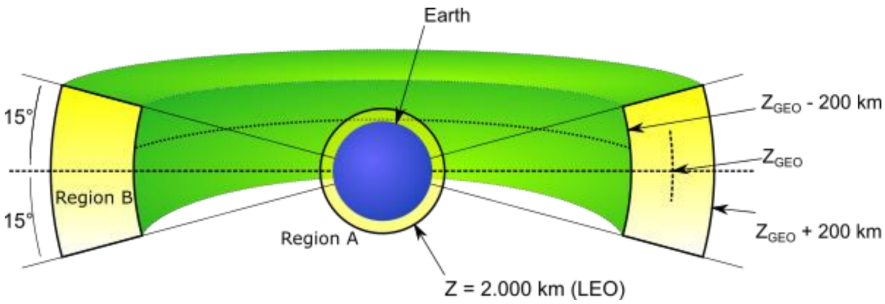


Figure 2.2: IADC protected orbital regions [21]

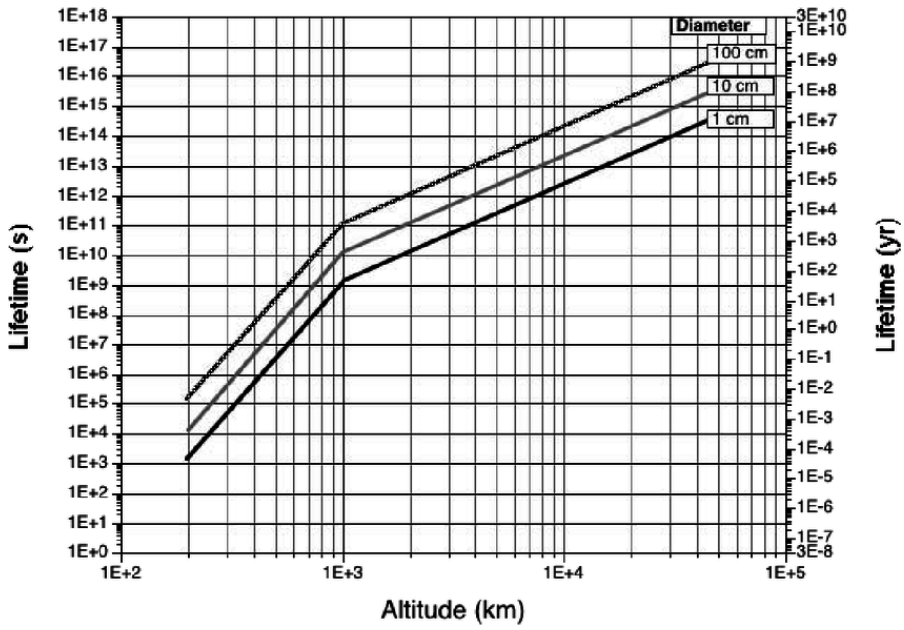


Figure 2.3: Expected orbital lifetime for three spherical debris fragments with diameter of 1, 10 and 100 cm, based on standard US atmosphere. The density of each object is $\rho = 0.2g/cm^3$ [41]

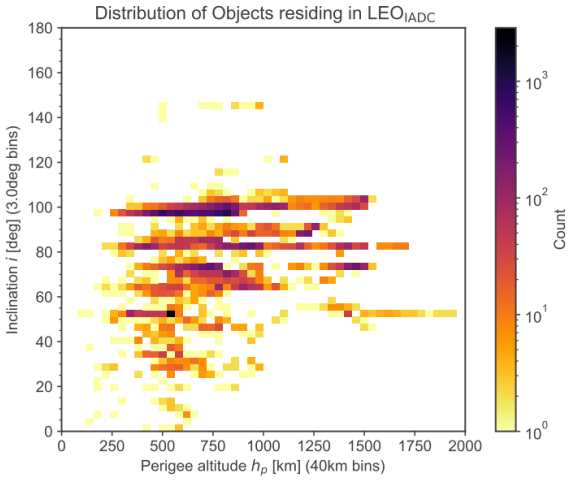


Figure 2.4: Spatial distribution of objects residing in LEO, in terms of attitude and inclination [36]

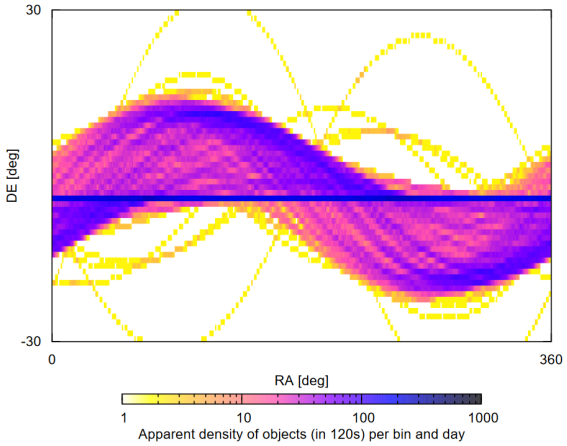


Figure 2.5: Spatial distribution of objects in GEO, in terms of right ascension and declination [18]

2.2. Space Debris

As per the IADC, space debris is defined as all artificial objects including fragments and elements thereof, in Earth orbit or re-entering the atmosphere, that are non-functional [21]. These remnants of past missions and collisions encompass everything from spent rocket stages and defunct satellites to the myriad of smaller fragments generated through collisions and disintegrations.

As of December 2023, the estimated number of debris objects in orbit, based on statistical models, is as follows: [37]:

- 36,500 objects larger than 10 cm;
- 1,000,000 objects from 1 cm to 10 cm;
- 130 million objects from 1 mm to 1 cm.

This becomes problematic when investigating the lifetime that space debris has at different orbital altitudes. Figure 2.3 shows the expected lifetime in seconds and years for three different spherical debris objects of 1, 10 and 100 cm of diameter with a density of $0.2 g/cm^3$. From this plot, it can be seen that a 100 cm object would spend 300 years at an altitude of 700 km, and more than a thousand at 1000 km altitude.

The spatial distribution and characteristics of space debris vary across different orbital regions. The LEO region, for instance, is particularly congested, housing a significant portion of these objects. Debris in LEO is distributed across various altitudes, with concentrations peaking around 700 to 1000 kilometers, and is commonly found in high-inclination orbits, which are optimal for a wide range of satellite applications, as can be seen in Figure 2.4. In contrast, the Geosynchronous Earth Orbit (GEO) hosts a smaller, though still concerning, amount of debris, primarily consisting of larger objects such as spent rocket bodies and defunct satellites, predominantly residing in the protected GEO region, as seen in Figure 2.5.

The presence of space debris poses multifaceted challenges to active satellites and space operations at large. At the velocities typical of orbital objects, even small pieces of debris carry enough kinetic energy to cause catastrophic damage upon collision, as depicted in Figure 2.7. The potential for such high-velocity impacts renders space debris a significant threat, not only to the physical integrity of satellites but also to the safety of crewed space missions. The dynamics of collisions and the subsequent evolution of debris clouds further intensifies this threat. When two objects collide in space, the resulting debris cloud rapidly spreads across the orbital shell, increasing the likelihood of subsequent collisions in a cascading effect known as the Kessler Syndrome. An example of this is the Cosmos-Iridium collision, which caused more than 2000 fragments larger than 10 cm [5], that quickly spread across the

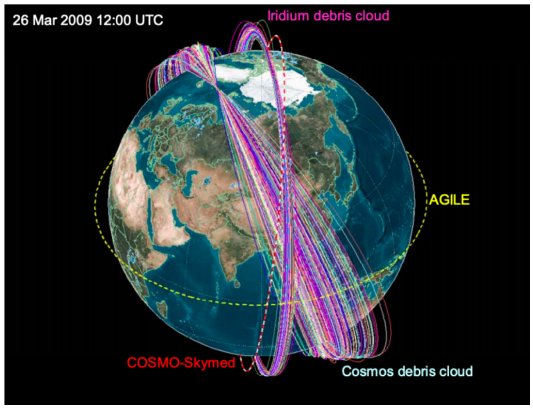


Figure 2.6: Debris cloud resulting from the Cosmos-Iridium collision, 1.5 month after the event [5]

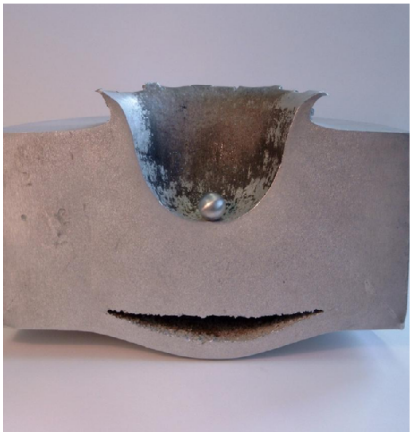


Figure 2.7: High velocity impact test. Impact of a 1.7 g Al-sphere travelling at 6.8 km/s on a 8.2cm-thick Al-block [24]

orbital shell, as shown in Figure 2.6. This phenomenon underscores a self-sustaining cycle of debris generation that could render certain orbital regions unusable if left unchecked.

In order to avoid collision with space debris, manoeuvres can be performed. This is introduced in the following section.

2.3. Collision Avoidance Manoeuvres

In the densely populated orbits around Earth, active satellites are constantly at risk of colliding with space debris. To mitigate this threat, satellites employ Collision Avoidance (CA) manoeuvres, strategic adjustments to their orbits designed to dodge potential collisions. These manoeuvres typically involve altering the satellite's trajectory to either raise or lower its orbit or, less commonly, to change its inclination, thus avoiding the path of an approaching debris object [60].

However, the decision to execute a CA manoeuvre is fraught with complexity, largely due to the incomplete and uncertain knowledge of the space debris population. The vast majority of space debris remains untracked, with current catalogues capturing less than 4% of objects larger than 1 cm [37]. This glaring gap in surveillance leaves satellite operators navigating a largely unseen landscape, making decisions based on limited and often inaccurate data.

This uncertainty can lead to unnecessary, resource-intensive manoeuvres that deplete a satellite's finite propellant reserves, shortening its operational lifespan and reducing its overall mission value. The process of deciding whether to undertake a CA manoeuvre involves assessing the probability of collision, with maneuvers typically initiated when this probability exceeds a certain threshold, set conservatively to minimize risk. However, given the lack of precise data on debris locations and trajectories, these thresholds are often based on incomplete information, potentially triggering unwarranted maneuvers that waste valuable resources.

In order to give an idea of the order of magnitudes of such events, for a Sentinel-like mission, there is 1 alert sent a month, and about 1 CA manoeuvre every 3 months. For each manoeuvre, there is about 8 hours of data outage, resulting in mission profit loss, and fuel usage [3].

2.4. Space-Situational Awareness

At the heart of addressing the challenge of space debris and the consequent risk to active satellites lies the field of Space Situational Awareness (SSA). SSA embodies the coordinated efforts to enhance the understanding and monitoring of the space environment, specifically focusing on the identification, tracking, and cataloguing of objects in Earth's orbit [16]. The primary objective of SSA is to reveal the previously unknown locations and trajectories of space debris. This provides the crucial information

necessary to develop collision avoidance strategies and protect operational satellites.

Traditionally, the bulk of SSA capabilities has been anchored to ground-based sensors. These earth-bound systems, while constituting the backbone of current surveillance networks, are inherently limited by their geographical and atmospheric constraints. Optical sensors, for example, are impeded by weather conditions and daylight, whereas radar systems, despite their all-weather capabilities, are limited by their power requirements and the attenuation of radio waves through the atmosphere.

In response to these limitations, and driven by the critical need for comprehensive space surveillance, the advent of space-based SSA sensors marks a significant evolution in the field. These pioneering systems, now in various stages of development and deployment, promise a paradigm shift in our ability to monitor space debris . Unlike their ground-based counterparts, space-based sensors have the unique advantage of an unobstructed view of the space environment, free from atmospheric distortions and the diurnal cycle. This vantage point allows for continuous and more precise tracking of objects, including those in high-inclination orbits that are challenging to monitor from the ground.

Furthermore, space-based SSA missions introduce a novel operational context for satellites. These missions entail satellites orbiting Earth while primarily focusing their observational instruments away from the Earth's surface and, in the case of optical systems, away from the Sun's direction. This orientation is critical for detecting and tracking space debris, which requires a clear line of sight to the vast expanse of space devoid of terrestrial and solar interference. At the same time, this difference with respect with conventional Earth-orbiting satellites, which predominantly point their instruments towards the Earth, leads to new challenges when it comes to scheduling the operations of the satellite. These challenges are introduced in the next section.

2.5. Satellite Scheduling Problem

A significant aspect identified in the literature study is the Earth-Observation Satellite Scheduling Problem (EOSSP), which emerges from the broader field of Operational Research. The EOSSP revolves around optimising task schedules for Earth-observing satellites to maximise data acquisition within operational constraints. This encompasses challenges such as managing limited on-board resources and ensuring timely data downlink. Traditionally, the emphasis within Operational Research has been on computational efficiency and handling large datasets, occasionally at the expense of detailed modelling of a satellite's operational constraints and attitude dynamics modelling.

2.6. Knowledge Gap

The crux of the knowledge gap addressed by this research is the implications of existing task scheduling frameworks for satellite operations, particularly in the context of space-based SSA missions. Unlike conventional Earth-observation satellites, SSA missions require monitoring space objects, necessitating sensor orientation away from both the Earth and, in some cases, the Sun. This unique operational mode necessitates a reevaluation of existing task scheduling frameworks to accommodate the specific requirements of space-based SSA missions.

On top of bridging this gap, this research aims at tackling the satellite scheduling problem from a more practical point of view than traditional approaches in the EOSSP. In order to achieve this, the constraints considered for the problem will be modelled accurately, through a synergistic use of a low-fidelity scheduler tool and a high-fidelity continuous-time simulator. This combination allows for a nuanced analysis that goes beyond scheduling efficiency to include the practical operational constraints of space missions, such as energy consumption, data storage capacities, and communication windows. By adopting this methodology, the research delves into the practicalities of scheduling tasks for space-based SSA missions, extending beyond conventional operational research perspectives to develop solutions that are pragmatically viable for real-world space operations.

3

Theoretical Background

This chapter goes over the theoretical background needed to fully understand the work that was conducted in this research. Section 3.1 and section 3.2 discuss orbital and attitude dynamics, respectively, which lay the foundation for understanding the forces governing the behaviour of a satellite and of space debris. Subsequently, section 3.4 focuses on SSA, and in particular to space surveillance practices that make use of an optical telescope. Section 3.5 examines the field of spacecraft operations scheduling, at first linking it to space surveillance, then diving into the Earth-Observation Satellite Scheduling Problem (EOSSP). Different optimisation methods used to solve the EOSSP, and their characteristics are then summarised in section 3.6, which concludes the chapter.

Note that most of the content of this chapter is taken from the literature study performed prior to this research [48].

3.1. Orbital Dynamics

This section reviews concepts of orbit dynamics, necessary for a full understanding of the work presented in the next chapters.

3.1.1. Reference Frames

In this section, the type of reference frames commonly used to define the orbital motion of the objects around the Earth are discussed, together with their origin and orientation. Note that there are multiple options available for the types of frames described below. The goal of this section is only to present the types of frames. The choice of which implementation is used for the simulation is explained in the following chapters.

Pseudo-inertial, geocentric frame

In order to follow Newton's laws of physics, physical quantities, such as velocities and forces, have to be defined in inertial, also called Newtonian, reference frames. These are frames defined using very distant celestial objects as reference points. As these objects are extremely far, they appear as stationary with respect to the center of the reference frame, and thus provide a pseudo-inertial reference system, with the axis not moving with respect to the objects in the vicinity of the origin. In this research, such a frame will be needed to specify the motion and forces acting on the Earth-orbiting space objects considered. This frame will be centred on the Earth, and will thus be an Earth-Centred Inertial (ECI) frame.

Geocentric, Earth-fixed, co-rotating frame

In order to define the position and motion of objects on the Earth surface, a frame that is centred on, and co-rotates with the Earth is required. This is useful when considering ground stations. Such frames are defined as Earth-Centred, Earth-Fixed (ECEF) frames, and are non-inertial, due to the frame rotating at the same rate as the Earth around its rotation axis.

Orbit-frame

The orbit-frame is popular in the spaceflight dynamics field. This frame is centred in the center of mass of an orbiting object, and travels along its orbit. Its three axis are defined as follows:

- X-axis: points in the velocity direction;
- Z-axis: points towards the center of the Earth;
- Y-axis: completes the right-handed coordinate system.

3.1.2. Representations

There are different ways in which the translational state of an object can be represented. Each representation has its level of complexity, advantages and limitations, as well as leading to different expressions for the translational dynamics modelling of an orbiting object.

Only representations which are deemed promising for the use-case at hand are considered. It is thus useful to remember that the main focus of the thesis is not precise orbital modelling, but that orbital motions will serve as the input for the observation strategy of the mission. Also, the simulation time will not be very long, and thus the stability of the representation in long periods of time is not of paramount importance. Thus the most important factors in the choice of representation are its robustness, and its computational efficiency.

A final selection of which representation to use for the modelling of translational dynamics is made in the following chapters, based on a quantitative analysis of the impact that using different ones has on the final result.

Cartesian state vector (Cowell Formulation)

The most straightforward and standard way to define the motion of an orbiting body in three-dimensional space, is using Cartesian coordinates. In this coordinate system, the state vector and its derivative have the following form:

$$\vec{x} = \begin{bmatrix} \vec{r} \\ \dot{\vec{r}} \end{bmatrix} \qquad \dot{\vec{x}} = \begin{bmatrix} \dot{\vec{r}} \\ \ddot{\vec{r}} \end{bmatrix} \tag{3.1}$$

The Cowell formulation has the advantage that it is straightforward, leading to simple equations, and it does not depend on any a priori assumptions on the behaviour. Additionally, it does not lead to singularities, and it is a robust formulation, since it is widely used in the field of spaceflight dynamics. On the other hand, it might lead to large values for state derivatives, leading to large numerical errors for large propagation time.

Classical Orbital Elements (Kepler Elements)

Kepler elements are a typical representation of an orbit due to a single point mass. A visual representation of these elements can be seen in Figure 3.1, which depicts the situation for a celestial body, the black dot, in an orbit with respect to a plane of reference, and reference direction. A description of each Keplerian element is presented below. Note that the symbols used here will not be used further in the report, but are used here to visualise the elements with the help of the figure.

- Eccentricity, e: describes how elliptical the orbit is, and describes how much it is elongated compared to a circle;

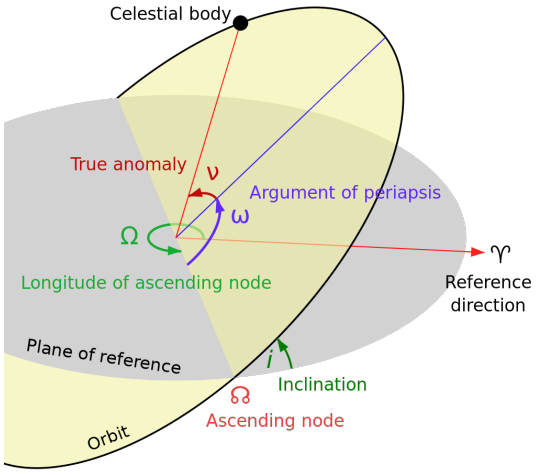


Figure 3.1: Keplerian elements of a celestial body in orbit

- Semi-major axis, a: describes the size of the orbit;
- Inclination, i: describes how "tilted" the orbit is. It represents the angle between a reference plane and the orbital plane of the motion;
- Right Ascension of the Ascending Node (RAAN), Ω: describes where the ascending node is with respect to a reference direction in the reference plane. The ascending node is the point in which the orbit passes from North to South through the reference plane. Ω is thus the angle between the reference direction and the ascending node;
- Argument of periapsis, ω: describes the orientation of the ellipse in the orbital plane. The periapsis is the point in the orbit in which the object passes closest to its central body. ω is the angle measured from the ascending node to this point in orbit;
- True anomaly, ν: defines the position of the object along its ellipse at a specific time. This "specific time" is called an *epoch*.

Kepler elements, despite being a popular way to represent an orbit in the classical two-body problem, lead to singularities. This is not desirable, and since the propagated Cartesian state following from the Cowell formulation can be mapped to Keplerian elements when post-processing the result, no benefits were seen in using Kepler elements.

3.1.3. Central Gravity

The main force acting on an object orbiting Earth, is the latter's gravitational attraction. The motion of such an object can be described by the following equation of motion:

$$\ddot{\vec{r}} = -\frac{\mu}{r^3} \cdot \vec{r} \tag{3.2}$$

where $\mu = GM_{Earth}$ is the gravitational constant of the Earth, $\vec{r}$ the position vector of the object, and r the distance from Earth's centre to the object. The velocity of an object undergoing such a motion can be described by the so-called *vis-viva* equation, below.

$$v^2 = \frac{2\mu}{r} - \frac{\mu}{a} \tag{3.3}$$

where v is the orbital velocity of the object, and a the semi-major axis of its orbit. When the object is in a perfectly circular orbit, which means that $r = a$ at all points along the orbit, Equation 3.3 can be rewritten to express the velocity of the object in the following way.

$$v = \sqrt{\frac{\mu}{a}} \tag{3.4}$$

3.1.4. Gravitational Perturbing Forces

There are two sources of gravitational perturbations: those due to irregularities in the central body itself, and those due to third bodies. These are presented below.

Non-Spherical Earth

The equations shown in subsection 3.1.3 assume the central body to be a point-mass, which means that the whole mass of the body is assumed to be concentrated in one point. In reality, this is not the case, as celestial bodies are non-spherical, and their mass distribution is not uniform.

The gravitational acceleration exerted by body B on body A is obtained from:

$$\ddot{\vec{r}} = \nabla U\left(\vec{r}\right) \tag{3.5}$$

where U is the gravitational potential of the central body, and ∇U its gradient. The way in which these quantities are formulated changes the degree of accuracy in which the gravitational acceleration is modelled. For point mass, the potential is expressed as $U = \frac{\mu}{r}\vec{r}$, which results in Equation 3.2, which itself results in Keplerian orbits.

This formulation of potential does not hold when there are latitudinal or longitudinal variations in the central body's mass distribution, which is the case for every real celestial body. A popular formulation to include these variations is that of using spherical harmonics. The formulation of the potential then becomes rather complicated:

$$U\left(\vec{r}\right) = \frac{\mu}{r}\sum_{l=0}^{\infty}\sum_{m=0}^{l}\left(\frac{R}{r}\right)^{l}\bar{P}_{lm}(\sin\phi)\left(\bar{C}_{lm}\cos m\theta + \bar{S}_{lm}\sin m\theta\right) \tag{3.6}$$

in which μ is the gravitational parameter, R is the reference radius of the central body, r is the distance from the center of mass of the orbiting and central body, $\bar{P}_{lm}$ are associated Legendre polynomials, C_{lm} and S_{lm} are the spherical harmonic coefficients, and the l and m summation indices are referred to as degree and order, respectively. Each combination of (l, m) represent a distinct effect of the gravity field variation. The higher the degree and order terms included, the better the gravity field variations are represented. On the other hand, including high degree and order coefficients in a numerical simulation increases the computational burden, and results in longer computation time.

For this reason, since precise orbit modelling is not the main focus of this research, only the terms having a *large* impact on the propagated orbit will be considered. An example of a coefficient that has a large impact on orbital dynamics is that of the $C_{2,0}$ spherical harmonic, also called J_2, which is due to Earth's flattening which results in more mass being present around the equator. This characteristic of Earth's gravitational field causes nodal precession, which allows for Sun-Synchronous Orbit (SSO).

Third Body Perturbations

On top of the Earth body, other celestial bodies exert forces on an Earth-orbiting object, impacting its dynamics. Taking as example Figure 3.2, the effect of the Moon's presence on an Earth-orbiting satellite can be observed. When the object is going *away* from the Moon, then the latter exerts a force opposing the satellite's motion, thus decelerating it. On the contrary, when the object is going *towards* the Moon, the latter exerts a force in the same direction as the satellite's motion, thus accelerating it. Similar effects are introduced by the gravitational attractions of other bodies in the solar systems.

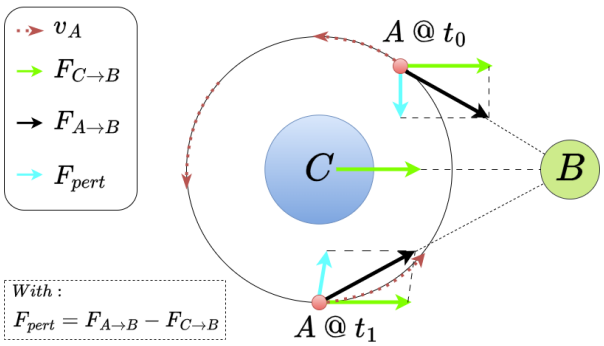


Figure 3.2: Third-body gravitational perturbation of the Moon (B) on an Satellite (A) orbiting around the Earth (C). The green arrows represent the direction and magnitude of the Moon's gravitational force on the Earth. The black arrows represent the direction and magnitude of the Moon's gravitational force on the satellite. The cyan arrows are the third-body perturbing force, obtained by subtracting the green arrow to the black arrow. From this, the different effect that this force has on the satellite in the two points in orbit can be visualised.

3.1.5. Non-Gravitational Perturbing Forces

There are many sources of perturbations which do not arise from gravitational forces. The most relevant are presented below, together with the effect these have on the translational dynamics of an Earth-orbiting satellite. Additional effects these forces have on rotational dynamics are discussed in section 3.2.

Atmospheric Drag

Although the atmospheric density decreases drastically when reaching orbital heights, there are still some atmospheric particles that exert a force in the direction opposite to the object velocity. The acceleration experienced by the orbiting object due to atmospheric drag is given by the following equation:

$$\vec{a}_{atm} = -\left(\frac{\rho v_{\text{air}}^2 C_D S_{ref}}{2m}\right)\hat{v}_{\text{air}} \tag{3.7}$$

where ρ is the atmospheric density at current body location, v_{air} is the body airspeed-based velocity, S_{ref} is the reference area, C_D is the drag coefficient, and m is the body's mass.

Solar Radiation Pressure (SRP)

SRP is caused by the photons emitted by the Sun. These have momentum and energy, which impart a force on a body when interacting with it. The photon can either be absorbed, reflected, or let pass through. The latter is the case for translucent surfaces. Using a first-order approximation, this acceleration can be expressed in the following formula:

$$\vec{a}_{SRP} = \left(\frac{P}{4\pi c}\right)\left(\frac{C_r S_{ref}}{m}\right)\frac{\hat{r}}{\|\vec{r}\|^2} \tag{3.8}$$

where C_r is the radiation pressure coefficient of the orbiting body, P is the total power output of the Sun, S_{ref} is the reference area, c is the speed of light in vacuum. As limit cases, for a $C_r = 1$ all radiation is absorbed, while for $C_r = 2$ all radiation is reflected.

Other Perturbations

On top of atmospheric drag and SRP, there are other causes of less significant perturbations. These are linked to forces due to tides, Earth radiation pressure, interaction between Earth's magnetic field

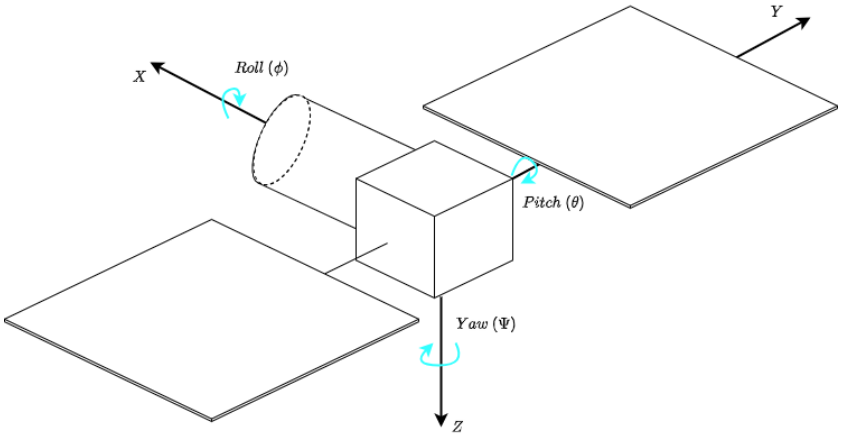


Figure 3.3: Typical body-axis definition

and the orbiting object, and antenna thrust, among others. These forces have very small effects on the dynamics of orbiting objects. Since precise modelling of orbital dynamics is not the main focus of this research, they will not be further discussed.

3.2. Attitude Dynamics

In order to check whether the operational schedules outputted by an optimisation algorithm are feasible in practice, the attitude dynamics of the satellite need to be modelled. Such a simulation requires an attitude model that combines a reference system, the rotational equations of motion, including both disturbance and control torques, and angular kinematics relations, which are presented in sections 3.2.1 to 3.2.3.

3.2.1. Reference Frames

A well-defined reference system is necessary to clearly express the equations of rotational motion and kinematic relations. The frame needed for the rotational aspects of this research is presented below, together with another useful concept for the tracking of other objects. Note that rotational aspects are important only for the observer spacecraft, and thus the frames will be presented assuming the body under analysis is a spacecraft.

Body-fixed Coordinate System

A body-fixed coordinate frame is necessary to express the rotational dynamics of a body. The axes of this frame are usually chosen to be the same as the principal body axes, as depicted in Figure 3.3, showing the roll, pitch, and yaw axes aligned with the principal body axes of the spacecraft. A body-fixed frame co-rotates with the spacecraft, and is thus a non-inertial reference system. Because of this, in order to express the equations of motion in a suitable reference system, a transformation between this body-fixed frame, and a Newtonian, inertial reference frame has to be performed. This process is explained in more detailed in the following section. The Newtonian reference frame used for attitude considerations is the ECI frame, introduced in subsection 3.1.1.

In the derivation of the equations of motion, the body-frame will be referred to as $\mathbb{B}$-frame, and the Newtonian, inertial, frame as $\mathbb{N}$-frame.

3.2.2. Attitude Kinematics

Attitude kinematics plays a pivotal role in the field of spacecraft attitude dynamics, offering a framework to describe the orientation of a spacecraft in three-dimensional space. It encompasses various repre-

sentation techniques, each with its advantages and limitations, tailored to specific applications within the broader scope of space missions. For the description of the concepts in this section, the work of Schaub and Junkins was followed [45].

Diverse Representations

The landscape of attitude representations is characterised by a rich diversity, including Direction Cosine Matrices (DCMs), Euler Angles, Quaternions, Rodrigues Parameters (RP), and Modified Rodrigues Parameters (MRP). Each of these methods offers a unique perspective on spacecraft orientation, catering to different requirements of accuracy, computational efficiency, and simplicity. In the interest of brevity, rather than exhaustively detailing the various attitude representations available, this discussion prioritises the reasons for selecting one particular representation.

Direction Cosine Matrices (DCMs) and **Euler Angles** are classical methods that provide intuitive frameworks for representing spacecraft orientations. However, DCMs can involve redundancies and significant computational overhead in inverse calculations, while Euler Angles are susceptible to singularities known as gimbal lock, which restrict their use in three-dimensional space where complex rotational maneuvers are required.

Quaternions offer a four-parameter representation that is free from the singularities affecting Euler Angles. This characteristic makes quaternions highly suited for the representation of large rotations, avoiding the gimbal lock issue and ensuring robustness in dynamic space environments. Quaternions are particularly advantageous for their computational efficiency and the ease with which they can represent composite rotations, making them ideal for simulations that require high fidelity and reliability.

Following quaternions, **Rodrigues Parameters (RP)** and **Modified Rodrigues Parameters (MRP)** serve as compact alternatives, each reducing the parameter set to just three components. While they eliminate some of the complexities associated with quaternions, RPs and MRPs do not completely avoid singularities and offer no significant advantages in terms of computational efficiency. Moreover, their usage is less standard within the space dynamics community, which can pose challenges for integration and support in widespread tools and applications.

Rationale for Choosing Quaternions

The decision to adopt quaternions for attitude representation in this research is informed by several key factors: their robustness against singularities, superior computational efficiency, and their standardization in space dynamics tools such as tudat and GMAT [8] [20]. The absence of gimbal lock and the ability to perform straightforward composite rotations make quaternions exceptionally well-suited for accurately simulating the complex dynamics of spacecraft attitude control. Additionally, their normalised nature and the extensive support they receive across various simulation platforms ensure that they integrate seamlessly into existing frameworks, enhancing both the reliability and the fidelity of simulations.

Kinematic Differential Equation

The kinematic differential equation for quaternions can be derived by differentiating each term of $\vec{q}$, which leads to four coupled kinematic differential equations, that can be elegantly expressed in the following matrix form:

$$\begin{pmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} \tag{3.9}$$

which by performing transmutation can be written as:

$$\begin{pmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} = \frac{1}{2} \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{pmatrix} 0 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} \tag{3.10}$$

In the above equations, $\vec{\omega}$ is the angular rate of the spacecraft. Equation 3.10 can be expressed as $\dot{\boldsymbol{q}} = \frac{1}{2}[B(\boldsymbol{q})] \cdot \vec{\omega_{\mathbb{B}}}$. The matrix $B(\boldsymbol{q})$ is called "quaternion rate matrix", and is orthogonal and singularity-free. The full derivation is these equations of motion can be found in the literature study performed for this research [48], which follows the work of Schaub and Junkins [45].

3.2.3. Derivation of Attitude Dynamics Equations

In this section, the attitude dynamics of a rigid body will be presented, with the goal of deriving the equations of motion needed to model the rotation of the spacecraft. In order to have such a model, it is necessary to find a relation between the state vector and its derivative. The derivations in this section follow the work of Wie "Space Vehicle Dynamics and Control" [61].

The rigid body rotation relative to an inertial reference frame is given by:

$$\vec{M} = \dot{\vec{H}} = \left.\frac{d\vec{H}}{dt}\right|_{\mathbb{N}} \tag{3.11}$$

where $\vec{H}$ is the angular momentum vector of the rigid body with respect to the $\mathbb{N}$-frame and $\vec{M}$ is the external moment vector acting on the body. Looking into the definition of the angular momentum, the following relation can be derived:

$$\begin{aligned}
\vec{H} &= \int \vec{\rho} \times \dot{\vec{R}} dm \\
&= \int \vec{\rho} \times \left(\dot{\vec{R}}_O + \dot{\vec{\rho}} \right) dm \\
&= \int \vec{\rho} \times \dot{\vec{R}}_O dm + \int \vec{\rho} \times \dot{\vec{\rho}} dm \\
&= \int \vec{\rho} dm \times \dot{\vec{R}}_O + \int \vec{\rho} \times \dot{\vec{\rho}} dm \\
&= 0 + \int \vec{\rho} \times \vec{\omega} \times \vec{\rho} dm
\end{aligned} \tag{3.12}$$

where $\vec{\rho}$ is the distance between the mass elements dm and the CoM and $\vec{R}$ the distance between the origin of the $\mathbb{N}$-frame and the mass element dm, as it can be seen in Figure 3.4. Note that the last line of Equation 3.12 follows from the definition of the center of mass of a body, for which $\int \vec{\rho}\, dm = 0$. Furthermore, the transport theorem states that:

$$\dot{\vec{\rho}} = \left.\frac{d\vec{\rho}}{dt}\right|_{\mathbb{N}} = \left.\frac{d\vec{\rho}}{dt}\right|_{\mathbb{B}} + \vec{\omega}^{\mathbb{B}/\mathbb{N}} \times \vec{\rho} \tag{3.13}$$

and due to the rigid body assumption, $\left.\frac{d\vec{\rho}}{dt}\right|_{\mathbb{B}} = 0$. Note that the notation $\vec{\omega}^{\mathbb{B}/\mathbb{N}}$ means "the angular velocity of the $\mathbb{B}$-frame with respect to the $\mathbb{N}$-frame".

To get rid of the integral in the last line of Equation 3.12, $\vec{\rho}$ and $\vec{\omega}$ are expressed in the $\mathbb{B}$-frame, and their cross product is worked out, which results in:

$$\vec{\rho} \times \vec{\omega} \times \vec{\rho} = \begin{pmatrix} \left(\rho_2^2 + \rho_3^2\right)\omega_1 - \rho_1\rho_2\omega_2 - \rho_1\rho_3\omega_3 \\ -\rho_1\rho_2\omega_1 + \left(\rho_1^2 + \rho_3^2\right)\omega_2 - \rho_2\rho_3\omega_3 \\ -\rho_1\rho_3\omega_1 - \rho_2\rho_3\omega_2 + \left(\rho_1^2 + \rho_2^2\right)\omega_3 \end{pmatrix} \tag{3.14}$$

Consider now the following definitions:

$$J_{11} = \int \left(\rho_2^2 + \rho_3^2\right) dm; \quad J_{22} = \int \left(\rho_1^2 + \rho_3^2\right) dm; \quad J_{33} = \int \left(\rho_1^2 + \rho_2^2\right) dm$$

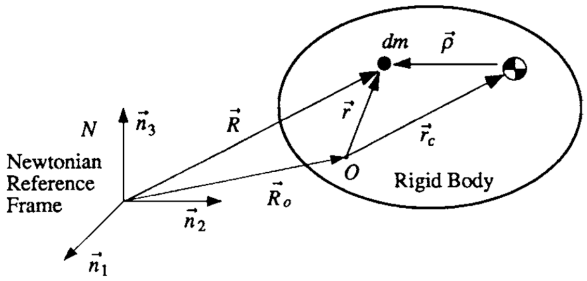


Figure 3.4: Rigid body in motion relative to a Newtonian reference frame [61]

$$J_{12} = J_{21} = -\int \rho_1 \rho_2 dm; \quad J_{13} = J_{31} = -\int \rho_1 \rho_3 dm; \quad J_{23} = J_{32} = -\int \rho_2 \rho_3 dm$$

The angular momentum can now be expressed in the $\mathbb{B}$-frame as follows:

$$\vec{H} = \hat{J}\vec{\omega} \tag{3.15}$$

Which in matrix form becomes:

$$\begin{bmatrix} H_1 \\ H_2 \\ H_3 \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix} \cdot \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{3.16}$$

According to the transport theorem:

$$\dot{\vec{H}} = \left.\frac{d\vec{H}}{dt}\right|_{\mathbb{N}} = \left.\frac{d\vec{H}}{dt}\right|_{\mathbb{B}} + \vec{\omega} \times \vec{H} \tag{3.17}$$

From this, Equation 3.11 can be rewritten as:

$$\begin{aligned} \vec{M} &= \frac{d}{dt}(\hat{J} \cdot \vec{\omega})|_{\mathbb{B}} + \vec{\omega} \times \hat{J} \cdot \vec{\omega} \\ &= \left.\frac{d\hat{J}}{dt}\right|_{\mathbb{B}} \cdot \vec{\omega} + \hat{J} \cdot \left.\frac{d\vec{\omega}}{dt}\right|_{\mathbb{B}} + \vec{\omega} \times \hat{J} \cdot \vec{\omega} \end{aligned} \tag{3.18}$$

For a rigid body $\left.\frac{d\hat{J}}{dt}\right|_{\mathbb{B}} = 0$ and $\left.\frac{d\vec{\omega}}{dt}\right|_{\mathbb{B}} = \left.\frac{d\vec{\omega}}{dt}\right|_{\mathbb{N}} = \dot{\vec{\omega}}$, so we obtain what is referred to as the Euler rotational equation of motion in vector form:

$$\vec{M} = \hat{J} \cdot \dot{\vec{\omega}} + \vec{\omega} \times \hat{J} \cdot \vec{\omega} \tag{3.19}$$

Final Differential Equations

In this section, the final set of differential equations that will be considered for the implementation of rotational dynamics in the numerical simulation are presented. In these equations, the derivative of the rotational state has to be expressed in terms of known quantities at the considered moment in time, such that numerical integration can be performed.

Euler's dynamic equation, given in Equation 3.19, can be re-arranged and be solved for $\dot{\vec{\omega}}$, which results in the following:

$$\dot{\vec{\omega}} = \hat{J}^{-1}\vec{M} - \hat{J}^{-1}(\vec{\omega} \times \hat{J} \cdot \vec{\omega}) \tag{3.20}$$

Inspecting the moment vector, $\vec{M}$, it can be noted that it can be split into two components: one including the control moments to be applied to achieve the required attitude, $\vec{M}_{control}$, and one including the disturbance moments, $\vec{M}_{disturbance}$.

$$\vec{M} = \vec{M}_{control} + \vec{M}_{disturbance} \tag{3.21}$$

Moving to the kinematics equations, the notation used from here onward will be that of expressing the quaternion as $\boldsymbol{q} = [q_0, \vec{q}_v]^T$, with q_0 being the scalar part, and $\vec{q}_v = [q_1, q_2, q_3]^T$ the vectorial part.

With this notation, Equation 3.10 can be rewritten in the following relations:

$$\dot{q}_0 = -\frac{1}{2}\vec{\omega}^T\vec{q}_v \tag{3.22}$$

$$\dot{\vec{q}}_v = \frac{1}{2}(q_0 \cdot \vec{\omega} - \vec{\omega} \times \vec{q}_v) \tag{3.23}$$

3.3. Control

In this section, a quick overview of control theory and how it applies to spacecraft ADCS is presented, followed by a section on the types of controllers considered.

3.3.1. Control Theory

The idea behind a control system is that of:

1. Defining a goal to be achieved by the system in some measurement frame;
2. Measuring the difference between the system's current and desired state;
3. Calculating the appropriate actions to be performed by the system in response to those differences;
4. Taking the calculated actions.

Consider a situation in which a spacecraft needs to keep in the field of view of its telescope another space-object, as in the tasked-tracking mode of the considered mission. Examples can now be made for each of the steps listed above.

Goal Definition

Step 1 consists in defining the desired rotational state the spacecraft should achieve in a certain measurement frame. In the use-case considered, this would mean keeping the spacecraft's body-axis aligned with its optical telescope pointed at the target object intended to be tracked. The desired attitude can be defined as $\boldsymbol{q}_{desired}$, assuming that quaternions are used as attitude representation, and the desired angular rate as $\vec{\omega}_{desired}$.

Attitude Determination

Step 2 consists in measuring the differences in terms of angles the spacecraft would have to rotate through to reach the desired attitude. This step includes measuring the current rotational state of the spacecraft, $\boldsymbol{q}_{current}$ and $\vec{\omega}_{current}$, by means of sensors, and calculating the difference with the desired rotational state. This difference can be referred to as the error rotational state, which can be split into the error quaternion, $\boldsymbol{q}_{error}$, and error angular velocity, $\vec{\omega}_{error}$. This step can be referred to as *attitude determination*.

Control Law

Step 3 consists in calculating the required actions to be taken in order to reduce the difference between the current and the desired rotational state. The set of rules used to determine the actions to be taken is referred to as *control law*.

Attitude Control Actuation

Step 4 consists in applying the calculated forces on the spacecraft by means of actuators, making it rotate by the desired amounts. This step takes the name of *actuation*.

Deterministic System Assumption

In the scope of this study, a fully deterministic system is assumed, characterised by its ability to possess precise, real-time knowledge of the spacecraft's current rotational state. This assumption entails that control laws are formulated based on access to an idealised, error-free representation of the spacecraft's state, eliminating the need to account for inaccurate or biased sensor data.

This assumption is made due to the specific context of satellite operations and tasking, where the process of attitude determination does not yield immediate, tangible consequences for the overall system. This is primarily because sensors can efficiently ascertain the spacecraft's attitude while it concurrently engages in tasks such as observing other objects or facilitating communication. Hence, attitude determination can be seamlessly integrated into other simultaneous operations.

In contrast, control actions carry greater significance than attitude determination tasks. This distinction arises from the fact that attitude control actions directly and mechanically influence the spacecraft's orientation, thereby exerting a substantial impact on the manoeuvres and actions the spacecraft can execute.

This is a simplification of how the real system works, and will be taken into account when drawing conclusions.

3.3.2. Control Laws

As stated in the section above, control laws relate the attitude determined by the sensors to the commands sent to the actuators. There are many types of attitude laws that determine how a controller will behave. In the context of this research, only simple, linear controllers will be considered. The reason behind this choice is that within the context of spacecraft operations scheduling, the attitude of the spacecraft is only important when it constraints other actions to be performed, but is not the main focus of the research. Considering complex attitude laws that would increase the complexity of the model, without adding meaning to the results, was thus discarded.

Amongst simple controllers, proportional-, PD-, and PID-controllers are presented. A schematic overview of their functioning is presented in Figure 3.5.

Proportional controller

The idea behind this type of controller is that the strength of the actuator response is proportional to the difference between the goal and the measured value. The output of such a controller would take the following form:

$$Output = K(x_{goal} - x_{measured})$$

where K is the gain of the system, which determines how strongly the system reacts to errors.

In terms of spacecraft ADCS, this could result in the following control law:

$$\vec{M}_{control} = K_p \, \boldsymbol{q}_{error} \tag{3.24}$$

where $\vec{M}_{control}$ is the control torque to be sent to the actuators, K_p the proportional gain of the controller, and $\boldsymbol{q}_{error}$ the error quaternion, representing the rotation to go through to reach the desired orientation.

An important note to be made is that, in order for this type of controller to be achieved, the strength of the actuator must be able to be gradually changed.

PD-Controller

PD-controllers stand for "Proportional and Derivative controllers". The idea behind this type of controllers is thus an extension to that of proportional controllers. The strength of the actuator response

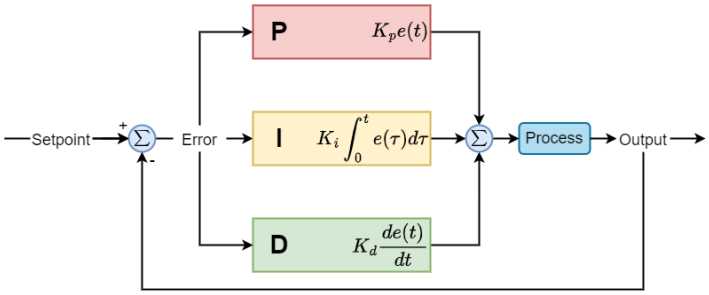


Figure 3.5: PID controller

is now not only dependent on the difference between current and desired state, but also to the rate of change of this difference. In terms of spacecraft ADCS, a possible PD-controller could follow the law:

$$\vec{M}_{control} = K_p \, \boldsymbol{q}_{error} + K_d \frac{d\boldsymbol{q}_{error}}{dt} \tag{3.25}$$

where K_d is the derivative gain of the controller.

PID-Controller

In a similar fashion, the idea can be extended such that the strength of the actuators is dependent also on the integral of the error. A possible control law would then have the following form:

$$\vec{M}_{control} = K_p \, \boldsymbol{q}_{error} + K_d \frac{d\boldsymbol{q}_{error}}{dt} + K_i \int_0^t \boldsymbol{q}_{error}(\tau)d\tau \tag{3.26}$$

where K_i is the integral gain of the controller.

Tolerance vs Resources

Looking at the control law equations presented above, it can be seen that the outputted control torque depends on how the error is defined. If the error is defined as the difference between the exact desired value and the measured value, then at every measurement step the control law will send a control torque to the actuators. This is because the desired and measured values will have a certain degree of difference at nearly every measurement epoch, due to errors in the measurements, or to disturbance torques that were unaccounted for. As a result, the ADCS actuators will be constantly executing commands, deteriorating their health and using resources.

For this reason, it is desired to have a margin in how exact the desired attitude needs to be. Such margin can be referred to as *tolerance*, which can be defined as the amount of deviation from the desired value for which no output needs to be generated. From this definition, it can be derived that the higher the tolerance, the lower the control resources needed to obey the control law, and vice-versa.

3.4. SSA

SSA encompasses the comprehensive understanding and proactive management of various threats that could compromise space systems. While SSA initially bore a military-centric connotation, focusing on discerning potential deliberate threats [18], the evolution of space exploration towards commercialisation necessitates a shift towards a civil perspective. This modern approach emphasises the accurate characterisation of space objects, including their operational capabilities, ownership, and potential for manoeuvring. These elements are crucial for developing effective collision avoidance strategies and for safeguarding active satellites from potential collisions [16].

3.4.1. Space Surveillance

At the heart of SSA lies space surveillance, a discipline dedicated to cataloguing space objects. This catalogue includes critical information such as the orbital elements of each object and the uncertainties

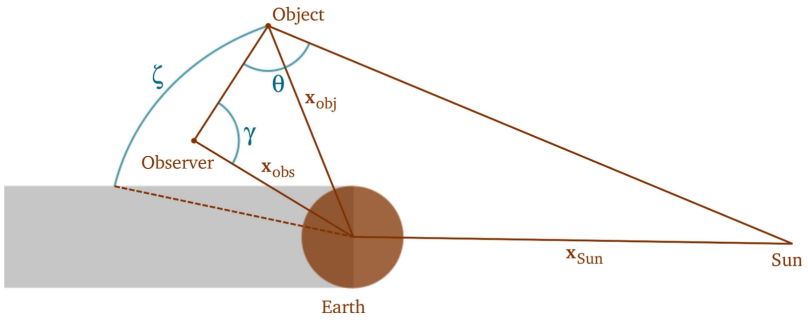


Figure 3.6: Geometry of visibility constraints for a space-based passive optical telescopes, assuming a cylindrical shadow model [18]

associated with them, which form the basis for collision avoidance maneuvers. Space surveillance is characterised by two principal activities: surveying and tracking. *Surveying* scans the skies for previously unidentified objects, while *tracking* focuses on re-observing known objects to refine their orbital data or monitor maneuvers. These tasks are underpinned by a network of sensors that capture observations, laying the groundwork for estimating the orbits and physical characteristics of space objects. Amongst the various types of SSA sensors, this research work focuses on the optical telescope, and its characteristics are presented in the next section.

3.4.2. Optical Telescope

An optical telescope works by collecting photons on a detector. The photons that are collected can be divided into two categories:

- "*Signal*" photons: emitted by the Sun and reflected by the target objects needed to be detected;
- "*Background*" photons: all the photons captured by the telescope which do not come from the target objects. These can come from different sources, such as directly from the Sun or other stars, reflected by celestial bodies, such as the Earth and the Moon, or, most notably, as light pollution from terrestrial sources.

In order to have a successful detection of an object, the Signal-to-Noise Ratio (SNR) has to be above a certain threshold. Otherwise, the signal is too low and cannot be distinguished from the noise, thus not producing a detection. Optical telescopes are passive sensors, as they do not require a signal to be first emitted by the sensor itself, but only light to be collected. For this reason, this technique can be referred to as "passive optical observation".

There are many parameters on which a successful detection depends. Many of these are related to the sensor design, such as the pixel size and dimension and the FoV. Others are related to the catalogue, as different catalogues have different thresholds for how strict to be in order to consider an observation valid, in terms of accuracy or timeliness. In this research, the focus is on operational tasks in terms of scheduling the actions of the observer. For this reason, only characteristics with direct implications on the task of the observer will be discussed. Examples of such constraints are the conditions to have a successful detection of an object, which are presented in the following sections.

Third Body Occultation Constraint

The most trivial constraint for optical telescopes is that of third body occultation. As photons do not travel through matter, there needs to be a direct Line-of-Sight (LoS) between the observer and the target. Considering the situation in which an optical telescope is located either on Earth's surface or in Earth orbit, and the targets are Earth-orbiting objects, then the only third body that could be in the LoS between observer and target is the Earth itself.

Illumination Conditions

In practical terms, what gets collected by the detector inside passive optical telescopes are photons. These photons are first emitted by the Sun, reflected by the surface of the target object, and then collected by the detector of the observer. This leads to different constraints, all related to the geometry of the problem, depicted in Figure 3.6.

Direct Sunlight

The most obvious is that the target object has to be in direct Sun light in order to be detected. If this condition is not satisfied, then the photons cannot reach the target, thus will not get reflected and will never reach the collector inside the telescope. This means that while target objects are in eclipse, no optical observation can be made. Looking at Figure 3.6, the angle ζ represents the angle between the position of the orbiting object to be observed, and the beginning of the eclipse cylindrical region. If ζ is below 0, then the object would lie in the eclipse of the Earth. Thus, the direct sunlight constraint can be expressed with the relation: $\zeta > 0^\circ$.

Phase Angles

On top of receiving direct light from the Sun, to have a successful detection, the illumination condition of the target object as seen from the observer must meet certain conditions. These derive from the requirement for the SNR to be above a certain threshold for an observation to result in a successful detection. Due to this, having excessive light coming from unwanted sources might lead to an unsuccessful detection.

Observer-Object-Sun Angle

The observer-object-Sun angle, the phase angle θ in Figure 3.6, must meet the condition $0 < \theta < \theta_{max}$, where the maximum allowed phase angle, θ_{max} depends on instrumental and operational constraints. This value is usually well below 180°, as the energy reflected by space debris is significantly lower than that of direct light coming from the Sun. In practical terms, this means that the observer satellite needs to look ”away from the Sun”, in order to achieve good illumination conditions for detection.

Earth-Observer-Object Angle

For space-based telescopes, also the Earth cannot be too close to the FoV of the telescope, as this would lead to excessive Sun light being reflected from the Earth into the telescope. Following the development of Flohrer [18], this means that the Earth-Observer-Object angle, indicated with the symbol γ in Figure 3.6, must meet the following condition:

$$\gamma > \arcsin \frac{r_{Earth}}{||\vec{x}_{obs}||} + \xi \tag{3.27}$$

where r_{Earth} is the radius of the Earth, $\vec{x}_{obs}$ the position of the observer, and ξ a parameter depending on the diameter of the telescope's FoV.

Observer-Object-Moon Angle

Depending on the sensitivity of the sensor, the Moon's presence in the FoV of the sensor might also need to be excluded to have successful detection. The condition for this angle is similar to that of the observer-object-Sun angle, but substituting the Moon for the Sun.

3.4.3. Common Pointing Strategies

In this section, two promising pointing strategies for a space-based passive optical observation sensor identified by Flohrer are presented [18]. Both these strategies are based on the sensor onboard a satellite orbiting Earth in LEO.

The first strategy consists in pointing the camera always away from the LoS with the Sun. In this way the satellite is always looking at the ”dark” portion of the sky, and the illumination condition of the space debris should be beneficial to achieve successful detections. In this strategy, the satellite is keeping an orientation that is fixed with respect to the ECI frame, and thus keeps an inertial pointing;

The second strategy consists in pointing the camera in a fixed position with respect to the orbital frame, and maintaining it fixed as the satellite orbits the Earth. In this strategy, the camera should be pointed in a direction approximately parallel to the velocity vector. In this way, the camera would be pointing at other objects in the LEO region as the satellite travels around the Earth. As discussed in the previous chapter, most space debris are present in LEO, and for this reason this strategy is deemed promising.

3.5. Spacecraft Operations Scheduling

In this section, the field of spacecraft operations scheduling is introduced. First, subsection 3.5.1 presents the problem of sensor management as an optimisation problem. Subsequently, subsection 3.5.2 analyses the research field of the Earth-Observation Satellite Scheduling Problem (EOSSP), highlighting its similarities with this research, and describing the different variations of the problem along with various approaches researchers have employed to address it over the years.

3.5.1. Sensor Management in Space Surveillance

Space surveillance missions are fundamentally reliant on effective sensor management, which entails the systematic scheduling of a sensor's activities over time. This necessity arises from the reality that the number of space objects significantly outstrips the available sensors. Moreover, as stated by Jaunzemis [26], observational constraints dictated by orbital mechanics limit the number of space objects that each sensor can observe at any given moment. This creates a scenario where it is not feasible for a network of sensors to monitor all space objects simultaneously. Consequently, sensor management within space surveillance cannot adhere to a one-size-fits-all approach. Instead, it is inherently an optimisation problem, requiring the development of varied strategies to either maximise a specific reward function or minimise associated costs. The Earth-Observation-Satellite-Scheduling-Problem (EOSSP) is a pertinent field in this context, focusing on the task scheduling of space-based sensors aboard satellites, and showcases the application of optimisation principles in managing the intricate demands of space surveillance missions.

3.5.2. EOSSP

Following the description proposed by Wang et al. (2020), the EOSSP is a class of problems that aims at scheduling satellite observation tasks in order to maximise the observation profit while satisfying the considered operational constraints [57]. This thesis project collocates itself as a continuation and expansion of this area of research, analysing the effect that observing space rather than the Earth has on the task scheduling. This section outlines the different approaches and modifications that have been developed throughout the years to tackle this problem.

Historical Evolution of EOSSP

The EOSSP, as delineated by contemporary literature, aims at scheduling satellite observation tasks to maximise observation profit within the bounds of operational constraints. This field has witnessed a marked evolution over the past 25 years, beginning with the seminal work of Gabrel et al. in 1997, who first addressed the EOSSP [19]. This was further developed by Pemberton, who conceptualised the problem within a constraint satisfaction framework, employing heuristic search and constraint propagation to navigate the complexities of satellite scheduling [40]. Initially, the focus was on satellites with limited rotational capabilities only around a single axis, typically the roll-axis, facilitating Earth scanning in a cross-flight direction, as shown in Figure 3.7. These satellites, referred to as Conventional Earth-Observation Satellites (CEOS), laid the groundwork for subsequent advancements in the field.

The introduction of Agile Earth-Observation Satellites (AEOS) by Lemaitre marked a significant paradigm shift within EOSSP [28]. Unlike their conventional counterparts, AEOS possess enhanced attitude adjustment capabilities, allowing for a broader observational scope that includes areas "ahead" and "behind" the nadir direction, as depicted in Figure 3.8. According to Liu, this flexibility significantly expands the observational opportunities but also introduces additional complexity to the scheduling problem [31]. The transition from CEOS to AEOS has catalysed a shift in research focus towards the

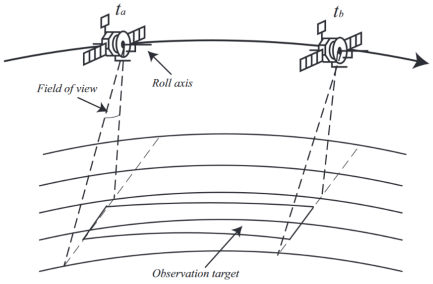


Figure 3.7: A Conventional Earth-Observation Satellite (CEOS), able to rotate only about its roll-axis, along the flight direction [57]

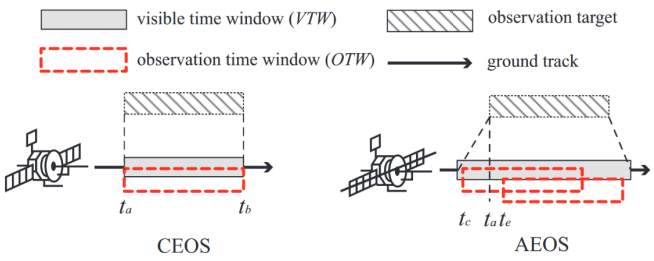


Figure 3.8: Difference between CEOS and AEOS. It can be noted that the AEOS has a longer visible time window, as it is able to look "ahead" and "behind" the nadir direction, as it can rotate also about the yaw-axis [57]

Agile Earth-Observation Satellite Scheduling Problem (AEOSSP), reflecting the capabilities of modern Earth-observation satellites.

Core Aspects of EOSSP Research

The diversity in research approaches within the EOSSP is primarily influenced by three factors: 1) the modelling assumptions related to mission constraints, 2) the definition of observation profit, and 3) the choice of optimisation frameworks. Researchers have explored various operational constraints, including communication, memory, energy, and attitude control, tailoring their models to reflect the needs of specific mission profiles. Similarly, the definition of observation profit, a central element in EOSSP, varies widely across studies, shaped by the unique objectives and targets of each Earth-observation mission. The optimisation frameworks employed to tackle the EOSSP are equally diverse, ranging from exact methods to heuristic and metaheuristic approaches, each chosen to best address the specific challenges posed by the problem at hand. In the following sections, these three factors will be investigated.

3.5.3. Modelling Assumptions on Mission Constraints

Different modelling assumptions lead to a different representation of the real-life problem. Which operational constraints to include, and the degree of accuracy with respect to the real system, have a big impact on the simulation. Based on a typical space mission, the different constraints that can be included are: communication-related, energy-related, memory-related, ADCS-related and thermal-related. Different researchers follow different approaches on how to model these constraints. A short review of how different authors include and model operational constraints is presented below.

Communication

The communication constraint has received a lot of attention by many researchers, as it is very relevant since a mission is only profitable if the data collected by the spacecraft is successfully sent to the ground. Various researchers approached this as a stand-alone problem, which has been referred to as the range satellite scheduling problem (RSSP), dealing with downlink of data and communications between ground stations and satellites [32] [63]. In such problems, the data link between a satellite and a network of ground stations needs to be modelled, representing both the uplink of commands to the spacecraft, and the downlink of scientific and telemetry data to the ground station.

One way to disregard this constraint and simplify the problem is to assume the usage of relay satellites to take care of the data downlink, as done by Rojanasoonthon [44]. Later research, such as that made by Cho et al. and by Spangelo, have tried to merge the two problems, formulating an integrated AEOSSP set-up, trying to find a set of observation- and corresponding data-downlink-tasks that maximises the mission return [7] [50].

Memory

Memory constraints are also taken into account by some research groups. The modelling of this constraint requires an assumption on the rate of produced data, together with a maximum onboard storage. Once the storage limit is reached, taking further observations does not lead to an increased observation profit.

There can be different approaches on how to model the acquisition of data. When the satellite is in a continuous observation mode, meaning that it is taking pictures at all times, a constant rate of acquired data can be assumed. This approached is followed by Qi et al. in their research that considers resources constraints, including memory and energy [42]. In the mission considered for this research, this assumption is representative for the surveillance mode. This is because in this mode, the satellite would keep a constant staring direction and a constant picture acquisition rate, thus making the linear increase in data generated feasible. On the other hand, this assumption might not be very representative for the tasked tracking case. This will be kept in mind, and different approaches will be tested in the simulation.

Energy

Energy constraints are included in the optimisation framework by many researchers [7] [50] [43]. This is done by modelling the energy produced by solar panels and a battery capacity, and assuming a certain energy consumption for each task. As sustained by Wang et al. (2020), data transmission is often not included in the energy budget, and it is assumed that there are enough resources for this [57].

Including this constraint is very relevant for satellites which are not equipped with rotating solar panels. If the solar panels are able to rotate, then they can power the spacecraft and recharge the batteries at all times, with the exception of eclipses, when the batteries would provide the power. If the solar panels are not able to rotate, which is the case for all body-mounted solar panels, then the solar panels can power the spacecraft and recharge the batteries only when the satellite is a certain attitude with respect to the Sun. For this reason, the spacecraft must periodically go into *sun-pointing mode*, where it points its solar panels towards the Sun to recharge the batteries. It then goes back to its nominal mode, where observations can be taken, until the batteries need to be recharged, and the cycle repeats. From this description, it is clear that this "energy cycle" can greatly affect the operations of a satellite.

Attitude

Attitude manoeuvre considerations have been added to this body of research in later years. Following the approach defined by Wertz, many researchers assume that the task transition time is composed of attitude manoeuvre time and attitude stabilisation time, and that these are known parameters [59]. Following this approach, the rotational dynamics of the spacecraft and its ADCS actuators does not need to be accurately modelled, nor does the performance of the control loop. Various researchers follow this idea, and incorporate attitude considerations in their work. The modelling of the spacecraft rotation is done by assuming a triangular or trapezoidal angular velocity profile, and the scheduling of the observation tasks takes into account the constraint of not having attitude manoeuvre conflicts [52] [58] [56].

The attitude manoeuvre time, and the stabilisation time depend on the choice of ADCS actuators, and the rotational characteristics of the spacecraft, such as its inertia matrix. On top of these, there is another operational constraint arising from ADCS actuators that was not considered in the inspected research. This is the desaturation, or momentum dumping, of the wheels onboard the spacecraft, involving the utilisation of supplementary actuators to counteract the wheels' angular momentum. When the spacecraft is performing momentum dumping, other operations might not be possible, and this might affect its capability to take pictures, ultimately reducing its observation profit.

Thermal

Constraints deriving from the thermal subsystem have not received much attention from researchers in this field, as it is thought to have a smaller impact on the operation scheduling than other subsystems. Despite this, Filho et al. have analysed the effects of including thermal considerations in the modelling

of spacecraft operations [17]. Their work focused on modelling the irradiation flux for a CubeSat in LEO in different attitudes, and analysing the impacts this had on the attitude, orbit, thermal balance, and power production capability of the spacecraft. The attitude and orbit of the CubeSat are affected by the irradiation flux through Solar Radiation Pressure (SRP), which is introduced in previous chapters. The thermal balance consideration is shown to have an impact on the amount of time the CubeSat can operate its subsystems. Each subsystem has a temperature range it can sustain, and if the temperature gets too high, the satellite might have to reduce its energy consumption to avoid raising the temperature even further. This means that for certain periods, only a few subsystems can be used. On the power production side, the performance of the solar cells depends on their temperature, producing less energy as their temperature increases. Based on the desired accuracy for the modelling of the solar panels, this effect can be included or not.

3.5.4. Definitions of Observation Profit

The way the observation profit is modelled also varies largely across literature. Since all the AEOSSP involve satellites with the ultimate goal of taking pictures and sending them back to Earth, the observation profit becomes of central importance, as it directly defines what the goal of the optimisation is.

The reason for the differences in observation profits comes from the fact that each Earth-orbiting satellite has a different mission, involving different targets to be observed. Each target has its own characteristics, and thus requires different observation strategies. For example, in Earth-observation, some targets might be too large to be fully observed in one frame, and thus require more than one picture to be taken to cover the full target. Following the naming convention of previous researchers, this large-scale target can be defined as area target [28] [53] [34]. Targets that can be observed fully in one picture are referred to as point targets. It can be desirable to observe such targets more than once, as done in the work of Wang et al. (2019) [56]. The reason for this can be to achieve time-series observations, which is also the case for optical space surveillance.

These different definitions play a big role in the outcome of the optimisation problem. The reason for this is that the observation profit is intrinsically part of the objective function to be optimised.

3.6. Optimisation Frameworks

Task scheduling problems are optimisation problems, and as such they can be formulated in many different ways. In this section, the main approaches employed in the AEOSSP are presented. Specifically, different definitions of the objective functions are presented in subsection 3.6.1, while subsection 3.6.2 presents several solution methods used in AEOSSPs. An overview of the procedure that can be used to make a decision on which objective function, or solution method to use is presented alongside each subsection.

3.6.1. Objective Function

The way in which the objective function for the optimisation problem is defined is one of the main drivers of the outcome.

The most straightforward way to define the function is as a single-objective function solely aiming at maximising one objective, which due to the nature of the problem is the observation profit. This approach was followed by Pemberton [40]. In the following years, other research groups followed this approach, and focused solely on maximising the observation profit.

In 2015, Tangpattanakul proposed a multi-objective function, defining the problem as a multi-user AEOSSP, with different users being interested in different targets [53]. Here, a multi-objective model was established, to address fairness between the users by concurrently 1) maximising the total observation profit and 2) minimising the profit difference between the users. Subsequently, most authors considered the AEOSSP as a multi-objective optimisation problem. Examples are Li et al. proposing a three-objective approach including profit, quality and timeliness, and Wang et al. (2019) proposing

a two-objective objective function maximising the observation targets and minimising the energy consumption for observation task transit [29] [56]. Other studies that employ multi-objective optimisation include [30] [62] [2] [13].

How to Choose an Objective Function

When setting up an optimisation problem, different objective functions can be developed. These are then evaluated by simulating the same models, each time using a different objective function. Shi et al., in their work comparing objective functions for community detection [47], use the same optimisation algorithms for each evaluation of a objective function. Then, some quantitative metric about each objective function is retrieved. Examples of such metrics are:

- *Convergence rate*: how quickly the optimisation problem converges to a solution. This metric intrinsically contains aspects related to the *computational complexity* of the objective function, which directly translates into how long it takes for the optimisation to converge;
- *Robustness*: the stability of the solution across different problem formulations. A robust objective function returns the same best solution for a number of problems with slightly different input parameters. On the contrary, a sensitive objective function does not stay stable across variations of the problem, but returns different solutions for different inputs;
- *Score diversity*: how spread the score of each solution is. A objective function that returns solutions with scores very close to each other does not allow for a good comparison. A good level of score diversity is thus desired for objective functions.

Other more specific metrics can be utilised, as shown in the aforementioned work of Shi et al. [47]. These metrics are specific to the type of problem at hand, and will thus be developed ad hoc when evaluating the objective functions. The chosen metrics can be used to evaluate the different objective functions. They can be combined to get a comprehensive view of the performance, as no single metric may capture all aspects of a objective function.

Note that more than one objective function, or combination thereof, can be used to find the best solution, or the best set of solutions for a problem. In this case, different objective functions are used to find optimal solutions, which are then stored. After this, the obtained solutions are analysed to gain insights on the problem.

3.6.2. Solution Methods

Many different solution methods have been used to tackle the EOSSP. These are: exact methods, heuristic, metaheuristic and machine learning. A quick overview of these classes of methods is given in this section, along with some examples of popular methods used in the EOSSP, and their main characteristics. Following this, a procedure that facilitates the selection of the most suitable solution method for addressing the problem analysed is provided. The work of Wang et al. (2020) provides a thorough review of the various solution methods used in the EOSSP, and was followed for this section [57].

It is worth mentioning that the AEOSSP is a highly combinatorial problem. This means that it is a problem where the solution space grows rapidly as the size of the problem increases. In such problems, evaluating all possible combinations might be unfeasible, as they would result in an excessive computational time. This is the reason why different researchers have employed different methods to tackle this problem. The methods can be divided into two main groups: those making use of hard computing, and those of soft computing. Hard computing refers to traditional computing methods that involve precise algorithms and mathematical models that find the globally best solution. Soft computing encompasses a set of computing approaches that deal with uncertainties, approximations, and partial truths. A schematic overview of these methods is depicted in Figure 3.9, which was made taking inspiration from Bara'a's work [6].

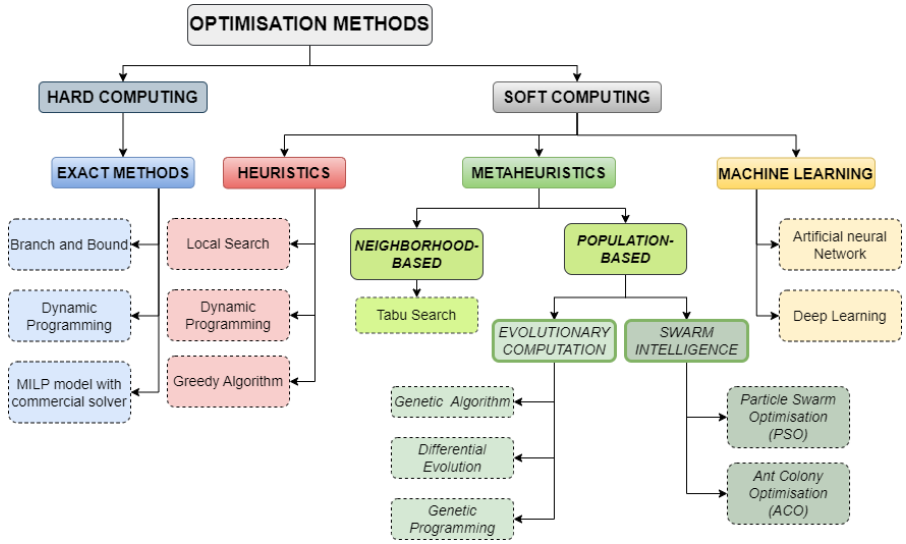


Figure 3.9: Overview of optimisation techniques used in the EOSSP research field.

Exact Methods

Exact methods aim at formulating the problem such that a unique and best solution can be identified. The exact methods that have been developed to this date are branch and bounds, and setting up a Mixed Integer Linear Programming (MILP) model. Gabrel was the first to employ a branch and bound method [19]. Cho used a MILP model developed as a commercial solver, to tackle a constellation of AEOS with various operational constraints [7]. She et al. regard the problem as a dynamical combinatorial optimisation problem, aiming at minimising the slew angle, while obtaining the highest observation profit [46].

Heuristic

Heuristic methods were used to speed up the process and find a satisfactory solution. As stated by Wang et al. (2020), such methods can typically provide high-quality solutions in a short time, even when complex operational constraints are considered [57]. Heuristics are problem-solving techniques that aim to quickly find good solutions, without the guarantee of optimality. They are based on being computationally efficient. These methods are used when traditional, hard computing methods are too slow.

Examples of such methods are constructive heuristics, that build a solution incrementally, step by step, and make decisions on how to construct the solution based on certain criteria. These methods often use a greedy approach, where at each step the algorithm makes a locally optimal choice. This means that at each timestep, the best available option is chosen, without considering the global consequence or future steps. This leads such methods to often converge to a sub-optimal solution.

Examples of heuristic methods used in the EOSSP are a greedy algorithm, dynamic programming and a local search algorithm used by Lemaitre, and a priority-based heuristic utilised by Wang et al. (2019) [28] [56].

Meta-heuristic

A metaheuristic optimisation method can be considered as a higher-level technique that seeks, generates or selects a heuristic method that may provide a sufficiently good solution to an optimisation problem [6]. They are general frameworks or strategies that guide the search for solutions across a broad class of optimisation problems, and can incorporate heuristics as components within their algorithms, to guide the search for a good solution. Differently from heuristics, metaheuristic methods are not tied to a specific problem, but can be adapted for use in various domains, providing a general approach to optimisation. Such methods were developed to address problems such as non-linearity

and multi-objectives, which is the case for the EOSSP [15]. In fact, metaheuristic methods make up the majority of the solution methods found in literature. Following the approach of Bara'a et al., these methods can broadly be categorised in neighbourhood-based and population-based algorithms [6].

Neighbourhood-based

Neighbourhood-based algorithms operate on the principle of iteratively exploring the solution space by examining the "neighbourhood" of current solutions. Starting from an initial solution, a neighbourhood is defined around the current solution, comprising solutions that are close or adjacent to the current one. These solutions are then evaluated based on an objective or objective function. This procedure is iterated until a certain convergence criteria is met. An example of such method is a "Tabu search", a method in which a concept of a memory is added to this algorithm, to avoid choosing solutions that were previously chosen. These techniques can be used together with other optimisation techniques, as it is the case for the work of Cui et al. [9], in which a Tabu search method is used jointly with an ant colony optimisation algorithm, explained below.

Population-based

Population-based algorithms are divided into two main classes, those based on evolutionary computation, and those on swarm intelligence. Evolutionary algorithms are inspired by the process of natural selection, and use populations of solutions and evolutionary operators, such as crossover and mutation, to iteratively improve the solutions. As stated by Wang et al. (2020), these methods have been used extensively with success in the scheduling domain. As such, many researchers in the AEOSSP have used these methods, such as the genetic algorithms used by Lemaitre and Du in their work [28] [12].

Swarm intelligence algorithms are inspired by the collective behaviour of social organisms, where a population of agents interacts locally to collaboratively search and optimize a solution space, with prominent examples being particle swarm optimisation (PSO) and ant colony optimisation (ACO). In these methods, each individual uses the collective knowledge of the total population to explore the solution space in a more efficient manner, leading to a better solution. He et al. employed an ACO method to tackle the AOESSP [25].

Machine Learning

In more recent years, machine learning approaches have been used in the AEOSSP. Examples are the works of Du et al. where data-driven techniques are used to find solutions [11] [12]. These cannot tackle operational constraints yet. For this reason, they are not deemed of high importance for the case considered for this research, and have only been reviewed for reference.

How to Choose a Solution Method

The process of choosing the most suitable solution method for the problem is similar to that for objective functions, explained in the previous section. Initially, the most promising methods can be shortlisted. This is done by looking in the existing literature at what methods were used to solve the problem at hand, and looking at the specific characteristics of the methods. Once several methods have been shortlisted, a quantitative comparison has to be performed. The metrics of *convergence rate* and *robustness*, explained in the previous section, can be used to compare solution methods. Additionally, the *scalability* of the method can be utilised. This is a measure of how well the algorithm performs when the size and complexity of the problem increases.

For a systematic comparison, it is imperative to employ the same objective function when executing different optimisation algorithms. This involves running all the shortlisted solution methods with identical objective functions, recording the performance metrics for each method, and conducting comparisons. This comparative process can be iterated across different objective functions, ensuring the selected method demonstrates enhanced robustness.

3.7. Collocation of the Thesis in the Broader Research Field

In this chapter, the theoretical background needed to fully understand the work performed in this research has been reviewed. It is thus now possible to collocate this research within the EOSSP research field that was identified during the literature study. This research considers a single, agile satellite, able to rotate about all axis, and focuses on practicality of the scheduling results. This is achieved by including operational constraints and combining the observation task scheduling problem with the downlinking scheduling problem, or RSSP. Additionally, this is done for a use-case in which the satellite's targets are orbiting Earth, as opposed to on its surface.

In the next chapter, the approach used to tackle this problem is presented.

4

Approach

In this chapter, the satellite scheduling problem considered for this research is defined, together with the description of the approach employed to solve it. The chapter is structured as follows. Section 4.1 presents the specific characteristics of the problem at hand. Next, the approach followed to model the dynamics of the resources onboard the spacecraft is presented in section 4.3. Section 4.4 examines the complications linked with scheduling problems, and proposes an approach to simplify the problem. To conclude the chapter, section 4.5 presents the complete pipeline of the method proposed in this thesis.

It is pointed out that the approach followed to tackle the problem, in particular that of simplifying the problem presented in section 4.4, is an adaptation of the work of Spangelo et al. *"Optimization-based scheduling for the single-satellite, multi-ground station communication problem"* [50]. In this work, the satellite is assumed to be by default in observation mode, and it has to be scheduled when the satellite has to downlink data, and to what ground station, so when it should be in *communication mode*. This problem was then adapted for the case of a satellite equipped with an optical telescope, tasked with space surveillance activities. The resulting framework was set-up such that additional operational modes than those considered in the work of Spangelo can be included, as is discussed in a following chapter.

4.1. Problem Description

The problem considered for this research consists in a single satellite orbiting the Earth, observing objects in Earth-bound orbits by means of an optical telescope, and downlinking scientific data to a number of ground stations. The totality of the latter are referred to as a Ground Station Network (GSN).

In the work of Spangelo, the aim of the problem is to devise a scheduling solution such that the satellite is able to downlink as much observation data as possible to the GSN, while satisfying certain operational constraints. These are the following:

- **Communication**: by scheduling the downlinking tasks of the satellite to a GSN;
- **Memory**: by including a limit on the data that can be stored onboard the satellite;
- **Energy**: by modelling the energy collected by the solar panels, the energy consumed by the satellite to perform each task, the maximum energy that can be stored onboard, and the maximum allowable Depth-of-Discharge (DoD), which is a measure of how much the battery can be depleted at every cycle;
- **Attitude**: by modelling the attitude dynamics of the satellite when performing the tasks defined by the scheduling solution.

Operational constraints due to thermal effects have not been investigated in this research.

The assumptions made for the satellite scheduling problem considered here are presented below, split into groups for ease of understanding.

The general description of the problem is as follows:

1. A single satellite orbits the Earth, and collects data, via an optical telescope, and energy via solar panels;
2. The satellite has a *default mode*, and the scheduling solution defines when it should go into a different *operational mode*;

The assumptions related to resource dynamics are as follows:

3. The satellite energy collection rate depends on the Line-of-Sight of the solar panels relative to the Sun;
4. The satellite energy consumption rate depends on its operational mode;
5. The satellite data collection rate depends on its operational mode;
6. The satellite data downlink rate depends on its operational mode;
7. Energy is required to perform nominal operations of the spacecraft. By *"nominal operations"*, it is meant the tasks that are carried out in the default mode of the satellite;
8. Energy is required to carry out tasks outside those included in the nominal mode. These include *communication tasks*;
9. The satellite has finite limits on the amount of energy and data that can be stored at any given time;

The assumptions related to the communication with ground stations are the following:

10. There are multiple ground stations as part of the GSN;
11. Each ground station periodically comes in- and out-of-view of the satellite. The period of time in which a ground station is visible is referred to as *communication window*;
12. The duration of these communication windows depends on the geometry of the orbit of the satellite relative to the ground station;
13. It can happen that more than one ground station is in view of the satellite at the same time. In this case, the satellite can downlink only to one ground station at the time;
14. Each ground station is assumed to have the same characteristics, in terms of rate at which they can receive data, energy utilisation from the satellite to send one bit of data, and the efficiency of the data downlink. The latter includes the losses due to communication system and transmission inefficiencies;
15. The satellite is capable of downlinking data to any visible ground stations for the entire duration of an interval or any fraction thereof.
16. The orbit of the satellite is assumed to be deterministic, and as such the communication windows to the GSN are known a priori;

To conclude, the assumption related to the ADCS subsystem is the following:

17. The single satellite is equipped with magnetorquers, such that it can constantly de-saturate its ADCS actuator. With this assumption, the angular momentum accumulated by the satellite along its orbit does not need to be taken into account.

4.2. Nomenclature

The outcome of the optimisation framework is an *operational schedule*, which is defined as the epochs in which the spacecraft should be in an operational mode different than its default mode. By doing this, the epochs in which the satellite should be in its default mode are automatically defined as well.

Thus, with the terms scheduling solution, or operational schedule, it is meant *"a timeline of modes the spacecraft should be operating in over time"*.

At every epoch, the satellite has the option to be in different operational modes, which can be referred to as *options*. The set of operational modes considered in a framework can be referred to as the *set of options*, O. The set of operational modes available at time t_i is referred to as $O(t_i)$. At time t_i, the operational mode the satellite is operating in is referred to as $o(t_i)$. An operational schedule is referred to as $o(t)$, as it defines what option, so what operational mode, the satellite should be in for every epoch.

The time for which the operational schedule is to be optimised, is referred to as *"mission scheduling horizon"* or *"mission planning horizon"*, and is indicated by $[0, T]$.

In the next section, the way in which the resources onboard the spacecraft are modelled is presented.

4.3. Resource Dynamics

In this section, the equations that govern the modelling of the onboard resources dynamics, namely energy and data, are presented. This is done by generalising the modelling approach proposed by Spangelo et al., in order to be able to include more operational modes than *observation* and *communication*. To allow for an easier understanding of how this is done, the modelling made by Spangelo et al. [50] is shown first, and then a method for generalising this is investigated.

4.3.1. Data Dynamics

In order to model the data dynamics of the spacecraft, it is helpful to identify the satellite's actions related to data dynamics. These are the following:

- Collecting data via optical telescope;
- Storing data via the on-board memory storage;
- Losing data due to data degradation and data expiration;
- Downlinking data to a ground station.

In Spangelo's work, given an initial memory storage level at the simulation start time, $t = 0$, and an operational schedule of when to perform downlinking tasks, the data stored in the memory at time t can be calculated as follows:

$$D(t) = min\left[D(t-\tau) + \left[\dot{D}^{+}(t-\tau) - \dot{D}^{-}_{loss}(t-\tau) - \dot{D}^{-}_{downlink}(t-\tau)\right] \cdot \tau, D_{max}\right] \tag{4.1}$$

where the terms in the equation, together with their definition and units, are presented in Table 4.1. The data loss defined in the table represents the data loss unrelated to data downlink (i.e. data degradation, expiration, ...).

Symbol	Definition	Unit
$D(t)$	Data storage level	Bits
τ	Unit of time approaching 0	Seconds
$\dot{D}^{+}(t)$	Rate of data collection at time t	Bits-per-second
$\dot{D}^{-}_{loss}(t)$	Rate of data loss at time t	Bits-per-second
$\dot{D}^{-}_{downlink}(t)$	Rate of data downlink	Bits-per-second
D_{max}	Data storage limit	Bits

Table 4.1: Definition of Terms in Data Dynamics Equation

At any time in the simulation, if the data level would exceed the memory capacity, then the excess is spilled. In the real physical system, this is equivalent to the satellite not storing the data produced.

$$D(t) > D_{max} \;\rightarrow\; H_D(t) = D(t) - D_{max} \tag{4.2}$$

Equation 4.1 can be generalised as follows:

$$D(t) = min\left[D(t-\tau) + \left[\dot{D}^{+}_{nom}(t-\tau) - \dot{D}^{-}_{nom}(t-\tau) - \dot{D}^{-}_{loss}(t-\tau) + \dot{D}^{+}_{o}(t-\tau) - \dot{D}^{-}_{o}(t-\tau)\right]\cdot\tau, D_{max}\right] \tag{4.3}$$

where the terms in red are linked with the default, nominal, operational mode considered in the problem. The term in cyan is related to data loss, which depends on the hardware specifications of the satellite and is thus independent from the operational mode. The terms in blue instead depend on the non-default operational mode of the satellite at time t.

4.3.2. Energy Dynamics

Similarly to the data dynamics, in order to model the energy dynamics of the orbiting spacecraft, it is helpful to identify the satellite's actions related to energy dynamics. These are as follows:

- Collecting energy via solar panels;
- Storing energy in the on-board battery;
- Consuming energy to support the default mode of the satellite;
- Consuming energy to support the non-default mode(s) of the satellite.

In Spangelo's work, given an initial battery State-of-Charge (SoC) at the simulation start time, $t = 0$, and defining a data downlink schedule, the energy stored in the battery at time t is computed as follows:

$$E(t) = min\left[E(t-\tau) + \left[\dot{E}^{+}(t-\tau) - \dot{E}^{-}(t-\tau) - \dot{E}^{-*}_{downlink}\cdot\dot{D}^{-}_{downlink}\right]\cdot\tau, E_{max}\right] \tag{4.4}$$

where the terms in the equation, together with their definition and units, are presented in Table 4.2.

Symbol	Definition	Unit
$E(t)$	Battery State of Charge (SoC)	Joules
τ	Unit of time approaching 0	Seconds
$\dot{E}^{+}(t)$	Rate of energy collection at time t	Joules-per-second
$\dot{E}^{-}(t)$	Rate of energy consumption for nominal (surveillance) functions	Joules-per-second
$\dot{E}^{-*}_{downlink}(t)$	Rate of energy consumption to downlink	Joules-per-bit
$\dot{D}^{-}_{downlink}(t)$	Rate of data downlink	Bits-per-second
E_{max}	Battery storage limit	Joules

Table 4.2: Definition of Terms in Energy Dynamics Equation

At any time in the simulation, if the energy level would exceed the battery capacity, then the excess will be spilled. In the real physical system, energy spillage is equivalent to the power generated by the solar panels being fed directly to the subsystems of the satellite. In case more power is generated by the solar panels than what is required by the satellite, and the battery is full, then the satellite the excess energy would be dispersed as heat, for example by means of radiators placed on the exterior of the spacecraft's body.

$$E(t) > E_{max} \;\rightarrow\; H_E(t) = E(t) - E_{max} \tag{4.5}$$

Equation 4.4 can be generalised as follows:

$$E(t) = min\left[E(t-\tau) + \left[\dot{E}^{+}(t-\tau) - \dot{E}^{-}_{nom}(t-\tau) - \dot{E}^{-}_{o}\right]\cdot\tau, e_{max}\right] \tag{4.6}$$

where the term in green depends on the satellite's illumination condition and its solar panel's orientation with respect to the Sun. The red term depends on the energy consumption linked with the default

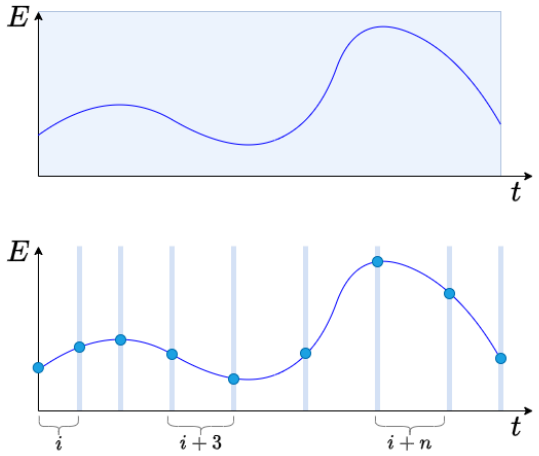


Figure 4.1: Example of going from continuous-time (top) to discrete-time (bottom).

mode of the satellite, while the term in blue depends on the non-default operational mode the satellite is functioning at time t.

How these equations can be set-up to accommodate for more operational modes to be considered is shown in chapter 6.

4.4. Simplifying the Problem

Scheduling problems often encounter problems due to two main causes. These are presented below, together with the problem they lead to.

- Too many decision variables $\rightarrow$ Computationally intractable problems;
- Discontinuities in the resource dynamics $\rightarrow$ Non-linear optimisation problems.

Both these issues result in optimisation problems that are difficult to solve. In this section, the approach proposed by Spangelo et al. [50] to tackle this problem is presented.

4.4.1. Under-Constrained Formulation

Scheduling problems consist in choosing one option amongst a number of available ones. If a problem is not discretised, and the time is considered as continuous, at every moment a decision has to be made. In the case of a numerical simulation, this would mean making a decision at every epoch of the simulation. This quickly leads to the problem size becoming too large, and difficult to solve. For this reason, the decision space has to be restricted. This is achieved by discretising the problem into a finite set of time periods. A visual example of what this means can be seen in Figure 4.1. In both plots, the energy profile is depicted, with the blue line representing the battery SoC. In the plot on top, the light-blue background symbolises that the resource constraint is applied on the continuous-time, so for every value of t. In the plot below, the simulation time is discretised into intervals, indicated with the letter i, and gradually increasing until $i+n$. In this case, the constraints are applied only at the discrete locations indicated by the light-blue vertical lines, with the blue dots indicating where the energy level will be checked against its constraints. This means that the constraints are imposed only at the start and end of each time interval, and *the resource dynamics within each interval are ignored*.

Since the resource constraints are no longer applied to the continuous-time, but only at certain epochs, this problem formulation can be referred to as an Under-Constrained Formulation (UCF) problem. The solution found by solving this problem can be referred to as a UCF solution.

As the resource dynamics within each interval have been ignored, it might happen that the constraint is violated. This makes the UCF solution infeasible in the continuous-time. This means that, in practice,

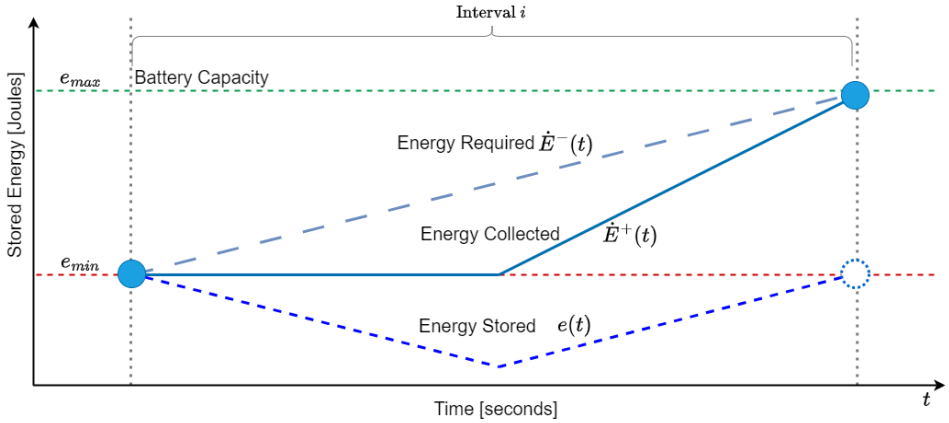


Figure 4.2: Single-interval example of the communication problem, where a feasible Under-Constrained Formulation results in infeasibilities when applied to the continuous-time dynamics.

the operational schedule would be impossible to implement. An example is shown in Figure 4.2. In this single-interval example, the UCF solution only applies the constraint on the stored energy level at the start and end of the interval. In the example, this means that the stored energy level is only checked where the light blue dots are, and the dynamics within the interval are ignored. Assume that the interval is 10 seconds long, and that the battery is empty at the start of the interval, $e(t) = 0$. The energy collected during the interval is of 0 Joules/second for the first half of the interval and of 2 Joules/second for the second half of the interval. This could be the case when the satellite is coming out of an eclipse window. Since the UCF solution only looks at the boundaries of the interval, it could output a solution that requires an energy consumption of 1 Joules-per-second for the whole interval. According to this solution, the energy collected throughout the interval is equal to the energy consumed, and thus the solution is feasible. When considering the continuous-time dynamics, it becomes apparent that this solution is not feasible. The reason for this is that for the first half of the interval, energy is required by the solution, but no energy is being collected. In practice, this means that the satellite would not be able to operate according to the planned schedule, due to a lack of energy stored on the onboard battery. This example, taken from the work of Spangelo et al. [50], makes it clear that a method is needed to check the feasibility of the solution in the continuous-time, to ensure that the constraints are respected throughout the complete scheduling horizon.

4.4.2. Assuming Piece-Wise Linear Dynamics

The reason why the UCF solution of the previous example was infeasible lies in the discontinuity in the energy collection rate within the interval. If the rate of energy collection would have been constant throughout the interval, the UCF solution would have been feasible in the continuous-time dynamics. This is because if the rates of energy and data collection, energy consumption, data downlink and data loss are all constant with respect to time throughout every interval of the scheduling horizon, then the UCF solution will satisfy the continuous-time dynamics. This is a sufficient condition to yield a feasible, and thus optimal, solution to the scheduling problem. A mathematical proof of this theorem can be found in Appendix A.

This concept can be understood visually by analysing Figure 4.4, which shows an idealised example of a single interval in which linear dynamics are assumed for the energy and data, respectively. From these plots it becomes apparent how, if the resource level at the start, $E(t_1), D(t_1)$, and end of each interval, $E(t_2), D(t_2)$, are both within the given limits, and a straight line connects these points, then the resources at every point within the interval will be also within the limits.

Following this reasoning, it makes sense to split time into intervals where these resource rates are constant. This is shown in Figure 4.3. In this figure, it can be seen how the smooth non-linear dynamics in the plot on top is assumed to be linear in the plot at the bottom. The way in which this is achieved in the satellite scheduling problem is presented in the next section.

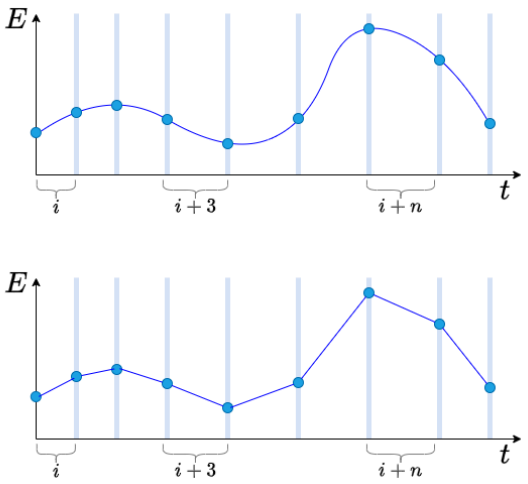


Figure 4.3: Going from discretised problem with non-linear dynamics (top) to a discretised problem with piece-wise linear dynamics (bottom).

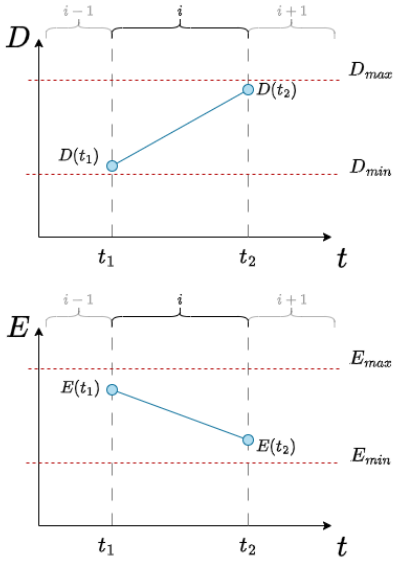


Figure 4.4: Single interval with an assumed linear dynamics for the data and energy profile

4.4.3. Dividing Time into Intervals

In order to simplify the problem and make it easier to solve, a set of intervals, I, needs to be defined within the planning horizon, $[0, T]$. As explained in the previous section, the intervals should be chosen such that the energy and data rates are constant. It thus makes sense to find intervals in which the following are constant:

- Data downlink rate, $\dot{D}^-$;
- Data collection rate, $\dot{D}^+$;
- Energy collection rate, $\dot{E}^+$;
- Energy consumption rate, $\dot{E}^-$.

The data loss rate is assumed to be independent of the operational mode, and will thus be constant regardless of how the intervals are chosen.

The first three depend on what operational mode the spacecraft is carrying out, as each mode is linked with a certain resources dynamics profile. In Spangelo's research, the two operational modes considered are SURVEILLANCE and COMMUNICATION, and result in the following profiles:

- Resources Dynamics Profile$|_{COMM} = \left[\dot{E}^-_{COMM},\ \dot{D}^-_{COMM},\ \dot{D}^+_{COMM}\right]$
- Resources Dynamics Profile$|_{SURV} = \left[\dot{E}^-_{SURV},\ \dot{D}^-_{SURV},\ \dot{D}^+_{SURV}\right]$

These profiles depend on which subsystems of the spacecraft are active during a certain operational mode. This will be explained in more detail in chapter 8, when discussing results.

Based on this information, it thus makes sense to **split time into intervals based on when the spacecraft can change mode**.

When considering SURVEILLANCE and COMMUNICATION modes, the spacecraft is always able to be in the former, since it is the default mode. Instead, the latter mode is only available when one (or more) ground stations are in view of the spacecraft. It thus makes sense to split the time based on when the visibility condition of a ground station changes. This is depicted in Figure 4.5.

The energy collection rate term, instead, depends on the illumination condition and orientation of the satellite. It thus makes sense to **split time into intervals based on when the illumination condition**

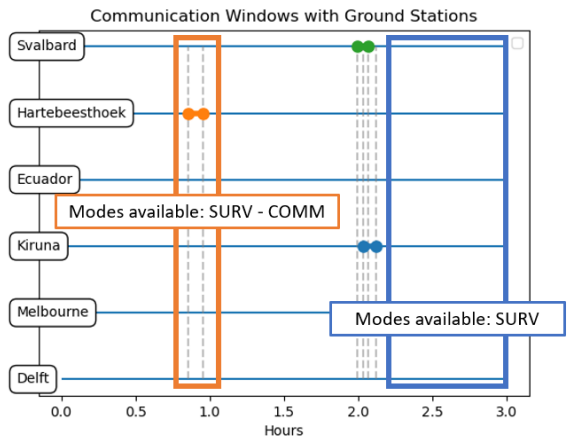


Figure 4.5: Communication windows between the ground stations part of the GSN and the observer satellite, with the available modes identified for two of the intervals.

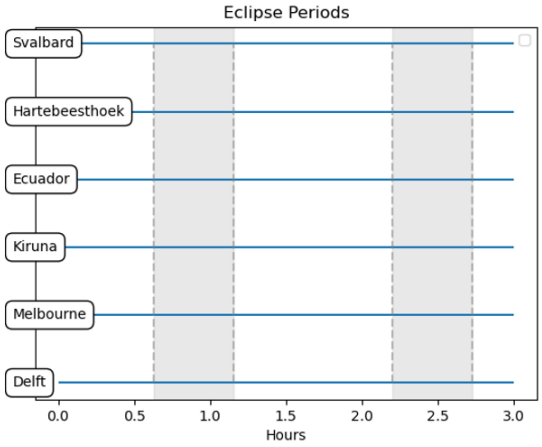


Figure 4.6: Eclipse windows for the observer satellite. The horizontal lines for the ground stations are plotted to allow for an easier comparison with the plot on the left.

of the spacecraft changes, as shown in Figure 4.6. In this research, two ways in which this can be achieved are proposed. The most simple is presented below. A second, more accurate method, will be presented in chapter 7.

Discretisation Scheme A

In this discretisation scheme, the energy collection rate is assumed constant when the satellite in in sunlight, and is assumed 0 when the satellite is in eclipse. The events that trigger the start of a new interval are:

- Whenever a ground station comes in- or out-of-view of the satellite;
- Whenever the illumination condition of the satellite goes from eclipse to non-eclipse, or vice-versa.

This can be seen in Figure 4.7. It can be seen now how the scheduling horizon can be split in such a way that makes the rates of change of the resources, namely energy and data, constant within a single interval.

Note that to perform the division in intervals following discretisation scheme A, there is no need to propagate the rotational dynamics of the satellite, since the eclipse condition is purely dependent on orbital dynamics. This division thus does not depend on the pointing strategy employed for surveillance mode.

After the scheduling horizon has been split into a set of intervals, I, within each interval, i, linear resource dynamics can be assumed. This means that, defining the initial energy and data levels at the start of the simulation, $E_{i=0}$ and $D_{i=0}$, the energy and data levels can be updated recursively as follows:

$$E_{i+1} = E_i + E_i^+ - E_i^- \tag{4.7}$$

$$D_{i+1} = D_i + D_i^+ - D_i^- \tag{4.8}$$

where the $i+1$ subscript refers to the *next* interval, and i refers to the *current* interval. The increase or decrease of the resource levels are then calculated as follows:

$$E_i^+ = \dot{E}_i^+ \cdot \Delta t_i \tag{4.9}$$

$$E_i^- = \dot{E}_i^- \cdot \Delta t_i \tag{4.10}$$

$$D_i^+ = \dot{D}_i^+ \cdot \Delta t_i \tag{4.11}$$

$$D_i^- = \dot{D}_i^- \cdot \Delta t_i \tag{4.12}$$

$$\tag{4.13}$$

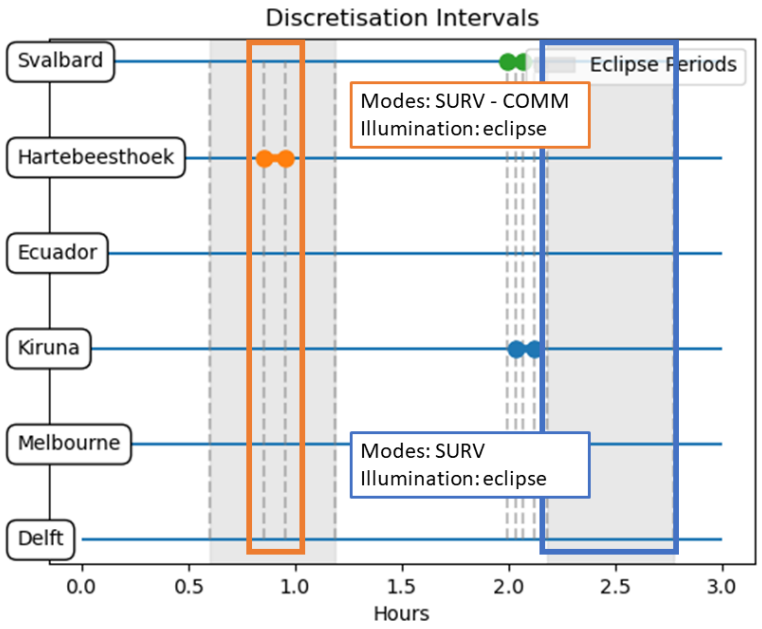


Figure 4.7: Division in intervals according to discretisation scheme A, with available modes and illumination condition indicated for two of the intervals.

where Δt_i is the duration of the current interval. As mentioned earlier, $\dot{E}_i^-$, $\dot{D}_i^+$ and $\dot{D}_i^-$ depend on the operational mode of the spacecraft, and $\dot{E}_i^+$ depends on its illumination condition. Again, these should be constant within an interval, as a result of how the scheduling horizon was split into intervals. By recursively solving the equation at every interval, a data and energy profile can be obtained. By analysing these profiles, the feasibility of the solution can be checked.

The Under-Constrained Formulation problem is now set-up. How this is integrated into a pipeline is explained in the following section.

4.5. Pipeline

This section deals with the pipeline of the approach used to solve the space-observing satellite scheduling problem. By pipeline, it is meant the collection of steps taken to go from the problem inputs to a solution.

The pipeline is depicted in Figure 4.8. The inputs are first used to discretise the planning horizon into a set of I_K intervals. Then, the UCF problem is solved by assuming linear resources dynamics over the intervals. This is done using a Mixed-Integer Linear Programming (MILP) algorithm, which will be explained in the following chapter. By solving the MILP problem, the UCF solution is generated. The latter is only guaranteed to be feasible at discrete times, as the resource dynamics are checked only at the boundaries of the intervals. Thus, the UCF solution needs to be checked for feasibility considering continuous-time dynamics.

If the solution is feasible, then this approach produced an optimal scheduling solution when considering continuous-time dynamics, which means that the operational schedule could be implemented in practice. If instead the continuous-time check results in the UCF solution not being feasible, a different discretisation scheme can be used, to try get a feasible solution with the given inputs.

If the UCF solution that made use of the newly discretised intervals is unfeasible, then a different discretisation scheme could be used, to try improve the solution. In case this step also fails, then the inputs could be changed, to try to come up with a different mission scenario or spacecraft characteristics that results in a feasible schedule. This is deemed outside the scope of this thesis and is therefore left as a recommendation for future work to continue and enhance this research. This thesis thus focuses on

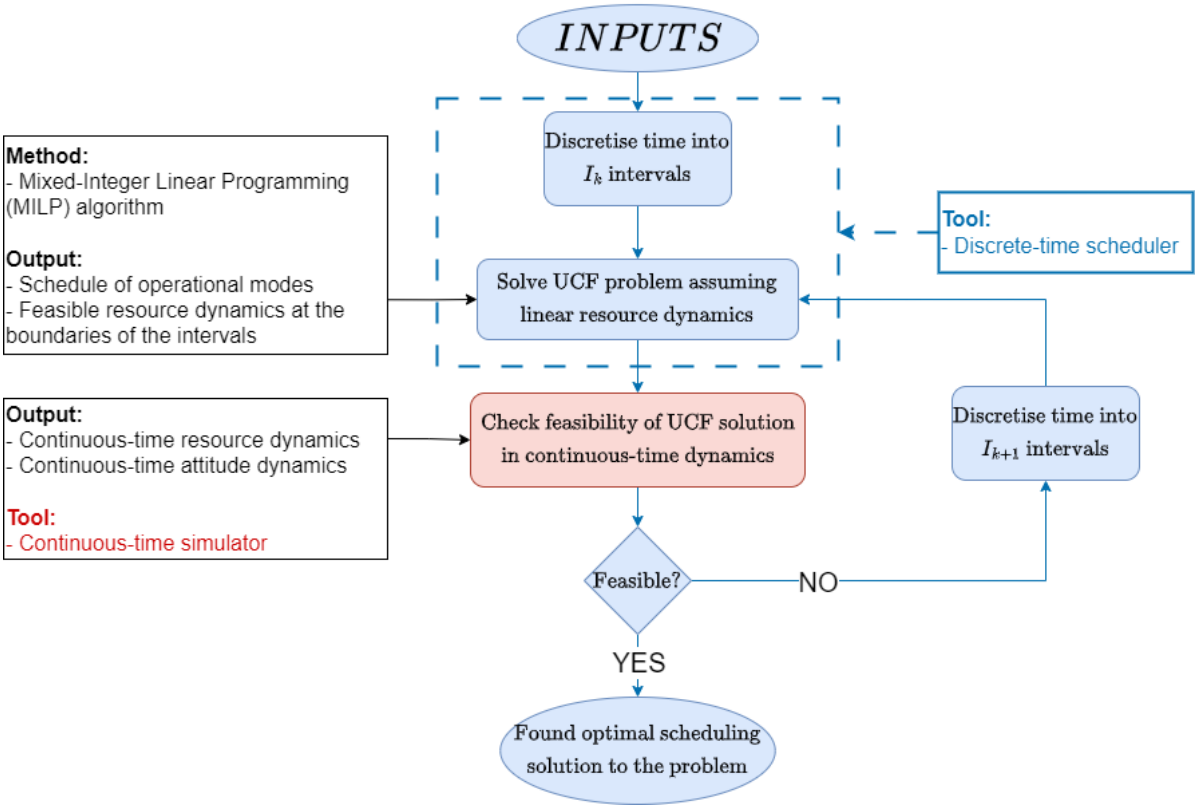


Figure 4.8: Pipeline of the approach followed

setting up a framework that allows to find feasible scheduling solutions given certain mission scenarios and constraints, as well as how sensitive this framework is to the various inputs. The optimisation of the inputs themselves are left for future work.

The pipeline presented above requires two tools to be carried out. These can be referred to as a *discrete-time scheduler* and a *continuous-time simulator*, namely.

The discrete-time scheduler takes as input the problem parameters, and outputs a schedule of operational modes, which is feasible at the boundaries of the intervals. It comprises of the discretisation into intervals, as well as the setting-up and solving of the Under-Constrained Formulation (UCF) problem. This tool will be described in the next chapter.

The continuous-time simulator needs to take as input the problem parameters, and the schedule of operational modes produced by the scheduler. It then needs to propagate the continuous-time attitude and resource dynamics, in order to check for the feasibility of the schedule.

These tools will be described in more detail in chapters 5 and 7, respectively.

5

Discrete-Time Scheduler

In this chapter, the discrete-time simulator tool introduced in chapter 4 is presented.

5.1. Overview

The discrete-time scheduler has to take as input the problem parameters, discretise the simulation time into intervals, set-up the Under-Constrained Formulation problem, and solve it using a MILP algorithm.

In order to achieve this, it was decided to use the python library tudat(py) (TU Delft Astrodynamics Toolbox), which is a set of libraries with built-in astrodynamics-modelling capabilities [8]. This library was chosen as it allows to easily propagate the translational dynamics of bodies around the Earth, bodies on the surface of the Earth, which can represent the ground stations part of the GSN, and celestial bodies. On top of this, tudat offers the capability to internally calculate when communication between a satellite orbiting the Earth and a single ground station is possible. Additionally, it allows for a minimum elevation angle constraint to be implemented. This means that the communication windows between a satellite and a ground station are easily determined. Combined with the flexibility offered by python, these reasons resulted in the decision to use tudat for this tool.

In the next sections, the various parts that make-up the workflow of the discrete-time scheduler are presented.

5.2. Workflow

In this section, the different steps that make-up the discrete-time scheduler are presented.

5.2.1. Create Orbital Environment

The first step is to take the problem parameters as input, and create the orbital environment in tudat. By orbital environment it is meant an environment where all the bodies involved, with the exception of the celestial bodies, are considered as point masses. In this case, this means that the observer satellite and all ground stations are considered as point masses, and as such only their translational dynamics are propagated. This is done as the rotational dynamics are not needed in the discrete-time scheduler.

The input parameters needed are presented in Table 5.1. In the table, by ”S/C” it is meant ”spacecraft”, and LLA stands for Latitude-Longitude-Altitude.

5.2.2. Split Time into Intervals

After the orbital environment has been propagated, the scheduling horizon has to be split into intervals. As described in the previous chapter, the first, simple, discretisation scheme that is used to generate a

Table 5.1: Input parameters needed for the discrete-time scheduler.

Sim Settings	GSN	S/C Physical Properties	S/C Initial State
Time-step	Ground Stations LLA	Dry mass	
Start Date	Min Elevation Angle	Drag coefficient	
Duration	Min Duration	Reference area	

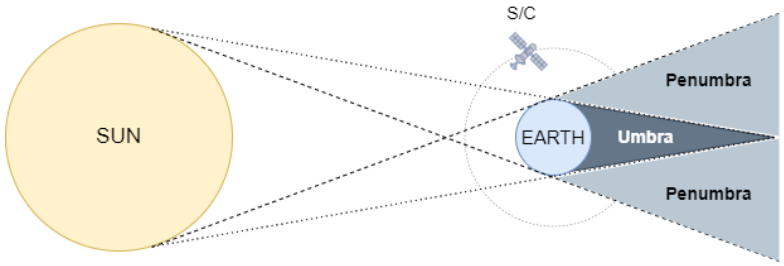


Figure 5.1: Sun-Earth-Satellite system for the determination of illumination condition of the satellite [27].

solution makes use of the communication windows between the observer satellite and the GSN, and the eclipse periods. Thus, these two quantities have to be calculated from the orbital environment.

For the determination of the ground station communication window, tudat's built-in functions for state estimation can be used. This is achieved by simulating observations between the ground station and the satellite. When modelling observations between two bodies, tudat allows for observation constraints to be applied. Amongst these constraints, a minimum elevation angle can be applied on the communication link between each ground station and the observer satellite. Tudat then internally simulates the observations between the satellite and each ground station, and returns the epochs at which communication was possible, effectively returning the communication windows.

The result of this can be seen in Figure 5.2.

Note that a minimum duration threshold is imposed on what communication windows to consider when splitting time into intervals. This is to avoid that the optimisation algorithm outputs solutions in which the satellite downlinks data only for a short time, as this is not desirable. This is a way to define observation profit for the problem, as short communication windows are deemed not profitable for the mission.

For the determination of the eclipse periods, the algorithm proposed by Xianghua et al. [27] is implemented. This algorithm takes as inputs the positions of the Sun and of the satellite, both expressed in the ECI frame, and some constant parameters, namely the radius of the Sun and of the Earth, and their distance. As a result, it outputs the illumination condition of the satellite, differentiating between *sunlight - umbra - penumbra*. The geometry of the problem is shown in Figure 5.1, and the resulting eclipse windows can be seen in Figure 5.2.

After determining the communication and eclipse windows, these are overlapped, in order to obtain the result shown in Figure 5.2. As shown in the figure, and following the approach described in the previous chapter, time has now been split into intervals in which the illumination condition, and the available operational modes are constant. In this way, the energy and data rates within each interval can be assumed as constant. The next section deals with the procedure of mathematically setting up the optimisation problem, using the intervals defined in this step.

5.2.3. Mathematical Formulation

After splitting the scheduling horizon into intervals, the Under-Constrained Formulation (UCF) problem can be expressed as a Mixed-Integer Linear Programming (MILP) problem. The notation used in this formulation is taken from the work of Spangelo et al. [50].

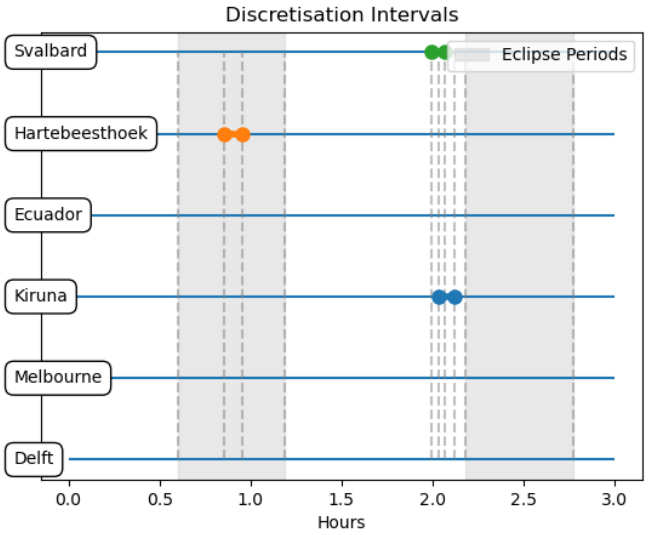


Figure 5.2: Intervals resulting from overlapping the communication and eclipse windows.

Parameters

The parameters used in the mathematical formulation are presented below. They follow the notation for which I is the set of intervals, O is the set of operational modes available and $O_i \subseteq O$ is the subset of the operational modes available during interval i, $\forall\, i \in I$. Using this notation results in the following parameters:

- $\eta_{i,o}$: efficiency of data downlink during interval i when downlinking using option o;
- t_i is the start time of interval i;
- t_{i+1} is the end time of interval i, which coincides with the start time of interval $i+1$;
- Δt_i is the duration of time interval i, and is equal to $t_{i+1} - t_i$;
- $\phi_{i,o}$ is the data rate associated with downlinking during interval i using option o;
- $\alpha_{i,o}$ is the energy per data associated with downlinking using option o during interval i;
- e_{min} and e_{max} are the minimum and maximum allowable amounts of energy that can be stored in the battery;
- e_{start} is the amount of energy stored in the battery at the beginning of the scheduling horizon;
- d_{min} and d_{max} are the minimum and maximum allowable data stored in the onboard memory;
- d_{start} is the amount of data stored in the onboard memory at the beginning of the scheduling horizon;
- δ_i^{e+} is the total amount of energy that can be acquired during interval i;
- δ_i^{e-} is the total amount of energy consumed during interval i for default operations (i.e. surveillance mode);
- δ_i^{d+} is the total amount of data that can be acquired during interval i;
- δ_i^{d-} is the total amount of data that is lost during interval i, independent of the operational mode. This includes for example data degradation and expiration.

Note that δ_i^{e-} includes both the energy required to operate the optical payload to take pictures, and that required to operate the life-supporting subsystems onboard the spacecraft.

Variables

The variables used in the problem are the following:

- $x_{i,o} \in 0, 1$: binary variable indicating the operational mode during some or all of interval i;
- $q_{i,o} \in \mathbb{R}^+$ is the amount of data downlinked during interval i while in operational mode o, $\forall i \in I, o \in O$;
- $e_i \in \mathbb{R}^+$ is the amount of energy available at the beginning of interval i, measured in Joules, $\forall i \in I$;
- $d_i \in \mathbb{R}^+$ is the amount of data available at the beginning of interval i, measured in bits, $\forall i \in I$;
- $h_i^e \geq 0$ is the amount of excess energy spilled during interval i, measured in Joules, $\forall i \in I$;
- $h_i^d \geq 0$ is the amount of excess data spilled during interval i, measured in bits, $\forall i \in I$.

Formulation

Using the parameters and variables defined above, the optimisation problem can be now formulated mathematically as a Mixed-Integer Linear Programming (MILP) problem.

Objective function:

$$\max \sum_{i \in I} \sum_{o \in O_i} \eta_{i,o} q_{i,o} \tag{5.1}$$

Subject to:

$$\sum_{o \in O_i} x_{io} \leq 1 \quad \forall i \in I \tag{5.2}$$

$$q_{i,o} \leq \Delta t_i \phi_{i,o} x_{i,o} \quad \forall i \in I, o \in O_i \tag{5.3}$$

$$e_0 = e_{start} \tag{5.4}$$

$$e_{min} \leq e_i \leq e_{max} \quad \forall i \in I \tag{5.5}$$

$$e_{i+1} = e_i + \delta_i^{e+} - \delta_i^{e-} - \sum_{o \in O_i} \alpha_{i,o} q_{i,o} - h_i^e \quad \forall i \in I \tag{5.6}$$

$$d_0 = d_{start} \tag{5.7}$$

$$d_{min} \leq d_i \leq d_{max} \quad \forall i \in I \tag{5.8}$$

$$d_{i+1} = d_i + \delta_i^{d+} - \delta_i^{d-} - \sum_{o \in O_i} q_{i,o} - h_i^d \quad \forall i \in I \tag{5.9}$$

$$x_{i,o} \in \{0, 1\} \quad \forall i \in I, o \in O_i \tag{5.10}$$

$$q_{i,o} \in \mathbb{R}^+ \quad \forall i \in I, o \in O_i \tag{5.11}$$

The objective function, Equation 5.1, aims at maximising the total amount of data downlinked over the scheduling horizon.

The constraints can be explained as follows:

- Equation 5.2 enforces that the satellite can downlink to one ground station at most per interval;
- Equation 5.3 enforces: 1) that data can be downlinked only using the chosen ground station, and 2) that the amount of data downlinked is limited by the duration of the interval, Δt_i and the data downlink rate of the chosen ground station, $\phi_{i,o}$;

- Equation 5.4 initialises the amount of energy stored in the battery at the start of the planning horizon;
- Equation 5.5 enforces that the energy at the start of every interval is within the battery limits;
- Equation 5.6 defines the amount of energy at the beginning of an interval to be equal to:
  - The amount stored at the start of the previous interval (+)
  - The amount acquired via the solar panels (-)
  - The amount consumed for nominal operations (-)
  - The amount used to downlink data (-)
  - The amount spilled
- Equation 5.7 initialises the amount of data stored in the onboard memory at the beginning of the scheduling horizon;
- Equation 5.8 enforces that the data at the start of every interval is within the acceptable memory limits;
- Equation 5.9 defines the amount of data at the beginning of an interval to be equal to:
  - The amount stored at the start of the previous interval (+)
  - The amount acquired (-)
  - The amount lost due to degradation or expiration (-)
  - The amount downlinked (-)
  - The amount spilled
- Equation 5.10 enforces the decision variables, $x_{i,o}$ to be binary;
- Equation 5.11 enforces the amount of downlinked data to be a positive, real number.

Two comments can now be made on the effects that the modelling choices made in the definition of the problem have on the fidelity of the results.

Energy Consumption per Data

Looking at the equation that governs the updating of the energy level at every interval, Equation 5.6, it can be seen that the amount of energy consumed for downlinking data depends on the amount of data downlinked. As the latter is expressed in $bits$, this means that the rate of energy consumption for downlinking data has to be expressed in $Joules/bit$. This measure of "energy consumption per data downlinked" is typically not provided as part of the characteristics of the hardware onboard the spacecraft. The determination of this parameter using the power consumption profile of the spacecraft in different operational modes is presented in chapter 6.

Mismodelling of Data Acquisition

The equation that governs the updating of the data level at every interval, Equation 5.9, also introduces an inaccuracy with respect to the physical system. In the equation, the rate of data acquisition for nominal operations, δ_i^{d+}, is multiplied by the total duration of the interval, Δt_i. This means that the spacecraft is assumed to be observing data continuously throughout the interval, regardless of the time the satellite spent in communication mode. The reason for this is that the time spent downlinking can be calculated only *after* the problem has been solved, as it is calculated by dividing the amount of data downlinked by the data Downlink rate. Thus, it is impossible to calculate the *effective observation time* while setting up the problem. As a result of this, the outputted schedule will overestimate the amount of observation data produced. The effects that this has on the fidelity of this approach are discussed in chapter 8.

5.3. Optimisation Algorithm

The optimisation problem defined in the previous section is solved as a Mixed-Integer Linear Programming (MILP) problem. The main characteristics of this method are explained in this section.

Table 5.2: Characteristics of the numerical model used for the discrete-time simulator, based on the analysis performed in Appendix E.

Characteristic	**Value**
Integrator type	Runge-Kutta-Felhberg
Order of integrator	4(5)
Time-step type	Fixed
Δt	10 seconds
State representation	Cowell
Accelerations included	Earth SH gravity Moon PM gravity Sun PM gravity Aerodynamic drag
Earth SH Degree and Order	(16,16)
Atmospheric Model	US-76
Benchmark Integrator	RKF7(8), $\Delta t = 4s$

MILP is a mathematical method used to model and solve optimisation problems that involve both continuous and discrete variables. In this method, the decision variables can be either integer or real numbers, and the objective is to optimise, minimise or maximise, a linear function subject to a set of linear constraints. This technique is highly effective in operational research, which often deals with complex decision-making processes that require a mix of quantitative and qualitative data. MILP is particularly suitable because it can handle the binary nature of decision variables, constraints of limited resources, and various operational rules which are linear or can be linearised.

Thus, MILP provides a structured way to approach the satellite scheduling optimisation problem, leveraging its capability to handle diverse and complex sets of linear constraints, which is pivotal for managing the intricate dynamics and operational restrictions of space missions. This makes MILP a highly suitable tool for enhancing the efficiency and effectiveness of satellite task scheduling in space situational awareness missions. For this reason, it was decided to solve the problem using a MILP algorithm.

In practice, the MILP was implemented using the Gurobi optimiser. Gurobi is a state-of-the-art optimisation solver specifically designed to solve large-scale linear, integer, and mixed-integer linear programming problems [22]. Recognised for its robust performance and efficiency, Gurobi leverages advanced mathematical and computational technologies to deliver rapid solutions to complex optimisation challenges. It is particularly well-suited for mixed-integer problems, where the ability to handle both continuous and discrete variables is essential. Gurobi's Python library, `gurobipy`, was used, allowing integration with the existing architecture of the discrete-time scheduler, which is fully implemented in python. The algorithm used by the Gurobi optimiser to solve the MILP is the branch-and-cut algorithm, which is the default for solving MILP problems [23].

5.4. Model Selection

The complete analysis performed to select the different aspects of the model for the discrete-time scheduler is shown in Appendix E. Here, a summary of the final model and its characteristics is reported, shown in Table 5.2.

5.5. V&V

In this section, the validation steps taken for the discrete-time scheduler are presented. The following steps will be made concerning validation:

- Results on the computational speed of solving the MILP will be compared with those presented by Spangelo et al. [50]. This will be presented in chapter 8;
- The operational schedule outputted by the discrete-time scheduler will be considered validated if the schedule gives realistic results. A realistic result is a schedule that can be satisfied with typical on-board resources. The results of this are presented in chapter 8;
- Intermediate results needed to set up the MILP problem will be used to validate the results obtained with tudat. These are the eclipse periods, and the ground station communication periods.

The General Mission Analysis Tool (GMAT) [20] has the capability to compute the eclipse periods of a spacecraft, and its communication windows with a set of ground stations. This is obtained by using its `contact locator` and `eclipse locator`. These output a .txt file, stating the epochs during which the spacecraft is visible with respect to a certain ground station, or the epochs during which the spacecraft is in umbra or penumbra condition. The information stored in these .txt files can be compared to the communication and eclipse windows calculated using tudat. Since these depend on the position of the propagated bodies (satellite and ground station for the communication windows, satellite and Sun for the eclipse periods), by checking the accuracy of the time windows, the accuracy of the propagated position is also validated.

Before comparing the results outputted by tudat and GMAT, it is necessary to ensure that the simulation settings are the same, to ensure a proper comparison of the results. An overview of the different settings is shown below:

- **Frame-origin**: Earth
- **Frame orientation**: EarthMJ2000Eq
- **Integrator**: RKF4(5) is not available in GMAT. RKF5(6) is instead available. Looking at the result of Figure E.10, it can be seen that the error behaviour of the RKF5(6) integrator is better the needed threshold. For this reason, it was decided to use this integrator, with a time-step of 10 seconds;
- **Accelerations**: Following from the results shown in Table 5.2, the following accelerations should be included: Earth Spherical Harmonic gravity, with degree and order up to (16,16), Moon and Sun Point Mass gravity and aerodynamic drag due to Earth's atmosphere. The Sun and Moon PM are implemented in the same way in both software, as they do not depend on a model. The gravitational pull of the Earth, and its aerodynamic drag, instead depend on the chosen model. For the gravity model of the Earth, the models readily available are: *EGM-96*, *JGM-3* and *JGM-2*. The most up-to-date model that is available also in tudat is the *EGM-96*. For this reason, both models were run with the *EGM-96* gravity model. In GMAT, the available atmospheric models are the *MSISE90* and the *JacchiaRoberts*. These are not available in tudat, as the only predefined atmospheric model available is the NRLMSISE-00. Following the discussion made in Appendix E, the position error due to uncertainty in the atmospheric model easily go above the desired threshold. For this reason, since only different models could be used between the two software, it was decided not to include the atmospheric drag.
- **Time-related parameters**: The simulation was set to start on the 1st of January 2025, at 00:00:00

The use-case considered for this comparison is the following:

- Single Satellite
- Altitude = 500 km
- Inclination = 97.8 $^\circ$ (SSO)
- Eccentricity = 0
- Simulation time = 7 days

- Fixed step-size = 10 seconds

Note that in GMAT, it is not possible to specify 0 for some of the Classical Orbital Elements (namely the eccentricity and true anomaly), while this is the case in tudat. For this reason, very small values, such as 10^{-6}, were used for some COEs for the initial state of the satellite in GMAT. The dry-mass of the satellite was set to 150 kg, its reference area for the drag to 1 m^2, and its drag coefficient to 2.0.

5.5.1. Ground Station Communication

The minimum elevation angle was set to 15° for both software. Also, the same Ground Station Network (GSN) was implemented, with 15 ground stations spread across the globe.

The `contact locator` of GMAT outputs a .txt file with: 1) Start Time, 2) Stop Time, 3) Duration of every communication window with each ground station. The same information is calculated by tudat. Thus, it makes sense to compare the two by: 1) calculating the difference in duration of the communication window, 2) calculating the difference between the Start and Stop Time of every window. To perform step 2, the epoch from both software was converted into *seconds since J2000*. Another metric that can be used to compare the two results is the amount of communication windows not identified by tudat. The metrics resulting from the comparison are the following:

- Successfully detected events: 99.16%
- Average absolute start-time error: 5.54 s
- Average absolute end-time error: 5.78 s
- Average duration error: 10.16 s

Figure 5.3 and Figure 5.4 show the absolute difference over time of the start- and end-time of the communication windows, respectively. It can be seen that the result for the end-time has higher spikes of difference with respect to the start-time. This is because, in order to check if a communication window was successfully detected, the start-time is used. The approach is to check for every communication pass outputted by GMAT's `contact locator` if the corresponding communication pass with the same ground station is present in the result of the discrete-time scheduler. A "corresponding communication pass" is defined as a pass in which the start time is within the defined threshold, either before or after the start of the event in GMAT. Since the threshold was set to 2 minutes, Figure 5.3 does not show high peaks. On the other hand, once a communication pass is considered successfully detected by looking at the start time, no check is imposed on the end-time. As a result of this, higher peaks are shown in Figure 5.4. This means that some communication windows are either quite shorter or longer than the result outputted by GMAT. The plot of the absolute difference in duration is almost identical to Figure 5.4, so it is not shown here.

It would be interesting to investigate the reason for this, analysing if these windows belong to certain specific ground stations. Due to a lack of time, and since precise modelling is not the goal of this thesis, the results presented here are considered satisfactory, as they closely represent the communication passes a satellite would encounter is real missions. A further investigation on the validation is left for future work.

5.5.2. Eclipse Periods

The `eclipse locator` of GMAT outputs a .txt file with: 1) Start Time, 2) Stop Time, 3) Duration, 4) Type of every eclipse period the satellite goes through during the simulation. The types of eclipse for a spacecraft in LEO are *umbra* and *penumbra*. For the type of orbits considered, a typical duration is of about 2000 seconds for the umbra period, and 10 seconds for the penumbra periods before and after the umbra. Thus, the penumbra makes up about 1% of the eclipse period. For this reason, the penumbra was left out from this comparison, as very precise modelling is outside the scope of this thesis. Nonetheless, the algorithm employed in tudat to determine the eclipse status of the satellite is able to calculate penumbras.

Note that the simulation day was set to 1st of January, with a RAAN of 75°. Under such conditions, the eclipse periods last slightly over 2000 seconds, or 33 minutes.

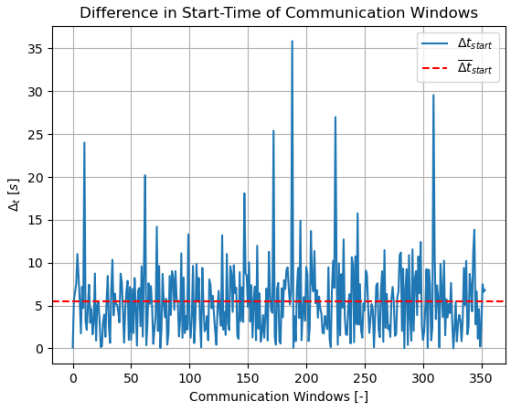


Figure 5.3: Time difference between the start-time of every communication window

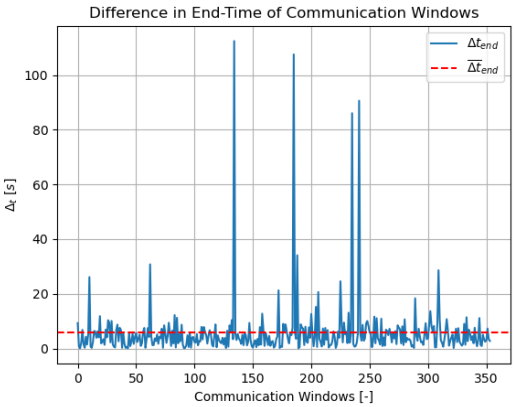


Figure 5.4: Time difference between the end-time of every communication window

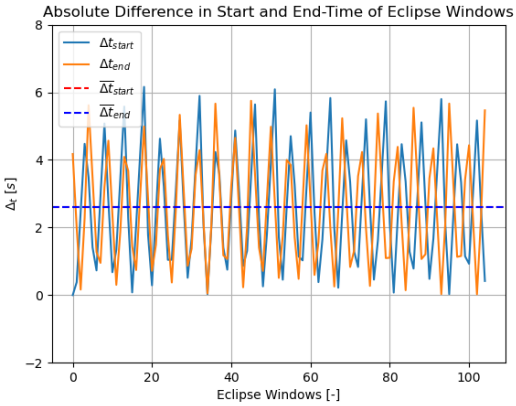


Figure 5.5: Time difference between start- and end-time of every eclipse window

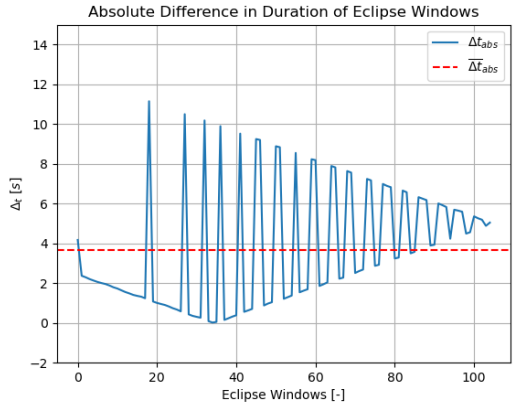


Figure 5.6: Absolute value of the duration difference of every eclipse window

Similarly to the communication passes, the metrics that can be used to quantify the accuracy of the model in tudat are: 1) number of eclipse periods successfully detected, 2) difference in start- and stop-time of each period, 3) difference in duration of each eclipse period. The simulation was run for seven days, and the metrics resulting from the comparison are the following:

- Successfully detected events: 100%
- Average absolute start-time error: 2.6 s
- Average absolute end-time error: 2.6 s
- Average absolute duration error: 3.7 s

Figure 5.5 and Figure 5.6 show the absolute difference in start- and end-time, and in duration, respectively, for all eclipse windows in the seven days of simulation. It can be seen that all average errors are smaller than the fixed step size used for numerical integration. For this reason, it can be inferred that the scheduler is successfully detecting the eclipse periods.

These results are considered satisfactory, and together with the results shown in the previous sections, validate the intermediate results of the discrete-time scheduler.

In the following chapter, the steps taken to expand the scheduling framework and include additional operational modes is discussed.

6

Expanding the Framework: Adding Operational Modes

This chapter deals with the steps taken to expand the scheduling framework proposed by Spangelo et al. [50] that was explained in the previous chapter.

In this chapter, the abbreviations used to indicate the different operational modes and frameworks are introduced. These are shown in Table 6.1, and are used throughout the remaining of the report.

Table 6.1: Abbreviations used throughout the report to indicate the different operational modes and frameworks

Term	Type	Abbreviation
SURVEILLANCE	Mode	SURV
COMMUNICATION	Mode	COMM
TASKED TRACKING	Mode	TT
SURVEILLANCE - COMMUNICATION	Framework	SC
IDLE - SURVEILLANCE - COMMUNICATION	Framework	ISC
IDLE - TASKED TRACKING - COMMUNICATION	Framework	ITC

6.1. Rationale for Adding Operational Modes

In the existing framework, only two operational modes are considered, namely one *observation* mode, in which the satellite is constantly acquiring data, and one *communication* mode, in which the satellite is downlinking data to a single ground station. In order to make this framework more suited to schedule the operations of space-based space-observation missions, it would be helpful to include additional operational modes. These are presented below.

TASKED TRACKING Mode

Based on the work of Olmos et al. [38], an efficient space-based space surveillance network could consist of four satellites equipped with optical telescopes. Amongst these satellites, three would be fully devoted to SURVEILLANCE, while one would be devoted to TASKED TRACKING. The latter mode consists in having a satellite point its optical telescope towards a certain target, and keep the target in the telescope's FoV by rotating about the satellite's body axes. Customers could send tracking requests to the ground segment, which could sort these requests, and send commands to the satellite to perform tasked tracking manoeuvres.

IDLE Mode

Another operational mode that can be implemented is an IDLE mode. In this mode, a satellite is not performing any scientifically-relevant activities, and limits itself to operate the critical subsystems. Such a mode is usually employed when a satellite does not have enough onboard resources to carry out observations or downlinking of data, and operates only certain subsystems to keep functioning while waiting for better conditions. This could be used as the default operational mode in the scheduling framework.

In the next section, the idea behind how to consider modes beyond those considered by Spangelo et al. [50] is presented.

6.2. Relating Energy Dynamics to Data Dynamics

In order to understand how more operational modes can be added in the scheduling framework, it is important to focus the attention on how the energy dynamics are related to the data dynamics in Spangelo's approach.

Looking at the unit of the terms listed in Table 4.2, it can be noted that the rate of energy consumption for downlinking data in communication mode is calculated by multiplying the following two terms:

- $\dot{E}^{-*}_{downlink}$: Rate of energy consumption to downlink [Joules-per-bit];
- $\dot{D}^{-}_{downlink}$: Rate of data downlink [bits-per-second].

The data downlink rate, expressed in bits-per-second, is linked with the specifications of the TT&C subsystem of the spacecraft, such as which antenna and transmitter are used. The $\dot{D}^{-}_{downlink}$ term thus arises directly from hardware specifications.

The rate of energy consumption to downlink, instead, is expressed in Joules-per-bit. This is not a usual unit of measure for energy consumption, and can thus not be retrieved by the specifications of hardware onboard the spacecraft. Instead, to determine this term the following procedure can be followed:

- Define the power required for the different operational modes considered in the framework. For example, in Spangelo's case this would mean defining i.e. $\dot{E}^{-}_{nominal} = 30W$ and $\dot{E}^{-}_{downlink} = 50W$;
- Relate the energy consumption rate of the non-default operational mode to its data dynamics rate. In Spangelo's case, this would mean calculating the difference between the power consumption in the two modes, and divide it by the data downlink rate, as follows:

$$\dot{E}^{-*}_{downlink} = \frac{\dot{E}^{-}_{downlink} - \dot{E}^{-}_{nominal}}{\dot{D}^{-}_{downlink}} \qquad \left[\frac{W}{bps} = \frac{\frac{J}{s}}{\frac{bit}{s}} = \frac{J}{\cancel{s}}\frac{\cancel{s}}{bit} = \frac{J}{bit} = \text{Joules-per-bit}\right]$$

- In this way, it is possible to relate the energy dynamics of the non-default mode to its data dynamics, finding an expression for $\dot{E}^{-*}_{downlink}$, which can be defined as the **"energy consumption rate per data"** associated with each mode.

This approach can be generalised, in order to allow for more operational modes to be considered. Each operational mode has a different resources dynamics profile, defined in terms of energy consumption rate, data downlink rate and data acquisition rate, as follows:

$$\text{Resource Dynamics Profile}_{mode\ 1} = \left[\dot{E}^{-}_{mode\ 1},\ \dot{D}^{-}_{mode\ 1},\ \dot{D}^{+}_{mode\ 1}\right]$$

The energy collection rate term, $\dot{E}^{+}$, depends on the illumination condition of the solar panels, and the data loss rate term, $\dot{D}^{-}_{loss}$ depends on the specifications of the satellite. Both these terms are independent of operational modes, and are thus not included in the resource profile.

The data dynamics term can be retrieved from the hardware specifications of the spacecraft. For example, $\dot{D}^{+}$ is the rate of data collection, which will be different from 0 only for modes in which the

Table 6.2: Example of resource dynamics rates for SURVEILLANCE and COMMUNICATION modes.

Resource Dynamics Rates	SURVEILLANCE	COMMUNICATION	Unit
$\dot{E}^-$	50	80	W
$\dot{D}^-$	0	200	Mbps
$\dot{D}^+$	1	0	Mbps

Table 6.3: Example of resource dynamics rates for IDLE, SURVEILLANCE, TASKED TRACKING and COMMUNICATION modes.

Resource Dynamics Rates	IDLE	SURV	TT	COMM	Unit
$\dot{E}^-$	10	50	60	80	W
$\dot{D}^-$	0	0	0	200	Mbps
$\dot{D}^+$	0	1	1	0	Mbps

satellite is performing observations. $\dot{D}^-$ is the rate of data downlink, which will be different from 0 only for modes in which the satellite is downlinking data.

The energy consumption term, $\dot{E}^-$, can be related to the data dynamics specific to that mode, and to the energy consumption typical of the nominal mode, as follows:

$$\dot{E}^{-*}_{mode\ o} = \frac{\dot{E}^-_{mode\ o} - \dot{E}^-_{nominal}}{\dot{D}_{mode\ o}} \tag{6.1}$$

Note that in the last equation, the data dynamics term in the denominator does not have a superscript ($^+$ or $^-$). This is deliberate, as different operational modes will have different data dynamics terms linked to them. For example, for an operational mode in which the satellite collects data via optical observations, the energy consumption term will be related to the data acquisition rate term, so $\dot{D}^+$. On the other hand, for an operational mode in which the satellite downlinks data, the energy consumption term will be related to the data downlink term, so $\dot{D}^-$. How this concept of relating the energy dynamics to the data dynamics can be used to include more operational modes will become clear when explaining the mathematical formulation of the problem.

Analysing the mathematical formulation of Spangelo allows to see in practice how this is implemented.

6.3. Example in Spangelo

Following the approach of Spangelo, assume that a satellite is in SURVEILLANCE mode by default, and has the energy consumption, data downlink and data acquisition rates shown in Table 6.2.

Let's re-write the energy update equation here for clarity.

$$e_{i+1} = e_i + \beta^{e+}_i - \beta^{e-}_i - \sum_{o \in O_i} \alpha_{i,o} q_{i,o} - h^e_i \quad \forall i \in I \tag{6.2}$$

In the equation above, the term α can be calculated as follows:

$$\alpha = \frac{\dot{E}^-_{COMM} - \dot{E}^-_{SURV}}{\dot{D}_{DOWNLINK}} = \frac{80W - 50W}{200 \cdot 10^6 bps} = \frac{30}{200 \cdot 10^6} \frac{J}{s} \frac{s}{bit} = 1.5 \cdot 10^{-7} \frac{J}{bit}$$

This idea can be extended to include different operational modes. Assume that a satellite has the resources dynamics rate linked with its operational modes shown in Table 6.3.

which can be retrieved by looking at the specifications of the hardware onboard the satellite. For example the energy consumption rate in the different modes depends what subsystems are turned on in each mode, and what their power consumption is, while the data acquisition rate in the observation modes depend on the specifications of the optical camera.

To summarise the concept, in order to relate data and energy dynamics, while solving the problem only for the (downlinked) data, the two quantities have to be related. So the idea is:

- Set a base energy consumption rate in the default mode, which needs to be the one with the lowest energy consumption rate;
- Relate the amount of data (collected or downlinked) in the other non-default modes to the amount of extra energy consumption rate with respect to the default mode.

The framework discussed until now consisted in the default mode being the *surveillance* mode, and the non-default mode being the *communication* mode. In the next section, two additional frameworks are proposed, in which different combinations of default and non-default modes are proposed.

6.4. New Operational Scheduling Frameworks

In this section, new scheduling frameworks are proposed. The effect they have on the scheduling results are reported in chapter 8. Note that only the changes with respect to the framework described in chapter 5 are presented in this section. The complete list of assumptions that describe each instance of the problem, and the respective mathematical formulations are shown in Appendix B and Appendix C.

6.4.1. IDLE - SURVEILLANCE - COMMUNICATION (ISC)

This framework is similar to that proposed by Spangelo et al. [50], with the difference that the default mode considered here is the IDLE mode, rather than the SURVEILLANCE mode. The full list of assumptions that make-up the problem, and its mathematical formulation can be found in Appendix B. Below, the assumptions and pieces of mathematical formulation that are new compared to the original problem are reported.

Different Assumptions

Compared to the framework that is only comprised of SURVEILLANCE and COMMUNICATION modes, the different assumptions are the following:

- The satellite's default mode is the IDLE mode, and the scheduling solution defines:
  - The amount of data that is acquired while in SURVEILLANCE mode, w;
  - The amount of data that is downlinked while in COMMUNICATION mode, q;
- Energy is required to carry out tasks outside those included in the default mode. These can include *communication tasks* as well as *observation tasks*.
- The satellite is capable of downlinking data to any visible ground stations for the entire duration of an interval or any fraction thereof.

Different Mathematical Formulation

With respect to the original framework, the energy per data associated with downlinking is now calculated as follows:

$$\alpha = \frac{\dot{E}^{-}_{COMM} - \dot{E}^{-}_{IDLE}}{\dot{D}^{+}_{COMM}} \tag{6.3}$$

where the difference lies in the default energy consumption being that for IDLE mode here.

Additionally, a new quantity has to be defined, which is the energy per data associated with acquiring surveillance data, calculated as follows:

$$\beta = \frac{\dot{E}^{-}_{SURV} - \dot{E}^{-}_{IDLE}}{\dot{D}^{+}_{SURV}} \tag{6.4}$$

Since in this new formulation the satellite is not acquiring data by default, a new constraint has to be added to the mathematical formulation, as follows:

$$w_{i,o} \leq \Delta t_i \dot{D}^{+}_{SURV} x_{i,o} \quad \forall i \in I, o \in O_i \tag{6.5}$$

Meaning: 1) surveillance data can only be acquired while the satellite is operating in SURVEILLANCE mode. 2) The amount of surveillance data acquired is limited by the interval time and the data acquisition rate.

The data update constraint needs to be reformulated, as follows:

$$d_{i+1} = d_i + \cancel{\delta_i^{d+}} - \delta_i^{d-} - \sum_{o \in O_i} q_{i,o} + \sum_{o \in O_i} w_{i,o} - h_i^d \quad \forall i \in I \tag{6.6}$$

where δ_i^{d+} is the data acquisition rate for the default mode, which is equal to 0 since the default mode is the IDLE mode. Here, the amount of surveillance data acquired during each interval is added to the total.

The energy update constraint needs to be reformulated, as follows:

$$e_{i+1} = e_i + \delta_i^{e+} - \delta_i^{e-} - \sum_{o \in O_i} \alpha_{i,o} q_{i,o} - \sum_{o \in O_i} \beta_{i,o} w_{i,o} - h_i^e \quad \forall i \in I \tag{6.7}$$

where at each interval, the energy required to acquire surveillance data is subtracted from the total.

Objective Function

The objective function is kept the same as for the SURVEILLANCE-COMMUNICATION case, which is to maximise the data downlink across the scheduling horizon. The reason for this is that, if the objective would be that of acquiring data, and acquiring data would require less energy than downlinking data, then the most optimal solution will always be to never downlink any data. In practice, such a schedule would result in no mission profit, and for this reason the downlinked data is set as the quantity to maximise.

Division in Intervals

In this framework, the division in intervals is the same as for the framework that includes only SURVEILLANCE and COMMUNICATION. The reason for this is that there are no other events that can be used to split the scheduling horizon. The difference with the case considered in Spangelo is that in this case, the minimum number of options in each interval possible is two, as the satellite can always be in IDLE or SURVEILLANCE mode, as they are both target independent modes. The impact this has on the computational complexity will be investigated in chapter 8.

6.4.2. IDLE - TASKED TRACKING - COMMUNICATION (ITC)

In this framework, the satellite is by default in IDLE mode, and the algorithm has to schedule when it goes in TASKED TRACKING and in COMMUNICATION mode. The full list of assumptions that make-up the problem, and its mathematical formulation can be found in Appendix C. Below, the assumptions and pieces of mathematical formulation that are new compared to the original problem are reported.

Different assumptions

The additional assumptions needed for this framework are as follows:

- The satellite's default mode is the IDLE mode, and the scheduling solution defines:
  - The amount of data that is acquired while in TASKED TRACKING mode, y;
  - The amount of data that is downlinked while in COMMUNICATION mode, q;

- Energy is required to carry out tasks outside those included in the default mode. These can include *communication tasks*, as well as *observation tasks*;
- During any interval i, the satellite is capable of downlinking data for the entire duration or any fraction thereof.
- There are multiple targets to be observed by the satellite, these are referred to as Resident Space Objects (RSOs);
- Each RSO periodically comes in- and out-of-view of the satellite. The period of time in which a RSO is visible is referred to as the *observation window*;
- The duration of these observation windows depends on the geometry of the orbit of the satellite relative to the RSOs;
- It can happen that more than one RSO is in view of the satellite at the same time. In this case, the satellite can only observe one target at a time;
- The satellite is capable of collecting tasked tracking data of any target RSO in view for the entire duration of an interval or any fraction thereof.
- The orbit of the satellite is assumed to be deterministic, and as such the communication windows to the GSN, and the observation windows to the RSOs are known a priori;

Resources Update Equations

A new quantity has to be defined, which is the energy per data associated with acquiring tasked tracking data, as follows:

$$\gamma = \frac{\dot{E}^{-}_{TT} - \dot{E}^{-}_{IDLE}}{\dot{D}^{+}_{TT}} \tag{6.8}$$

Since in this new formulation the satellite is not acquiring data by default, a new constraint has to be added to the mathematical formulation, as follows:

$$y_{i,o} \leq \Delta t_i \dot{D}^{+}_{TT} x_{i,o} \quad \forall i \in I, o \in O_i \tag{6.9}$$

Meaning: 1) tasked tracking data can only be acquired while the satellite is operating in TASKED TRACKING mode. 2) The amount of tasked tracking data acquired is limited by the interval time and the data acquisition rate.

The data update constraint needs to be reformulated, as follows:

$$d_{i+1} = d_i + \cancel{\delta^{d+}_i} - \delta^{d-}_i - \sum_{o \in O_i} q_{i,o} + \sum_{o \in O_i} y_{i,o} - h^d_i \quad \forall i \in I \tag{6.10}$$

where the amount of tasked tracking data acquired during each interval is added to the total. Like in the previously shown framework, δ^{d+}_i is equal to 0, as no data is acquired in IDLE mode.

The energy update constraint needs to be reformulated, as follows:

$$e_{i+1} = e_i + \delta^{e+}_i - \delta^{e-}_i - \sum_{o \in O_i} \alpha_{i,o} q_{i,o} - \sum_{o \in O_i} \gamma_{i,o} y_{i,o} - h^e_i \quad \forall i \in I \tag{6.11}$$

where the energy required to acquired tasked tracking data is subtracted from the total.

Objective Function

In this framework, the objective function is still that of downlinking as much data as possible across the scheduling horizon. The same reasoning as for the IDLE - SURVEILLANCE - COMMUNICATION can be made.

The division in intervals for this framework presents some differences with respect to the other frameworks presented above. For this reason, the following section is dedicated to how this is performed.

Table 6.4: Input parameters needed for the ITC framework.

Sim Settings	GSN	S/C Properties	TT targets	Optical Telescope
Time-step	Ground Stations LLA	Initial State	Initial State	Min Phase Angles
Start Date	Min Elevation Angle	Dry Mass		
Duration	Min Duration	Drag Coefficient		
		Reference Area		

6.5. Modified Workflow to Include TASKED TRACKING

The workflow for the IDLE - SURVEILLANCE - COMMUNICATION framework is the same as that described in chapter 5. The IDLE - TASKED TRACKING - COMMUNICATION framework, instead, requires a slightly different workflow, which is presented in this section.

The difference is due to the fact that in TASKED TRACKING mode, observations are carried out knowing the position of the target, contrary to the SURVEILLANCE mode, in which observations are made without a-priori knowledge of the target, and thus no such thing as an *"observation window"* can be defined.

Thus, the satellite can only go into TASKED TRACKING mode whenever a tasked tracking target is visible. For a satellite equipped with an optical telescope, a target is visible when the visibility constraints introduced in subsection 3.4.2 are respected. These will be quickly re-listed here, for clarity.

- **Third body occultation**: there cannot be any third body in between the observer satellite and the target object;
- **Direct sunlight**: since the optical telescope works by collecting photons emitted by the Sun and reflected by the target, the latter must be in direct sunlight to be observed;
- **Phase angles**: to avoid excessive straylight to enter the FoV of the optical telescope, the illumination condition of the target as seen from the observer must meet certain conditions. These conditions are imposed in order to avoid excessive light coming from celestial bodies. In the case of an Earth-orbiting satellite, the relevant celestial bodies are the Sun, the Earth and the Moon. Depending on the characteristics of the telescope, different values for the phase angles with these bodies can be tolerated.

Similarly to the communication window, tudat's capabilities can be exploited to determine the visibility window of the TT-targets as seen from the observer satellite. In this case, the problem parameters needed are shown in Table 6.4.

Following the same approach defined in chapter 4, the scheduling horizon should be split into intervals in which the set of operational modes the satellite can be functioning, and the illumination condition, are both constant. In this way, the resource dynamics can be assumed linear, and the UCF solution is more likely to be feasible when considering continuous-time dynamics.

A new discretisation scheme that can be used for this framework has to be devised. The events that trigger the start of a new interval now are:

- Whenever a ground station comes in- or out-of-view of the satellite;
- Whenever a tasked tracking target comes in- or out-of-view of the satellite;
- Whenever the illumination condition of the satellite goes from eclipse to non-eclipse, or vice-versa.

Then, the Under-Constrained Formulation (UCF) problem can be solved using a MILP algorithm, in the same way as the other frameworks presented before.

The solutions resulting from solving the UCF versions of the scheduling problem are then fed to the continuous-time simulator, discussed in the next chapter, which propagates the attitude and resource dynamics based on the discretised solution.

The results of the complete pipeline are presented in chapter 8.

7

Continuous-Time Simulator

This chapter presents the continuous-time simulator, which was developed in order to check the feasibility of the scheduling solutions. This chapter is structured as follows: section 7.1 reviews the reason why this tool is needed. Section 7.2 outlines the different steps involved in this simulator. Subsequently, section 7.3 describes the environmental torques considered, and how they are modelled. Section 7.4 describes the aspects related to control, while section 7.5 deals with the different pointing directions linked with the operational modes considered. Then, section 7.6 examines how the feasibility of the scheduling solution is checked in practice. Section section 7.7 concludes the chapter, describing how the result of the continuous-time simulator can be used to improve the time discretisation into intervals.

7.1. Rationale for Continuous-Time Simulator

In operational research, simplifications are often made to speed up the process of finding a solution, and to be able to tackle larger problems. This comes at the expense of a lower accuracy of the representation of the real physical system, due to a mismodelling of the attitude and resources dynamics.

For this thesis, it was decided to develop and make use of a continuous-time simulator, in which the discretised solution of the UCF problem, which assumes linear resources dynamics within each interval, is used as input, and a continuous-time resource and attitude profile is generated.

The latter is then used to check the feasibility of the schedule according to the operational constraints, and to improve the discretisation of the time intervals, in case the scheduling solution turns out to be unfeasible.

7.2. Workflow

This simulator is a 3-degrees-of-freedom (DoF) attitude simulator, in which the attitude dynamics are propagated independently from the translational dynamics.

The workflow of the simulator comprises a number of steps. These can be divided into pre-processing steps, achieved using a MATLAB script, and a simulation step, achieved using a Simulink model. The pre-processing steps include:

- Importing data from the discrete-time scheduler;
- Importing data linked with rotational dynamics;
- Linearising the equations of motion;
- Discretising a CAD input into a mesh of triangles;
- Calculating the gains of the PD controller by means of an LQR algorithm;
- Calculating the reference attitude for the simulation.

The Simulink model consists in propagating the attitude and energy dynamics, based on the operational schedule outputted by the scheduler.

Note that the reference frames used in this simulator are the ones described in section 3.2, and that the simulator uses a fixed-time-step for the numerical integration.

7.3. Environmental Torques Considered

The environmental torques that are included in the simulator are the following:

- Aerodynamic torque;
- SRP torque;
- Gravity gradient torque;

For the aerodynamic and SRP torque, the simulator allows for a 3D model to be imported as an .stl file, and discretised into a mesh of triangles. The torque generated by the each individual triangle is calculated, and then summed to obtain the total torque. The shadowing effects of solar panels and of the spacecraft body are not considered.

7.3.1. Aerodynamic Torque

The aerodynamic torque is the torque that the spacecraft is subject to when travelling through Earth's atmosphere. It is caused by the aerodynamic drag, introduced in subsection 3.1.5, when its resultant force does not pass through the center of gravity of the spacecraft, generating a torque. The total aerodynamic torque acting on the spacecraft can be calculated as follows:

$$T_{aero} = \frac{1}{2}\rho V^2 C_d \sum_{i=1}^{N} A_i \left(\hat{u}_v \times \hat{s}_{cp_i}\right) \tag{7.1}$$

where:

- T_{aero} is the aerodynamic torque;
- ρ denotes the atmospheric density, assumed constant at specific altitudes based on the densities during mean solar activities predicted by the MSISE-90 model;
- V is the velocity of the satellite;
- C_d is the drag coefficient;
- A_i represents the area of the i^{th} triangle;
- $\hat{u}_v$ represents the unit velocity vector, expressed in the body-frame $\mathbb{B}$;
- $\hat{s}_{cp_i}$ is the vector from the center of pressure of the i^{th} triangle to the center of mass of the spacecraft, expressed in the body frame $\mathbb{B}$;

At each time-step, the contribution of each triangle of the mesh is calculated by calculating the dot product between the velocity vector and the surface normal. If the triangle is facing "into the wind", the dot product between the normal and the velocity is greater than 0, and the triangle contributes to the total torque. If the dot product is smaller than 0, the triangle is facing "away from the wind", and does not contribute to the total torque.

7.3.2. SRP Torque

The SRP torque is caused by the solar radiation pressure, introduced in Equation 3.1.5. Similarly to the aerodynamic torque described above, it is a result of the SRP exerting a force away from the center of gravity of the satellite, causing the latter to rotate. The total SRP torque can be calculated as follows:

$$T_{SRP} = \frac{Flux_{Sun}}{c} C_r \sum_{i=1}^{N} A_i \left(\hat{u}_{solar} \times \hat{s}_{cp_i} \right) \tag{7.2}$$

where:

- $Flux_{Sun}$ is the solar constant;
- c is the speed of light;
- C_r is the coefficient of reflectivity;
- $\hat{u}_{\text{solar}}$ is the unit vector pointing towards the Sun;

and A_i and $\hat{s}_{cp_i}$ are the same as for the aerodynamic torque. The solar constant is a flux density that measures the mean solar electromagnetic radiation per unit area. It is measured at 1 Astronomical Unit (AU), which is approximately the distance from the Earth to the Sun, where it has a value of 1358 W/m^2 [60].

7.3.3. Gravity Gradient Torque

Gravity gradient torque is a perturbing torque that arises due to the differential gravitational pull exerted by a celestial body, such as Earth, on the spacecraft's various components. This torque emerges because different parts of the spacecraft are subject to varying gravitational forces owing to their differing distances from the Earth. The effect becomes significant especially in elongated spacecraft with non-symmetric inertia matrices, where the difference in distance from the Earth's centre of mass to the ends of the spacecraft can be considerable. The gravity gradient torque tends to align the satellite axis with the smallest inertia with the nadir axis. It can be calculated as follows:

$$T_{gg} = \frac{3\mu}{R^3} \hat{u}_n \times J\hat{u}_n \tag{7.3}$$

where:

- T_{gg} is the gravity gradient torque,
- μ is the gravitational constant of the Earth,
- R is the radial distance from the Earth's center to the satellite,
- J represents the inertia matrix of the satellite,
- $\hat{u}_n$ is the nadir unit vector.

Since the orbits considered for this work are all circular, the radial distance from the Earth's center to the satellite will be constant for every scenario. Within a single simulation, this torque depends only on the satellite's attitude.

Now that the environmental torques implemented in the simulator have been described, the implementation of the attitude control system can be investigated.

7.4. Control

In order to simulate the attitude dynamics of the satellite, a controller has to be implemented. For this simulator, it was chosen to implement a Proportional-Derivative (PD) controller, as introduced in subsection 3.3.2. To find the gains of the PD controller, a Linear Quadratic Regulator (LQR) algorithm is used. As this algorithm is designed for Linear-Time-Invariant (LTI) systems, the system has to first be linearised.

7.4.1. Linearisation about the Operating Point

The gains of the PD controller are calculated using a LQR technique. This technique is an optimal control algorithm that aims at finding control inputs that minimise a quadratic cost function while considering the linear dynamics of a system. LQR is designed to obtain constant gains for Linear-Time Invariant (LTI) systems. These are systems whose dynamics can be described by linear differential equations, and thus characterised by linear relationships between the system's state, inputs and outputs, that stay constant over time. For a rigid-body equipped with reaction wheels, which is the use-case considered for this work, considering the system as linear is a valid assumption (CITATION NEEDED).

Since an equilibrium point has to be chosen for the linearisation procedure, it makes sense to first discuss how this point is chosen, before explaining how the LTI system is set-up.

Operating Point Choice

For systems that have a predominant rotational equilibrium condition, it is easy to select a linearisation point. Examples of this are an aeroplane or a drone performing horizontal flight; These two systems will be in equilibrium at a certain angle of attack with respect to the incoming flow, where they will spend most of the time when in-flight. In these kind of systems, it makes sense to linearise the equations of motion around this equilibrium point, such that the dynamics of the system will be accurately represented for most of the time, except when deviating from this equilibrium condition.

For a spacecraft orbiting the Earth, there is no single orientation leading to *"the most"* stable equilibrium. This is due to the state of constant free fall experienced by objects orbiting celestial bodies. In such a state, the predominant force acting on the system is the gravitational pull of the central body, that changes direction as the object moves around the celestial body. For this reason, it is not as trivial as for other systems to select a linearisation point.

The spacecraft considered for this research has to point in different directions, based on the operational mode. Despite this, in the scheduling framework considered, a *default mode* is assumed, where the spacecraft is expected to spend most of its time. To decrease the linearisation error, it thus makes sense to select the linearisation point to be the orientation described by this mode. For the mission considered, the default is the *SURVEILLANCE mode*, in which the spacecraft is scanning the sky with a fixed orientation with respect to the orbit-frame, and thus with respect to Earth's horizon, searching for space objects. Thus, the linearisation point is chosen to be the following:

$$x_{linear} = [q_{1,lin}\ q_{2,lin}\ q_{3,lin}\ q_{4,lin}\ \omega_{1,lin}\ \omega_{2,lin}\ \omega_{3,lin}]$$

where $q_{lin} = [q_{1,lin}\ q_{2,lin}\ q_{3,lin}\ q_{4,lin}]$ is the quaternion expressing the nominal orientation of the satellite in the the orbit-frame, and $\omega_{lin} = [\omega_{1,lin}\ \omega_{2,lin}\ \omega_{3,lin}]$ is the angular velocity vector expressing the nominal angular velocity of the satellite in the orbit-frame.

When considering a satellite pointing its camera in the direction of its velocity vector, this results in the following operational point:

$$\begin{aligned} q_{linear} &= [0\ 0\ 0\ 1] \\ \omega_{linear} &= [0,\ -n_0,\ 0] \\ x_{linear} &= [0,\ 0,\ 0,\ 1,\ 0,\ -n_0,\ 0] \end{aligned}$$

where n_0 is the angular velocity of the spacecraft assuming a perfectly circular orbit, which is given by:

$$v = \sqrt{\frac{\mu}{a}} \tag{7.4}$$

$$n_0 = \frac{v}{r} \tag{7.5}$$

$$r = a \tag{7.6}$$

$$\therefore n_0 = \frac{1}{a}\sqrt{\frac{\mu}{a}} = \sqrt{\frac{\mu}{a^3}} \tag{7.7}$$

where v is the translational velocity of the orbiting body, r its distance to the central body, a its semi-major axis, which is equal to r for perfectly circular orbits, and μ is the gravitational parameter of the Earth.

In this case shown above, the linear quaternion is the unit quaternion, $[0\ 0\ 0\ 1]$, because the spacecraft is assumed to be pointing its camera in the velocity direction, and thus the body-frame and the orbit-frame would coincide. If a different pointing direction is considered for the nominal mode, a quaternion expressing this attitude needs to be used for linearisation.

LTI Systems

Now that the procedure for choosing an equilibrium point has been described, the explanation of the procedure carried out to linearise the system can be continued.

The state-space representation of an LTI system is typically given by:

$$\dot{\vec{x}} = A\ \vec{x} + B\ \vec{u} \tag{7.8}$$

$$\vec{y} = C\ \vec{x} + D\ \vec{u} \tag{7.9}$$

where $\vec{x}$ is the state vector, $\vec{u}$ is the control input vector, $\vec{y}$ the output vector, and A, B, C and D the matrices defining the system dynamics.

Linear A Matrix

The procedure for calculating the linearised A matrix is as follows. By inspecting Equation 7.8, it can be seen that the A matrix relates the state vector, $\vec{x}$ to its derivative, $\dot{\vec{x}}$. The A matrix is thus the partial derivative of $\dot{\vec{x}}$ with respect to $\vec{x}$, or $\frac{\partial \dot{\vec{x}}}{\partial \vec{x}}$

Following the procedure of Wie in his work on "Space Vehicles Dynamics and Control", the rates of change of the quaternions, with scalar-last notation, are given as follows [61]:

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{7.10}$$

Considering the angular velocity for linearisation to be $\vec{\omega}_{linear} = [\omega_1,\ (\omega_2 - n_0),\ \omega_3]^T$, the derivative of each quaternion is given by:

$$\dot{q}_1 = \frac{1}{2} \cdot [\omega_3 \cdot q_2 + (-\omega_2 + n_0) \cdot q_3 + \omega_1 \cdot q_4]; \tag{7.11}$$

$$\dot{q}_2 = \frac{1}{2} \cdot [-\omega_3 \cdot q_1 + \omega_1 \cdot q_3 + q_4 \cdot (\omega_2 + n_0)]; \tag{7.12}$$

$$\dot{q}_3 = \frac{1}{2} \cdot [(\omega_2 - n_0) \cdot q_1 - \omega_1 \cdot q_2 + \omega_3 \cdot q_4]; \tag{7.13}$$

$$\dot{q}_4 = -\frac{1}{2} \cdot [\omega_1 \cdot q_1 + (\omega_2 + n_0) \cdot q_2 + \omega_3 \cdot q_3]. \tag{7.14}$$

Following the derivation of rigid-body dynamics from Wie [61], for a principal-axis reference frame with a set of basis vectors, such as the body-frame $[b_1, b_2, b_3]$, Euler's rotational equations of motion become:

$$J_{11}\dot{\omega}_1 - (J_{22} - J_{33})\omega_2\omega_3 = M_1 \tag{7.15}$$

$$J_{22}\dot{\omega}_2 - (J_{33} - J_{11})\omega_3\omega_1 = M_2 \tag{7.16}$$

$$J_{33}\dot{\omega}_3 - (J_{11} - J_{22})\omega_1\omega_2 = M_3 \tag{7.17}$$

$$\tag{7.18}$$

where J_{11}, J_{22}, and J_{33} are the principal moments of inertia, M_i the momentum along the three principal axis. These are three coupled, nonlinear ordinary differential equations for state variables ω_1, ω_2, and ω_3 of a rigid body.

The normalised difference between the moments of inertia along the different axes can be defined as follows:

$$k_x = \frac{J_{22} - J_{33}}{J_{11}}; \tag{7.19}$$

$$k_y = \frac{J_{11} - J_{33}}{J_{22}}; \tag{7.20}$$

$$k_z = \frac{J_{22} - J_{11}}{J_{33}}. \tag{7.21}$$

The rates of change of the individual angular velocity components along one axis, due to angular velocity components along the other two axes are defined as follows:

$$\dot{\omega}_1 = k_x \cdot \omega_2 \cdot \omega_3 \tag{7.22}$$

$$\dot{\omega}_2 = k_y \cdot \omega_1 \cdot \omega_3 \tag{7.23}$$

$$\dot{\omega}_3 = k_z \cdot \omega_1 \cdot \omega_2 \tag{7.24}$$

$$\tag{7.25}$$

A vector, $\dot{\vec{x}}$, can now be defined, containing the dynamics of the state of the system, as follows:

$$\dot{\vec{x}} = [\dot{q}_1\ \dot{q}_2\ \dot{q}_3\ \dot{q}_4\ \dot{\omega}_1\ \dot{\omega}_2\ \dot{\omega}_3] \tag{7.26}$$

The state vector of the system can be defined as follows:

$$\vec{x} = [q_1\ q_2\ q_3\ q_4\ \omega_1\ \omega_2\ \omega_3] \tag{7.27}$$

Then the Jacobian of vector $\vec{f}$ with respect to the state vector $\boldsymbol{x}$ can be calculated, as follows:

$$A = \frac{\partial \dot{\vec{x}}}{\partial \vec{x}} \tag{7.28}$$

which in symbols can be expressed as:

$$A = \begin{bmatrix} \frac{\partial \dot{q}_1}{\partial q_1} & \frac{\partial \dot{q}_1}{\partial q_2} & \frac{\partial \dot{q}_1}{\partial q_3} & \frac{\partial \dot{q}_1}{\partial q_4} & \frac{\partial \dot{q}_1}{\partial \omega_1} & \frac{\partial \dot{q}_1}{\partial \omega_2} & \frac{\partial \dot{q}_1}{\partial \omega_3} \\ \frac{\partial \dot{q}_2}{\partial q_1} & \frac{\partial \dot{q}_2}{\partial q_2} & \frac{\partial \dot{q}_2}{\partial q_3} & \frac{\partial \dot{q}_2}{\partial q_4} & \frac{\partial \dot{q}_2}{\partial \omega_1} & \frac{\partial \dot{q}_2}{\partial \omega_2} & \frac{\partial \dot{q}_2}{\partial \omega_3} \\ \frac{\partial \dot{q}_3}{\partial q_1} & \frac{\partial \dot{q}_3}{\partial q_2} & \frac{\partial \dot{q}_3}{\partial q_3} & \frac{\partial \dot{q}_3}{\partial q_4} & \frac{\partial \dot{q}_3}{\partial \omega_1} & \frac{\partial \dot{q}_3}{\partial \omega_2} & \frac{\partial \dot{q}_3}{\partial \omega_3} \\ \frac{\partial \dot{q}_4}{\partial q_1} & \frac{\partial \dot{q}_4}{\partial q_2} & \frac{\partial \dot{q}_4}{\partial q_3} & \frac{\partial \dot{q}_4}{\partial q_4} & \frac{\partial \dot{q}_4}{\partial \omega_1} & \frac{\partial \dot{q}_4}{\partial \omega_2} & \frac{\partial \dot{q}_4}{\partial \omega_3} \\ \frac{\partial \dot{\omega}_1}{\partial q_1} & \frac{\partial \dot{\omega}_1}{\partial q_2} & \frac{\partial \dot{\omega}_1}{\partial q_3} & \frac{\partial \dot{\omega}_1}{\partial q_4} & \frac{\partial \dot{\omega}_1}{\partial \omega_1} & \frac{\partial \dot{\omega}_1}{\partial \omega_2} & \frac{\partial \dot{\omega}_1}{\partial \omega_3} \\ \frac{\partial \dot{\omega}_2}{\partial q_1} & \frac{\partial \dot{\omega}_2}{\partial q_2} & \frac{\partial \dot{\omega}_2}{\partial q_3} & \frac{\partial \dot{\omega}_2}{\partial q_4} & \frac{\partial \dot{\omega}_2}{\partial \omega_1} & \frac{\partial \dot{\omega}_2}{\partial \omega_2} & \frac{\partial \dot{\omega}_2}{\partial \omega_3} \\ \frac{\partial \dot{\omega}_3}{\partial q_1} & \frac{\partial \dot{\omega}_3}{\partial q_2} & \frac{\partial \dot{\omega}_3}{\partial q_3} & \frac{\partial \dot{\omega}_3}{\partial q_4} & \frac{\partial \dot{\omega}_3}{\partial \omega_1} & \frac{\partial \dot{\omega}_3}{\partial \omega_2} & \frac{\partial \dot{\omega}_3}{\partial \omega_3} \end{bmatrix} \tag{7.29}$$

The software Maple was used to symbolically solve the Jacobian, which lead to the following result:

$$A = \frac{1}{2}\begin{bmatrix} 0 & \omega_3 & -\omega_2 + n_0 & \omega_1 & q_4 & -q_3 & q_2 \\ -\omega_3 & 0 & \omega_1 & \omega_2 + n_0 & q_3 & q_4 & -q_1 \\ \omega_2 - n_0 & -\omega_1 & 0 & \omega_3 & -q_2 & q_1 & q_4 \\ -\omega_1 & -\omega_2 - n_0 & -\omega_3 & 0 & -q_1 & -q_2 & -q_3 \\ 0 & 0 & 0 & 0 & 0 & 2k_x\omega_3 & 2k_x\omega_2 \\ 0 & 0 & 0 & 0 & 2k_y\omega_3 & 0 & 2k_y\omega_1 \\ 0 & 0 & 0 & 0 & 2k_z\omega_2 & 2k_z\omega_1 & 0 \end{bmatrix} \tag{7.30}$$

the worksheet used to produce this result can be found in Appendix D.

Now, the A matrix of the LTI system can be calculated, by substituting the operating point in Equation 7.30.

Linear B Matrix

Next, the B matrix needs to be calculated, as it is a required input to calculate the gains using the `lqr` function provided by MATLAB. In an LTI system, the B matrix represent the input matrix, defining how the system responds to inputs. The B matrix thus relates system inputs to the rate of change of the state variables, as follows:

$$B = \frac{\partial \dot{\vec{x}}}{\partial \vec{u}} \tag{7.31}$$

Substituting the control inputs, $\vec{u} = [T_x\ T_y\ T_z]$, and the state vector gives:

$$B = \begin{bmatrix} \frac{\partial \dot{q}_1}{\partial T_x} & \frac{\partial \dot{q}_1}{\partial T_y} & \frac{\partial \dot{q}_1}{\partial T_z} \\ \frac{\partial \dot{q}_2}{\partial T_x} & \frac{\partial \dot{q}_2}{\partial T_y} & \frac{\partial \dot{q}_2}{\partial T_z} \\ \frac{\partial \dot{q}_3}{\partial T_x} & \frac{\partial \dot{q}_3}{\partial T_y} & \frac{\partial \dot{q}_3}{\partial T_z} \\ \frac{\partial \dot{q}_4}{\partial T_x} & \frac{\partial \dot{q}_4}{\partial T_y} & \frac{\partial \dot{q}_4}{\partial T_z} \\ \frac{\partial \dot{\omega}_1}{\partial T_x} & \frac{\partial \dot{\omega}_1}{\partial T_y} & \frac{\partial \dot{\omega}_1}{\partial T_z} \\ \frac{\partial \dot{\omega}_2}{\partial T_x} & \frac{\partial \dot{\omega}_2}{\partial T_y} & \frac{\partial \dot{\omega}_2}{\partial T_z} \\ \frac{\partial \dot{\omega}_3}{\partial T_x} & \frac{\partial \dot{\omega}_3}{\partial T_y} & \frac{\partial \dot{\omega}_3}{\partial T_z} \end{bmatrix} \tag{7.32}$$

From the kinematic equations of quaternions, shown in Equation 7.10, it is clear that $\dot{q}$ depends only on the angular velocity, ω and the current attitude quaternion, q. Thus, the partial derivative of $\dot{q}$ with respect to the control input vector will be zero, as follows:

$$\dot{q} \propto (q, \vec{\omega}) \quad \rightarrow \quad \dot{q} \not\propto \vec{u} \quad \rightarrow \quad \frac{\partial \dot{q}}{\partial \vec{u}} = 0$$

From the dynamic equation of a body subject to torque, shown in Equation 3.20 and reported here for clarity, it can be seen that the rate of change of the angular velocity, $\dot{\omega}$ is dependent on the torque. Thus, the partial derivative of $\dot{\omega}$ with respect to the control input vector, $\vec{u}$, will be related to the inertia matrix of the body, as follows:

$$\dot{\vec{\omega}} = \hat{J}^{-1}\vec{M} - \hat{J}^{-1}(\vec{\omega} \times \hat{J} \cdot \vec{\omega})$$

$$\dot{\omega} \propto (\hat{J}, \vec{T}, \omega) \quad \rightarrow \quad \dot{\omega} \propto \vec{u} \quad \rightarrow \quad \frac{\partial \dot{\omega}}{\partial \vec{u}} = \frac{\partial \dot{\omega}}{\partial \vec{T}} = \hat{J}^{-1}$$

By plugging in the partial derivative values that were just shown, the result is the following:

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{\partial \dot{\omega}_1}{\partial T_x} & \frac{\partial \dot{\omega}_1}{\partial T_y} & \frac{\partial \dot{\omega}_1}{\partial T_z} \\ \frac{\partial \dot{\omega}_2}{\partial T_x} & \frac{\partial \dot{\omega}_2}{\partial T_y} & \frac{\partial \dot{\omega}_2}{\partial T_z} \\ \frac{\partial \dot{\omega}_3}{\partial T_x} & \frac{\partial \dot{\omega}_3}{\partial T_y} & \frac{\partial \dot{\omega}_3}{\partial T_z} \end{bmatrix} \tag{7.33}$$

In the process of linearising the system, consideration is given only to the moments of inertia around the principal axes, disregarding the products of inertia. This simplification arises from assuming the alignment of the body-axes with the principal moments of inertia, leading to the formulation of the following inertia matrix:

$$\hat{J} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \quad \rightarrow \quad \hat{J}_{linear} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{7.34}$$

Thus, after linearising, only the partial derivatives of angular velocities and torques about the same axes are kept, resulting in the following B matrix:

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{\partial \dot{\omega}_1}{\partial T_x} & 0 & 0 \\ 0 & \frac{\partial \dot{\omega}_2}{\partial T_y} & 0 \\ 0 & 0 & \frac{\partial \dot{\omega}_3}{\partial T_z} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ I_{xx}^{-1} & 0 & 0 \\ 0 & I_{yy}^{-1} & 0 \\ 0 & 0 & I_{zz}^{-1} \end{bmatrix} \tag{7.35}$$

For completeness, the output equation of a state-space system, given in Equation 7.9, can be set-up. Since the only output we are interested in is the state of the system, $\vec{x}$, the C matrix will be the identity matrix, $\boldsymbol{I}$, while the D matrix will be the zero matrix, $\boldsymbol{O}$. In equation form:

$$\vec{y} = C\vec{x} + D\vec{u} \tag{7.36}$$

$$\vec{y} = \vec{x} \tag{7.37}$$

$$\therefore \vec{y} = \boldsymbol{I}\vec{x} + \boldsymbol{O}\vec{u} \tag{7.38}$$

7.4.2. Tuning the Gains - LQR algorithm

After defining the LTI system, MATLAB's `lqr` function can be used to find the gains of the PD controller [33]. This function uses the LQR algorithm to tune the gains of a controller. The `lqr` function, included in MATLAB's Control System Toolbox, accepts the matrices A and B , as previously described, along with two additional matrices, Q and R. The Q matrix, known as *state-cost weighted matrix*, assigns weights to the significance of deviations in the state. It is an n-by-n matrix, where n is the number of states. Bryson's rule can be used to set the initial values of Q, which is given by:

$$Q_{i,i} = \frac{1}{\text{maximum acceptable value of}(error_{states,i})^2}, \; i \in \{1, 2, \dots, n\} \tag{7.39}$$

and the resulting matrix has the following form:

$$Q = \begin{bmatrix} Q_{1,1} & 0 & \cdots & 0 \\ 0 & Q_{2,2} & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & Q_{n,n} \end{bmatrix} \tag{7.40}$$

The R matrix is referred to as the *input-cost weighted matrix*, and has size m-by-m, where m is the number of inputs. As the name suggests, this matrix weighs the control effort of the controller. Bryson's rule can be used to set the initial value of R, which is given by:

$$R_{j,j} = \frac{1}{\text{maximum acceptable value of}(input_{max,i})^2},\ j \in \{1, 2, \dots, m\} \tag{7.41}$$

and the resulting matrix has the following form:

$$R = \begin{bmatrix} R_{1,1} & 0 & \cdots & 0 \\ 0 & R_{2,2} & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & R_{m,m} \end{bmatrix} \tag{7.42}$$

Note that in the Q and R matrices, the diagonal terms each relate to one state or one control input, respectively. For example $Q_{1,1}$ refers to the quaternion q_1, and $Q_{5,5}$ refers to the angular velocity about the 1^{st} body axis, ω_1. Similarly, the term $R_{1,1}$ refers to the control torque about the 1^{st} body axes. If more information would be provided on the pointing requirements along the different axes, different values could be used for each entry of the diagonals. Since this information is not provided, these matrices are assumed to have the same value for all diagonal terms.

After defining the R and Q matrices, the LQR algorithm computes the state-feedback control $\vec{u} = -K\vec{x}$ that minimises the following quadratic function:

$$J(u) = \int_0^\infty \left(x^T Q x + u^T R u \right) dt \tag{7.43}$$

subject to the system dynamics described by the state-space equation $\dot{\vec{x}} = A\vec{x} + B\vec{u}$. As a general rule of thumb, larger values of R relative to Q will result in lower gains, as the algorithm is giving more importance to keeping the control effort low. On the contrary, if Q has larger values than R, the gains will be higher, as the algorithm is prioritising keeping the state deviation low.

Additionally to the state-feedback gain, K, this function returns the solution, S, of the associated algebraic Riccati equation, which is defined as follows:

$$A^T S + SA - SBR^{-1}B^T S + Q = 0 \tag{7.44}$$

Additionally, it returns the closed-loop poles $P = eig(A - BK)$, where $eig()$ is a function that calculates the eigenvalues of a matrix.

After calculating S, the gain matrix K is derived as follows:

$$K = R^{-1}B^T S \tag{7.45}$$

The gain matrix, K has the following form:

$$\begin{bmatrix} K_{1,1} & . & . & K_{1,4} & . & . \\ . & K_{2,2} & . & . & K_{2,5} & . \\ . & . & K_{3,3} & . & . & K_{3,6} \end{bmatrix} \tag{7.46}$$

where the first 3 columns are the proportional gains, and the last 3 columns are the derivative gains.

The control torque is then calculated as follows:

$$\vec{T_c} = K_p \odot \vec{q}_{err_v} + K_d \odot \vec{\omega} \tag{7.47}$$

where $\odot$ symbolises ”element-wise multiplication”.

The vectorial part of the error quaternion is defined as follows:

$$\vec{q}_{err,v} = [q_{err,1},\ q_{err,2},\ q_{err,3}]^T \tag{7.48}$$

The proportional and derivative gains are as follows:

$$K_p = \begin{bmatrix} K_{1,1} \\ K_{2,2} \\ K_{3,3} \end{bmatrix} \qquad K_d = \begin{bmatrix} K_{1,4} \\ K_{2,5} \\ K_{3,6} \end{bmatrix} \tag{7.49}$$

Then, the control torque is calculated as follows:

$$\vec{T_c} = \begin{bmatrix} T_{c,x} \\ T_{c,y} \\ T_{c,z} \end{bmatrix} = \begin{bmatrix} K_{1,1} \cdot q_{err,1} \\ K_{2,2} \cdot q_{err,2} \\ K_{3,3} \cdot q_{err,3} \end{bmatrix} + \begin{bmatrix} K_{1,4} \cdot \omega_1 \\ K_{2,5} \cdot \omega_2 \\ K_{3,6} \cdot \omega_3 \end{bmatrix} = \begin{bmatrix} K_{1,1} \cdot q_{err,1} + K_{1,4} \cdot \omega_1 \\ K_{2,2} \cdot q_{err,2} + K_{2,5} \cdot \omega_2 \\ K_{3,3} \cdot q_{err,3} + K_{3,6} \cdot \omega_3 \end{bmatrix} \tag{7.50}$$

Inspecting the equation, it can be seen that only the vectorial part of the quaternion is used to calculate the control torque. The reason for this is that each quaternion is linked to the other three quaternion components through their basic relation, which is $q_1 + q_2 + q_3 + q + 4 = 1$. Thus, any of the four quaternions can be disregarded, and it was chosen to consider only the vectorial components.

The derivative gains of the LQR algorithm are applied to the angular velocity, ω, of the satellite to mitigate oscillations. This process involves multiplying the angular velocity by the derivative gains determined by the LQR, which are calculated to correct deviations from the desired state. The target state for the satellite comprises the desired attitude orientation, q_{ref}, and the desired rotational rate, ω_{ref}. Ideally, the satellite should maintain a stable orientation directed at its target, necessitating a reference rotational rate of [0, 0, 0] radians per second to ensure it remains steadily pointed without oscillating. This approach helps to smoothly align the satellite with its target by adjusting the oscillations through controlled angular velocity adjustments

7.5. Linking Operational Modes and Pointing Directions

This section presents the different steps needed to go from a discretised schedule of operational modes that considers only translational dynamics, to a continuous-time simulation that considers attitude dynamics. Subsection 7.5.1 presents the pointing directions linked with each operational mode considered for this research, with focus on the communication frame and tasked tracking frame, while subsection 7.5.2 introduces the approach employed to go from the discrete-time scheduling solution to the input needed by the continuous-time simulator.

Looking at Equation 7.50, it can be seen that the control torques are calculated based on the angular velocity and attitude error. As explained in section 3.3, the attitude error is found by calculating the difference between the current attitude and the desired attitude. The current attitude is propagated by numerically integrating the attitude dynamics and kinematics equations in Simulink, given an initial condition and external torques. The desired attitude, instead, needs to be calculated during pre-processing, using the results imported from the discrete-time scheduler.

7.5.1. Linking Operational Modes and Pointing Directions

Every operational mode is linked with a specific pointing direction. Section 3.4.3 shows examples of some common pointing strategies that are could be considered for the SURVEILLANCE mode of a space-based SSA mission.

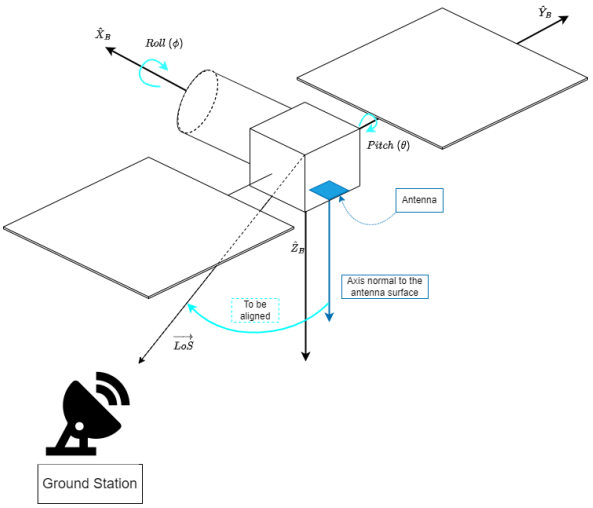


Figure 7.1: Visualisation of communication frame

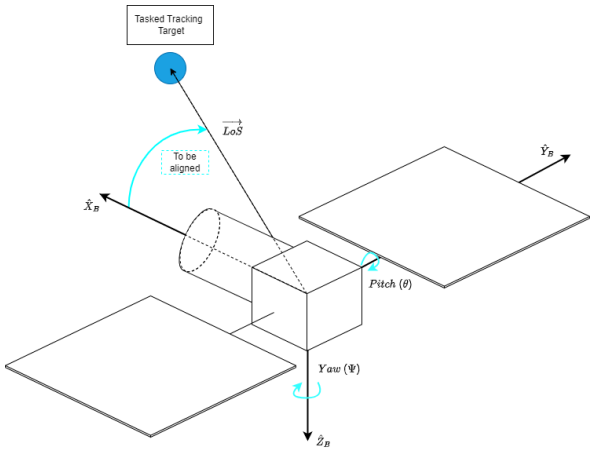


Figure 7.2: Visualisation of tasked tracking frame

Let's consider the case in which the satellite can operate in SURVEILLANCE, COMMUNICATION and TASKED TRACKING modes. Assume that in SURVEILLANCE mode, the second strategy introduced in subsection 3.4.3 is used. This strategy consists in pointing the telescope in the velocity direction. The reference attitude for these operational modes would be:

- SURVEILLANCE: The reference attitude in this mode would be following the orbit frame. In this case, the B_3 axis would be pointed towards the center of the Earth, B_1 axis would be pointed in the velocity direction, and B_2 would complete the right-handed system.
- COMMUNICATION: The reference attitude in this mode is to point the antenna, placed on the exterior of the satellite, towards the ground station that the spacecraft wants to communicate with. Given the location of the antenna on the satellite body, the reference attitude is such that the axis normal to the antenna in the body-frame is aligned with the LoS with the ground station, as shown in Figure 7.1;
- TASKED TRACKING: The reference attitude in this mode is to point the optical telescope in the direction of the target. As shown in Figure 7.2, this would mean aligning the x-axis of the $\mathbb{B}$-frame to the LoS with the target.

For the last two modes, the reference attitude has been defined only by aligning one axis with a certain direction. Such an orientation can be achieved in infinitely many ways, as the other two axis could be arranged in any arbitrary orientation that results in a right-handed system. It thus makes sense to investigate in practice how these "target-dependent" reference frames can be calculated.

Reference Frame for TASKED TRACKING mode

Consider the situation depicted in Figure 7.2, where the optical camera is aligned with the $\hat{x}_B$ axis. The unit vector coming out of the camera thus has coordinates $[1, 0, 0]$ in the body-frame.

By retrieving the positions of the observer satellite and of the optical target object, the Line-of-Sight vector from the observer to the latter can be calculated as follows:

$$\overrightarrow{LoS} = \vec{x}_{target} - \vec{x}_{s/c} \tag{7.51}$$

The $\hat{x}_B$ axis can then be set equal to this vector, such that the camera is aligned with the target.

The unit relative velocity vector between the observer satellite and the target object can be calculated as follows:

$$\hat{v}_{rel} = \frac{\vec{v}_{target} - \vec{v}_{s/c}}{\left\| \vec{v}_{target} - \vec{v}_{s/c} \right\|} \tag{7.52}$$

Taking the cross product between the LoS vector and the relative velocity vector, $\left(\widehat{LoS} \times \hat{v}_{rel}\right)$, gives the axis of rotation about which the satellite needs to rotate to keep the $[0, 0, 1]$ axis aligned with the $\widehat{LoS}$. This is depicted in Figure 7.3, where the satellite is depicted as stationary, such that the relative velocity between the satellite and the target is equal to the velocity of the target.

This vector can now be set equal to any arbitrary axis, about which the satellite will rotate to point its camera towards the target. For simplicity, let's consider the case in which one of the remaining body-axes, $\hat{z}_B$, is set equal to the negative of the cross product, as shown in Figure 7.3. The remaining body axis, $\hat{y}_B$, completes the right hand coordinate frame.

The physical meaning of this axes selection is that the satellite rotates about its yaw axis to point the camera towards the target.

The resulting axis selection for this mode is shown in Table 7.1.

Reference Frame for COMMUNICATION mode

Consider the situation depicted in Figure 7.1. In this figure, it is assumed that the antenna that the spacecraft uses to downlink data is positioned at the "bottom" of the spacecraft, towards the positive z-axis. So the unit vector normal to the antenna, expressed in the body-frame, has coordinates $[0, 0, 1]$.

By retrieving the positions of the observer satellite, $\vec{x}_{s/c}$ and of the ground station, $\vec{x}_{GS}$, the LoS vector from the satellite to the ground station can be calculated as follows:

$$\overrightarrow{LoS} = \vec{x}_{GS} - \vec{x}_{s/c} \tag{7.53}$$

This vector can then be used to define the orientation that the axis normal to the antenna should have. Thus:

- The $[0, 0, 1]$ axis in the body-frame should be aligned with the $\overrightarrow{LoS}$ calculated above.

The unit relative velocity vector between the observer satellite and the ground station can be calculated as follows:

$$\hat{v}_{rel} = \frac{\vec{v}_{GS} - \vec{v}_{s/c}}{\left\|\vec{v}_{GS} - \vec{v}_{s/c}\right\|} \tag{7.54}$$

Similarly to the TASKED TRACKING case, taking the cross product between the $\widehat{LoS}$ vector and the relative velocity vector, $\hat{v}_{rel}$, gives the axis of rotation about which the satellite can rotate to keep the $[0, 0, 1]$ axis aligned with the $\widehat{LoS}$.

This unit vector expressed in the body-frame can be set equal to any arbitrary axis. For simplicity, let's consider the case in which the $\hat{y}_B$ axis is aligned with this axis. The physical meaning this is that in order to keep the $\hat{z}_B$ axis pointed towards the antenna, the satellite rotates about its y-axis, pitching. Instead, setting $\hat{x}_B = \hat{v}_{rel}$ would lead the satellite to rotate about its x-axis, rolling to keep the antenna pointed in the correct direction. After setting the $\hat{y}_B$ axis, the $\hat{x}_B$ axis follows automatically, completing the right-hand system. The resulting axis selection for this mode is shown in Table 7.1.

Please be aware that in actual missions, the manoeuvres required to align an instrument with a specific target can become significantly complex due to the demands of the mission. For instance, the tracking frame might be configured to maximise solar panel exposure to the Sun. Conversely, for payloads that are highly sensitive, direct exposure to the Sun could damage the sensors. In such scenarios, the tracking frame is adjusted to prevent this. In this study, the models described previously are employed because they simplify the issues while still fulfilling their objectives.

Examples of reference frames describing the desired orientation for each operational mode considered have been defined in this section. Next, the way in which the operational scheduling solution generated by the discrete-time scheduler is used to calculate the reference attitude for each mode is presented.

Table 7.1: Example of ways to set-up the reference frame for target-dependent modes

TASKED TRACKING	COMMUNICATION
$\hat{x}_B = \widehat{LoS}$	$\hat{z}_B = \widehat{LoS}$
$\hat{z}_B = -\left(\hat{x}_B \times \hat{v}_{rel}\right)$	$\hat{y}_B = -\left(\hat{z}_B \times \hat{v}_{rel}\right)$
$\hat{y}_B = -(\hat{x}_B \times \hat{z}_B)$	$\hat{x}_B = (\hat{y}_B \times \hat{z}_B)$

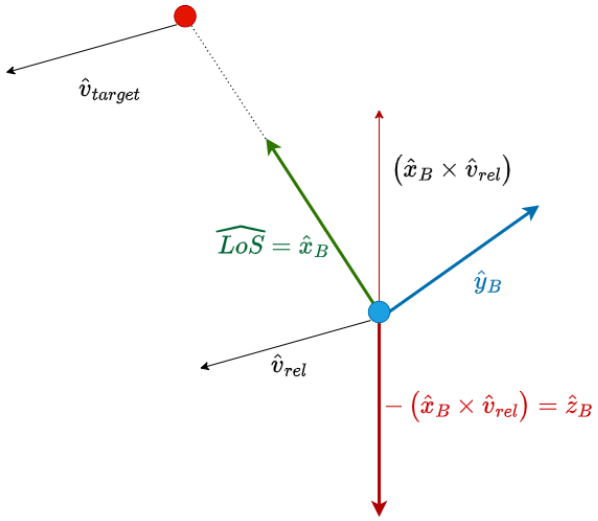


Figure 7.3: Example of axes selection for TASKED TRACKING mode

7.5.2. Going from Discrete-Time Scheduling Solution to Continuous-Time Simulator Input

The scheduling solution generated by the discrete-time scheduler consists of:

- Set of time intervals, I;
- What operational mode the spacecraft is operating in for each interval i;
- The target, in case the satellite is operating in a target-dependent mode;
- The amount of time spent in the non-default operational mode, if any.

In the simulator, a certain orientation is achieved by feeding the corresponding reference attitude to the PD controller. Then, the PD controller outputs a control torque such that the orientation of the body-frame becomes aligned with the reference attitude. A way to be able to switch between operational modes that require different reference attitude is to generate 1) a continuous-time reference attitude for each operational mode, and 2) a continuous-time "switch array" that indicates what operational mode to be. Then, based on the operational mode indicated by the *switch array*, the corresponding reference attitude is fed to the PD controller. This approach can be expanded to include as many operational modes as desired, as long as their reference attitudes can be calculated with the information calculated in the discrete-time scheduler.

The reference attitudes linked with each operational mode considered have been described before. For the SURVEILLANCE mode, and more in general for any target-independent modes, the reference attitude can be propagated based on the position of the spacecraft along its orbit. For the target-dependent modes, as shown above, the reference attitude is calculated based on the position of the observer satellite and of the target, it being an optical target or a ground station. Both of these are imported from the discrete-time scheduler output.

In the case in which the scheduling solution includes intervals in which a non-default operational mode is

Table 7.2: Resources dynamics terms for SURVEILLANCE and COMMUNICATION mode.

Resource Dynamics Rate	SURV	COMM	Unit
$\dot{E}^-$	$\dot{E}^-_{SURV}$	$\dot{E}^-_{COMM}$	Joules-per-second
$\dot{D}^+$	$\dot{D}^+_{SURV}$	0	bit-per-second
$\dot{D}^-$	0	$\dot{D}^-_{COMM}$	bit-per-second
$\dot{D}^-_{loss}$	const.	const.	bit-per-second

scheduled for a smaller duration than the whole duration of the interval (i.e. $t_{COMM,i} < \Delta t_i$ or $t_{TT,i} < \Delta t_i$), then it needs to be decided *when, within the interval, the operational mode is to be carried out*. For simplicity, and also following the approach of some researchers in the EOSSP [7] [52], the following assumption is considered:

- "Whenever an operational mode is scheduled for a duration smaller than the full duration of the interval, the operational mode will be carried out at the first possible opportunity"

Such an approach can be referred to as a *greedy* approach, as the satellite performs its task at the first possible opportunity.

In practical terms, this means that if interval i has a duration of 5 minutes, and the discrete-time scheduler outputs a solution that states "in interval i the satellite goes in COMMUNICATION mode for 3 minutes", then the satellite will spend the first 3 minutes of interval i in COMMUNICATION mode, and the remaining 2 minutes in the default mode.

7.6. Checking Feasibility

The primary goal of the continuous-time simulator is to check the feasibility of the solution generated by the discrete-time scheduler. This means checking whether the resources constraints imposed in the problem definition have been respected. In this work, it is assumed that the angular momentum built-up by the reaction wheel onboard the spacecraft is continuously dumped by the use of magnetorquers. In this way, the momentum build-up does not need to be taken into account. The two resources that need to be modelled are thus the energy and the data. Since the modelling done here is slightly different than in the discrete-time scheduler, a dedicated section for each resource is presented below.

Before investigating the resource dynamics, it is worth listing the different components that make-up the resource dynamics profile for the two operational modes considered, shown in Table 7.2.

7.6.1. Data Dynamics

Given an initial condition, the data stored in the onboard memory is modelled by recursively computing the following equation:

$$D(t+1) = min\left[D(t) + \left(\dot{D}^+(t) - \dot{D}^-(t)\right) \cdot \Delta t, D_{max}\right] \tag{7.55}$$

where:

- Δt is the fixed-time-step;
- D_{max} the maximum data level, which represents the onboard memory storage;
- $D(t)$ and $D(t+1)$ are the data levels at time t and $t+1$, respectively, and represents the onboard memory storage;
- $\dot{D}^+(t)$ is the rate of data collection at time t;
- $\dot{D}^-(t)$ is the rate of data dumping at time t.

In the case where both data loss and data downlink are considered, the last term is composed of two different terms: $\dot{D}^{-}_{downlink}$ and $\dot{D}^{-}_{loss}$. From the point of view of the onboard memory storage, these two have the same effect of reducing the amount of onboard data, and are thus lumped together in the equation above.

The rate of data loss is not dependent on the operational mode of the satellite, and is thus constant throughout the simulation. The total amount of data lost can thus be calculated simply by multiplying the data loss rate by the scheduling horizon duration:

$$D_{lost} = \dot{D}^{-}_{loss} \cdot T \tag{7.56}$$

where T is the duration of the scheduling horizon.

The rate of data downlink instead depends on the operational mode of the satellite. As a result of this, the total amount of data downlinked throughout the scheduling horizon can be calculated as follows:

$$D_{downlinked} = \int_0^T \dot{D}^{-}_{downlink}(t)\ dt \tag{7.57}$$

$$\tag{7.58}$$

In the same way as for the discrete-time scheduler, the excess data is calculated as follows:

$$H^D(t) = max\left[D(t) - D_{max}, 0\right] \tag{7.59}$$

7.6.2. Energy Dynamics

Given an initial condition, the energy onboard the spacecraft is modelled by recursively calculating the SoC of the onboard battery. This can be done by calculating the following equation:

$$E(t+1) = min\left[E(t) + \left(\dot{E}^{+}(t) - \dot{E}^{-}(t)\right) \cdot \Delta t, E_{max}\right] \tag{7.60}$$

where:

- Δt is the fixed-time-step;
- E_{max} the maximum energy level, which represents the battery capacity;
- $E(t)$ and $E(t+1)$ are the energy levels at time t and $t+1$, respectively, and represents the battery SoC;
- $\dot{E}^{+}(t)$ is the rate of energy collection at time t, which represents the power production of the solar arrays;
- $\dot{E}^{-}(t)$ is the rate of energy consumption at time t, which represents the power consumption of the spacecraft in the current operational mode.

In the same way as for the discrete-time scheduler, the excess spilled energy is calculated as follows:

$$H^E(t) = max\left[E(t) - E_{max}, 0\right] \tag{7.61}$$

Energy Consumption Rate Term

In the above equation, $\dot{E}^{-}(t)$ depends on the operational mode the spacecraft is operating in. This is a result of the fact that in each operational modes, different subsystems are operating, and at different levels of power consumption. For example, in SURVEILLANCE mode the optical payload is going to consume a high amount of energy, while in COMMUNICATION it will not consume any. On the other hand, the TT&C subsystem will consume more energy in COMMUNICATION than in SURVEILLANCE.

As this term is fully dependent on the operational mode of the satellite, the values of $\dot{E}^{-}$ throughout the whole scheduling horizon can be calculated once the scheduling solution is generated.

Energy Collection Rate Term

The energy collection rate term, $\dot{E}^{+}(t)$ depends instead on the orientation of the satellite with respect to the Sun. Provided the location of the solar arrays on the exterior of the satellite, as well as their area and efficiencies are known, the power generated by the satellite at any moment throughout the scheduling horizon can be determined. A representation of the situation is depicted in Figure 7.4. In order to calculate the power generated by each solar panel, the dot product between the normal to each panel and the Line-of-Sight vector with the Sun has to be calculated. If this product is greater than 0, this means that the solar panel is facing into the Sun, and it will generate power, as for panel 1 in Figure 7.4. If it is smaller than 0, then the solar panel is not facing the Sun, and will not generate any power, as for panel 2. The total power generated at any time t can be then found by summing the contribution of each solar panel.

This can be expressed with the following equations:

$$P_i = S_{Sun}\ A_i\ (\hat{n}_i \cdot \widehat{LoS}_{Sun}) \tag{7.62}$$

$$P_i = S_{Sun}\ A_i\ \cos\theta_i \tag{7.63}$$

$$P_{tot} = \sum_{i=1}^{N} P_i \tag{7.64}$$

where:

- S_{Sun} is the Sun power flux at 1 AU from the Sun, in W/m^2;
- A_i is the area of the i^{th} panel;
- $\hat{n}_i$ is the unit vector perpendicular to the i^{th} panel;
- $\widehat{LoS}_{Sun}$ is the unit Line-of-Sight vector from the satellite to the Sun;
- θ_i is the angle between the $\hat{n}_i$ and $\widehat{LoS}_{Sun}$;
- P_i is the power generated by the i^{th} panel;
- P_{tot} is the total power generated.

$\widehat{LoS}_{Sun}$ can be retrieved by using the position vectors of the satellite and of the Sun, which are imported from the discrete-time scheduler. $\hat{n}_i$ can be retrieved by the propagated attitude of the spacecraft, based on the location of the solar panel on the spacecraft's body. From this, a continuous-time power generation profile can be generated. This can be used to generate a new discretisation scheme for the MILP problem, as discussed in the following section.

It has been shown how all the terms of the data- and energy-update equations can be calculated for every time-step of the simulation. From this, the continuous-time battery SoC and the onboard memory level can be calculated. If, at any time t, the SoC goes below the minimum threshold imposed by the operational constraints, or the memory level goes below the minimum threshold, the scheduling solution is unfeasible.

7.7. Using the Results of the Continuous-Time Simulator

In this section, a different way of discretising time into intervals then the one proposed in subsection 4.4.3 is presented. This new discretisation scheme uses the result of the continuous-time simulator to increase the fidelity of the model in the discretised, under-constrained formulation. The aim of this procedure is to improve the solution generated by the MILP algorithm, and to achieve feasibility when scheme A fails to do so.

Discretisation Scheme B

In this discretisation scheme, the continuous-time energy production rate profile is used. The events that trigger the start of a new interval are:

- Whenever a ground station comes in- or out-of-view of the satellite;

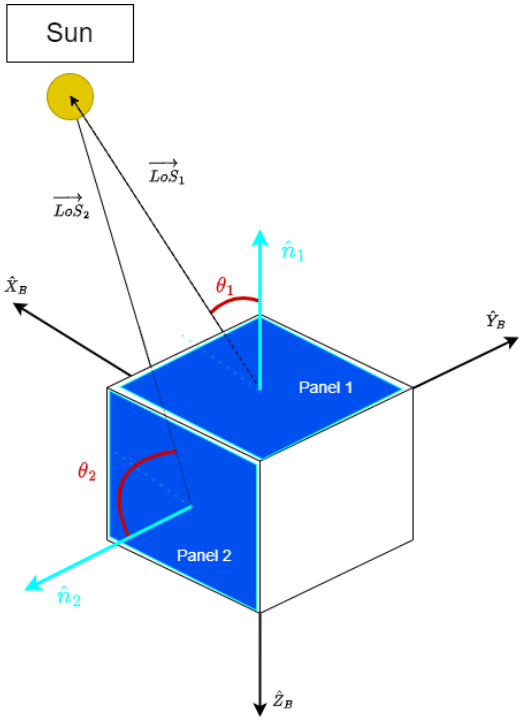


Figure 7.4: Visualisation of how the energy collection term can be determined, based on the angle between the $\overrightarrow{LoS}$ with the Sun and the axes normal to each solar array. The camera is omitted in this depiction, for clarity.

- Whenever the power generation profile of the satellite changes by a pre-determined amount;
- Whenever the illumination condition of the satellite goes from eclipse to non-eclipse.

Throughout the rest of the report, **the power step-size used to discretise the power generation profile when using discretisation scheme B is referred to as Γ.**

Note that the actual power generation profile is dependent on the operational schedule of the satellite. The reason for this is that, depending on what operational mode the satellite undertakes, it will perform a different manoeuvre, which will lead to a different orientation with respect to the Sun, which in turn will lead to a different illumination condition of the solar arrays. An example of this is illustrated in Figure 7.5, which compares the power generation profiles of two operational schedules: one where the spacecraft remains in SURVEILLANCE mode by default but performs occasional tracking manoeuvres, and another where the spacecraft stays in SURVEILLANCE mode for the entire scheduling horizon under the same conditions. Since in SURVEILLANCE mode the satellite maintains a fixed orientation with respect to the orbit frame, and the satellite is in a Sun-Synchronous Orbit, it makes sense that the power generation profile is periodic. The portions of the plot in which the power generated is equal to 0 correspond to eclipses periods. By comparing the two plots, it is clear to see when in the scheduling horizon the satellite performed tracking manoeuvres.

Figure 4.8 shows the pipeline of the approach followed. The procedure currently described is represented by the ”Discretise time into I_{k+1} intervals”. Looking at the pipeline, it can be noticed that the newly discretised intervals will become an input to the discrete-time scheduler, and will be used by the MILP algorithm to find a scheduling solution. This means that the operational schedule that will be generated using this discretisation scheme will be different from the one generated using discretisation scheme A. This means that discretisation scheme B should be constructed without considering the solution of the discrete-time scheduler, but should be solution-independent.

To achieve this, the fact that a default operational mode is considered in the scheduling frameworks developed is exploited. Since the default mode is linked with a certain orientation, a continuous-time power generation profile can be generated, by considering a schedule in which the satellite operates only in its default operational mode. Figure 7.5 gives a visual representation of how the power genera-

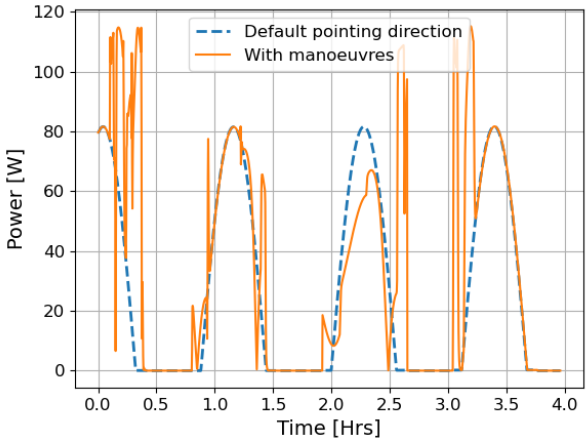


Figure 7.5: Power generation profile outputted by the continuous-time simulator for two cases: first for the case in which the default pointing direction is maintained for the whole scheduling horizon; second for the case in which tracking manoeuvres are considered.

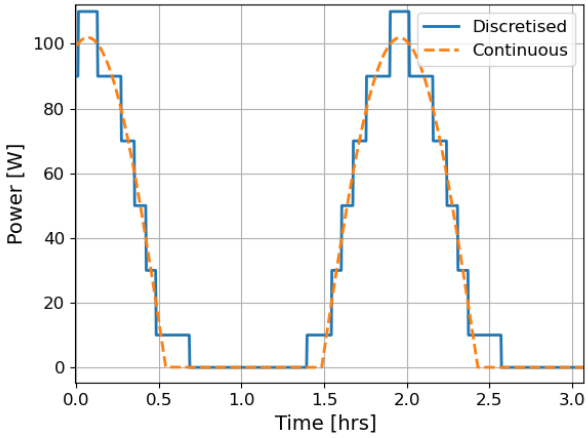


Figure 7.6: Continuous power generation profile vs power generation profile discretised into constant segments. Γ = 20 W

tion profile is approximated.

Once the power generation profile resulting from the satellite maintaining the orientation defined by its default mode is obtained, this is discretised into ”constant power generation segments”. For example, using a value of 20 W for Γ, the time can be discretised as follows:

- $0W < P \leq 20W \rightarrow P = 10W$;
- $20W < P \leq 40W \rightarrow P = 30W$;
- $40W < P \leq 60W \rightarrow P = 50W$;
- $60W < P \leq 80W \rightarrow P = 70W$;
- ...

This discretisation into segments of constant values can be visualised in Figure 7.6. If the solution generated with this new discretisation scheme results unfeasible, then a lower value of Γ can be used, to use a more refined discretisation scheme.

In the next chapter, the results of the complete pipeline of the different frameworks introduced in the previous chapters are presented.

8

Results

In this chapter, the results obtained through the implementation of the method introduced in the rest of the report are presented. The chapter's structure is as follows: section 8.1 presents a recap of the rationale for conducting this research, such that the reason behind the results shown in the rest of the chapter is clearly indicated. Subsequently, an overview of all the input parameters for the problem is presented in section 8.2, briefly describing their characteristics. The input parameters have been divided into three groups, for clarity. Then, section 8.3 introduces the metrics that are used to assess the quality of the scheduling solutions. Section 8.4 shows the results of the nominal run considered for the SURVEILLANCE-COMMUNICATION (SC) framework, and describes the plots that can be used to gain insights on the behaviour of the satellite throughout the scheduling horizon. Next, subsection 8.4.7 shows the results of the sensitivity analysis of the SC framework. Subsequently, section 8.5 discusses the attempt made to incorporate the IDLE and TASKED TRACKING modes into a single framework, while section 8.6 and section 8.7 discuss the results of the two novel frameworks developed for this thesis. Section 8.8 concludes the chapter, presenting an analysis of the computational performance of the three frameworks.

The chapter is structured such that it follows the research questions, which are answered explicitly in the next chapter. For clarity, the research questions are listed below:

Research Questions

RQ1: *"What is the best scheduling strategy for SURVEILLANCE mode, that maximises the amount of data downlinked while satisfying the energy and data requirements?"*

RQ2: *"To what spacecraft and operational constraints is the scheduling framework most sensitive?"*

RQ3: *"How do initial conditions and operational scenarios affect resource management along the (continuous) scheduling horizon?"*

RQ4: *"What are the impacts of including additional operational modes (i.e. IDLE and TASKED TRACKING) in the scheduling framework?"*

Note that the Canadian space-based surveillance satellite Sapphire is used as a case study for this result chapter [10]. When specific parameter characteristics for the Sapphire mission were unavailable, values aligned with commercially available hardware for the same class of space missions were considered.

8.1. Recap

In this section, a brief recap of the rationale for producing the results is presented, by summarising the most relevant points from the previous chapters.

8.1.1. Motivation: Shifting the Focus from Earth-Observation to Space-Based SSA

This thesis addresses a gap in satellite operations scheduling by developing a framework explicitly for space-based SSA missions, utilising existing scheduling approaches. Unlike Earth-Observation satellites, which typically perform observation and communication tasks with their instruments directed in a fixed orientation, SSA missions for satellites equipped with optical telescopes require consideration of multiple observation modes such as SURVEILLANCE and TASKED TRACKING. Additionally, incorporating an IDLE mode could enhance the flexibility of the scheduling framework. This study investigates these aspects to develop a robust and adaptable scheduling framework for space-based SSA missions.

8.1.2. Limitations of the Approach Followed

The approach taken from the field of the EOSSP, based on the work of Spangelo et al. [50], presents some limitations. The main issue of the framework is that it solves the scheduling problem only at discretised time steps, assuming linear dynamics within these intervals. This can result in infeasible operational scheduling when considering continuous-time attitude and resource dynamics onboard the spacecraft. Consequently, a continuous-time simulator was developed to ensure feasibility and operational accuracy when considering continuous-time constraints. This advancement allowed for more precise and realistic scheduling, reflecting the true operational dynamics of the system. Furthermore, the approach exhibits limitations due to the mismodelling of the data collection term, assuming continuous data collection even during communication, which results in infeasible downlink scheduling. Additionally, the framework only considers two operational modes: continuous data collection and downlink to a ground station. For typical space-based SSA missions, a distinction between data collection modes, such as SURVEILLANCE and TASKED TRACKING are usually considered. The inclusion of an IDLE mode in the framework, while not unique to SSA missions, allows for greater flexibility and adaptability in scheduling.

8.1.3. Integration and Problem Re-Orientation

After integrating the discrete-time scheduler with the continuous-time simulator considering only SURVEILLANCE and COMMUNICATION modes, it became apparent that further investigation into this integration was less critical than addressing the broader problem of incorporating various operational modes typical of SSA missions. Therefore, the focus shifted towards developing a framework that could handle multiple operational modes, including IDLE and TASKED TRACKING, to optimise the scheduling of SSA tasks comprehensively.

Rationale for IDLE Mode

The integration of an IDLE mode into the scheduling framework allows the satellite to go into a mode where only the critical subsystems are operational, and no scientific activities are carried out. The inclusion of this mode allows the framework to find feasible solutions in scenarios where the energy resource constraint is more stringent, as the satellite can strategically schedule periods of low-power activities in non-favourable illumination conditions, to recharge the batteries and collect, or downlink, scientifically relevant data when better illumination conditions arrive.

A practical scenario in which the integration of this mode could be useful is as follows:

- End-Of-Life (EOL) Operations: At the end of a satellite's operational life, the degradation of solar panels or batteries often occurs. Despite such deterioration, mission extensions are possible provided the other subsystems remain functional, allowing the satellite to continue operating. This scenario necessitates the satellite to function in an environment where its energy production is below its designed capacity. Consequently, an optimisation framework that assumes continuous

Table 8.1: Overview of constants used for the simulation

Definition	Symbol	Value	Unit
Earth Gravitational Parameter	μ	$3.986004418 \cdot 10^{14}$	m^3/s^2
Radius of the Earth	R_E	6371000	m
Radius of the Sun	R_{Sun}	696000000	m
Astronomical Unit	AU	149597870691	m
Speed of Light	c	299792458	m/s
Mean Solar Flux @ 1 AU	$Flux_{Sun}$	1358	W/m^2

payload usage would fail to generate feasible solutions. To address this challenge, the implementation of an IDLE mode is essential, enabling the satellite to manage reduced energy availability effectively.

Rationale for TASKED TRACKING Mode

As mentioned in the work in Olmos et al. [38], a space-based SSA network could comprise satellites fully devoted to surveillance tasks, while others could additionally be devoted to tasked tracking tasks. In the latter, users could request an observation of a certain target. Whenever the conditions to be able to detect the target object are met, the satellite performs a manoeuvre to point the telescope towards it. In this way, the state of the target object can be detected, and its uncertainty can be reduced. Since this operational mode is typical of space-based SSA missions, incorporating this into a scheduling framework was deemed important for this work.

These points collectively underscore the progression from a narrow focus on discrete-time scheduling to a more robust and versatile framework capable of supporting the complex operational needs of space-based SSA missions. This evolution reflects the growing importance of SSA in maintaining space environment safety and highlights the necessity for advanced scheduling solutions.

8.2. Inputs Overview

In this section, the inputs that are considered for the simulations are presented.

8.2.1. Constants

In order to be able to replicate the results, the definition, symbol, value and unit of the constants used in the simulations are presented in Table 8.1 [60].

8.2.2. Constant Input Parameters

The parameters that are kept constant throughout the simulations are mostly related to the satellite's physical properties. These are based on the Sapphire satellite, which consists of a 1x1x1 m cube of approximately 150 kg [10]. The drag and reflectivity coefficients are chosen both equal to 2, to consider a case in which disturbance torques have a large impact on the dynamics. For the CAD model, required for the aerodynamic and SRP torque calculation in the continuous-time simulator, a cube of 1 m per side is used, for simplicity. For the linear inertia matrix, that of a cube, with a value of $\frac{m_{sat}s^2}{6}$ was used, where s is the length of one side of the body. For the Centre-of-Mass (CoM) location, a value of [0.5, 0, 0] m value was used, assuming a symmetric distribution of the mass about two of the body-axes. These are shown in Table 8.2.

Note: the CAD model will be kept as for body-mounted solar panels. By varying the size of the solar panels, the user is able to set a size that is larger than one of the sides of the satellite. In this way, the effect of having a deployable solar array can be investigated.

Table 8.2: Input parameters that will be kept constant

Definition	Symbol	Value	Unit
Satellite mass	m_{sat}	150	kg
Satellite size	s x s x s	1 x 1 x 1	m x m x m
Satellite drag coefficient	C_d	2	–
Satellite reflectivity coefficient	C_r	2	–
Reference area	A	1	m^2
Satellite CAD model	–	Cube	–
Satellite linear inertia matrix	J	$(m_{sat}s)/6\mathbb{I}$	kg m^2
Satellite Centre-of-Mass location	CoM	[0.5, 0, 0]	[m, m, m]
Initial angular velocity	ω_{initial}	0	rad/s
Desired angular velocity	ω_{desired}	0	rad/s

8.2.3. Input Parameters Group A

The inputs on which the division of the scheduling horizon depends on are the following:

- Simulation start date;
- Simulation duration;
- Satellite initial state;
- GSN;
- Target object initial states (for TASKED TRACKING only);
- Payload optical characteristics (for TASKED TRACKING only).

Simulation Start Date

Figure 8.1 shows the variation of the eclipse duration throughout a year. The plot was produced for an orbit at 800km, with a RAAN of 75°. By changing the simulation start date, and keeping the RAAN constant, the Local Time of the Ascending Node (LTAN) is varied. It can be seen that the eclipse duration has quite a big variation, going from 0 to over 35 minutes. This is expected to have a big impact on the scheduling solution, as the power generated by the solar arrays will be smaller, allowing for less tasks to be completed. In the plot, four different ”eclipse regimes” are identified, based on the duration of the eclipse periods. To check the impact this has on the solution, a simulation for each of the four start days will be performed. These days correspond to the following: 03/06 - 03/08 - 11/08 - 07/09 (DD/MM format).

Mission Duration - Scheduling Horizon

The mission duration is expected to also have a big effect on the scheduling solution. In mission planning, different planning horizons can be considered, based on what wants to be investigated. Short mission planning horizons can be as small as several hours, while longer planning horizons can be up to a week. Longer planning horizons are usually not considered in scheduling the operations of Earth-orbiting satellites [50].

Satellite's Initial State - Satellite Orbit

The satellite’s initial state dictates its orbit for the duration of the simulation. Different orbits result in a different ground track, effectively resulting in different contact times with the GSN. Additionally, they result in different illumination conditions. The communication windows and illumination condition are both key factors in how the scheduling horizon is split into intervals, which lays the foundation of how the MILP problem is set-up. Thus, in order to check the impact of other parameters on the scheduling solution, the satellite’s orbit needs to be kept the same. For this reason, a nominal orbit will be considered, based on promising orbits for space-based SSA mission, and only a few orbital elements will be changed.

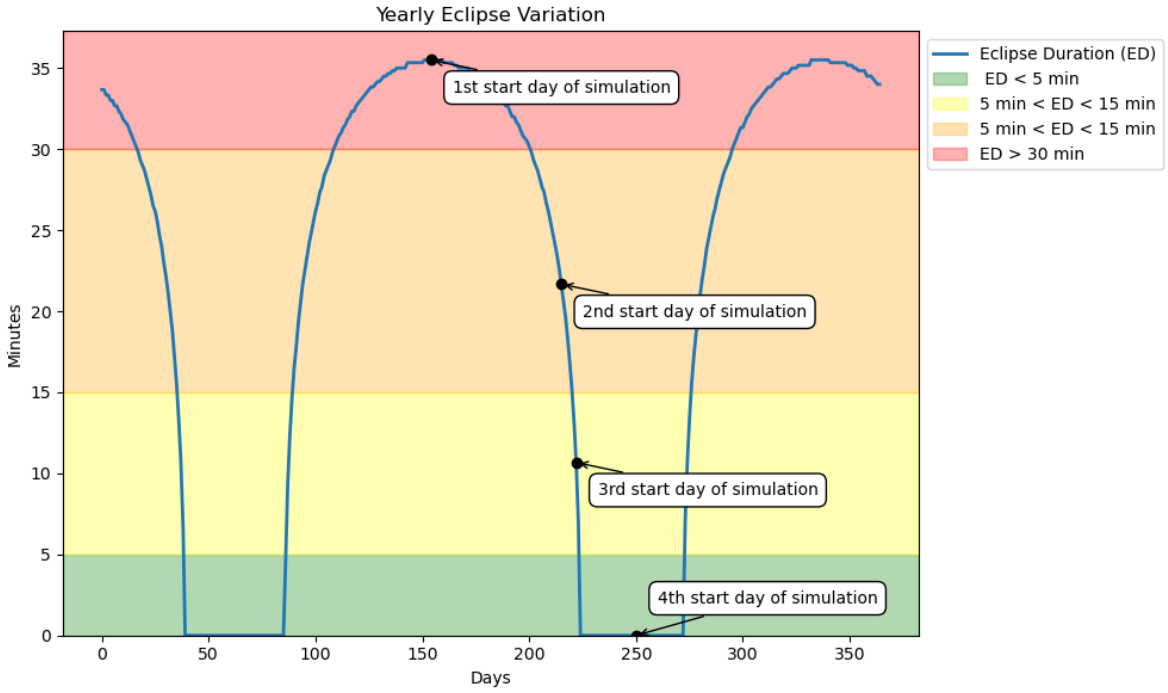


Figure 8.1: Variations in eclipse duration throughout a year of simulation. Start date: 12:00:00 01-01-2025 - Altitude=800km - Eccentricity=0 - Inclination=97.8° - AoP=0° - RAAN=75° - True Anomaly=0°

Table 8.3: Classical Orbital Elements of the nominal orbit considered for the results [18]

COEs	Value	Unit
Altitude	800	km
Eccentricity	0	-
Inclination	98.57	°
Argument of Periapsis	0	°
Right Ascension of Ascending Node	75	°
True Anomaly	0	°

The nominal orbit is taken from the work of T. Flohrer ”Optical survey strategy and their application to space surveillance” [18]. This scenario consist of an SSO orbit at 800 km. The corresponding orbital elements are listed in Table 8.3.

Since the orbit considered is circular, the argument of periapsis and true anomaly have a small effect on the resulting orbit, and for this reason are set to 0°. The RAAN, instead, determines the illumination condition of the satellite, together with the orientation of the Earth with respect to the Sun, which is itself dependent on the simulation start date. Since different combinations of start date and RAAN would result in different illumination conditions, the RAAN is kept fixed. The result of changing the start date while keeping the RAAN fixed at 75° can be seen in Figure 8.1.

According to Flohrer, the advantage of SSO orbits comes from the good accessibility to downlink data given by high-latitude ground stations in terms of timeliness of ground contacts and duration of passes. Additionally, the orbit being Sun-Synchronous allows the illumination conditions of the satellite to be periodic throughout subsequent orbits, allowing for a better planning of its tasks.

Note that, among all the initial orbital elements, only the satellite’s altitude will be varied to address the second research question, which is to investigate which operational constraint the scheduling framework is sensitive to. The remaining orbital elements will be held constant. This approach allows for the isolation and more precise analysis of the impact that variations in altitude have on the scheduling framework.

GSN

For the GSN, three different providers are considered, to check what the effect of changing the GSN has on the scheduling solution. The location of the ground stations that make up the three GSN are presented in the confidential annex, which is not included in this version of the report, where they will be referred to as GSN provider A, B and C. A minimum elevation angle of 15° is assumed for all ground stations.

8.2.4. Input Parameters Group B

The inputs below are not needed to discretise time into intervals, do not come directly from hardware specifications, but affect the feasibility of the scheduling solution. These parameters usually arise from mission requirements.

- Maximum allowable DoD;
- Minimum allowable memory storage;
- “Pointing direction” for SURVEILLANCE mode;
- “Pointing direction” for IDLE mode;
- “Tracking strategy” for COMMUNICATION mode;
- “Tracking strategy” for TASKED TRACKING mode.

With “tracking strategy” it is meant the way in which the reference frame for the desired attitude is constructed. This is explained in more detail in subsection 7.5.1. For COMMUNICATION mode, this is not expected to have a big impact on the scheduling solution. The reason for this is that the antenna for downlink is usually placed on the“Earth-facing” side of the satellite, and thus only a small rotation is needed to point it towards the ground station.

In this work, it is assumed that the satellite maintains the same orientation in IDLE mode as it does in SURVEILLANCE mode. In this way, it is assumed that only the ADCS subsystem continues to operate when in IDLE mode, keeping the satellite in the desired orientation.

Nominal Values for Input Parameters Group B

- $DoD_{max} = 30\%$;
- Memory is allowed to be completely emptied when downlinking;
- In SURVEILLANCE mode, the satellite points in the direction of the velocity vector;
- In IDLE mode, the satellite keeps the same direction as in SURVEILLANCE mode. This holds also for the ITC framework;
- The tracking strategies considered for COMMUNICATION and TASKED TRACKING are those shown in Table 7.1.

8.2.5. Input Parameters Group C

The inputs below arise directly from the characteristics of the hardware onboard the satellite. These characteristics are shown in Table 8.4, together with the physical quantity that they determine.

The data loss rate is also assumed constant, for simplicity. The characteristics of the other parameters are going to be varied, to investigate the sensitivity of the framework to these parameters.

Simplified Energy Collection Rate Modelling

In order for the solution of the discrete-time scheduler to be feasible when using discretisation scheme A, its energy collection rate term for when the satellite is in sunlight, $\dot{E}^{+}$, should be realistic. This can be achieved by relating this term to the solar array characteristics. Considering a satellite of cubical shape with 3 body-mounted solar panels, the following approach is proposed.

Table 8.4: Input parameters derived from characteristics of hardware onboard the spacecraft. *The solar array characteristics that can be chosen are: 1) their area, 2) their efficiencies, 3) their location on the satellite's body. How the power is calculated from these parameters is explained in subsection 7.6.2.

Hardware	Physical Quantity	Symbol
Battery capacity	Maximum Energy Storage	E_{max}
Solar arrays*	Energy Collection Rate	$\dot{E}^{+}$
Memory storage	Maximum Data Storage	D_{max}
TT&C antenna	Data downlink rate	$\dot{D}^{-}_{downlink}$
	Energy consumption rate for downlinking	$\dot{E}^{-}_{downlink}$
Payload	Data acquisition rate	$\dot{D}^{+}_{observing}$
	Energy consumption rate for observing	$\dot{E}^{-}_{observing}$
Onboard computer	Data loss rate	$\dot{D}^{-}_{loss}$

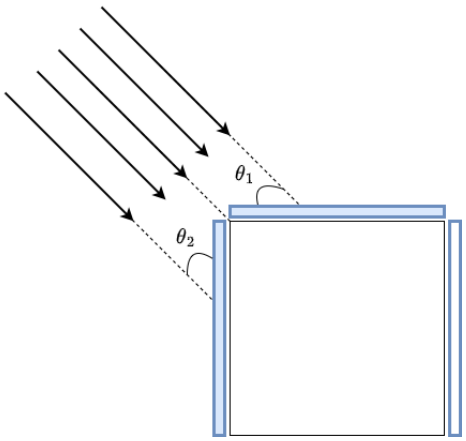


Figure 8.2: Best-case illumination condition for 3 body-mounted solar panels.

Assume three solar panels, on three sides of the satellite. With this configuration, the best illumination condition possible is achieved when the Sun is at 45° with respect to two panels. This situation is depicted in Figure 8.2. Defining A_1 and A_2 as the areas of the two solar panels, $\theta_1 = \theta_2 = 45°$ as the incident angle, and η as their efficiency, their power generated can be calculated as follows:

$$P_1 = A_1 \cdot Flux_{Sun} \cdot \eta \cdot \cos\theta_1 \tag{8.1}$$

$$P_2 = A_2 \cdot Flux_{Sun} \cdot \eta \cdot \cos\theta_2 \tag{8.2}$$

$$P_{tot} = P_1 + P_2 \tag{8.3}$$

$$P_{tot} = (A_1 + A_2) \cdot Flux_{Sun} \cdot \eta \cdot \cos 45° \tag{8.4}$$

$$P_{tot} = (A_1 + A_2) \cdot Flux_{Sun} \cdot \eta \cdot \frac{\sqrt{2}}{2} \tag{8.5}$$

where η is the efficiency of the solar panels. For the efficiency, a fixed value of 30% is assumed, based on the high-efficiency solar panels offered by Blue Canyon Technologies [54].

What is shown above is the highest power generated by the solar panels, with the most ideal incident angle. What discretisation scheme A needs is an average value for when the satellite is in illumination condition. Looking at an example of continuous power generation profile, such as that shown in Figure 7.6, it can be seen that, when coming out of an eclipse period, the power increases rapidly. A simplifying assumption can thus be made, in which the total power generated in the best illumination case is multiplied by a reducing factor, and this value is used as the constant energy production rate

Table 8.5: Nominal numerical values for the resource profiles of the different operational modes considered for the scheduling frameworks.

Mode	$\dot{E}^-$	$\dot{D}^+$	$\dot{D}^-_{downlink}$	$\dot{D}^-_{loss}$	E_{max}	D_{max}
SURVEILLANCE	45 W	15 Mbps	0	–	–	–
COMMUNICATION	70 W	0	200 Mbps	–	–	–
Common	–	–	–	10 kbps	600 Wh	150 GB

when the satellite is in sunlight. Arbitrarily, a value of 2/3 is selected for this reducing factor, and a brief discussion on the effect this has on the scheduling solution is made in the following sections. By comparing the values of the total energy collected throughout the planning horizon, $\int_0^T \dot{E}^+$, outputted by the discrete-time scheduler and the continuous-time simulator, the validity of this assumption is checked. If the two values present a large difference, then this simplified assumption is not representative of the real situation. This only holds when using discretisation scheme A, as in discretisation scheme B, the continuous-time power generation profile is used to get the $\dot{E}^+$ term.

Nominal Values for Resource Dynamics

The values considered for the parameters of group C are shown in Table 8.5. These values are based on commercially available hardware components, typical for small spacecrafts, such as the Canadian Sapphire satellite [10]. Additionally, three body-mounted solar arrays with an area of 0.2 m^2 are considered. More on this is presented when investigating the sensitivity to the eclipse variation.

8.2.6. Initial Resources Levels

The initial energy and data levels, $E_{i=0}$ and $D_{i=0}$, can also be varied. By default, the energy is set to 90%, and the data to 30%. The initial energy level is set-up such that different levels of DoD_{max} can be tested, without worrying about the solution being infeasible from the start due to the DoD_{max} constraint. The initial data level has been set arbitrarily.

8.3. Metrics

In order to evaluate the performance of the optimisation framework, some metrics have to be devised. Two types of metrics can be identified, one related to computational performance, and one related to the quality of the scheduling solution.

8.3.1. Computational Performance

As mentioned in section 2.6, researchers in the Earth-Observation Satellite Scheduling (EOSSP) problem field mainly focus their efforts on the computational performance of different algorithms. Typical metrics that are used to evaluate this performance are:

- Solve Time vs Number of Intervals;
- Solve Time vs Planning Horizon;
- Solve Time vs Average Number of Options per Interval;

These metrics are used by several researchers in the EOSSP [50] [7] [52], and will be used in this chapter to evaluate the different frameworks.

8.3.2. Quality of Scheduling Solution

In scheduling optimisation problems, part of the quality of the scheduling solution is captured by the definition of the objective function. By defining the latter, it is decided what parameter to either maximise or minimise, in this way defining what is meant with a “good quality solution”. For example, in the

scheduling framework introduced in chapter 5, the objective function is to maximise the amount of downlinked data. From this, it can be inferred that the quality of the scheduling solution outputted by the optimisation algorithm is defined by the amount of data downlinked to a GSN.

However, this approach narrowly evaluates the solution using just one parameter, while other factors are only considered for constraint checks. This means significant information linked to the scheduling solutions is overlooked, potentially excluding important insights. For instance, the resource utilisation profiles, which can be generated post-solution, offer deeper insights into the solution and reveal parameters to which the framework is sensitive.

Therefore, after maximising or minimising a specific cost function to generate a solution, it is important to examine the resulting resource profile for further insights.

Energy Dynamics Metrics

The work of Rigo et al. performed such an analysis for the energy profile [43]. In this analysis, the battery SoC, the power generated by the solar arrays and the power required by the satellite are plotted against the scheduling horizon. In this way, the relation between these can be visualised. The total energy used, $\int_0^T \dot{E}^-$, and total spilled energy, $H_{E,tot}$, which are calculated following the steps shown in section 7.6, can also be used to gain more insights on the solution.

The amount of "Equivalent Battery Cycles (EBC)" can potentially be used as a performance metric. The EBC represents a way to quantify battery cycling when DoD constraints limit the full usage of a battery's capacity. Instead of counting the number of full cycles (from 0% to 100% SoC and back), EBC counts how many times the battery is cycled through its allowable DoD range. For instance, if the maximum DoD is set to 10%, the battery is only allowed to cycle between 90% and 100% SoC. In practice, this metric is calculated as follows:

- The battery SoC is determined for the whole scheduling horizon;
- All consecutive local maximum and minimum (peaks and valleys) are determined, based on a desired accuracy level;
- The values of the SoC at the subsequent peaks and valleys is stored in an array;
- The difference between each successive peak and valley is calculated and added together, as follows:
  - When going from a valley to a peak, the absolute value of the difference is added to the cumulative charge, Δ_{tot}^+;
  - When going from a peak to a valley, the absolute value of the difference is added to the cumulative discharge, Δ_{tot}^-;
- Based on the maximum allowed DoD, a number of "equivalent battery cycles" can be determined, as follows:

$$nr.\ cycles = \frac{\Delta_{tot}^+ + \Delta_{tot}^-}{DoD_{max}} \tag{8.6}$$

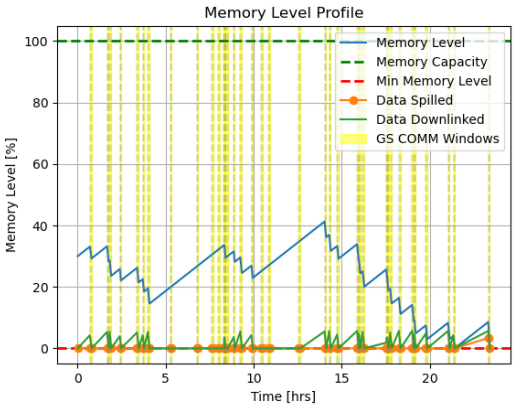
This provides a measure of how many times the battery has been utilised in terms of its allowable DoD, offering a practical assessment of battery wear and performance in constrained cycling scenarios.

Quantitative metrics related to the energy dynamics that can be used to determine the quality of a scheduling solution have thus been identified.

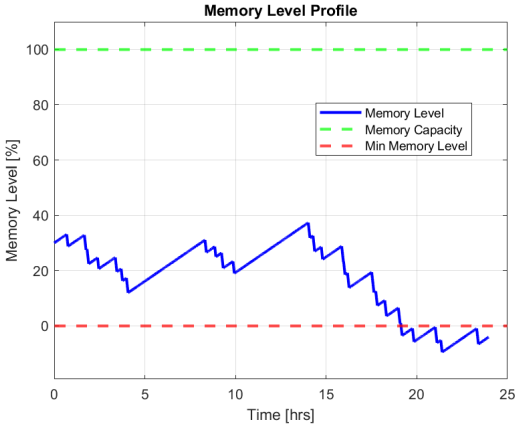
Data Dynamics Metrics

When considering the downlinking of data, the following two metrics can be considered:

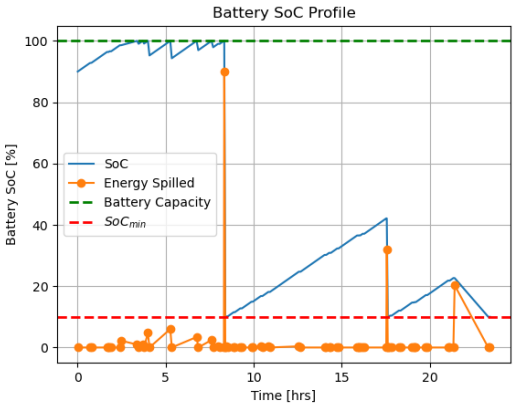
- *Maximum downlink interval*: the longest duration between two consecutive data downlinks;
- *Communication windows availability*: how many communication windows there are over the scheduling horizon;



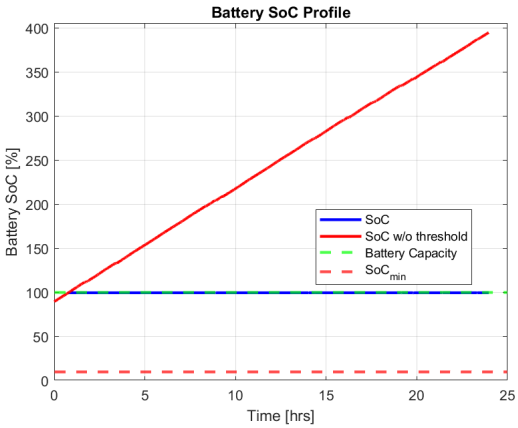
(a) Discrete-time data downlink solution, scheme A



(b) Continuous-time data downlink solution, scheme A



(c) Discrete-time energy profile, scheme A



(d) Continuous-time energy profile, scheme A

Figure 8.3: Resulting profiles for nominal run with discretisation scheme A

Note that the communication windows availability only depends on the satellite initial state and the GSN characteristics, and is thus not a very good metric to evaluate the scheduling solution, but rather to choose a certain GSN over another. The maximum downlink interval, instead, depends on the outputted schedule. This quantity should not be too big, as it is desirable to have information about the spacecraft downlinked regularly. Additionally, the uncertainty linked with the state of the objects that are detected in SURVEILLANCE mode, and tracked in TASKED TRACKING mode, increases with time. For this reason, it is desirable that the data collected is downlinked to a ground station as soon as possible. This metric can also be referred to as *data timeliness*.

8.4. SURVEILLANCE - COMMUNICATION

In this section, the results of the SURVEILLANCE-COMMUNICATION framework with the nominal inputs are presented. The simulation was run for the best illumination case, which is the rightmost point shown in Figure 8.1. The results are shown in Figure 8.3, which shows the data and energy profile outputted by the discrete-time scheduler, and the continuous-time simulator. Additionally, Figure 8.4 shows an overview plot that is obtained by the continuous-time simulator. Note that only the results using discretisation scheme A are shown, as the solution obtained using scheme B is almost identical. A discussion on the difference between the solutions using the two discretisation schemes for the various frameworks is made in section 8.8.

Let's now analyse the plots that will be used throughout the chapter to gain insights on the behaviour of the satellite throughout the planning horizon. Figure 8.3 shows four plots. The top two plots show

the level of data stored on the onboard memory storage over time. The memory level is indicated as a percentage of the total available onboard memory storage. The two plots on the bottom depict the battery SoC as a function of time. The two plots on the left are the result of the discrete-time scheduler, while the two plots on the right are the result of the continuous-time simulator. Analysing the plots one by one allows to gain insights on the simulation results.

8.4.1. Discrete-Time Onboard Memory Profile Plot

Figure 8.3a shows the onboard memory profile throughout the scheduling horizon. The blue line indicates the onboard memory level, indicated as a percentage of the memory capacity. The green line indicates the downlinked data, with the spikes representing how much data is downlinked per interval. The dashed green line indicates the memory capacity, and the red dashed line represents the minimum acceptable memory, which is set equal to 0% in this case. The scatter plot in orange shows the amount of spilled data throughout the simulation, with each point corresponding to an interval resulting from the discretisation scheme used. The yellow vertical lines represent the available communication windows. These lines can be used to determine which communication passes are used to downlink data, and which ones are skipped to continue collecting data.

8.4.2. Continuous-Time Onboard Memory Profile Plot

Figure 8.3b shows the onboard memory level throughout the planning horizon, as outputted by the continuous-time simulator. Similarly to the discrete-time case, the blue line represents the onboard memory level, as percentage of the capacity, and the green and red dashed lines represent the maximum and minimum memory levels, respectively.

Data Collection Term Overestimation

By comparing this figure with Figure 8.3a, some comments on this framework can be made. The two plots have the same shape, which makes sense as they result from the same data downlink schedule. The difference lies in the data profile in the continuous-time going below the minimum allowed memory level of 0%. This is due to the mismodelling of the data collection term in the modelling approach used for this framework, introduced in subsection 5.2.3. When updating the data level at every interval, the framework assumes that the satellite collects data continuously throughout the interval, multiplying the constant data acquisition rate, $\dot{D}^+$, by the interval duration, Δt_i. Then, using this information, the amount of data to be downlinked, and the intervals in which to do so, are scheduled. This is equivalent to assuming that the satellite continues to collect data while it is in COMMUNICATION mode, which in practice does not happen. The practical impacts of this is that the downlink solution is unfeasible, since at the 20^{th} hour of the scheduling horizon, the satellite would be attempting to downlink data that it has not collected.

To overcome this issue, a lower data collection rate than the actual one could be used when setting-up the discrete-time problem, such that in the continuous-time this over-estimation might be less severe. The issue with this approach is that the amount of data downlinked can only be determined after the optimisation is performed, and is thus hard to predict what reducing factor to apply to the $\dot{D}^+$ term. The proposed novel framework which includes IDLE mode is proposed as a solution to tackle this problem, as explained in subsection 8.8.3.

8.4.3. Discrete-Time Battery SoC Plot

Figure 8.3c shows the battery SoC profile for both discrete-time and continuous-time energy profiles under scheme A. In the discrete-time energy profile (left graph), the blue line represents the battery SoC over the planning horizon, while the orange scatter line indicates energy spilled. The green dashed line represents the battery capacity, and the red dashed line indicates the minimum allowed SoC, SoC_{min}, derived from the DoD_{max} constraint. The MILP algorithm finds a solution where the battery SoC at the end of the simulation is exactly at the minimum allowed value. This behaviour is a numerical artefact of the algorithm ensuring the optimal exact solution, rather than reflecting the physical reality where the battery might not always end at the minimum threshold. Due to this artefact, it can happen that energy

is spilled at discrete moments throughout the scheduling horizon, such that the battery SoC drops to exactly the SoC_{min} value.

Comparing this with the continuous-time profile, Figure 8.3d, where the SoC line shows a smoother trend without the abrupt drops seen in the discrete-time graph, it becomes clear that the ”energy spilled” in the discrete-time formulation represents energy directly powering subsystems or discarded when the battery is full. The continuous-time profile avoids these jumps, providing a more realistic depiction of the battery’s behaviour over time.

8.4.4. Continuous-Time Battery SoC Plot

Figure 8.3d shows the battery SoC profile resulting from the continuous-time propagation of the resource and attitude dynamics. In this plot, the green and red dashed line represent the maximum and minimum allowed SoC values, respectively. The blue line, which is horizontal and overlaps with the battery capacity, is the actual battery SoC, bounded by the battery capacity. The red line, which grows linearly over the simulation time, is the level that the SoC would reach if there would be no battery capacity. This additional line is plotted to give an idea of the energy that is being accumulated over time by the satellite. If this line would not be plotted, the SoC profile for scenarios with advantageous illumination conditions would result in a horizontal line, constantly at the maximum level.

8.4.5. Continuous-Time Overview Plot

Figure 8.4 presents an overview of the resource dynamics and operational schedule throughout the scheduling horizon. At the top, the energy dynamics is presented. There are two y-axes, the one of the left side dedicated to the power generation and consumption, and the one on the right indicating the battery SoC in percentage. The cyan line indicates the battery SoC, and is at 100% for the full duration of the simulation. The blue line represents the power generated by the solar panels, the $\dot{E}^{+}$ term. It can be seen that this is smooth when the satellite is in SURVEILLANCE mode, while it is discontinuous when the satellite is in COMMUNICATION mode. This is due to the satellite performing manoeuvres to point the antenna towards the ground stations. It can be seen how the solar panel power presents little variation across the scheduling horizon, due to the absence of eclipse periods. The red line indicates the power required by the payload. This value corresponds to the $\dot{E}^{-}$ term associated with each operational mode considered. For this reason, this term can only take two discrete values in this framework, corresponding to SURVEILLANCE and COMMUNICATION modes. The area in between the solar panel power and the payload power is shaded in green, indicating that the difference between the energy collected and the energy consumed can be used to charge the battery. If the solar panel power would be lower than the payload power, such as shown in Figure 8.5, the area between the lines would be shaded in red, indicating that not enough energy is being collected by the solar power, and that energy is required from the battery to operate the payloads. This corresponds to the battery being discharged.

Below this plot, the onboard memory profile is presented, which is identical to that shown in Figure 8.3b, together with a color bar to indicate the schedule of the different operational modes, which in this case are COMMUNICATION and SURVEILLANCE.

This plot allows to have a comprehensive view of the dynamics of the onboard resource, together with the operational schedule.

Please note that the spikes observed in the solar panel power generation plot are caused by the use of the same set of gains for the PD controller in both COMMUNICATION mode and TASKED TRACKING mode. These gains were specifically tuned for effective TASKED TRACKING, causing oscillations during antenna pointing manoeuvres towards ground stations in COMMUNICATION mode, which in turn results in the observed power generation spikes. To address this issue, distinct sets of gains should be employed for each mode. This recommendation is further elaborated in section 9.3, which discusses suggestions for future improvements.

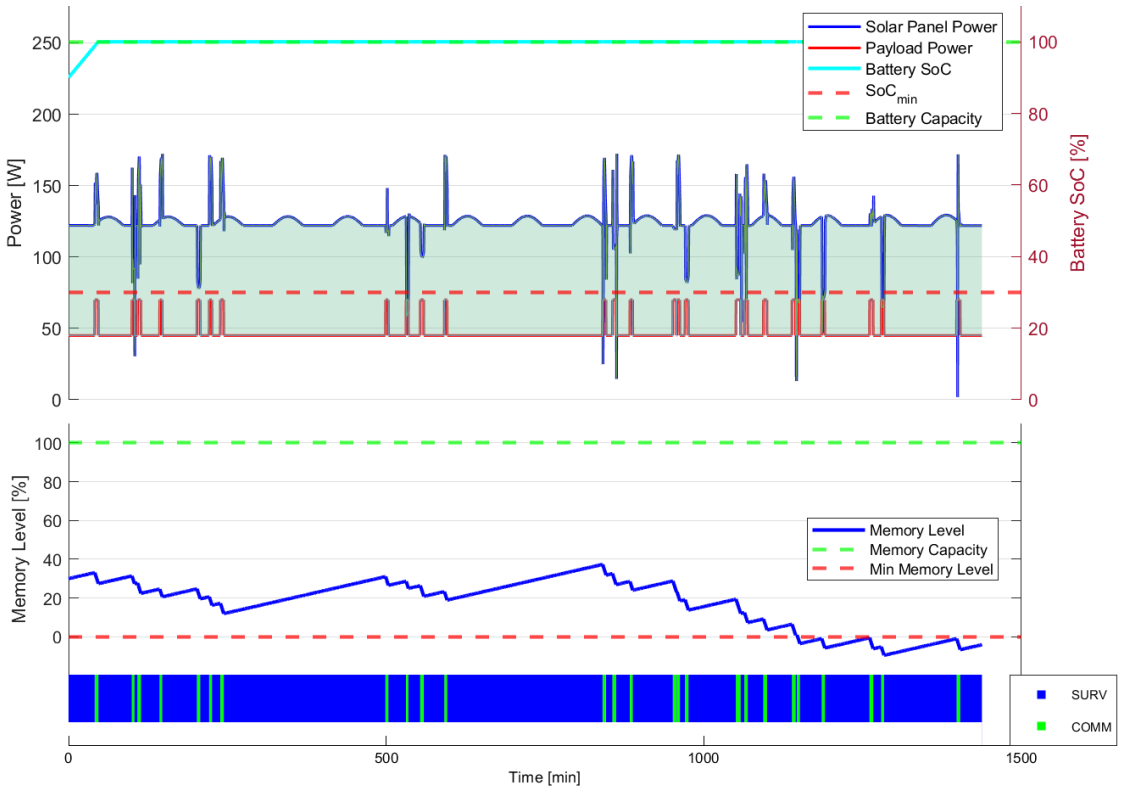


Figure 8.4: Overview of the continuous-time resource dynamics across the full scheduling horizon. Results for the nominal run of the SURVEILLANCE-COMMUNICATION framework.

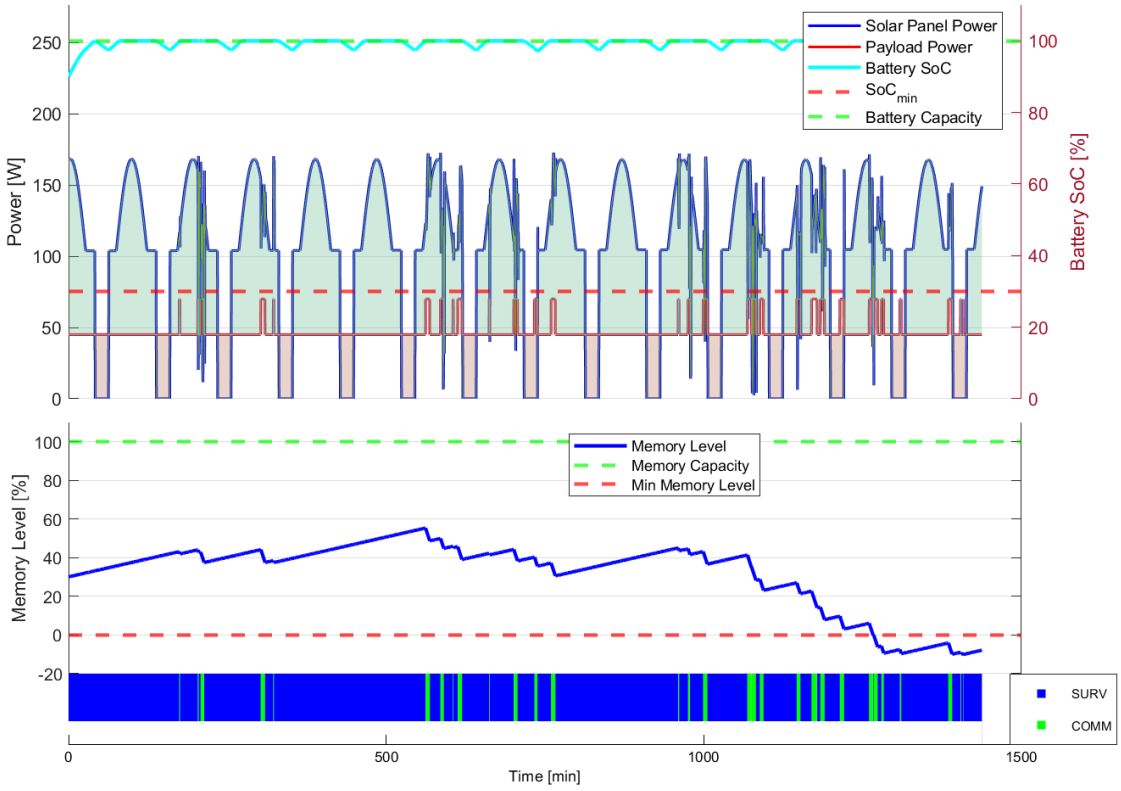


Figure 8.5: Continuous-time resource dynamics across the full scheduling horizon. Results for the SURVEILLANCE-COMMUNICATION framework on the 03-08 (3^{rd} longest eclipse).

Table 8.6: Metrics for the nominal run of the SURVEILLANCE-COMMUNICATION framework.

Metric	Value	Unit	Metric	Value	Unit
EBCs	0.16	-	*Max Comm Interval*	5.24	hrs
Energy Produced	2953	Wh	*Data Downlinked*	205	GB
Energy Consumed	1135	Wh	*Data Produced*	146.4	GB
MILP Run Time	0.008	s	*Nr Intervals*	95	-
Avg Nr Options	1.6	-	*MILP Opt Gap*	0	%
Comm Win Avail	47	-	*Comm Win Used*	28	-
t_{SURV}	90.5	%	t_{COMM}	9.5	%
E_{SURV}	96	%	E_{COMM}	14	%

8.4.6. Nominal Run

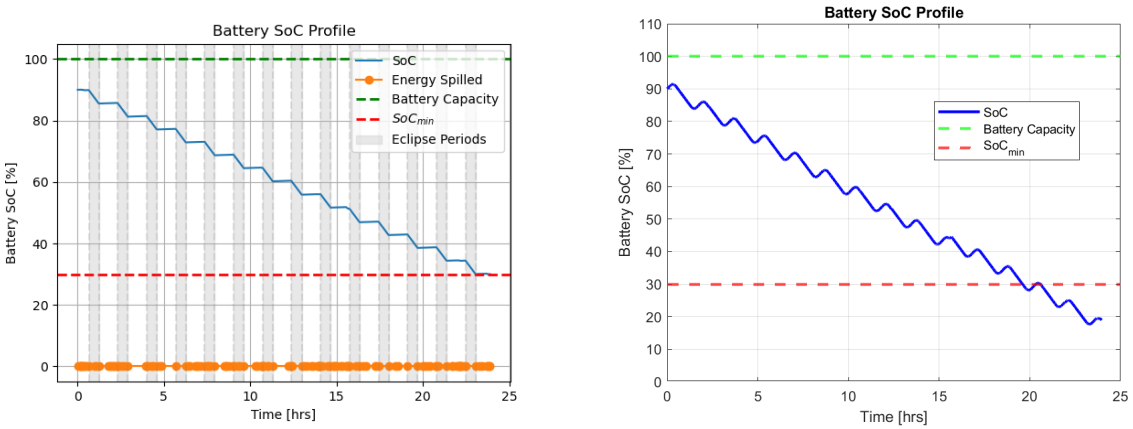
The values of the metrics identified in the previous section for the nominal run of the SC framework are presented in Table 8.6. In this scenario, the satellite was able to downlink 100% of the data collected, spending 90.5% of the time in SURVEILLANCE, and 9.5% of the time in COMMUNICATION. As a result of the battery SoC staying close to the battery capacity at all times, the EBC is equal to 0.16, which means that the battery is barely discharged in this scenario. The maximum downlink interval of more than 5 hours is not desirable, as it means not receiving information about the state of the satellite for many orbits. In the following sections, which parameters can be changed to achieve a shorter maximum interval downlink is investigated.

8.4.7. Sensitivity Analysis

This section discusses the results of the sensitivity analysis performed on the SC scheduling framework. The analysis was performed by changing one parameter at a time, keeping the others equal to their nominal values. In this way, the effect of each individual parameter on the scheduling solution is investigated. A downside of this approach is that possible coupling effects resulting from changing multiple parameters at the same time might be overlooked. Note that a sensitivity analysis has been performed for all the input parameters presented in section 8.2. Quantitative results are shown for parameters that have a large effect on the result, while a qualitative discussion is made for less important parameters.

It is worth noting that, since the MILP algorithm used to solve the optimisation problem is an exact method, only feasible solutions are returned by the discrete-time scheduler. This is because the algorithm is either able to find a closed solution that satisfies all constraints, or it does not return a solution due to infeasibility. Partially feasible solutions are thus not returned by the discrete-time scheduler. This means that, for the cases in which changing a parameter leads to infeasibility, it is not possible to produce quantitative results to show this. To tackle this, the same sensitivity analysis, with the same values for each parameter, was performed for the framework that additionally considers IDLE mode. In this way, whenever the SC framework does not achieve feasibility for a certain set of parameters, the result of the novel ISC framework can be used to gain insights on the cause of the infeasibility. Additionally, when feasibility could not be achieved due to the battery SoC going below the threshold, the simulation was attempted for a shorter scheduling horizon. In this way, the algorithm outputs a feasible solution, and it is possible to see the reason behind infeasibility for longer scheduling horizons.

In the following sections, the most significant results derived from varying the input parameters are presented.



(a) Discrete-time SoC profile

(b) Continuous-time SoC profile

Figure 8.6: Difference in SoC profile between discrete-time scheduler and continuous-time simulator, using discretisation scheme A.

Eclipse Seasonal Variation

To investigate the sensitivity of the framework to the variation in eclipse duration, the optimisation was repeated for the three other illumination conditions shown in Figure 8.1. Figure 8.5 shows the overview resource dynamics plot for the illumination condition on the 3^{rd} of August, which is the 3^{rd} longest eclipse with a duration of approximately 22 minutes per orbit. Figure Figure 8.6 instead shows the battery SoC profile in the discrete-time scheduler and the continuous-time simulator during the worst illumination condition, with more than 35 minutes of eclipses per orbit.

The summarised results are as follows:

- The solution of the discrete-time scheduler gradually spills less energy as the illumination condition worsens;
- At the same time, the SoC in the continuous-time simulator stops being always at 100%, which means that power starts to be required by the batteries to power the payload. This can be seen in the top plot of Figure 8.5, where the area between the solar panel power and the payload power is shaded in red, indicating battery discharge. As a direct result of this, the EBC number increases with longer eclipse duration;
- There are not substantial changes between the 2^{nd} and 3^{rd} longest eclipse;
- When testing the worst-case illumination condition, which is for the longest eclipse duration, more interesting results are seen:
  - The UCF solution outputs a feasible solution, in which the battery SoC gradually decreases, until it reaches just above the SoC_{min} level at the end of the scheduling horizon;
  - When this is fed to the continuous-time simulator, the resulting schedule is unfeasible, because the SoC goes below the allowed threshold;
  - When running the optimisation with discretisation scheme B, feasibility is not achieved;

In this case, it can be seen how running the continuous-time simulator showed that the scheduling solution outputted with scheme A was unfeasible, and did a better modelling of the $\dot{E}^+$ term. When running the discrete-time scheduler with scheme B, then the optimisation is unfeasible. Despite the goal of achieving an optimal scheduling solution not being attained, it is clear that running the continuous-time simulator aids the discrete-time scheduler in assessing whether a given scenario can result in feasible schedules with the available resources. Additionally, by comparing the two figures shown in Figure 8.6, it becomes apparent that the discrete-time scheduler is over-estimating the power generated by the solar panels, and therefore the $\dot{E}^+$ term.

By investigating the effect of the seasonal variations on the scheduling solution, it is shown how periods with eclipses are more interesting from an operational point of view, as the satellite is designed to be

able to survive in the worst-case situations. It makes sense that the results shown in optimal illumination conditions are more trivial and easy to solve, as the EPS is oversized for these cases. Following this reasoning, in the next sections, the worst-case illumination condition is considered unless specified.

Solar Array Sizing

Before investigating the sensitivity to other parameters, it is worth noting that, in order to be able to show the impact of the eclipse period variations, the effect of changing the solar panels size had to be investigated. This was performed by following these steps:

- Running the simulation with the nominal parameters, with a certain initial solar panel size;
- If the solution is unfeasible, increasing the solar panel size;
- If the solution is feasible, but the solution is spilling a lot of energy, decreasing the solar panel size;
- Repeating until a desired behaviour is obtained, which allows to assess the impact of the eclipse variations.

Note that the intent of this test case is to demonstrate an unfeasible solution in the continuous-time simulator for the longest eclipse case, shown in Figure 8.6b, which can be addressed by the inclusion of IDLE mode, as discussed in subsection 8.8.3.

This analysis revealed the sensitivity of the solution to the size of the solar arrays. As a general trend, it is noticed that either the satellite has a combination of solar panel size and illumination condition resulting in the SoC staying close to 100% at all times, or the framework is not able to output a feasible solution. This result is expected, as in this framework the satellite does not have the possibility to stop operating its payload, but only to decide when to downlink data. For this reason, either there is enough energy produced by the solar panels to constantly power the payload, or the framework is not able to output more advanced solutions. This is because the least energy-intensive solution that the MILP can output is to be collecting data at all times. The necessary conditions to achieve feasible solutions for the SC framework is thus $\int_0^T \dot{E}^+ dt \geq \int_0^T \dot{E}^-_{SURV} dt$.

Since the solar panels were sized such that the solution in the continuous-time simulator was unfeasible, in the following runs their size is increased from 0.2 to 0.3 m^2 on each side, to help the optimisation achieve feasibility.

GSN

The results of the sensitivity analysis performed for the three different GSN providers is shown in Table 8.7. From this analysis, it is clear that the framework is highly sensitive to the GSN provider. For GSN provider A, the scheduling solution using scheme A fails, while that using scheme B results in a feasible solution, but results in a data downlink of half that for GSN provider C. For GSN provider B, both discretisation schemes lead to unfeasible solutions. The best solutions are obtained for provider C, which results in a data downlink amount of 205 GB. The EBC value is similar for all feasible solutions, at around 6. The maximum downlink interval for all feasible solutions is above 6 hours, which is not desirable.

Observer Orbit Altitude

The data presented in Table 8.8 indicates that the scheduling framework shows sensitivity to variations in the observer satellite altitude. This result suggests that while the framework can produce feasible solutions across the LEO regime, there are notable variations, especially in the maximum data downlink interval. An analysis of the data shows that lower altitudes tend to result in a greater number of communication windows being utilised, thereby reducing the maximum downlink interval while maintaining a high volume of data downlinked. This outcome is advantageous because shorter downlink intervals enhance the operational efficiency of satellite missions.

Regarding the discretisation schemes, scheme A generally yields better solutions when feasible, characterised by the use of more communication passes, which results in increased data downlinked and

Table 8.7: Performance metrics when varying the GSN.

	GSN Providers				
Metric	**A**	**B**	**C**	**C**	**Unit**
Scheme A?	Infeasible	Infeasible	Feasible	-	-
Scheme B?	Feasible	Infeasible	-	Feasible	-
Comm Win Avail	59	-	35	35	-
Comm Win Used	20	-	23	12	-
D_{down}	103	-	205	205.5	GB
EBC	6.12	-	6.12	5.97	-
$\Delta t_{downlink,max}$	7.8	-	9.02	6.94	hrs

Table 8.8: Performance metrics when varying the observer satellite altitude.

	Observer Satellite Altitude						
Metric	**400 km**	**500 km**	**600 km**	**700 km**	**800 km**	**900 km**	**Unit**
Scheme A?	Feasible	Feasible	Feasible	Feasible	Unfeasible	Unfeasible	-
Scheme B?	Infeasible	Feasible	Feasible	Feasible	Feasible	Feasible	-
Comm Win Avail	35	38	49	56	54	62	-
Comm Win Used	33	31	38	36	17	25	-
D_{down}	188	206	204	204	92	73	GB
EBC	6.09	6.17	6.14	5.97	6.04	6	-
$\Delta t_{downlink,max}$	1.88	3.2	4.09	4.88	4.18	6.96	hrs

shorter maximum downlink intervals. However, scheme A occasionally fails to find a solution at higher altitudes, such as 800 km and 900 km, where scheme B successfully identifies feasible solutions, albeit with fewer communication windows and slightly longer maximum downlink intervals.

Overall, the scheduling framework demonstrates a degree of sensitivity to altitude variations within the LEO regime. While it can accommodate various altitudes, the best solutions are typically achieved at lower altitudes, and there is room for improvement in handling higher altitudes more effectively.

Maximum Depth-of-Discharge

Within the SURVEILLANCE-COMMUNICATION framework, the maximum allowed DoD primarily influences the feasibility of the solution, while the other metrics remain unchanged. This occurs because, as long as the battery SoC remains above the specified threshold, the algorithm continues to generate the same solution. The scenario remains unchanged except for the DoD_{max}, so the optimal solution remains consistent. However, when the threshold is increased to a level where the satellite lacks sufficient resources to operate its instruments while adhering to the battery constraint, the framework cannot generate a scheduling solution.

This issue can be addressed by considering the IDLE mode as the default mode, as further discussed in section 8.6.

Scheduling Horizon

Varying the scheduling horizon produced unexpected results on the feasibility of the solution. For scheduling horizons longer than 2 days, the framework with the nominal parameters is unable to find feasible solutions. The continuous-time solution for 48 hours is shown in Figure F.3. The battery SoC plot shows that this solution is stable, with the SoC varying from 100% to 95% at each EBC. The data plot shows quite large data intervals, up to 12 hours, which is not a desirable result, but does not present

Table 8.9: Test cases for sensitivity analysis of payload (PL) characteristics.

	Payload				
Term	**PL 1**	**PL 2**	**PL 3**	**PL 4**	**Unit**
$\dot{E}^{-}_{SURV}$	35	45	55	65	*W*
$\dot{D}^{+}_{SURV}$	10	15	20	25	*Mbps*
$\dot{D}^{-}_{COMM}$	200	200	200	200	*Mbps*
$\dot{E}^{-}_{COMM}$	70	70	70	70	*W*
$\dot{E}*^{-}_{COMM}$	0.18	0.13	0.08	0.03	*J/Mb*

clear reasons on why the solution should be infeasible. This analysis lead to think that the reason for the infeasibility is not due to the scheduling horizon itself, but might be due to a coupling effect of the altitude of the observer, which was previously shown to have a large effect on the feasibility of the solution for higher altitudes in the LEO regime. Following this reasoning, the simulation was repeated for an altitude of 500 km. In this case, the framework is able to find feasible solutions up to 60 days. This analysis confirms the sensitivity of the framework to the observer altitude, while it shows that, for a certain observer altitude range, it is able to generate feasible solutions for up to a few weeks.

The reason for the instability of this framework to the altitude is further investigated in section 8.8.

Payload

To investigate the effect of equipping the satellite with different payloads, the $\dot{E}^{+}_{SURV}$ and $\dot{E}^{-}_{SURV}$ terms were varied. The cases considered for this analysis are shown in Table 8.9. Given the significant variation in power required to operate each payload, each case was evaluated with two solar panel sizes, 0.3 m^2 and 1.0 m^2. This approach was employed to avoid infeasible solutions due to energy constraints and to gain insights into the effects of varying payload characteristics.

For the nominal case of the payload (PL 2 in Table 8.9), a solar panel size of 0.3 m^2 was selected. Altering the payload maintaining a panel size of 0.3 m^2 significantly impacts the battery SoC, which progressively decreases for payloads consuming more power, as expected. This is evident when comparing figures F.4 and F.5, which illustrate the continuous-time resource dynamics for PL 1 and PL 3, respectively. For PL 1, all collected data is downlinked within the scheduling framework. When using PL 3, the higher data acquisition rate leads to more data collection, and the algorithm schedules a solution with more data downlinked compared to the first case, but the memory is not fully emptied. The downlinking strategies are similar, with the solution waiting for the memory to fill up before performing multiple subsequent downlinks, downlinking many GB of data at once. Additionally, the battery SoC plot for Figure F.5 demonstrates how the DoD_{max} constraint will be violated over longer scheduling horizons due to excessive energy consumption in the default mode.

Increasing the solar panel size to 1.0 m^2 results in different data downlink schedules with respect to the previous results. The continuous-time resource dynamics profile resulting from the same payloads investigated for the smaller solar panel case, PL 1 and PL 3, can be seen in Figure F.6 and F.7, respectively. The battery SoC oscillates about 100% in both cases, but using PL 3 results in larger oscillations, a result of the higher power consumption. Additionally, the data downlink profile of both solutions is different from those of the solutions for smaller solar panels. This downlinking solutions result in a much more frequent downlink, reducing the maximum downlink interval, and benefitting data timeliness.

The main takeaway is that with this framework, good solutions in terms of data timeliness are achieved if the system is not tightly constrained by power. This is evident when comparing cases where the same payload is used with two different solar panel sizes, and the battery SoC is not a significant constraint. In such scenarios, the algorithm produces better solutions when more power is available.

Table 8.10: Performance metrics when varying the optical payload.

	Payload				
	1	3	1	3	
	Solar Panel Size				
Metric	0.3 m^2	0.3 m^2	1.0 m^2	1.0 m^2	**Unit**
Scheme A?	Infeasible	Infeasible	Infeasible	Infeasible	-
Scheme B?	Feasible	Feasible	Feasible	Feasible	-
Comm Win Avail	54	54	54	54	-
Comm Win Used	22	28	38	44	-
$D_{collected}$	100	197	100	190	GB
D_{down}	152	185	151	258	GB
EBC	2.25	2.78	2.11	3.25	-
$\Delta t_{downlink,max}$	**10.34**	**6.98**	**1.75**	**1.62**	hrs
t_{SURV}	93	91.4	93	85.3	%
t_{COMM}	13	8.6	13	14.7	%

Default Pointing Direction

Since target objects are not simulated in the frameworks that include SURVEILLANCE mode, altering the surveillance pointing direction primarily impacts power generation due to the varying orientation of the solar panels relative to the Sun. This difference is illustrated in two plots shown in Appendix F, namely Figure F.1 and Figure F.2. In the orbit-following scenario, the power profile is smooth, reflecting the satellite's attitude change relative to the LoS with the Sun as it moves along its orbit. Conversely, for the away-from-the-Sun pointing direction, the power generation profile appears almost discrete, with power output rapidly shifting from zero to maximum shortly after the satellite exits the eclipse.

These observations can be summarised as follows:

- If the default pointing direction results in a sub-optimal power generation profile, then the more manoeuvres (for COMM or TT modes) the satellite performs, the higher the resulting SoC will be compared to maintaining the default orientation at all times. This is evident in Figure F.1, where manoeuvres enable the satellite to achieve better illumination conditions.
- Conversely, if the default pointing direction results in optimal power generation, as shown in Figure F.2, then performing manoeuvres will cause the satellite to deviate from this optimal illumination condition, resulting in reduced power collection.

This effect should be considered when scheduling satellite tasks. If a sub-optimal illumination condition is associated with the default pointing direction, then the continuous-time battery SoC resulting from satellite manoeuvres would be higher that what outputted by the solution. Consequently, a tighter DoD_{max} constraint could be imposed. Conversely, if the battery SoC resulting from tracking manoeuvres is lower due to optimal illumination conditions in the default orientation, a more relaxed DoD_{max} constraint should be considered.

Initial Battery SoC

Varying the initial battery SoC reveals the limitations of the SC framework. Different initial SoC values, ranging from 10% to 100% in 10% increments, were tested. The scheduling solutions obtained from this sensitivity analysis are largely similar; thus, only a qualitative discussion is presented. For initial SoC values above 30%, the algorithm successfully finds an exact solution. However, the SoC gradually decreases in all cases, but the scheduling horizon is too short for the SoC to fall below the minimum threshold. When the initial SoC is set to 30%, the algorithm fails to find a feasible solution,

as the minimum SoC constraint is violated. This indicates that the current framework can only generate scheduling solutions when sufficient resources are available to continuously operate at the lowest power consumption, which is for SURVEILLANCE mode in this case. This necessary condition to have feasible solutions with the SC framework can be expressed as follows:

$$\int_0^T \dot{E}^+ dt \geq \int_0^T \dot{E}^-_{SURV} dt \tag{8.7}$$

This limitation is specifically addressed by incorporating the IDLE mode into the framework, as discussed in section 8.6.

Initial Data

The sensitivity of the framework to the initial onboard memory level was investigated by testing progressively increasing values from 10% to 100%. The results across different runs are similar and can be categorised into two distinct groups. For initial data levels ($D_{initial}$) less than or equal to 50% of the memory capacity, the solution waits for the memory to fill up before initiating data downlink. For higher initial data levels, the solution begins downlinking data at the start of the planning horizon and continues to downlink more regularly.

Summary of Sensitivity Analysis

The sensitivity analysis conducted in this study examined the impact of various parameters on the scheduling framework's performance. The analysis included variations in seasonal illumination conditions, solar panel size, scheduling horizon, observer satellite altitude, maximum DoD, payload characteristics, default pointing direction, initial battery SoC and initial onboard memory level.

The findings revealed that the framework is sensitive to the altitude of the observer satellite, with better solutions obtained at lower altitudes in the LEO regime due to more frequent communication windows. The framework's sensitivity to the maximum DoD showed that feasible solutions are only possible when the battery SoC remains above a certain threshold. Varying payload characteristics demonstrated that higher power-consuming payloads necessitate larger solar panels to maintain operational feasibility, and that solutions that are not constrained in power result in better data timeliness. The default pointing direction significantly impacts power generation, influencing the battery SoC based on whether the orientation maximises or minimises solar exposure. Initial battery SoC variations indicated that feasible solutions require sufficiently charged batteries, with constraints becoming critical below 30% SoC. Initial onboard memory levels affected downlink strategies, with higher initial data levels prompting more frequent downlinking. Seasonal variations highlighted that eclipse periods present more operational challenges, with the framework struggling to maintain feasibility under prolonged eclipses. Lastly, the solar panel size directly influenced the feasibility of the solutions, requiring adjustments to ensure continuous power supply during varying illumination conditions.

The analysis of the scheduling horizon revealed that the framework struggled to find feasible solutions for horizons longer than two days with nominal parameters, though a 48-hour horizon showed stable battery SoC and large data intervals. Further investigation suggested that the issue might be linked to the altitude of the observer satellite. At an altitude of 500 km, the framework could find feasible solutions for up to 20 days, indicating a strong sensitivity to observer altitude. This demonstrates that while the framework can handle extended scheduling horizons, its performance is significantly affected by the satellite's altitude.

Overall, the sensitivity analysis underscored the importance of optimising power generation and storage capabilities to maintain the feasibility of satellite operations under diverse mission scenarios and highlighted areas for further improvement, such as the integration of an IDLE mode to enhance stability over extended scheduling horizons.

8.5. Extending the Framework to Additional Operational Modes

Following the successful integration of the discrete-time scheduler and continuous-time simulator for a scheduling framework encompassing SURVEILLANCE and COMMUNICATION modes, and after

Table 8.11: Nominal numerical values for the resource profiles of the different operational modes considered for the scheduling frameworks.

Mode	$\dot{E}^-$	$\dot{D}^+$	$\dot{D}^-_{downlink}$
SURVEILLANCE	45 W	15 Mbps	0
COMMUNICATION	70 W	0	200 Mbps
IDLE	10 W	0	0
TASKED TRACKING	50 W	15 Mbps	0

evaluating the framework's performance across various scenarios as demonstrated in the sensitivity analysis in the previous section, it was decided to explore extending the framework to incorporate additional operational modes rather than further investigating this problem. The rationale behind this decision is that there is extensive literature on scheduling algorithms that only consider observation and communication modes, whereas frameworks that account for multiple additional modes are less studied.

In order for a framework to consider all modes considered in a typical space-based space surveillance mission, all four modes should be included. Ideally, the satellite would be in IDLE mode by default, and the solution would schedule when it goes in SURVEILLANCE, TASKED TRACKING or COMMUNICATION mode.

This was attempted during the research, but the algorithm developed was not able to discern between data acquired while in SURVEILLANCE mode and while in TASKED TRACKING mode, and resulted in a double counting of the data collected. This distinction between type of observation data collected is necessary for the successful implementation of the framework. The reason for this is that collecting surveillance data usually costs less energy than collecting tasked tracking data, as the ADCS consumes more power while operating in the latter mode, and the payload consumes the same power in both modes. Thus, two different weights need to be given to collecting the two types of data, as otherwise the framework would stick to only collecting the data that costs less energy. In order to be able to assign different weight, discerning between the types of data is necessary.

A successful implementation of such a framework was not achieved in this research, and is left as recommendation for future work. Despite this, two different frameworks were implemented, following the steps shown in chapter 6. Both assume an IDLE mode as the default, and schedule when the satellite goes in COMMUNICATION mode, and in *observation* mode. In one framework, the *observation* mode is SURVEILLANCE, and in the other it is TASKED TRACKING. The results of both frameworks are reported in the following sections. The values used for the resource profile of each operational mode considered are listed in Table 8.11.

8.6. IDLE - SURVEILLANCE - COMMUNICATION

In this section, the effects of integrating an IDLE mode into the framework discussed in Spangelo et al. [50] are examined.

8.6.1. Improvements Compared to SC

As previously established, the SC framework exhibits several limitations. The primary constraint is that it can only identify feasible solutions when there is ample energy to continuously operate the payload. Consequently, the framework fails to produce solutions when the DoD_{max} constraint is active. Instances of such scenarios have been demonstrated, including cases with smaller solar panels, more powerful payloads, low initial energy levels, or low DoD_{max} values. The integration of an IDLE mode lowers the energy requirement, as the necessary condition for the ISC framework is the following:

$$\int_0^T \dot{E}^+ dt \geq \int_0^T \dot{E}^-_{IDLE}\, dt \tag{8.8}$$

Table 8.12: Differences between the nominal and the energy-constrained scenario.

Parameter	Nominal	Energy-Constrained	Unit
Scheduling Horizon	24	48	hrs
Battery capacity	600	400	Wh
Solar Panel Size	0.3	0.25	m^2
SoC_{min}	30	70	%
$\dot{E}^{-}_{SURV}$	45	55	W
$\dot{D}^{+}_{SURV}$	15	25	Mbps

which means that, as long as the satellite collects enough energy throughout the planning horizon to operate in IDLE mode, then the framework can be used to schedule scientifically-relevant tasks. This is useful in scenarios in which the energy constraint is active, as is shown in the example below.

Furthermore, the SC framework's sensitivity to observer altitude is significant, rendering it ineffective for scheduling horizons exceeding two days at altitudes above 700 km. This issue is solved by including the IDLE mode in the framework, enabling feasible solutions for the whole LEO regime for several weeks.

Moreover, the SC framework overestimates the data collection, resulting in infeasible data downlink schedules that require post-processing adjustments before transmission to a satellite. This issue is solved by considering the IDLE mode as default, by following the procedure defined in chapter 6.

Finally, the pipeline for the ISC framework can be made iterative, by using a progressively more refined discretisation scheme at each iteration, until the energy constraint is met. In this way, the loop shown in Figure 4.8 can be used to find feasible solutions to a wider range of scenarios.

An example that shows all the benefits aforementioned is presented below, while the implications that modifying the framework have on the computational performance are discussed in section 8.8.

8.6.2. Example: Energy-Constrained Scenario

In this section, an example of how the ISC framework can be used to find scheduling solutions when the SC framework fails to do so is presented, together with a proposed method to iteratively improve the discretisation scheme to find better solutions.

The scenario considered is a deviation from the nominal scenario presented in section 8.4. The differences are listed in Table 8.12. As a reminder, the nominal scenario considers the longest eclipse periods, of approximately 35 minutes.

Discretisation Scheme A Solution

The scheduling solution obtained using discretisation scheme A is shown in Figure 8.7. Looking at the solution, a few comments can be made. First, looking at the memory level profile over the scheduling horizon, it can be seen that the memory level never drops below 0, as in this framework the downlinking schedule is calculated based on the actual data collected, contrary to the SC framework, where continuous data collection is assumed.

Additionally, it can be seen that the framework is schedules periods of IDLE mode during eclipses, in order to try and keep the battery SoC above the threshold. Despite this, the constraint is violated three times, as indicated by the red ellipses in the plot.

Following the pipeline shown in Figure 4.8, the next step is to use discretisation scheme B to try achieve feasibility. In order to create the new discretisation scheme, a value for "*what power step-size to use to discretise the power generation profile*" has to be picked. This value is referred to as Γ. In the following subsection, the effect that choosing progressively smaller values of Γ has on the scheduling solution is discussed.

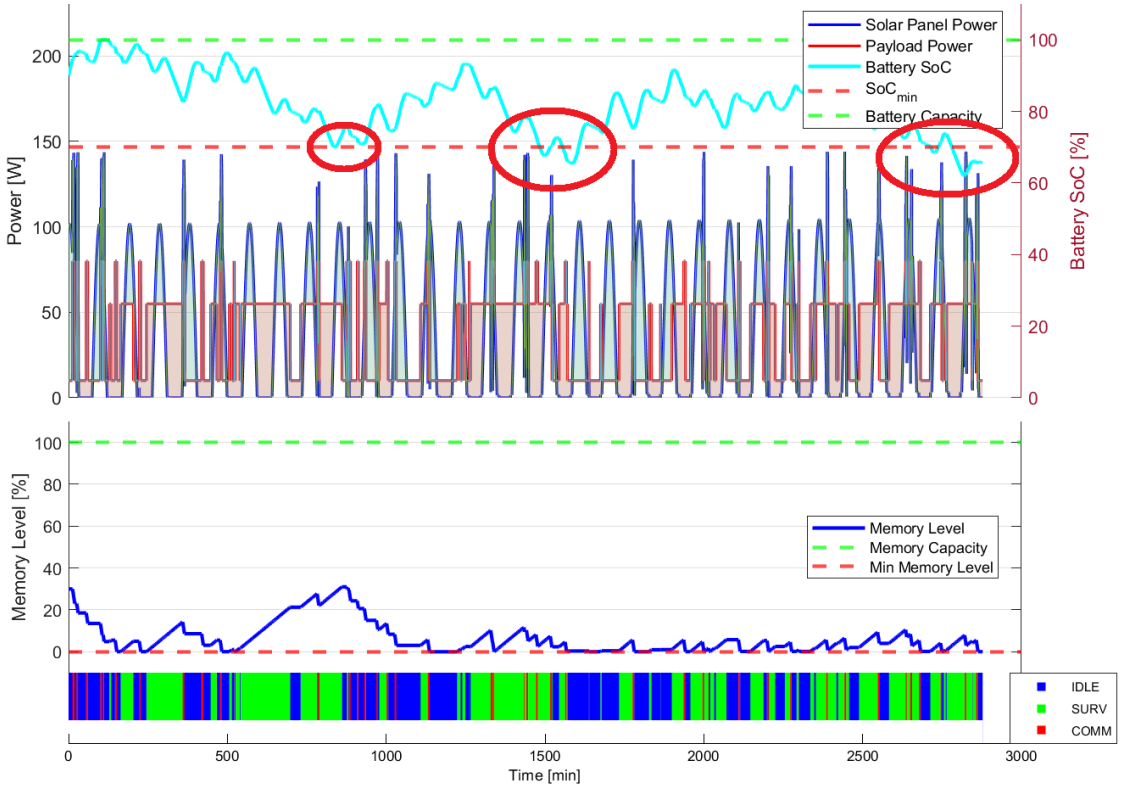


Figure 8.7: Continuous-time overview of solution obtained with discretisation scheme A.

Discretisation Scheme B Solution - Γ Variations

Since the power generation profile resulting from maintaining the default pointing direction oscillates between 100W and 0 W, an initial guess of 40 W is used for Γ. The scheduling solution resulting from using 40 W as the Γ value to discretise the scheduling horizon is shown in Figure 8.8. By looking at the battery SoC, it can be seen that for the first part of the simulation, the solution is feasible, but that the SoC_{min} constraint is violated in the second half of the scheduling horizon.

The next step in the framework's pipeline involves employing a new discretisation scheme based on the energy production profile, utilising a different value for Γ. Given that the solution with $\Gamma = 40$ W was unfeasible, it is logical to use a lower Γ value to discretise the power generation profile into smaller intervals, enabling the framework to make more informed decisions about the timing of scientific activities. Consequently, a value of 20 W was chosen for Γ, resulting in the scheduling solution shown in Figure 8.9.

The battery SoC plot illustrates how the framework now effectively schedules IDLE periods, ensuring the SoC_{min} constraint is met throughout the entire planning horizon. **This demonstrates that the novel ISC framework successfully adapts the operational schedule in energy-constrained scenarios, allowing scientific activities to be performed despite the reduced energy availability.**

It is now worth investigating what effect further decreasing Γ has on the scheduling solution. The most relevant metrics of the scheduling solutions, which are the amount of data downlinked and the percentage of time spent in IDLE mode, have been plotted in Figure 8.10a. To allow for a more comprehensive understanding, the battery SoC for different values of Γ are plotted together in Figure 8.10b. From the figures, it can be seen that further decreasing the value of Γ results in three effects: 1) an increase of the percentage of the planning horizon spent in IDLE mode, 2) a decrease in the data downlinked, 3) An increase in the minimum battery SoC reached. As the value of Γ is decreased past 5 W, the solutions converge to the same values for these metrics.

For excessively high values of Γ, the framework inadequately schedules periods of IDLE mode, leading

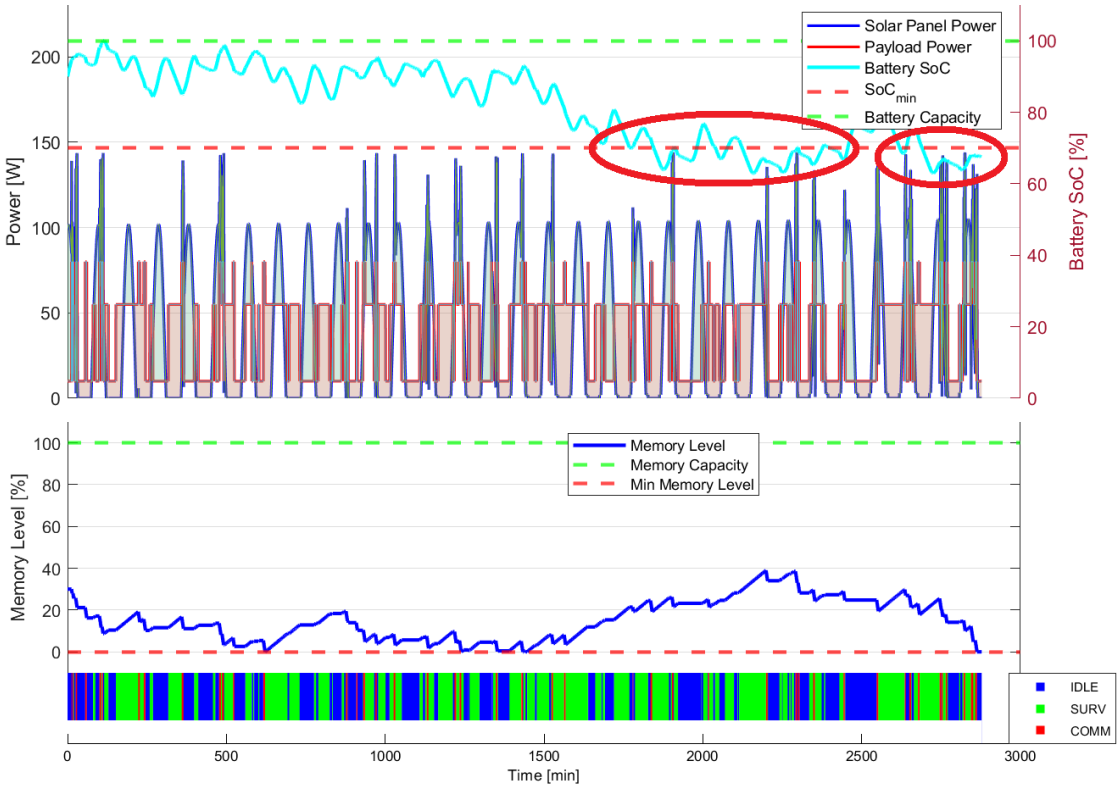


Figure 8.8: Continuous-time overview of solution obtained with discretisation scheme B, Γ = 40 W.

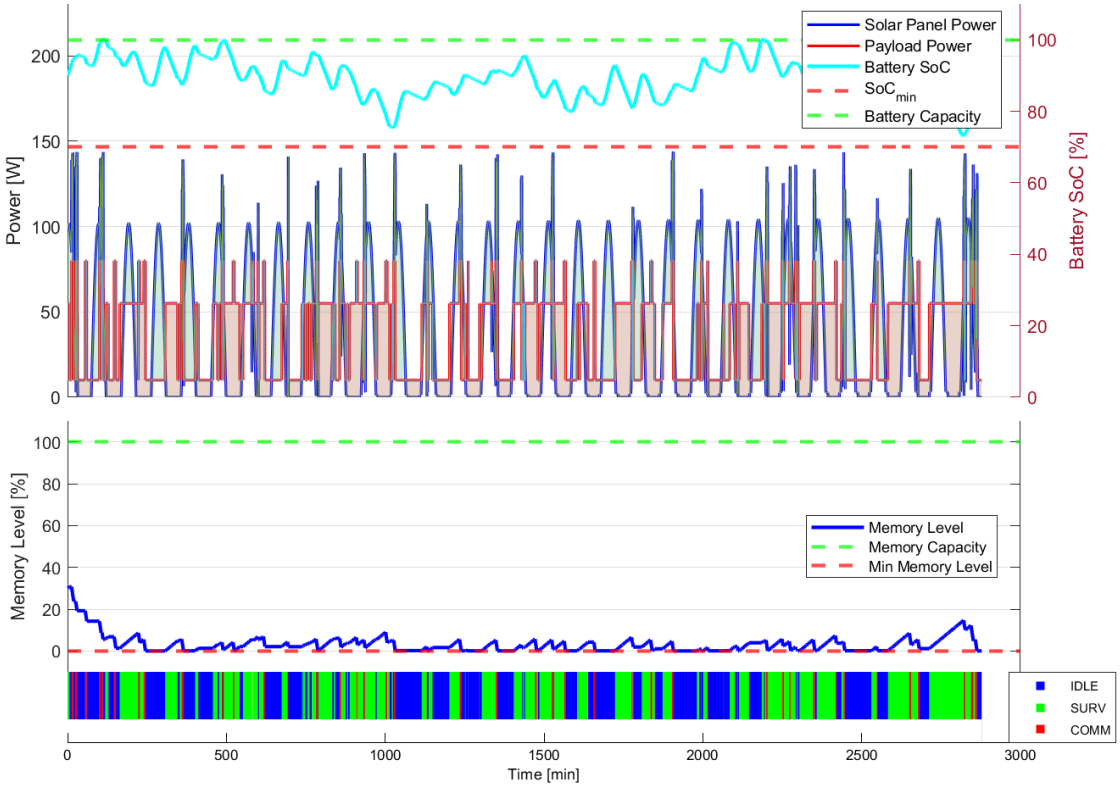


Figure 8.9: Continuous-time overview of solution obtained with discretisation scheme B, Γ = 20 W.

Table 8.13: Distributions of Classical Orbital Elements for the target objects

	LEO		MEO		GEO		
COEs	μ	σ	μ	σ	μ	σ	**Units**
Altitude	800	200	20,000	2,000	35,786	100	km
Eccentricity	0.01	0.005	0.01	0.005	0.0001	0.0001	-
Inclination	98	2	55	10	0	0.1	°
AoP	0	180	0	180	0	180	°
RAAN	0	180	0	180	0	180	°
True Anomaly	0	180	0	180	0	180	°

to an increased amount of data downlinked but resulting in unfeasible schedules due to violations of the SoC_{min} constraint. Conversely, when Γ is too low, the framework becomes overly conservative, significantly reducing the execution of scientific activities and yielding a lower mission profit than what is achievable while still adhering to the SoC_{min} constraint. The figure highlights the existence of an optimal Γ value, where the SoC_{min} constraint is just satisfied, resulting in the maximum possible data downlinked. This suggests the potential for an iterative algorithm that decreases Γ until the SoC_{min} constraint is met, optimising data downlink and mission profit. However, due to time constraints at the end of this thesis project, a detailed exploration and development of such an algorithm were not feasible and are recommended for future research.

The effects that optimising Γ has on the computational performance of the framework are discussed in section 8.8.

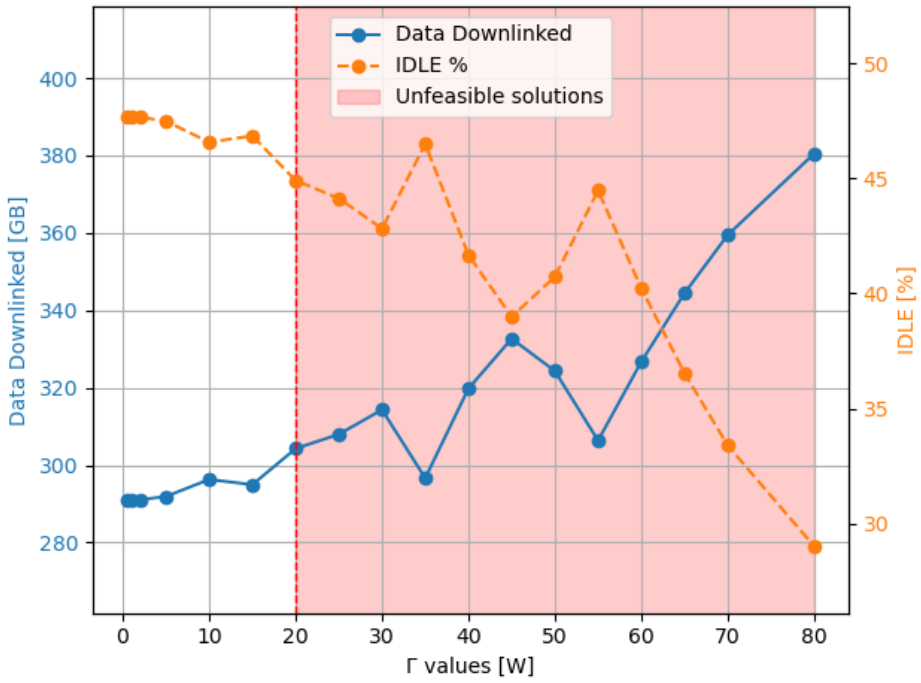
8.7. IDLE - TASKED TRACKING - COMMUNICATION

This section presents the results of the framework that follows from the discussion made in chapter 6, including TASKED TRACKING mode.

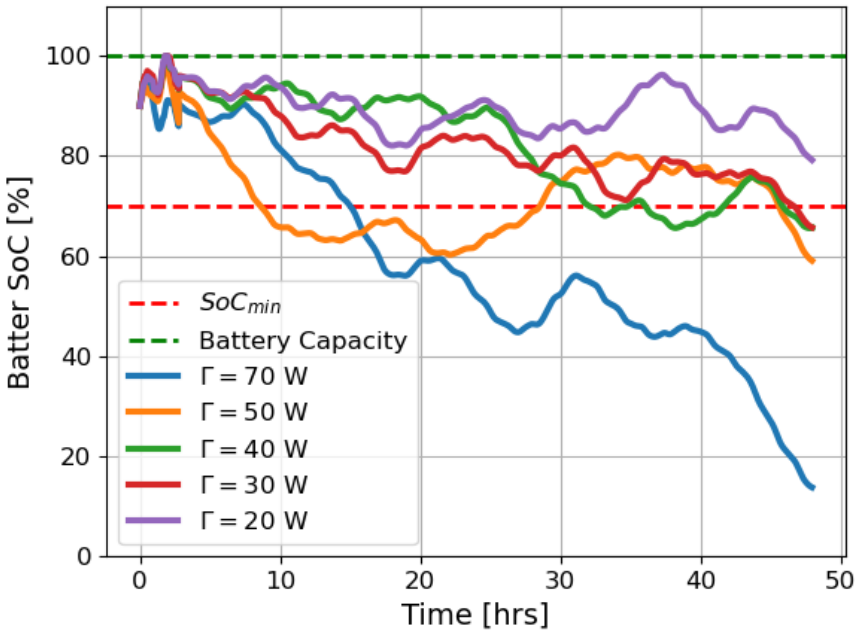
When including TASKED TRACKING mode, additional input parameters are needed, in order to determine the observation windows of each target as seen by the observer satellite. These are the minimum phase angle between the observer, the target and the Sun, the minimum phase angle between the observer, the target and the Earth's horizon and the initial state of the target objects. For these two phase angles, a value of 90° and 3° was used, respectively, based on typical values suggested by Flohrer [18]. For the target objects generation, a normal distribution was assumed for each orbital regime. The standard deviation, σ, and mean, μ, values used for each classical orbital element in each regime are shown in Table 8.13. The accelerations considered to be acting on the targets for the numerical propagation are the same as those acting on the observer satellite, as listed in Table 5.2.

The results of this framework depend on the amount of targets considered in the environment, and on the resulting observation windows. A metric that can be used to judge the result of the framework is the amount of observation tasks in which the satellite is continuously observing the same target for a time longer than a certain threshold. For example, a continuous observation time of 200 seconds could be considered. The reason for this is that, in order to be able to determine the orbit of the target object, a series of pictures has to be taken in succession, resulting in an illumination streak that makes determining the orbit of the target object possible. The information gathered from the orbit determination can then be used to insert or to update the state of the tracked object in a catalogue.

The discretisation of the scheduling horizon for this framework is different with respect to that presented in the other two frameworks, which include SURVEILLANCE mode. This is because, on top of the communication windows, the visibility windows of the target objects are used to split time into intervals. An example of this can be seen in Figure 8.11, which shows the onboard memory profile throughout the scheduling horizon, with the yellow lines indicating communication windows with ground stations, and the red lines indicating observation windows with the target objects. Note that the visibility windows are not very realistic, and there was not enough time to validate them. Thus, the results presented in this section serve to verify the feasibility of the approach, but more work is required to ensure the fidelity of



(a) Each scatter point represents a solution, with the total amount of data downlinked and percentage of time spent in IDLE mode indicated. The solutions to the right of the red dashed line are unfeasible due to the SoC_{min} constraint.



(b) Each line represents the battery SoC profile of a solution. For progressively smaller value of Γ, the SoC shifts upwards, until it satisfies the SoC_{min} constraint for the full scheduling horizon at Γ = 20 W. Note that the SoC profiles have been smoothed using a moving average with a window size of 1000 time-steps to remove high-frequency oscillations and allow for easier distinction of the different lines by inspecting the general trend.

Figure 8.10: Effect of iteratively decreasing the value of Γ when discretising the scheduling horizon into intervals using scheme B.

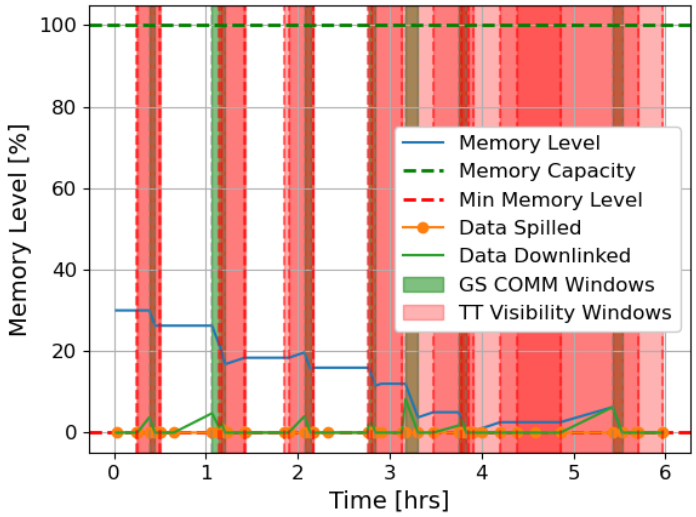


Figure 8.11: Data profile in the discrete-time scheduler, with communication windows and visibility windows indicated. Use-case showed: 3 targets in GEO

the target visibility windows.

Satellite Loaded with Right Amount of Observation Tasks

When considering scenarios in which the satellite is not overloaded with conflicting observation tasks at the same time, the framework is able to output operational schedules that result in continuous observations of the same target. When such schedules are plugged in the continuous-time simulator, the satellite performs tracking manoeuvres, to maintain the target in the FoV of its telescope. This can be seen in Figure 8.13b, where the dots in each line correspond to the location of the target object being tracked as seen by the telescope at consecutive points in time. From this figure, it is clear how such an observation could be used to produce a streak, as the targets stay within the 3° FoV of the camera for multiple consecutive time steps. Thus, for scenarios in which the satellite is not loaded with conflicting observation requests, the framework is able to schedule the observation and communication tasks of the satellite, resulting in continuous illumination streaks, and feasible memory and battery SoC levels, as shown in Figure 8.12. In the figure, it is possible to see how the framework schedules TASKED TRACKING periods when there is enough energy available during visibility windows, and goes into IDLE mode after about 250 minutes, to save energy during eclipses and do not let the SoC drop below the minimum threshold, in this way respecting the DoD_{max} constraint.

As an example of this result, for a scheduling horizon of 24 hours, and a total of 45 targets, 20 in LEO, 5 in MEO and 20 in GEO, the scheduling solution resulted in 56 continuous observations of the same target longer than 200 seconds. 25 targets were observed at least once continuously for a duration above the threshold, corresponding to 56% of the total. Overall, the satellite spent 429 minutes carrying out continuous observations, which corresponds to 29% of the planning horizon. In this scenario, the solution schedules a total of 39% of the total time for TASKED TRACKING mode, which means that 10% of the allocated observation time the satellite observes targets for a shorter duration than the threshold, not leading to a successful detection.

This use-case highlights the limitations of the framework. For scenarios where the satellite is not loaded with overlapping requests, the framework operates as follows: it schedules TASKED TRACKING periods during visibility windows whenever there is sufficient onboard battery SoC, and schedules IDLE periods when there is not. Additionally, it determines which communication windows to utilise for data downlink. In this situation, the framework's strength lies in scheduling TASKED TRACKING periods only when adequate energy is available. However, the framework does not need to decide between targets to observe, as the number of targets is insufficient to cause overlapping visibility windows. The

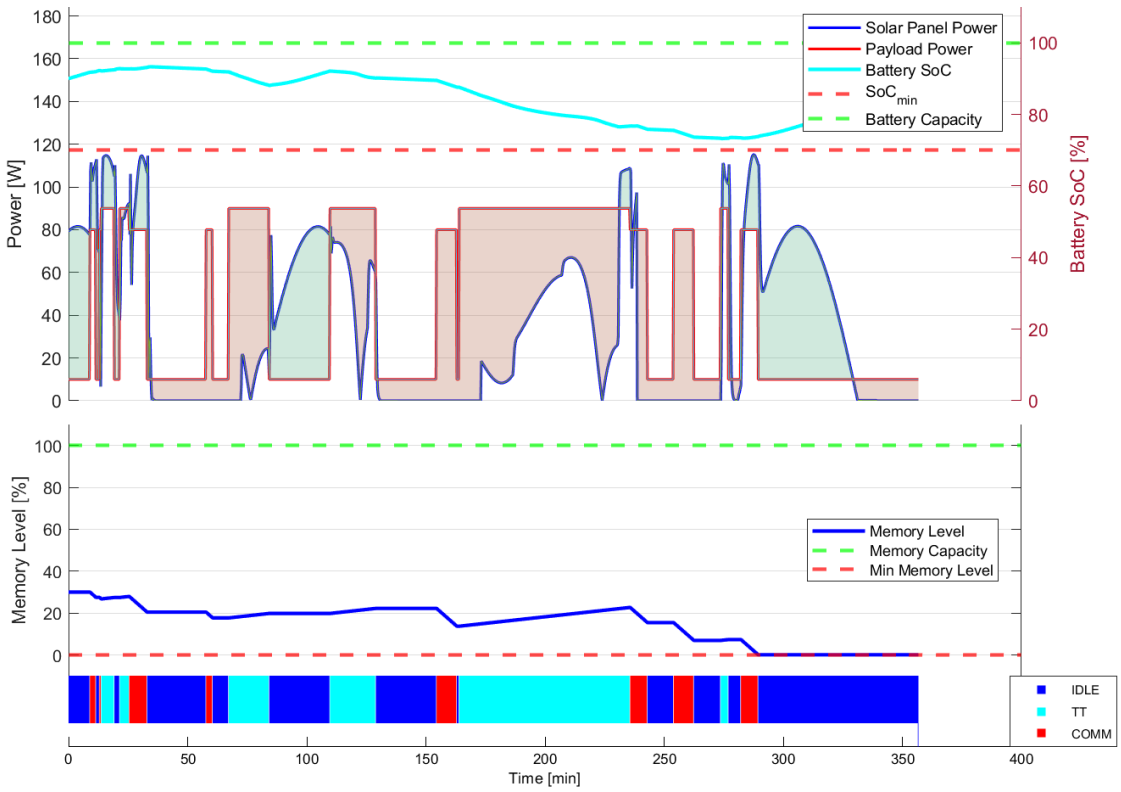


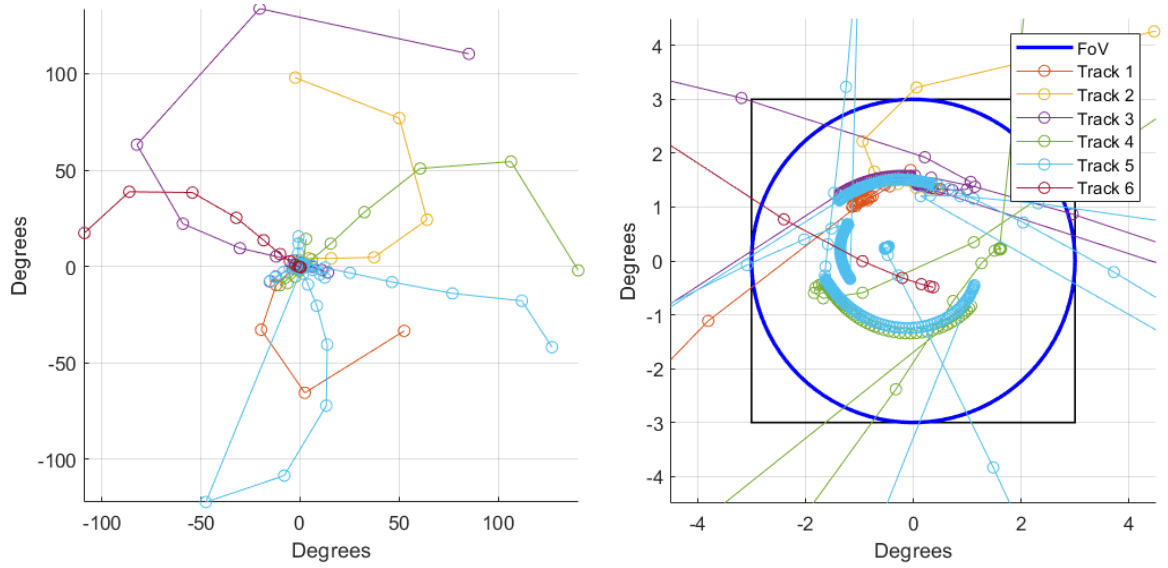
Figure 8.12: Example of a continuous-time solution for the IDLE-TT-COMM framework. The same scenario as that shown in Figure 8.11 is considered

implications of scenarios involving overlapping visibility windows are discussed in the following section.

Satellite Overloaded with Conflicting Observation Tasks

In the case where the orbital positions of the targets result in overlapping and conflicting observation windows, the framework fails to produce continuous observations of the same target. This means that the satellite does not keep the telescope pointed towards the targets for a long enough time, not producing illumination streaks. Such observations do not allow to determine the orbital state of the target object, and do not allow for its position to be added to a surveillance catalogue, not increasing the mission profit.

The reason for this limitation is that no measure of priority has been implemented in the framework, nor has a minimum threshold duration for each observation tasks. A practical example can help understand this better. Let's assume a certain population of target objects results in conflicting observation windows, in which in the same 5 minutes, 4 different objects are visible, and require different pointing directions. Following the approach presented in previous chapters, the scheduling horizon is discretised into intervals based on the observation windows, and the satellite is able to choose between being in IDLE mode, or tracking one of the targets visible in the current interval. If the observation windows of multiple targets overlap, the overlapping windows result in intervals that are shorter than the minimum threshold to produce an illumination streak. As a result of this, the algorithm outputs scheduling solutions in which the satellite observes one target for a short time, and then another one, and so on, leading to observations that do not allow to determine the orbit of the target. This happens because, from the perspective of the algorithm, there is no reason to prefer long continuous observations of the same target over subsequent, shorter, observations of different targets, as the mission profit is set as the total amount of data downlinked. Implementing a measure of target priority, or a minimum duration threshold for observation tasks, is thus necessary to help overcome this problem.



(a) Zoomed Out

(b) Zoomed In

Figure 8.13: Optical targets as seen by the observer satellite while in TASKED TRACKING mode. FoV = $6°$. Results from the implementation of the scheduling solution shown in Figure 8.12.

Target Object Path in Telescope FoV

Figure 8.13 shows the target objects as seen by the FoV of the camera while the satellite is in TASKED TRACKING mode, for the scheduling solution shown in Figure 8.12. Figure 8.13a shows how the satellite rotates about its body-axes by large angles to align the camera boresight with the LoS to the target object, as soon as satellite enters the TASKED TRACKING mode. Figure 8.13b then shows how the satellite keeps rotating, to keep the target within the FoV of its camera. With a specific frame rate of the optical camera, the image in Figure 8.13b would produce streaks, facilitating orbit determination of the target object and enabling its inclusion in a catalogue of space objects. As stated in subsection 3.4.1, adding states of an object to such a catalogue is the main goal of space surveillance activities.

This outcome serves as a verification of the tool and framework developed, demonstrating that the implemented steps culminate in this cataloguing result, representing the scientific contribution of space-based space-surveillance missions. Additionally, it shows that the gains of the PD controller were determined correctly for the TASKED TRACKING mode, as they achieve their goal of correctly changing the attitude of the satellite and keeping target objects within the FoV of the camera.

8.8. Computational Results

This section discusses the computational performance of the different frameworks. First, subsection 8.8.1 focuses on the parameters affecting computational performance. Subsequently, sections 8.8.2 to 8.8.4 present the results for the three frameworks considered.

8.8.1. Parameters Influencing Computational Performance

Computational performance is primarily influenced by the scheduling horizon, the number of intervals, and the average number of options per interval, as analysed by Spangelo et al. [50]. The following sections discuss these.

Number of Intervals

The number of intervals depends on the scenario, discretisation scheme, and the framework used. For the SURVEILLANCE - COMMUNICATION (SC) and IDLE - SURVEILLANCE - COMMUNICATION (ISC) frameworks, intervals are triggered by ground station visibility and the observer satellite's illumi-

Table 8.14: Average number of options per interval variation for different eclipse conditions, observer satellite altitude and discretisation schemes. Results shown for SC and ISC frameworks

		SURV-COMM		IDLE-SURV-COMM	
Eclipses	**Altitude**	**Avg Nr Opt (A)**	**Avg Nr Opt (B)**	**Avg Nr Opt (A)**	**Avg Nr Opt (B)**
Yes	500 km	1.44	1.25	2.44	2.25
No	500 km	1.56	1.56	2.56	2.56
Yes	800 km	1.62	1.45	2.62	2.45
No	800 km	1.72	1.72	2.72	2.72

nation condition. Discretisation scheme A triggers new intervals based on ground station visibility and eclipse conditions, while scheme B also considers changes in power generation. The IDLE - TASKED TRACKING - COMMUNICATION (ITC) framework includes additional intervals based on visibility windows of target objects. The number of intervals scales linearly with the scheduling horizon due to periodic visibility changes of targets and ground stations.

Average Number of Options Per Interval

This metric depends on several factors: the framework, illumination condition, observer satellite altitude, discretisation scheme, and the number of target objects. Illumination conditions affect intervals and the average number of options per interval, with more intervals occurring in scenarios with eclipses, leading to a lower average number of options per interval. Higher observer altitudes increase the duration of communication windows, leading to more situations in which multiple ground stations are visible at the same time, thus increasing the average number of options per interval. The framework impacts the minimum number of options available per interval, with SC and ITC having a minimum of one and ISC having a minimum of two options. The ITC framework's complexity depends on the number of target objects and their orbits.

The discretisation scheme also significantly impacts the number of intervals while keeping the communication and visibility windows constant. There are two schemes considered: A and B. Scheme B takes into account variations in power generation, resulting in more intervals during eclipse periods compared to scheme A, and consequently a lower average number of options per interval. This distinction highlights the importance of the chosen discretisation scheme in determining the scheduling and operational strategy of the satellite.

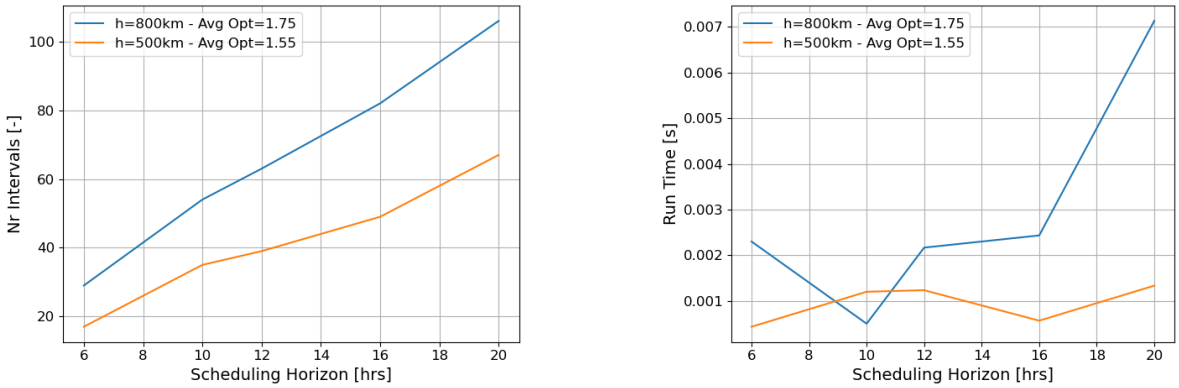
An example overview of the average number of options per interval for the SC and ISC frameworks for different combinations of illumination condition, altitude and discretisation schemes is shown in Table 8.14.

8.8.2. SURVEILLANCE-COMMUNICATION

In this section, the computational results of the SC framework are presented.

Effect of altitude

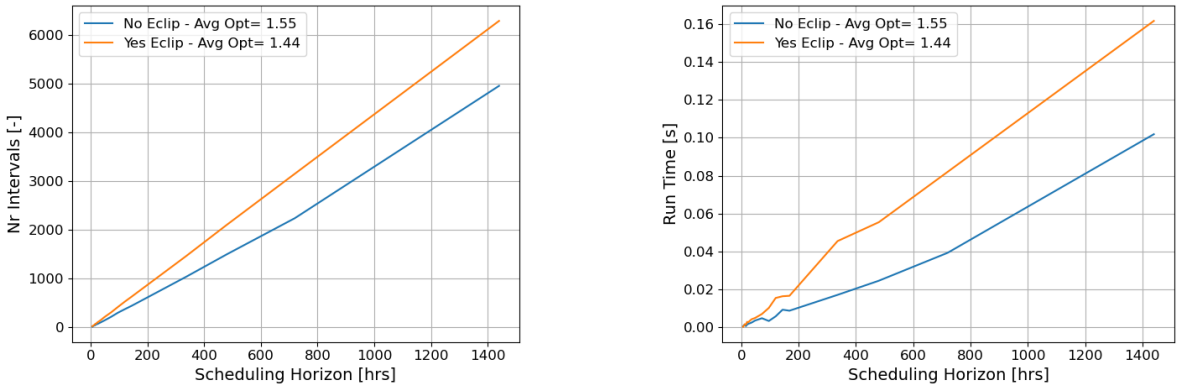
From the sensitivity analysis presented in subsection 8.4.7, it was demonstrated that the SC framework is highly sensitive to the observer satellite altitude. The previous section indicated that a higher observer satellite altitude results in a greater average number of options per interval, potentially impacting computational performance. Therefore, the number of intervals over the scheduling horizon and the run time over the scheduling horizon were plotted for altitudes of 500 km and 800 km. The plots reveal that the average number of options is 13% higher at the higher altitude. Additionally, the computational time for shorter scheduling horizons increases more rapidly at the higher altitude. This may explain why the framework struggles to find feasible solutions for longer scheduling horizons.



(a) Number of intervals over scheduling horizon.

(b) Run time over scheduling horizon.

Figure 8.14: Comparison of number of intervals and run time per scheduling horizon for different observer satellite altitude. Result shown for SC framework, altitude of 500 km and no eclipses.



(a) Number of intervals over scheduling horizon.

(b) Run time over scheduling horizon.

Figure 8.15: Comparison of number of intervals and run time per scheduling horizon for different illumination conditions. Result shown for SC framework, altitude of 500 km. The ”Yes Eclip” scenario is that with the longest eclipse considered.

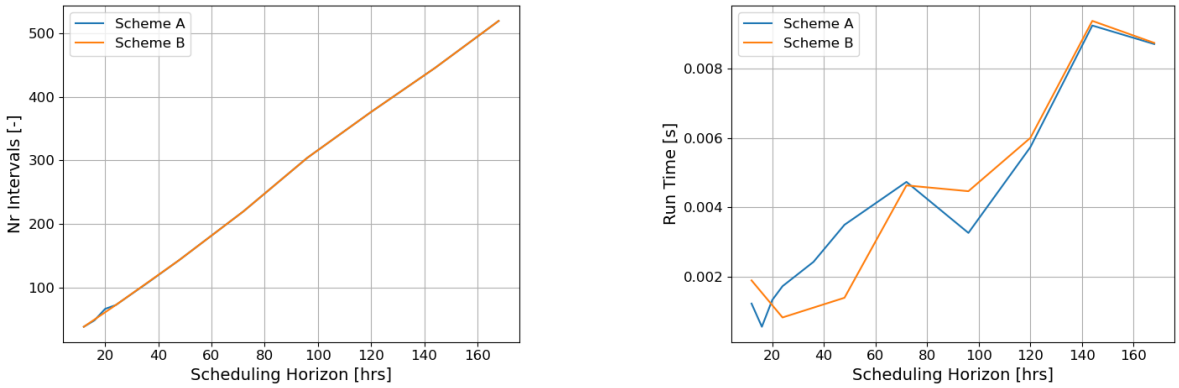
Effect of eclipses

Similar plots were generated for scenarios with and without eclipses at an altitude of 500 km, as shown in Figure 8.15a and Figure 8.15b. It was observed that the number of intervals scales linearly with the scheduling horizon. Additionally, the run time was found to scale linearly with the number of intervals. Consequently, it is logical that the framework requires more time to find a solution when more intervals are included, as is the case when eclipses are considered.

Effect of Discretisation Scheme

Comparing discretisation schemes A and B for the case without eclipses yields similar results, as shown in Figure 8.16. This occurs because, when the power generated does not exhibit significant variations, the two discretisation schemes become more alike. Consequently, using scheme B in the absence of eclipses is unnecessary.

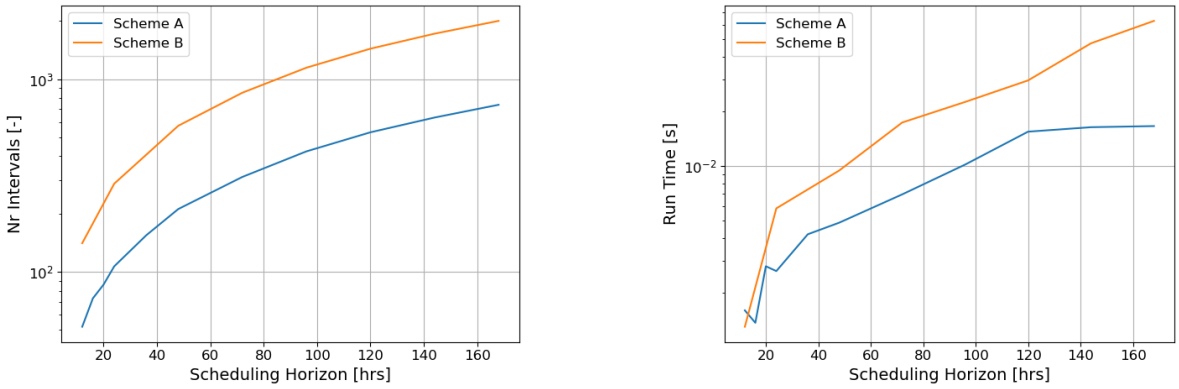
In contrast, comparing discretisation schemes A and B for the case with eclipses reveals notable differences, as illustrated in Figure 8.17. It is evident that the run time for scheme B is higher than that for scheme A when eclipses are considered. Examination of the scheduling solutions from both schemes indicates that when the SoC_{min} constraint is not stringent, the solutions from schemes A and B are very similar. However, scheme B requires propagation of the satellite's attitude and incurs higher run times to solve the MILP problem due to the increased number of intervals. Additionally, the SC framework can perform limited actions with the energy information provided by scheme B because it only



(a) Number of intervals over scheduling horizon.

(b) Run time over scheduling horizon.

Figure 8.16: Comparison of number of intervals and run time per scheduling horizon when using discretisation scheme A and B. Result shown for SC framework, altitude of 500 km and no eclipses.



(a) Intervals comparison

(b) Run time comparison

Figure 8.17: Comparison of number of intervals and run time per scheduling horizon when using discretisation scheme A and B. Result shown for SC framework, altitude of 500 km and longest eclipses.

schedules periods for data downlink, restricting the flexibility to change the power consumption.

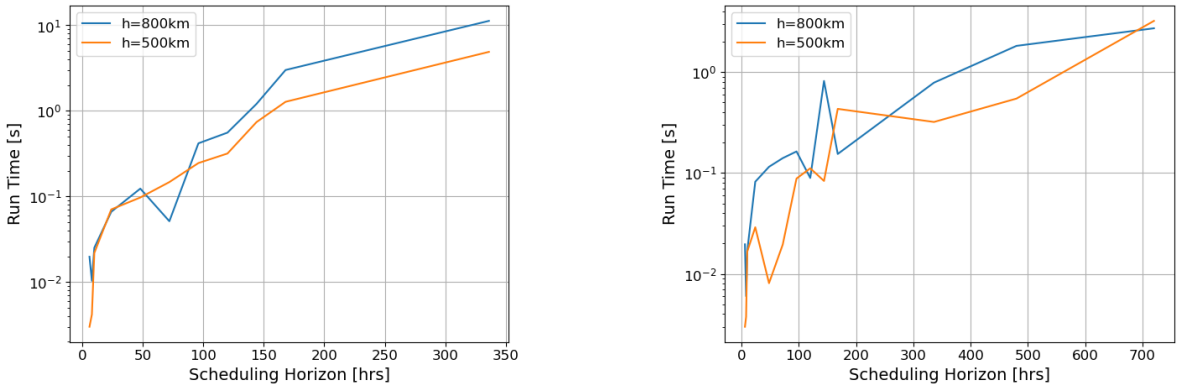
Thus, for the SC framework, using discretisation scheme B over A is beneficial when scheme A fails to produce feasible solutions. In such cases, the increased computational time of scheme B is justified by obtaining feasible solutions for a broader range of scenarios. However, when there is no variation in power generated (i.e., full sunlight), scheme B is redundant as it converges with scheme A. Scheme B is approximately ten times slower than scheme A. Therefore, if scheme A yields feasible solutions, scheme B does not enhance the solution and is unnecessary. Conversely, if scheme A is infeasible, scheme B can help achieve feasibility.

8.8.3. IDLE - SURVEILLANCE - COMMUNICATION

In this section, the computational results of the ISC framework are presented.

Comparison with SC

When comparing the difference in computational time between the SC and ISC framework for an altitude of 500 km, and the illumination condition with longest eclipse periods, the result shown in Figure 8.18 is obtained. From this result, it can be seen that the ISC is approximately 10 times more computationally expensive than the SC framework. This computational difference remains stable and constant over extended scheduling horizons, up to 60 days.



(a) No eclipses

(b) Longest eclipse scenario

Figure 8.19: Comparison of run time per scheduling horizon for different altitudes. Result shown for ISC framework.

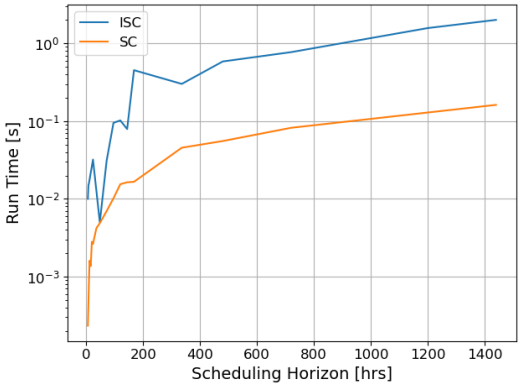


Figure 8.18: Computational Time difference - SC vs ISC - altitude = 500 km - yes eclipses

Effect of Observer Satellite Altitude

The effect of altitude on the ISC framework is analysed in Figures 8.19a and 8.19b. The run-time is similar for altitudes of 500 km and 800 km, in both cases with and without eclipses. This confirms that the ISC framework is less sensitive to observer altitude compared to the SC framework.

Effect of Discretisation Scheme - Γ Optimisation

In subsection 8.6.2 it was shown that, when considering the ISC framework, using discretisation scheme B can be of great help in improving the scheduling solution, achieving feasibility in scenarios with tight energy constraints. The effect that considering different values of Γ to obtain different discretisation schemes has on the quality of the solution was discussed in subsection 8.6.2, here, the effect this has on the computational performance is discussed.

Figure 8.20 provides a comprehensive analysis of the computational performance of the scheduling framework for varying values of Γ. The number of intervals increases as the value of Γ decreases, demonstrating an inverse relationship. Correspondingly, the run time also increases, affirming the proportional relationship between the number of intervals and the computational time required. An interesting observation is the inverse proportionality between run time and the average number of options per interval. This is evident from the specular nature of the two plots: at $\Gamma = 35$ W, a sudden increase in the average number of options corresponds to a notable decrease in run time. This anomaly warrants further investigation to fully understand the underlying cause. Additionally, as the scheduling horizon is discretised into progressively smaller intervals, both the number of intervals and the run time escalate, which is detrimental to computational efficiency. Coupled with the earlier finding that the data downlinked decreases with smaller intervals, it is clear that *the optimal value of* Γ *is just*

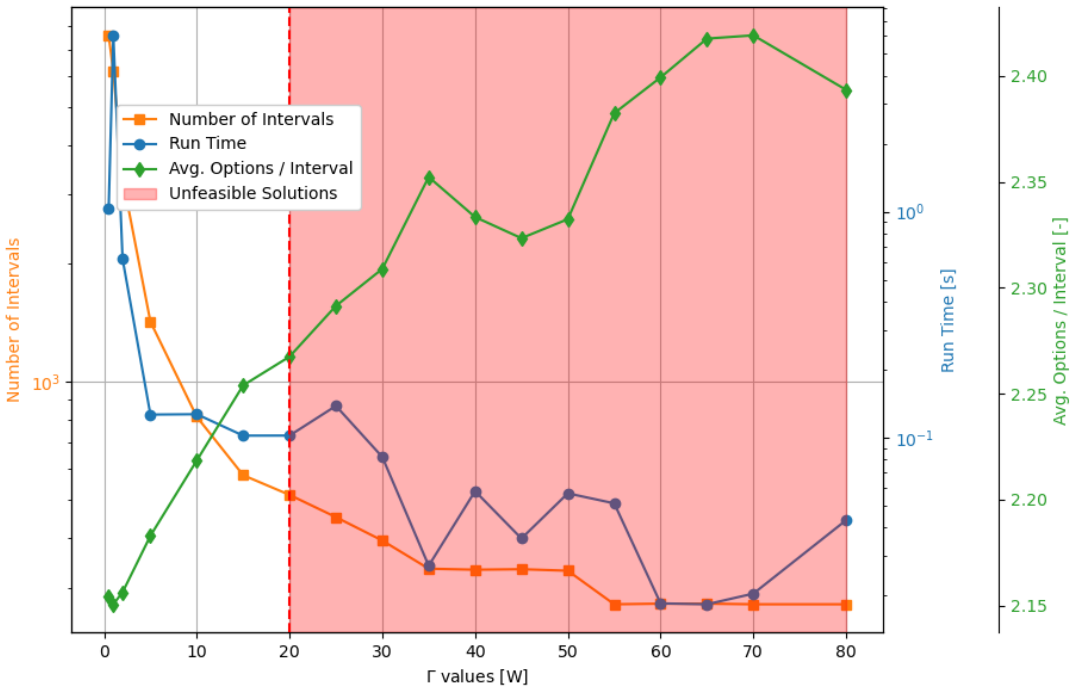


Figure 8.20: Computational performance of the framework for decreasing values of Γ.

at the point where the solution transitions to feasibility. This optimal point minimises run time while maximising the amount of data downlinked, striking a balance between computational efficiency and mission performance.

8.8.4. IDLE - TASKED TRACKING - COMMUNICATION

As discussed in subsection 8.8.4, the ITC framework is less mature than the SC and ISC frameworks, and was implemented primarily to assess the feasibility of the approach, rather than to optimise performance. As such, the computational performance of this framework was evaluated to provide an indicative understanding of its efficiency and scalability with an increasing number of targets. To achieve this, various simulations were conducted with differing scheduling horizons and numbers of target objects, thereby varying the average number of options per interval. The results, depicted in Figure 8.21, illustrate that the framework's run time increases as the number of targets—and consequently, the number of average options per interval—grows. This trend highlights the preliminary nature of this framework and its potential areas for improvement in handling larger, more complex scheduling tasks involving many overlapping visibility windows.

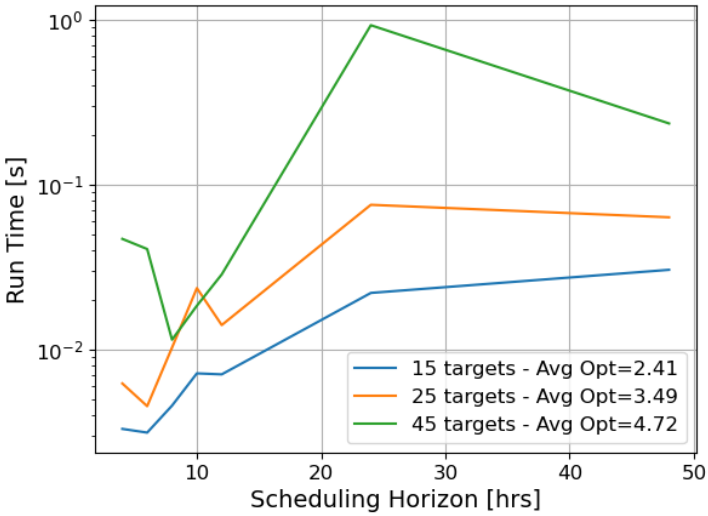


Figure 8.21: Computational time to get solutions for the ITC framework. Comparison for different number of targets

8.8.5. Concluding Remarks

The analysis of the SURVEILLANCE - COMMUNICATION framework demonstrates that scheduling the downlinking of data by only considering ground station passes results in the fastest solution, as evidenced by the lowest computational times in scenarios without eclipses. This approach aligns with the method used by Spangelo et al. If the solution provided by discretisation scheme A is feasible, there is no benefit in employing scheme B, as it does not enhance the solution. However, in cases where the solution is infeasible with scheme A, scheme B can help achieve feasibility. The framework's sensitivity to the altitude of the observer satellite is evident, as at an altitude of 800 km, the run time increases rapidly compared to 500 km, and all solutions become infeasible for planning horizons exceeding 24 hours.

In contrast, the IDLE - SURVEILLANCE - COMMUNICATION framework exhibits similar performance at altitudes of 500 km and 800 km, indicating less sensitivity to altitude. While scheme A results in the ISC framework being approximately ten times slower than the SC framework, it also produces solutions that are infeasible with respect to the battery SoC constraint. Consequently, to maximise the effectiveness of the ISC framework, scheme B must be employed in scenarios with tight energy constraints, despite its computational time being slower than scheme A. This analysis suggests that when the SC framework fails to produce feasible solutions, the ISC framework should be utilised, with discretisation scheme B applied in tight energy scenarios. The value of Γ used to discretise the scheduling horizon into segments using the power generation profile needs to then be adjusted iteratively such that the SoC_{min} constraint is satisfied, as shown in subsection 8.8.3. In such cases, the increased computational time is a necessary condition to achieve feasible solutions. In case the solution using discretisation scheme B results unfeasible, then a more refined discretisation scheme B can be used.

Regarding the IDLE - TASKED TRACKING - COMMUNICATION framework, it is less mature compared to the SC and ISC frameworks and was primarily implemented to assess the feasibility of the approach rather than to optimise performance. The computational performance of the ITC framework was evaluated to provide an indicative understanding of its efficiency and scalability with an increasing number of targets. The results show that the framework's run time increases with the number of targets, highlighting areas for improvement in handling larger, more complex scheduling tasks.

In this chapter, the results obtained following the proposed method have been presented. In the next chapter, the main findings of this research are presented, showing how these enable to answer the research questions of this thesis. Additionally, a number of recommended steps to continue investigating this topic are listed.

9

Conclusion

The congestion of Earth's orbit, particularly in LEO, is increasing. This is largely due to the proliferation of mega constellations, such as those planned and launched by Starlink, OneWeb, and Project Kuiper [51] [39] [4]. The high uncertainty in the positional data of space objects is attributable to the insufficient number of SSA sensors. This uncertainty leads to numerous unnecessary collision avoidance manoeuvres, which are performed based on imprecise data and conservative risk thresholds, thereby consuming valuable fuel and shortening the operational lifespan of satellites, ultimately decreasing revenue for satellite owners.

Space-based SSA offers a potential solution to mitigate these issues by providing more accurate data on the location and trajectory of space objects, thus reducing the need for unnecessary manoeuvres. In recent years, commercial companies such as Vyoma GmbH have been planning such missions [14]. However, there is a notable gap in the literature regarding scheduling the operations for such missions because the field of operational scheduling has been focused only on Earth-observation satellites. This research aims to fill this gap by applying existing task scheduling frameworks from the Earth-Observation Satellite Scheduling Problem (EOSSP) field to the problem of space-based SSA, focusing on missions that consider satellites equipped with an optical telescope.

This research addresses this gap by investigating how to schedule the operations of space-based SSA missions. A combination of low-fidelity and high-fidelity simulators has been employed to gain practical insights into the dynamics of onboard resources during missions, enhancing the understanding of resource management and optimisation in space operations.

9.1. Steps Taken

A literature study was first conducted to identify suitable frameworks, leading to the adoption and implementation of Spangelo et al. [50] approach using a discrete-time scheduler. To address potential unfeasible solutions due to continuous-time dynamics, a continuous-time attitude simulator was developed, enabling precise modelling of onboard resource dynamics and more accurate feasibility determinations. This comprehensive framework was tested on a number of scenarios typical of space-based SSA missions, yielding operational schedules, that were shown to be feasible when considering continuous-time dynamics, meaning they could be implemented in practice.

Additionally, this study explored the incorporation of additional operational modes into the scheduling framework. Although the complete implementation of the four modes typical of space-based SSA mission concepts (IDLE, SURVEILLANCE, TASKED TRACKING, and COMMUNICATION) was not achieved, two novel frameworks were developed and tested under different scenarios, showing promising results for space-based SSA mission scenarios, especially for energy-constraint cases.

9.2. Answers to Research Questions

The results obtained allow to answer the research questions that guided this work. Below, these questions are re-stated, together with the best answer that this research has provided.

RQ1

"What is the best scheduling strategy for SURVEILLANCE mode, that maximises the amount of data downlinked while satisfying the energy and data requirements?"

This study demonstrated that there is no universal strategy that produces the best outcome in all scenarios. The operational schedule must be adapted based on the tightness of onboard resource constraints. For instance, if the energy constraint is not tight, a scheduling strategy with frequent downlinks can be achieved by equipping the satellite with the appropriate combination of payload and solar panels. This approach is desirable as it reduces the maximum downlink intervals, ensuring good timeliness of the data.

Conversely, when the energy constraint is stringent, frequent downlinks are not feasible due to an unmanageable battery SoC profile. In such cases, incorporating an IDLE mode makes the scheduling framework more flexible. This allows the satellite to plan periods of non-scientific activities to recharge batteries, resulting in feasible schedules despite the resource constraints. Here, the "best" strategy involves compromising on data timeliness to achieve feasibility, ensuring the operational schedule can be practically performed by the satellite.

In summary, the definition of the "best" strategy varies depending on the severity of the energy constraint. The developed framework is capable of producing optimal scheduling solutions for both relaxed and stringent energy constraints

RQ2

"To what spacecraft and operational constraints is the scheduling framework most sensitive?"

To answer this question, a sensitivity analysis was performed for the two frameworks developed, that comprising of SURVEILLANCE and COMMUNICATION mode (SC), and that additionally considering the IDLE mode (ISC).

By "spacecraft constraint" it is meant a physical characteristic of the hardware onboard the satellite. With this definition, it was shown that the scheduling framework is most sensitive to the payload and the solar panels. By selecting a good combination of these two parameters, the scheduling framework is able to output solutions which are optimal in terms of data downlink quantity and data timeliness, while satisfying the energy and memory constraint requirements. Other physical constraints, such as the size of the battery or that of the memory, were shown to have less of an impact on the solutions.

By "operational constraint" it is meant a characteristic of the mission that does not arise directly from the spacecraft's hardware, nor it depends on the moment in time in which the mission is analysed. Instead, these are parameters that arise from mission requirements. The scheduling framework was shown to be highly sensitive to the observer satellite's orbital altitude, as less communication windows are utilised to downlink data as the altitude increases, leading to longer downlink intervals. The SC framework is also highly sensitive to the GSN provider, as no feasible solutions can be found for one of the providers, and very different performance is observed for the other two providers. This sensitivity is tackled by adding the IDLE mode, which results in feasible solutions for all providers. The SC framework also suffers from high sensitivity to the duration of the scheduling horizon when coupled with high altitudes, leading to unfeasible solutions for horizons longer than two days for an altitude above 700 km. This is improved when adding the IDLE mode, as feasible solutions can be achieved for up to 60 days, for a range of altitudes. Another parameter that plays an important role in the scheduling framework is the default pointing direction. Altering the latter significantly impacts the power generation profile due to changes in solar panel orientation relative to the Sun. If the default pointing direction results in sub-optimal illumination, manoeuvres increase power collection, allowing a tighter SoC constraint to be imposed. Conversely, if the default pointing direction results in optimal illumination, manoeuvres

decrease power collection, necessitating a more relaxed SoC constraint.

RQ3

"How do initial conditions affect resource management along the (continuous) scheduling horizon?"

By "initial conditions" it is meant parameters that vary depending on the moment in time in which the mission is analysed. For example, the illumination condition for a mission changes in different periods of the year, and for this reason the eclipse condition is considered an initial condition.

Varying the eclipse duration significantly impacts the scheduling framework's feasibility and performance. As eclipse duration increases, power generation decreases, requiring the batteries to supply more power to the payload. This leads to a higher Equivalent Battery Cycles (EBC). During worst-case illumination conditions, the continuous-time simulator often shows unfeasible schedules due to SoC dropping below the threshold, indicating the framework's sensitivity to prolonged eclipses. For such cases, larger solar panel sizes or larger batteries can help achieve feasible solutions. The initial battery SoC also has a large effect on the scheduling framework. When this value becomes too low (i.e. below 40%), while the illumination condition is sub-optimal, then the framework without IDLE mode is not able to find feasible solutions. The framework with IDLE mode is instead able to schedule periods of non-scientific activity to recharge the batteries, and obtain feasible solutions also for such more tight scenarios. The onboard memory level instead does not have a large effect on the solution, affecting mainly what the short-term downlink strategy is, based on how much data is available at the beginning of the optimisation.

RQ4

"What are the impacts of including additional operational modes (i.e. IDLE and TASKED TRACKING) in the scheduling framework?"

During this thesis, the effect of including the IDLE and TASKED TRACKING mode to the scheduling framework was investigated.

Incorporating the IDLE mode into the framework allows the satellite to operate in two modes within the same interval. By designating IDLE as the default mode, the satellite can alternate portions of each interval between SURVEILLANCE and COMMUNICATION modes, and then return to IDLE. This inclusion has been shown to enhance the framework's flexibility, enabling it to produce feasible solutions even under highly constrained energy resources, and making it less sensitive to input parameters, thereby achieving feasible solutions for up to 60 days. Furthermore, the ISC framework allows for an iterative approach. If the initial solution is unfeasible due to energy constraints, the scheduling horizon can be divided into smaller segments, which has been demonstrated to improve the solution and achieve feasibility in power-limited scenarios. However, this framework is approximately 10 times more computationally expensive than its counterpart without the IDLE mode, and it has been shown that smaller discretisation steps require progressively longer computation times.

Substituting TASKED TRACKING for the SURVEILLANCE mode has significant implications for the framework's functioning. The discretisation scheme changes, using target visibility windows and GSN communication windows to split the scheduling horizon into intervals. The framework is highly sensitive to the number of target objects considered. When too many targets cause overlapping visibility windows, the framework outputs a schedule with the satellite switching rapidly between targets, which is undesirable as continuous observation is required for profitable tracking. Thus, the framework can schedule satellite tasks for tasked tracking and data communication, yielding satisfactory results when the satellite is not overloaded with conflicting tasks. To ensure effectiveness, the framework should enforce a minimum observation time per target, or a measure of target priority.

During the thesis, an attempt was made to incorporate all four modes typical of a space-based SSA mission into one scheduling framework. This is discussed in the next section, when making recommendations on how to continue this work.

9.3. Recommendation for Future Work

In this section, proposed recommendations to continue the work commenced with this thesis are presented.

9.3.1. IDLE - SURVEILLANCE - TASKED TRACKING - COMMUNICATION

In order for a task scheduling framework to consider all modes typical of a space-based SSA mission, four operational modes should be included. Ideally, the satellite would be in IDLE mode by default, and the solution would schedule when it goes in SURVEILLANCE, TASKED TRACKING or COMMUNICATION mode.

This was attempted during the research, by setting up the problem following a similar approach to that explained in chapter 6, and solving it as a MILP using Gurobi. Unfortunately, the algorithm was not able to discern between data acquired while in SURVEILLANCE and while in TASKED TRACKING, and resulted in a double counting of the data collected. This distinction between type of observation data collected is necessary for the successful implementation of the framework. The reason for this is that collecting surveillance data usually costs less energy than collecting tasked tracking data, due to the ADCS consuming more power while the satellite operates in TASKED TRACKING mode, and the payload consuming the same power in both modes. Thus, two different weights need to be assigned to collecting the two types of data, as otherwise the framework would stick to only collecting the data that costs less energy, as it would bring the same profit. In order to be able to assign different weight, discerning between the types of data is necessary. As a next step, the implementation of the complete framework could be investigated, paying close attention to the problem of doubling counting the data collected.

9.3.2. Minimum Observation Duration for Tasked Tracking Targets

To enhance the effectiveness of the proposed framework incorporating the TASKED TRACKING mode, it is recommended to implement a measure ensuring that the satellite observes each target for a minimum duration. This adjustment will address the issue of the satellite rapidly switching between targets when visibility windows overlap, which is currently undesirable for profitable tracking. By enforcing a minimum observation time per target, the framework will better support continuous and effective data collection, ultimately improving the overall performance of space-based SSA missions involving TASKED TRACKING. Future work should focus on developing and integrating this constraint into the scheduling algorithm to ensure more reliable and efficient satellite operations.

9.3.3. Gain Scheduling

In the current work, the same gains are used by the PD controller for both TASKED TRACKING manoeuvres and COMMUNICATION manoeuvres. However, these gains were tuned based on the performance of the TASKED TRACKING mode. Consequently, the manoeuvres performed to point the antenna towards ground stations are not as accurate. This issue could be addressed by using two different sets of gains: one for TASKED TRACKING and one for COMMUNICATION. The technique of using different sets of controller gains at predetermined times is known as gain scheduling. This approach would allow for an optimised performance in both modes.

9.4. Open Science Contribution

During this thesis, an open-source set of modules to set-up and solve satellite scheduling problems for a range of mission scenarios was developed. Of the complete pipeline, shown in Figure 4.8, the steps made using python, so everything concerning the discrete-time scheduler, has been made open-source.

These modules use libraries and material taught in MSc courses at the aerospace faculty of TU Delft, namely:

- The propagation of the environment, and discretisation of the scheduling horizon into intervals is made using `tudat`. Relevant background to understand these steps is taught in the courses "Numerical Astrodynamics (AE4868-1)", "Space Debris Tracking and Mitigation (AE4898)" and "Propagation and Optimisation in Astrodynamics (AE4866-1)" in the Spaceflight Dynamics track;
- The MILP is set-up and solved using `gurobipy`, which is available under the TU Delft licence. The implementation of the MILP follows the material taught in the course "Operations Optimisation (AE4441-16)", in the Control and Operations track.

Therefore, this thesis has developed a framework that future students can utilise and build upon, fostering open science and encouraging continued contributions to this field of research.

References

[1] NASA Orbital Debris Program Office (ODPO). “Orbital Debris Quarterly News”. en. In: 28.2 (Apr. 2024), p. 10.

[2] A. Abbas et al. “Agile Mission Optimization for a Constellation of Earth Observation Satellites”. In: *IOP Conference Series: Materials Science and Engineering*. Vol. 610. 1. 2019, p. 012059.

[3] ESA - European Space Agency. *Sentinel Online*. Accessed on October 26, 2023. 2023. URL: https://sentinels.copernicus.eu/web/sentinel/home.

[4] Amazon. *Project Kuiper*. https://www.aboutamazon.com/what-we-do/devices-services/project-kuiper. Accessed: 2024-05-20.

[5] L. Anselmo and C. Pardini. “Analysis of the Consequences in Low Earth Orbit of the Collision between Cosmos 2251 and Iridium 33”. In: *Proceedings of the 21st International Symposium on Space Flight Dynamics*. Centre Nationale d’Etudes Spatiales. Paris, France, 2009, pp. 1–15.

[6] A Attea Bara’a et al. “A review of heuristics and metaheuristics for community detection in complex networks: Current usage, emerging development and future directions”. In: *Swarm and Evolutionary Computation* 63 (2021), p. 100885.

[7] Doo-Hyun Cho et al. “Optimization-Based Scheduling Method for Agile Earth-Observing Satellite Constellation”. In: *Journal of Aerospace Information Systems* 15.11 (2018). Publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/1.I010620, pp. 611–626. ISSN: 1940-3151. DOI: 10.2514/1.I010620. (Visited on 07/11/2023).

[8] tudat community. *tudat.space documentation*. Accessed on November 14, 2023. 2023. URL: https://docs.tudat.space/en/latest/_src_user_guide/state_propagation/environment_setup/frames_in_environment.html#quaternion-definition.

[9] K. Cui, J. Xiang, and Y. Zhang. “Mission Planning Optimization of Video Satellite for Ground Multi-object Staring Imaging”. In: *Advances in Space Research* 61.6 (2018), pp. 1476–1489.

[10] Canada Department of National Defence. *A Gem in Space: Canada’s Sapphire Satellite Continues to Shine*. Accessed: 2024-05-01. 2023. URL: https://www.canada.ca/en/department-national-defence/maple-leaf/rcaf/2023/02/gem-in-space-canada-sapphire-satellite-continues-to-shine.html.

[11] Yonghao Du et al. “A Data-Driven Parallel Scheduling Approach for Multiple Agile Earth Observation Satellites”. In: *IEEE Transactions on Evolutionary Computation* 24.4 (Aug. 2020). Conference Name: IEEE Transactions on Evolutionary Computation, pp. 679–693. ISSN: 1941-0026. DOI: 10.1109/TEVC.2019.2934148.

[12] Yonghao Du et al. “Integrated agile observation satellite scheduling problem considering different memory environments: a case study”. en. In: *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 42.1 (Jan. 2020), p. 76. ISSN: 1806-3691. DOI: 10.1007/s40430-019-2121-0. (Visited on 07/26/2023).

[13] D. Eddy and M. Kochenderfer. “Markov Decision Processes for Multi-Objective Satellite Task Planning”. In: *arXiv preprint arXiv:1910.08419* (2019).

[14] EnduroSat. *EnduroSat to Build Vyoma’s Space Situational Awareness Constellation*. Accessed: 2024-05-01. 2023. URL: https://www.endurosat.com/news/endurosat-to-build-vyomas-space-situational-awareness-constellation/.

[15] Sabah S Fayaed, Ahmed El-Shafie, and Othman Jaafar. “Reservoir-system simulation and optimization techniques”. In: *Stochastic environmental research and risk assessment* 27 (2013), pp. 1751–1772.

[16] M. Ferrazzani. “Preliminary Considerations on the European Preparatory Programme on Space Situational Awareness”. In: *Proc. of the 61st International Astronautical Congress*. IAC-10-E7.3. Prague, Czech Republic, Oct. 2010.

[17] Edemar Morsch Filho et al. “Irradiation Flux Modelling for Thermal–Electrical Simulation of CubeSats: Orbit, Attitude and Radiation Integration”. en. In: *Energies* 13.24 (Jan. 2020). Number: 24 Publisher: Multidisciplinary Digital Publishing Institute, p. 6691. ISSN: 1996-1073. DOI: 10.3390/en13246691. (Visited on 07/20/2023).

[18] Tim Flohrer. *Optical survey strategies and their application to space surveillance*. de. Geodätisch-geophysikalische Arbeiten in der Schweiz Siebenundachtzigster Band. Zürich: Schweizerische Geodätische Kommission, 2012. ISBN: 978-3-908440-34-5.

[19] Virginie Gabrel et al. “A new single model and derived algorithms for the satellite shot planning problem using graph theory concepts”. en. In: *Annals of Operations Research* 69.0 (Jan. 1997), pp. 115–134. ISSN: 1572-9338. DOI: 10.1023/A:1018920709696. (Visited on 08/08/2023).

[20] *General Mission Analysis Tool (GMAT) User Guide*. NASA Goddard Space Flight Center. Greenbelt, MD, USA, 2020.

[21] IADC Steering Group and Working Group 4. *IADC Space Debris Mitigation Guidelines*. Tech. rep. IADC-02-01. Revision 2. Inter-Agency Space Debris Coordination Committee, Mar. 2020.

[22] LLC Gurobi Optimization. *Mixed-Integer Programming (MIP) – A Primer on the Basics*. Accessed: 2024-05-15. 2024. URL: https://www.gurobi.com/resources/mixed-integer-programming-mip-a-primer-on-the-basics/.

[23] *Gurobi Optimizer Python API Overview*. Accessed: 2024-05-02. 2024. URL: https://www.gurobi.com/documentation/11.0/refman/py_python_api_overview.html#sec:Python.

[24] Aridon Gwenaëlle et al. “Selection of shielding materials and configurations for particle debris impacts of future LEO satellites”. In: vol. 705. Sept. 2012.

[25] L. He et al. “Hierarchical Scheduling for Real-time Agile Satellite Task Scheduling in a Dynamic Environment”. In: *Advances in Space Research* 63.2 (2019), pp. 897–912.

[26] Andris D. Jaunzemis, Marcus J. Holzinger, and Moriba K. Jah. “Evidence-based Sensor Tasking for Space Domain Awareness”. In: *Copyright © 2016 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*. www.amostech.com. 2016.

[27] Xianghua Jia et al. “Eclipse Prediction Algorithms for Low-Earth-Orbiting Satellites”. In: *IEEE Transactions on Aerospace and Electronic Systems* 53.6 (2017), pp. 2963–2975. DOI: 10.1109/TAES.2017.2722518.

[28] Michel Lemaitre et al. “Selecting and scheduling observations of agile satellites”. en. In: *Aerospace Science and Technology* 6.5 (Sept. 2002), pp. 367–381. ISSN: 1270-9638. DOI: 10.1016/S1270-9638(02)01173-2. (Visited on 07/20/2023).

[29] Longmei Li et al. “Preference incorporation to solve multi-objective mission planning of agile earth observation satellites”. In: *2017 IEEE Congress on Evolutionary Computation (CEC)*. June 2017, pp. 1366–1373. DOI: 10.1109/CEC.2017.7969463.

[30] Z. Li and X. Li. “A Multi-Objective Binary-Encoding Differential Evolution Algorithm for Proactive Scheduling of Agile Earth Observation Satellites”. In: *Advances in Space Research* 63.10 (2019), pp. 3258–3269.

[31] Xiaolu Liu et al. “An adaptive large neighborhood search metaheuristic for agile satellite scheduling with time-dependent transition time”. en. In: *Computers & Operations Research* 86 (Oct. 2017), pp. 41–53. ISSN: 0305-0548. DOI: 10.1016/j.cor.2017.04.006. (Visited on 08/08/2023).

[32] Fabrizio Marinelli et al. “A Lagrangian heuristic for satellite range scheduling with resource constraints”. en. In: *Computers & Operations Research* 38.11 (Nov. 2011), pp. 1572–1583. ISSN: 0305-0548. DOI: 10.1016/j.cor.2011.01.016. (Visited on 07/20/2023).

[33] MathWorks. *Linear-Quadratic Regulator (LQR)*. Accessed: 2023-04-26. 2023. URL: https://nl.mathworks.com/help/control/ref/lti.lqr.html.

[34] Xiaonan Niu, Hong Tang, and Lixin Wu. “Satellite scheduling of large areal tasks for rapid response to natural disaster using a multi-objective genetic algorithm”. en. In: *International Journal of Disaster Risk Reduction* 28 (June 2018), pp. 813–825. ISSN: 2212-4209. DOI: 10.1016/j.ijdrr.2018.02.013. (Visited on 07/21/2023).

[35] *Northrop Grumman Space Tracking and Surveillance System*. Accessed on 27-10-2023. URL: https://www.northropgrumman.com/space/stss.

[36] ESA Space Debris Office. *ESA’S ANNUAL SPACE ENVIRONMENT REPORT*. LOG 7.0. ESA ESOC, Robert-Bosch-Strasse 5, D-64293 Darmstadt, Germany: European Space Agency (ESA), June 2023.

[37] ESA Space Debris Office. *MASTER-8 (Meteoroid and Space Debris Terrestrial Environment Reference model)*. https://sdup.esoc.esa.int/master/. Version 8.0.3. 2022.

[38] Diego Escorial Olmos et al. “SPACE-BASED SPACE SURVEILLANCE OPERATIONAL AND DEMONSTRATION MISSIONS”. In: *6th European Conference on Space Debris*. European Space Agency (ESA). Apr. 2013. URL: https://conference.sdo.esoc.esa.int/proceedings/sdc6/paper/174/SDC6-paper174.pdf.

[39] OneWeb. *Our Network*. https://oneweb.net/our-network. Accessed: 2024-05-20.

[40] Joseph C Pemberton and Flavius Galiber. “A constraint-based approach to satellite scheduling”. In: *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* 57 (2001), pp. 101–114.

[41] Claude Phipps et al. “ORION: Clearing near-Earth space debris using a 20-kW, 530-nm, Earth-based, repetitively pulsed laser”. In: *Laser and Particle Beams* 14 (Mar. 1996), pp. 1–44. DOI: 10.1017/S0263034600009733.

[42] Juntong Qi et al. “A Cooperative Autonomous Scheduling Approach for Multiple Earth Observation Satellites With Intensive Missions”. In: *IEEE Access* 9 (2021). Conference Name: IEEE Access, pp. 61646–61661. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3075059.

[43] Cezar Antônio Rigo et al. “A nanosatellite task scheduling framework to improve mission value using fuzzy constraints”. en. In: *Expert Systems with Applications* 175 (Aug. 2021), p. 114784. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2021.114784. (Visited on 07/11/2023).

[44] S Rojanasoonthon, J F Bard, and S D Reddy. “Algorithms for parallel machine scheduling: a case study of the tracking and data relay satellite system”. In: *Journal of the Operational Research Society* 54.8 (Aug. 2003). Publisher: Taylor & Francis, pp. 806–821. ISSN: 0160-5682. DOI: 10.1057/palgrave.jors.2601575. (Visited on 08/09/2023).

[45] Hanspeter Schaub and John L. Junkins. *Analytical mechanics of space systems*. 2nd ed. AIAA Education Series. AIAA, Jan. 2009. DOI: 10.2514/4.867231.

[46] Yuchen She, Shuang Li, and Yanbin Zhao. “Onboard mission planning for agile satellite using modified mixed-integer linear programming”. en. In: *Aerospace Science and Technology* 72 (Jan. 2018), pp. 204–216. ISSN: 1270-9638. DOI: 10.1016/j.ast.2017.11.009. (Visited on 08/09/2023).

[47] Chuan Shi et al. “Comparison and selection of objective functions in multiobjective community detection”. In: *Computational Intelligence* 30.3 (2014), pp. 562–582.

[48] Pietro Silvagni. “Agile Space-Observation Satellite Task Scheduling Optimisation: A Literature Study”. Master’s thesis Literature Study, Faculty of Aerospace Engineering, Delft University of Technology. Dec. 2023.

[49] *Space-Based Space Surveillance Fact Sheet - Space Based Space Surveillance (SBSS)*. Accessed on 27-10-2023. URL: https://www.spaceforce.mil/About-Us/Fact-Sheets/Article/2197743/space-based-space-surveillance/.

[50] Sara Spangelo et al. “Optimization-based scheduling for the single-satellite, multi-ground station communication problem”. en. In: *Computers & Operations Research* 57 (May 2015), pp. 1–16. ISSN: 0305-0548. DOI: 10.1016/j.cor.2014.11.004. (Visited on 07/11/2023).

[51] Starlink. *Technology*. https://www.starlink.com/technology. Accessed: 2024-05-20.

[52] Yosuke Takeo and Saburo Matunaga. “Multi-objective Optimization of Observation Scheduling Avoiding Attitude Maneuvering Conflict”. en. In: *TRANSACTIONS OF THE JAPAN SOCIETY FOR AERONAUTICAL AND SPACE SCIENCES* 65.2 (2022), pp. 66–75. ISSN: 0549-3811, 2189-4205. DOI: 10.2322/tjsass.65.66. (Visited on 07/11/2023).

[53] Panwadee Tangpattanakul, Nicolas Jozefowiez, and Pierre Lopez. “A multi-objective local search heuristic for scheduling Earth observations taken by an agile satellite”. en. In: *European Journal of Operational Research* 245.2 (Sept. 2015), pp. 542–554. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2015.03.011. (Visited on 07/26/2023).

[54] Blue Canyon Technologies. *Power Systems*. Accessed: May 2, 2024. 2024. URL: https://www.bluecanyontech.com/components.

[55] UCS. *Satellite Database*. https://www.ucsusa.org/resources/satellite-database. Accessed on [29/05/2024].

[56] Shuo Wang et al. “Task scheduling and attitude planning for agile earth observation satellite with intensive tasks”. en. In: *Aerospace Science and Technology* 90 (July 2019), pp. 23–33. ISSN: 1270-9638. DOI: 10.1016/j.ast.2019.04.007. (Visited on 07/11/2023).

[57] Xinwei Wang et al. “Agile Earth Observation Satellite Scheduling Over 20 Years: Formulations, Methods, and Future Directions”. In: *IEEE Systems Journal* 15.3 (Sept. 2021). Conference Name: IEEE Systems Journal, pp. 3881–3892. ISSN: 1937-9234. DOI: 10.1109/JSYST.2020.2997050.

[58] Zhuo Wang and Rui Xu. “Rapid search method for a spacecraft attitude maneuver path with multiple constraints”. en. In: *Aerospace Science and Technology* 117 (Oct. 2021), p. 106904. ISSN: 1270-9638. DOI: 10.1016/j.ast.2021.106904. (Visited on 07/11/2023).

[59] J. Wertz. *Spacecraft Attitude Determination and Control*. Springer, 1978.

[60] J.R. Wertz, D.F. Everett, and J.J. Puschell. *Space Mission Engineering: The New SMAD*. Space technology library. Microcosm Press, 2011. ISBN: 9781881883159. URL: https://books.google.it/books?id=VmQmtwAACAAJ.

[61] Bong Wie. *Space vehicle dynamics and control*. Aiaa, 1998.

[62] W. Yang et al. “The Bi-Objective Active-Scan Agile Earth Observation Satellite Scheduling Problem: Modeling and Solution Approach”. In: *Congress on Evolutionary Computation*. IEEE. 2018, pp. 1–6.

[63] Nicolas Zufferey, Patrick Amstutz, and Philippe Giaccari. “Graph colouring approaches for a satellite range scheduling problem”. en. In: *Journal of Scheduling* 11.4 (Aug. 2008), pp. 263–277. ISSN: 1099-1425. DOI: 10.1007/s10951-008-0066-8. (Visited on 08/08/2023).

A

Linear Resources Dynamics within each Interval as a Sufficient Condition for the Feasibility and Optimality of the Scheduling Solution

A.1. Theorem

"Having linear energy and data dynamics within each interval in the UCF problem is a sufficient condition to yield a feasible, and thus optimal, solution to the satellite scheduling problem when considering the continuous-time dynamics."

Spangelo et al. [50]

A.2. Proof

For simplicity, the discussion presented here is focused on the energy dynamics, as the same logic can be applied to the data dynamics.

Assume an instance of the scheduling problem in which:

- The rates of energy and data acquisition, π_{e+} and π^{d+}
- The rate of nominal energy consumption, π^{e-}
- The rate of data loss, π^{d-loss}

are all constant with respect to time over the duration of any interval $i \in I$.

Since the energy consumption and acquisition rates are constant with respect to time throughout the interval, $\pi^{e+}(t)$ and $\pi^{e-}(t)$ can be substituted with π_i^{e+} and π_i^{e-}. By the first lemma, shown below in section A.3, any solution in which data is downloaded for only a portion of interval i at the chosen rate can be converted into an equivalent solution where data is downloaded for the whole duration of the interval, at a lower constant rate. The same can be done for the energy spillage rate, as shown in section A.4.

After performing these two conversions, the rates at which energy is acquired, consumed for nominal operations, consumed to download data and spilled are all constant through the entire duration of interval i. This means that the net change in energy within the interval is linear with respect to time.

This can be expressed as:

$$e(t) = e_i + \left[\pi_i^{e+} - \pi_i^{e-} - \Theta_i^e - \Gamma_i^e\right] \cdot \Delta t_i \qquad \forall\, i \in I, t \in [t_i, t_{i+1}] \tag{A.1}$$

As a result of this, the energy at any time within the interval, $t \in [t_i, t_{i+1}$, will lie on the line-segment connecting the two points (t_i, e_i) and (t_{i+1}, e_{i+1}). Because the energy level at times t_i and t_{i+1} are within the limits, due to the constraint imposed by Equation 5.5, the energy level at any point along this line segment will be within the feasible range as well.

A.3. 1^{st} Lemma

Assume there exists a solution to an instance of the UCF problem in which, during an interval i:

- The interval has a duration of Δt_i;
- A data downlink rate, $\phi_{i,o}^*$ is used to download;
- Amount of data downloaded is equal to a total of $q_{i,o}^*$;
- Data is downloaded for a time $t_d < \Delta t_i$

, where the superscript * indicates the solution.

The fraction of interval i during which download occurs can be calculated as follows:

$$p = \frac{q_{i,o}^*}{\phi_{i,o}^* \cdot \Delta t_i} \tag{A.2}$$

An equivalent solution can be constructed, in which data is downloaded at a *lower* constant rate of:

$$\Theta_i^d = \phi_{i,o}^* \cdot p = \phi_{i,o}^* \cdot \frac{q_{i,o}^*}{\phi_{i,o}^* \cdot \Delta t_i} = \cancel{\phi_{i,o}^*} \cdot \frac{q_{i,o}^*}{\cancel{\phi_{i,o}^*} \cdot \Delta t_i} = \frac{q_{i,o}^*}{\Delta t_i} \tag{A.3}$$

over the entire duration of the interval.

If the energy per data downloaded stays constant, this means that in the new solution, a constant rate of energy consumption for downloading data can be defined, Θ_i^e

A.3.1. Proof

A solution to the UCF problem specifies the amount of data to be downloaded during an interval, $q_{i,o}^*$ and the rate at which this download occurs, $\phi_{i,o}^*$. For the cases in which the specified amount of data and download rate does not fill the entire duration of the interval (i.e. $t_{download} < \Delta t_i$, or $0 < p < 1$), the solution does not specify *when* within the interval the download must occur. Additionally, *the physics of the system do not require that the download occurs over a single continuous interval*. It can thus be assumed, without loss of generality, that the full interval is split into smaller sub-intervals of equal duration, and that in each sub-interval the satellite downloads for the first p fraction of time, and goes back to the nominal mode for the remainder of the sub-interval. As the size of these sub-intervals approaches zero, the download rate throughout the entire interval i approaches the constant download rate $\phi_{i,o}^* \cdot p$.

A.4. 2^{nd} Lemma

Assume there exists a solution to an instance of the UCF problem in which, during an interval i:

- The interval has a duration of Δt_i;
- A total amount of h_i^{e*} is spilled;
- The spilling rate, ξ_i^*, is used to spill energy;

- Energy is spilled for a time $t_s < \Delta t_i$;

, where the superscript * indicates the solution.

The fraction of interval i during which energy spillage occurs can be calculated as follows:

$$f = \frac{h^{e*}}{\xi_i^* \cdot \Delta t_i} \tag{A.4}$$

An equivalent solution can be constructed, in which energy is spilled at a constant rate of:

$$\Gamma_i^e = \xi_i^* \cdot f = \xi_i^* \frac{h_i^{e*}}{\xi_i^* \cdot \Delta t_i} = \cancel{\xi_i^*} \frac{h_i^{e*}}{\cancel{\xi_i^*} \cdot \Delta t_i} = \frac{h_i^{e*}}{\Delta t_i} \tag{A.5}$$

A.4.1. Proof

The same approach as the ones taken in Lemma A.1 can be used to prove the current Lemma.

B

IDLE - SURVEILLANCE - COMMUNICATION

In this chapter, the scheduling framework that comprises of the idle, surveillance and communication mode is presented. First, the assumptions that describe this version of the scheduling problem are presented. Then, the problem is formulated mathematically. To help visualise the difference with the original problem, the parts that were added or modified are in blue.

B.1. Assumptions

The assumptions made for the satellite scheduling problem considered here are as follows:

- A single satellite orbits the Earth, and collects data, via an optical telescope, and energy via solar panels;
- The satellite is by default in IDLE mode, and the scheduling solution defines :
  - The amount of data that is acquired while in SURVEILLANCE mode;
  - The amount of data downlinked while in COMMUNICATION mode;
- The satellite energy collection rate depends on the Line-of-Sight of the solar panels relative to the Sun;
- The satellite energy consumption rate depends on its operational mode;
- The satellite data collection rate depends on its operational mode;
- The satellite data downlink rate depends on its operational mode;
- Energy is required to perform nominal operations of the spacecraft. By *"nominal operations"*, it is meant the tasks that are carried out in the IDLE mode of the satellite;
- Energy is required to carry out tasks outside those included in the nominal mode. These include *communication tasks* as well as *observation tasks*;
- The satellite has finite limits on the amount of energy and data that can be stored at any given time;
- There are multiple ground stations as part of the GSN;
- Each ground station periodically comes in- and out-of-view of the satellite. The period of time in which a ground station is visible is referred to as *communication window*;
- The duration of these communication windows depends on the geometry of the orbit of the satellite relative to the ground station;
- It can happen that more than one ground station is in view of the satellite at the same time. In this case, the satellite can download only to one ground station at the time;

- Each ground station is assumed to have the same characteristics, in terms of rate at which they can receive data, energy utilisation from the satellite to send one bit of data, and the efficiency of the data download. The latter includes the losses due to communication system and transmission inefficiencies;
- The satellite is capable of downlinking data to any visible ground stations for the entire duration of an interval or any fraction thereof.
- The satellite is capable of collecting surveillance data for the entire duration of an interval or any fraction thereof.
- The orbit of the satellite is assumed to be deterministic, and as such the communication windows to the GSN are known a priori;
- The single satellite is equipped with magnetorquers, such that it can constantly de-saturate its ADCS actuator. With this assumption, the angular momentum accumulated by the satellite along its orbit does not need to be taken into account.

B.2. Mathematical Formulation

In this section, the scheduling problem is formulated as a Mixed-Integer Linear Programming (MILP) problem. Most of the notation used in this formulation is taken from the work of Spangelo et al. [50].

Parameters

The parameters used in the mathematical formulation are presented below. They follow the notation for which I is the set of intervals, O is the set of operational modes available and $O_i \subseteq O$ is the subset of the operational modes available during interval i, $\forall\, i \in I$. Using this notation results in the following parameters:

- $\eta_{i,o}$: efficiency of data download during interval i when downlinking using option o;
- t_i is the start time of interval i;
- t_{i+1} is the end time of interval i, which coincides with the start time of interval $i+1$;
- Δt_i is the duration of time interval i, and is equal to $t_{i+1} - t_i$;
- $\phi_{i,o}$ is the data rate associated with downloading during interval i using option o;
- $\alpha_{i,o}$ is the energy per data associated with downlinking using option o during interval i;
- $\xi_{i,o}$ is the data rate associated with acquiring surveillance data during interval i using option o;
- $\beta_{i,o}$ is the energy per data associated with acquiring surveillance data using option o during interval i;
- e_{min} and e_{max} are the minimum and maximum allowable amounts of energy that can be stored in the battery;
- e_{start} is the amount of energy stored in the battery at the beginning of the scheduling horizon;
- d_{min} and d_{max} are the minimum and maximum allowable data stored in the onboard memory;
- d_{start} is the amount of data stored in the onboard memory at the beginning of the scheduling horizon;
- δ_i^{e+} is the total amount of energy that can be acquired during interval i;
- δ_i^{e-} is the total amount of energy consumed during interval i for default operations (i.e. idle mode);
- δ_i^{d+} is the total amount of data that is acquired during interval i by default (=0);
- δ_i^{d-} is the total amount of data that is lost during interval i, independent of the operational mode. This includes for example data degradation and expiration.

Note that δ_i^{e-} includes only the energy required to operate the life-supporting subsystems onboard the spacecraft.

Variables

The variables used in the problem are the following:

- $x_{i,o} \in 0, 1$: binary variable indicating the operational mode during some or all of interval i;
- $q_{i,o} \in \mathbb{R}^+$ is the amount of data downloaded during interval i while in operational mode o, $\forall i \in I, o \in O$;
- $w_{i,o} \in \mathbb{R}^+$ is the amount of surveillance data acquired during interval i while in operational mode o, $\forall i \in I, o \in O$;
- $e_i \in \mathbb{R}^+$ is the amount of energy available at the beginning of interval i, measured in Joules, $\forall i \in I$;
- $d_i \in \mathbb{R}^+$ is the amount of data available at the beginning of interval i, measured in bits, $\forall i \in I$;
- $h_i^e \geq 0$ is the amount of excess energy spilled during interval i, measured in Joules, $\forall i \in I$;
- $h_i^d \geq 0$ is the amount of excess data spilled during interval i, measured in bits, $\forall i \in I$.

Formulation

Using the parameters and variables defined above, the optimisation problem can be now formulated mathematically.

Objective function:

$$\max \sum_{i \in I} \sum_{o \in O_i} \eta_{i,o} q_{i,o} \tag{B.1}$$

Subject to:

$$\sum_{o \in O_i} x_{io} \leq 1 \quad \forall i \in I \tag{B.2}$$

$$q_{i,o} \leq \Delta t_i \phi_{i,o} x_{i,o} \quad \forall i \in I, o \in O_i \tag{B.3}$$

$$w_{i,o} \leq \Delta t_i \xi_{i,o} x_{i,o} \quad \forall i \in I, o \in O_i \tag{B.4}$$

$$e_0 = e_{start} \tag{B.5}$$

$$e_{min} \leq e_i \leq e_{max} \quad \forall i \in I \tag{B.6}$$

$$e_{i+1} = e_i + \delta_i^{e+} - \delta_i^{e-} - \sum_{o \in O_i} \alpha_{i,o} q_{i,o} - \sum_{o \in O_i} \beta_{i,o} w_{i,o} - h_i^e \quad \forall i \in I \tag{B.7}$$

$$d_0 = d_{start} \tag{B.8}$$

$$d_{min} \leq d_i \leq d_{max} \quad \forall i \in I \tag{B.9}$$

$$d_{i+1} = d_i + \cancel{\delta_i^{d+}} - \delta_i^{d-} - \sum_{o \in O_i} q_{i,o} + \sum_{o \in O_i} w_{i,o} - h_i^d \quad \forall i \in I \tag{B.10}$$

$$x_{i,o} \in \{0, 1\} \quad \forall i \in I, o \in O_i \tag{B.11}$$

$$q_{i,o} \in \mathbb{R}^+ \quad \forall i \in I, o \in O_i \tag{B.12}$$

$$w_{i,o} \in \mathbb{R}^+ \quad \forall i \in I, o \in O_i \tag{B.13}$$

The new constraints can be explained as follows:

- Equation B.4 enforces that:
  - Surveillance data can only be acquired if the surveillance option has been chosen in interval i;
  - The amount of surveillance data acquired is limited by the duration of the interval, Δt_i and the data acquisition rate, $\xi_{i,o}$;
- Equation B.7 defines the amount energy at the beginning of an interval to be equal to:
  - The amount stored at the start of the previous interval (+)
  - The amount acquired via the solar panels (-)
  - The amount consumed for nominal operations (-)
  - The amount used to downlink data (-)
  - The amount used to acquire surveillance data (-)
  - The amount spilled
- Equation B.10 defines the amount of data at the beginning of an interval to be equal to:
  - The amount stored at the start of the previous interval (+)
  - The amount acquired during IDLE mode (=0) (-)
  - The amount lost due to degradation or expiration (-)
  - The amount downloaded (+)
  - The amount of surveillance data acquired (-)
  - The amount spilled
- Equation B.13 enforces the amount of acquired surveillance data to be a positive, real number.

C

IDLE - TASKED TRACKING - COMMUNICATION

In this chapter, the scheduling framework that comprises of the idle, tasked tracking and communication mode is presented. First, the assumptions that describe this version of the scheduling problem are presented. Then, the problem is formulated mathematically. To help visualise the difference with the original problem, the parts that were added or modified are in blue.

C.1. Assumptions

The assumptions made for the satellite scheduling problem considered here are as follows:

- A single satellite orbits the Earth, and collects data, via an optical telescope, and energy via solar panels;
- The satellite's default mode is the IDLE mode, and the scheduling solution defines:
  - The amount of data that is acquired while in TASKED TRACKING mode, y;
  - The amount of data that is downlinked while in COMMUNICATION mode, q;
- The satellite energy collection rate depends on the Line-of-Sight of the solar panels relative to the Sun;
- The satellite energy consumption rate depends on its operational mode;
- The satellite data collection rate depends on its operational mode;
- The satellite data downlink rate depends on its operational mode;
- Energy is required to perform nominal operations of the spacecraft. By *"nominal operations"*, it is meant the tasks that are carried out in the IDLE mode of the satellite;
- Energy is required to carry out tasks outside those included in the nominal mode. These include *communication tasks* as well as *observation tasks*;
- The satellite has finite limits on the amount of energy and data that can be stored at any given time;
- There are multiple ground stations as part of the GSN;
- Each ground station periodically comes in- and out-of-view of the satellite. The period of time in which a ground station is visible is referred to as *communication window*;
- The duration of these communication windows depends on the geometry of the orbit of the satellite relative to the ground station;
- It can happen that more than one ground station is in view of the satellite at the same time. In this case, the satellite can download only to one ground station at the time;

- Each ground station is assumed to have the same characteristics, in terms of rate at which they can receive data, energy utilisation from the satellite to send one bit of data, and the efficiency of the data download. The latter includes the losses due to communication system and transmission inefficiencies;
- There are multiple targets to be observed by the satellite, these are referred to as Resident Space Objects (RSOs);
- Each RSO periodically comes in- and out-of-view of the satellite. The period of time in which a RSO is visible is referred to as *observation window*;
- The duration of these observation windows depends on the geometry of the orbit of the satellite relative to the RSOs;
- It can happen that more than one RSO is in view of the satellite at the same time. In this case, the satellite can only observe one target at the time;
- The satellite is capable of collecting tasked tracking data of any target RSO in view for the entire duration of an interval or any fraction thereof;
- The satellite is capable of downlinking data to any visible ground stations for the entire duration of an interval or any fraction thereof.
- The orbit of the satellite is assumed to be deterministic, and as such the communication windows to the GSN, and the observation windows of the RSOs are known a priori;
- The single satellite is equipped with magnetorquers, such that it can constantly de-saturate its ADCS actuator. With this assumption, the angular momentum accumulated by the satellite along its orbit does not need to be taken into account.

C.2. Mathematical Formulation

In this section, the scheduling problem is formulated as a Mixed-Integer Linear Programming (MILP) problem. Most of the notation used in this formulation is taken from the work of Spangelo et al. [50].

Parameters

The parameters used in the mathematical formulation are presented below. They follow the notation for which I is the set of intervals, O is the set of operational modes available and $O_i \subseteq O$ is the subset of the operational modes available during interval i, $\forall\, i \in I$. Using this notation results in the following parameters:

- $\eta_{i,o}$: efficiency of data download during interval i when downlinking using option o;
- t_i is the start time of interval i;
- t_{i+1} is the end time of interval i, which coincides with the start time of interval $i+1$;
- Δt_i is the duration of time interval i, and is equal to $t_{i+1} - t_i$;
- $\phi_{i,o}$ is the data rate associated with downloading during interval i using option o;
- $\alpha_{i,o}$ is the energy per data associated with downlinking using option o during interval i;
- $\kappa_{i,o}$ is the data rate associated with acquiring tasked tracking data during interval i using option o;
- $\gamma_{i,o}$ is the energy per data associated with acquiring tasked tracking data using option o during interval i;
- e_{min} and e_{max} are the minimum and maximum allowable amounts of energy that can be stored in the battery;
- e_{start} is the amount of energy stored in the battery at the beginning of the scheduling horizon;
- d_{min} and d_{max} are the minimum and maximum allowable data stored in the onboard memory;
- d_{start} is the amount of data stored in the onboard memory at the beginning of the scheduling horizon;
- δ_i^{e+} is the total amount of energy that can be acquired during interval i;
- δ_i^{e-} is the total amount of energy consumed during interval i for default operations (i.e. IDLE mode);

- δ_i^{d+} is the total amount of data acquired during interval i by default (=0);
- δ_i^{d-} is the total amount of data that is lost during interval i, independent of the operational mode. This includes for example data degradation and expiration.

Note that δ_i^{e-} includes only the energy required to operate the life-supporting subsystems onboard the spacecraft.

Variables

The variables used in the problem are the following:

- $x_{i,o} \in 0, 1$: binary variable indicating the operational mode during some or all of interval i;
- $q_{i,o} \in \mathbb{R}^+$ is the amount of data downloaded during interval i while in operational mode o, $\forall i \in I, o \in O$;
- $y_{i,o} \in \mathbb{R}^+$ is the amount of tasked tracking data acquired during interval i while in operational mode o, $\forall i \in I, o \in O$;
- $e_i \in \mathbb{R}^+$ is the amount of energy available at the beginning of interval i, measured in Joules, $\forall i \in I$;
- $d_i \in \mathbb{R}^+$ is the amount of data available at the beginning of interval i, measured in bits, $\forall i \in I$;
- $h_i^e \geq 0$ is the amount of excess energy spilled during interval i, measured in Joules, $\forall i \in I$;
- $h_i^d \geq 0$ is the amount of excess data spilled during interval i, measured in bits, $\forall i \in I$.

Formulation

Using the parameters and variables defined above, the optimisation problem can be now formulated mathematically.

Objective function:

$$\max \sum_{i \in I} \sum_{o \in O_i} \eta_{i,o} q_{i,o} \tag{C.1}$$

Subject to:

$$\sum_{o \in O_i} x_{io} \leq 1 \quad \forall i \in I \tag{C.2}$$

$$q_{i,o} \leq \Delta t_i \phi_{i,o} x_{i,o} \quad \forall i \in I, o \in O_i \tag{C.3}$$

$$y_{i,o} \leq \Delta t_i \kappa_{i,o} x_{i,o} \quad \forall i \in I, o \in O_i \tag{C.4}$$

$$e_0 = e_{start} \tag{C.5}$$

$$e_{min} \leq e_i \leq e_{max} \quad \forall i \in I \tag{C.6}$$

$$e_{i+1} = e_i + \delta_i^{e+} - \delta_i^{e-} - \sum_{o \in O_i} \alpha_{i,o} q_{i,o} - \sum_{o \in O_i} \gamma_{i,o} y_{i,o} - h_i^e \quad \forall i \in I \tag{C.7}$$

$$d_0 = d_{start} \tag{C.8}$$

$$d_{min} \leq d_i \leq d_{max} \quad \forall i \in I \tag{C.9}$$

$$d_{i+1} = d_i + \cancel{\delta_i^{d+}} - \delta_i^{d-} - \sum_{o \in O_i} q_{i,o} + \sum_{o \in O_i} y_{i,o} - h_i^d \quad \forall i \in I \tag{C.10}$$

$$x_{i,o} \in \{0, 1\} \quad \forall i \in I, o \in O_i \tag{C.11}$$

$$q_{i,o} \in \mathbb{R}^+ \quad \forall i \in I, o \in O_i \tag{C.12}$$

$$y_{i,o} \in \mathbb{R}^+ \quad \forall i \in I, o \in O_i \tag{C.13}$$

The new constraints can be explained as follows:

- Equation C.4 enforces that:
  - Tasked Tracking data can only be acquired if the tasked tracking option has been chosen in interval i;
  - The amount of tasked tracking data acquired is limited by the duration of the interval, Δt_i and the data acquisition rate, $\kappa_{i,o}$;
- Equation C.7 defines the amount energy at the beginning of an interval to be equal to:
  - The amount stored at the start of the previous interval (+)
  - The amount acquired via the solar panels (-)
  - The amount consumed for nominal operations (-)
  - The amount used to downlink data (-)
  - The amount used to acquire tasked tracking data (-)
  - The amount spilled
- Equation C.10 defines the amount of data at the beginning of an interval to be equal to:
  - The amount stored at the start of the previous interval (+)
  - The amount acquired during IDLE mode (=0) (-)
  - The amount lost due to degradation or expiration (-)
  - The amount downloaded (+)
  - The amount of tasked tracking data acquired (-)
  - The amount spilled
- Equation C.13 enforces the amount of acquired tasked tracking data to be a positive, real number.

D

Maple Worksheets

```
> with(VectorCalculus):
```

Quaternion Kinematic Equations --- \dot(q)_i , i = 1,2,3,4

```
> eq1 := 0.5·( w3·q2 + (−w2 + n0)·q3 + w1·q4)
```

$$eq1 := 0.5\, w3\, q2 + 0.5\, (-w2 + n0)\, q3 + 0.5\, w1\, q4 \quad (1)$$

```
> eq2 := 0.5(−w3·q1 + w1·q3 + q4·(w2 + n0))
```

$$eq2 := -0.5\, w3\, q1 + 0.5\, w1\, q3 + 0.5\, q4\, (w2 + n0) \quad (2)$$

```
> eq3 := 0.5((w2 − n0)·q1 − w1·q2 + w3·q4)
```

$$eq3 := 0.5\, (w2 - n0)\, q1 - 0.5\, w1\, q2 + 0.5\, w3\, q4 \quad (3)$$

```
> eq4 := −0.5·(w1·q1 + (w2 + n0)·q2 + w3·q3)
```

$$eq4 := -0.5\, w1\, q1 - 0.5\, (w2 + n0)\, q2 - 0.5\, w3\, q3 \quad (4)$$

Angular velocity dynamics Equations --- Due to the asymmetric moment of inertia distribution about the body-axes --- \dot{\omega}_i , i = x, y, z

```
> eq5 := kx·w2·w3
```

$$eq5 := kx\, w2\, w3 \quad (5)$$

```
> eq6 := ky·w1·w3
```

$$eq6 := ky\, w1\, w3 \quad (6)$$

```
> eq7 := kz·w1·w2
```

$$eq7 := kz\, w1\, w2 \quad (7)$$

Make a list with all the equations representing the dynamics of the system --- \dot{\vec{x}} , \vec{x} = state vector

```
> f := [eq1, eq2, eq3, eq4, eq5, eq6, eq7]
```

$$f := [0.5\, w3\, q2 + 0.5\, (-w2 + n0)\, q3 + 0.5\, w1\, q4,\ -0.5\, w3\, q1 + 0.5\, w1\, q3 + 0.5\, q4\, (w2 + n0),\ 0.5\, (w2 - n0)\, q1 - 0.5\, w1\, q2 + 0.5\, w3\, q4,\ -0.5\, w1\, q1 - 0.5\, (w2 + n0)\, q2 - 0.5\, w3\, q3,\ kx\, w2\, w3,\ ky\, w1\, w3,\ kz\, w1\, w2] \quad (8)$$

Make a list with the variables we want to take the Jacobian of --- in this case 'v' is the state vector \vec{x}

```
> v := [q1, q2, q3, q4, w1, w2, w3]
```

$$v := [q1, q2, q3, q4, w1, w2, w3] \quad (9)$$

Calculate the Jacobian of "f" w.r.t. "v"

```
> Jacobian(f, v)
```

$$[[0,\ 0.5\, w3,\ -0.5\, w2 + 0.5\, n0,\ 0.5\, w1,\ 0.5\, q4,\ -0.5\, q3,\ 0.5\, q2],\ [-0.5\, w3,\ 0,\ 0.5\, w1,\ 0.5\, w2 + 0.5\, n0,\ 0.5\, q3,\ 0.5\, q4,\ -0.5\, q1],\ [0.5\, w2 - 0.5\, n0,\ -0.5\, w1,\ 0,\ 0.5\, w3,\ -0.5\, q2,\ 0.5\, q1,\ 0.5\, q4], \quad (10)$$

$$\left[\begin{array}{l}
[\,-0.5\,w1,\ -0.5\,w2-0.5\,n0,\ -0.5\,w3,\ 0,\ -0.5\,q1,\ -0.5\,q2,\ -0.5\,q3\,],\\
[\,0,\ 0,\ 0,\ 0,\ 0,\ kx\,w3,\ kx\,w2\,],\\
[\,0,\ 0,\ 0,\ 0,\ ky\,w3,\ 0,\ ky\,w1\,],\\
[\,0,\ 0,\ 0,\ 0,\ kz\,w2,\ kz\,w1,\ 0\,]\,]
\end{array}\right]$$

list **(11)**

E

Model Selection - Extended Analysis

In this appendix, the complete procedure to select the different aspects of the model used for the discrete-time scheduler is presented.

In order to schedule the operations of a satellite, a model that represents the dynamics of the system has to be numerically integrated. There are different aspects that make-up such a model, and each of these can be chosen amongst a number of options. These aspects can be grouped into four groups, as follows:

- Choice of numerical integrator;
- Choice of representation;
- Choice of acceleration models;
- Choice of environment models;

To be able to make an informed choice on the above aspects, metrics have to be used to compare the various options. Two metrics that can be used are the *accuracy* of the model and the *computational time* spent to output a solution.

The accuracy of the model is composed of the physical model accuracy, and the numerical model accuracy. The former refers to the degree of fidelity of the solution with respect to the real physical system. The latter refers to the approximations made in the numerical solution due to numerical integration. To guarantee that the outcomes are influenced primarily by the precision of the physical model rather than the approximations inherent in the numerical solution, the accuracy of the numerical model should be approximately one order of magnitude higher than that of the physical model. In order to compare models when making choices, a benchmark model is needed. A benchmark model is a model which is sufficiently accurate, and can thus be used to check the performance of different models. To ensure the “sufficient accuracy”, the benchmark accuracy must be 2 or 3 orders of magnitude stricter than for the numerical model.

The accuracy of the physical model can be calculated based on the required degree of fidelity desired in modelling the operations of the spacecraft. In the use-case considered in this research, the communication window between a satellite in LEO and a ground station can be used as an indicator of accuracy.

E.1. Position-Related Requirement

The duration of a communication window between a satellite and a ground station is an important parameter for satellite task scheduling. This duration depends on the geometry of the problem, and is calculated based on the position of the satellite and that of the ground station. The degree of accuracy of these two positions thus directly effects the accuracy of the duration of the communication window.

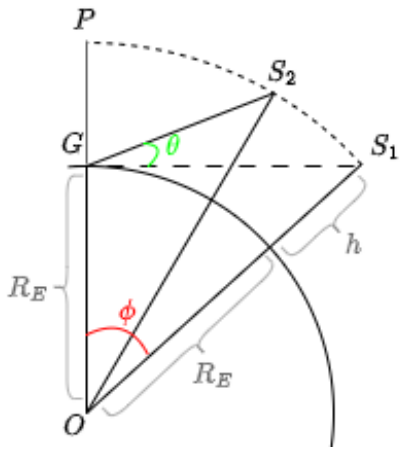


Figure E.1: Simplified geometry for calculating the communication window between the observer satellite and a ground station

Thus, by imposing an accuracy requirement on the communication window, we can determine the required accuracy on the propagated position.

Different sets of observer satellite orbits will be investigated, with varying altitudes. When determining the requirement on the propagated position, it is logical to choose the case in which an error in the position has the biggest effect on the communication window duration. This is the case for the lowest orbit considered, since the lower the orbit, the shorter the communication windows. As a consequence, an error in the propagated position results in a larger relative error in the communication window duration. Thus, an altitude of 500 km will be used for the use-case.

For simplicity, in the following derivation of the position-accuracy requirement, the following is assumed:

- Spherical Earth;
- Perfectly circular orbit;
- Satellite passes exactly over the ground station;
- Spacecraft orbit altitude is equal to 500 km.

The 2D geometry resulting from the assumptions mentioned above is depicted in Figure E.1. In this figure, O is the center of the Earth. G is the location of the ground station on the Earth's surface. S_1 is the position of the spacecraft along its orbit (the arc $P - S_2 - S_1$) when it enters the local horizon as seen from the ground station. S_2 is the position of the spacecraft when it is at an elevation angle θ, as seen from the ground station. P is the azimuth point with respect to the ground station. R_E is the radius of the Earth, h is the altitude of the spacecraft above Earth's surface.

The triangle $\triangle GOS_1$ is a right triangle. From this, the angle $\angle GOS_1$ can be calculated as follows:

$$\phi = \cos^{-1}\left(\frac{R_E}{R_E + h}\right) \tag{E.1}$$

which for an altitude of 500 km gives an approximate value of $22°$.

Considering the satellite in a perfectly circular orbit, the amount of kilometers along the orbit that correspond to an rotation around the Earth of 1° can be calculated as follows:

$$\frac{2\pi(R_E + h)}{360°} \tag{E.2}$$

where $2\pi(R_E + h)$ is the circumference of the satellite's orbit, thus the total distance travelled in one revolution around the Earth. For an altitude of 500 km, this gives a value of 120 km/°.

The arc that needs to be travelled to span half of the local horizon as seen from the ground station, $\widehat{PS_1}$, can be calculated as follows:

$$\frac{\ell}{2} \approx 22° \cdot 120 \text{ km}/° \approx 2640\ km \tag{E.3}$$

Table E.1: Accuracy requirements for the various models

Requirement	Physical Model	Numerical Model	Benchmark	Unit
Position Accuracy	15,000	1500	1.5	m

The equation for the circular velocity of an object around a celestial body, using the gravitational parameter of the Earth, and 500 km as altitude, can be used to approximate the velocity of the satellite, as follows:

$$v_{circular} = \sqrt{\frac{\mu}{r}} \approx \sqrt{\frac{3.986004418{\times}10^{14}\ m^3/s^2}{6878\ m}} \approx 7.6\ km/s \tag{E.4}$$

The time spent by the spacecraft to cross the half of the local horizon as seen from the ground station can be calculated as follows:

$$\frac{\Delta t}{2} \approx \frac{\ell/2}{v_{circular}} \approx \frac{2640\ km}{7.6\ km/s} \approx 350\ s \tag{E.5}$$

It is now assumed that the 180° of local horizon seen from the ground station are spanned by the satellite at a constant rate, giving the following rates:

$$\frac{2640\ km}{90^\circ} \approx 29.3\ km/^\circ \tag{E.6}$$

$$\frac{350\ s}{90^\circ} \approx 3.9\ s/^\circ \tag{E.7}$$

Which means that the satellite travels 29.3 km to move of 1° as seen from an observer standing at the ground station on the Earth's surface, and it takes about 4 seconds to do so.

A common value for the minimum elevation angle of ground station is 10° (CITATION NEEDED), which means that the satellite spans $180^\circ - 2 \cdot 10^\circ = 160^\circ$ of the local horizon while being inside the communication window. Using the value calculated above, this results in $160^\circ \cdot 3.9\ s/^\circ \approx 622\ s$, which is the total communication window time for the simplified case considered.

If the duration of the communication window wants to be captured at an accuracy of ≈ 1%, then the duration needs to be captured with an accuracy of $\approx 6\ s$. Since the satellite travels at $7.6\ km/s$, this results in a position accuracy of $\approx 45\ km$. Following this reasoning, the position accuracy requirement should be of approximately $45\ km$.

The above discussion was made for an idealised case, in which the satellite passes exactly over the ground station, resulting in a communication pass longer than 10 minutes. In reality, communication passes for LEO spacecrafts are shorter, due to the geometry of the satellite and ground stations. A minimum duration constraint can be imposed, in order to only consider communication passes of a certain duration, and discard the shorter ones. The reason for this is that scheduling a communication task only for few seconds is not beneficial, as it would mean changing frequently operational mode. A typical value for this minimum duration threshold is of 3 minutes, or 180 seconds (CITATION NEEDED). Following the same reasoning as above, to stay within the 1% of this communication window, an accuracy of $\approx 2\ s$ needs to be achieved. Using the circular velocity of a satellite at 500 km, the following position accuracy is obtained: $2\ s \cdot 7.6\ km/s \approx 15\ km$. This can now be used as the required accuracy in propagated distance for the physical model. From this, the requirement on the numerical model and benchmark model can be determined, resulting in Table E.1. Note that the requirement on the position accuracy is to be taken at the end of the propagation, and thus depends on the simulation-time chosen for the test-case.

E.1.1. Time-Related Requirement

On top of the position accuracy requirement derived above, the accuracy of the simulation is constrained by the fact that the communication windows have to be capture with a minimum desired accuracy.

For example, if the desired position accuracy, shown in Table E.1, is achieved by setting the numerical integration time-step size to 30 seconds, then a communication pass of 180 seconds duration could potentially be mismodelled by up to $2 \cdot 30/180 = 33.3\%$. This is not desirable, as in this way the operations of the satellite cannot be optimally scheduled, which is the final goal of this work. For this reason, a minimum threshold on the step-size to be used for the numerical integration is imposed to be 5% of a communication pass of 180 seconds, thus:

$$\Delta t \geq 9s$$

CPU Run Time

The CPU run time is an important metric for optimisation problems, as too long computational times can result in exploring a small portion of the desired design space. For this reason, the CPU run time will be used as a metric when choosing between the different integrators, representations, acceleration and environment models.

E.2. Test-Case

The test-case that will be used to compare the different models is chosen to be representative of the mission considered, and has the following characteristics:

- Single Satellite
- Altitude = 500 km
- Inclination = 97.8 $^\circ$ (SSO)
- Eccentricity = 0
- Simulation time = 7 days

A simulation time of 7 days is slightly longer compared to typical mission scheduling horizon, which is usually of a few days (CITATION NEEDED). This was chosen deliberately, to ensure the accuracy of the solution is within the desired range.

E.3. Benchmark Selection

In this section, a benchmark solution has to be selected, in order to later be able to assess the performance of the different integrators and propagators, as no analytical solution is available for the case-study. The accuracy requirement for the benchmark is shown in Table E.1. For the benchmark, the integrator RKF7(8), with fixed time-step, will be used. To find the time-step size that satisfies the accuracy requirement, different simulations are performed, each with a time-step double that of the previous run. The result can be seen in Figure E.2 and Figure E.3, which show results of the same simulation runs. The former shows the final position error increasing as the time-step increases, while the latter shows the position error over time for all simulations. It can be seen that the position error progressively decreases by 10^2 when going from Δt of 256 s to Δt of 32 s. It then keeps decreasing down to Δt of 4 s, and shows chaotic behaviour for Δt of 2 s, as the position error increases again. From this analysis, it can be seen that most of the the time-steps sizes tested satisfy the benchmark accuracy requirement, and that a $\Delta t \leq 64\ s$ needs to be selected for the benchmark.

A change in trend was noted in Figure E.2 as the time-step increases. Thus, it was decided to further inspect the behaviour of the errors for Δt of 4, 8, 16 and 32 seconds. The result is reported in Figure E.4 and Figure E.5. Note that a sample accuracy requirement of 1 mm was plotted, to give reference of the scale of the errors, as plotting a horizontal line at 1.5 m would make the scale of the y-axis not easy to interpret. From these results, it can be seen that the error shows the desired behaviour, with the final position difference plot linearly increasing when setting both x- and y-axes scales to *logarithmic*.

The computational times of each run was then inspected, to make a good selection of a benchmark that guarantees a reasonable solve time. Based on the result obtained, ti was decided to use the following as a benchmark: RKF7(8) fixed-step integrator, with $\Delta t = 4\ s$.

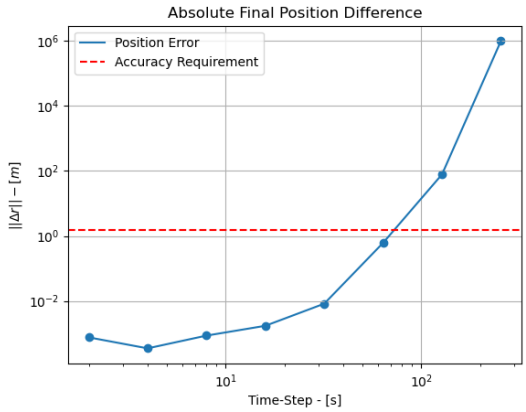


Figure E.2: Final Position Error vs Time-Steps.

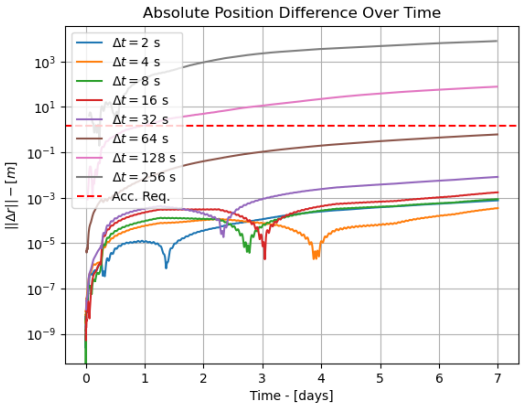


Figure E.3: Position Error Over Time for various Time-Steps

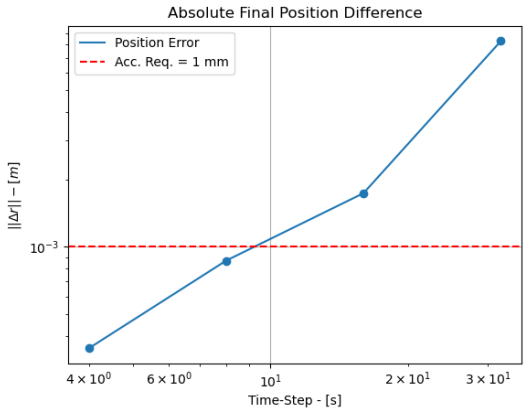


Figure E.4: Final Position Error vs Time-Steps - Zoomed

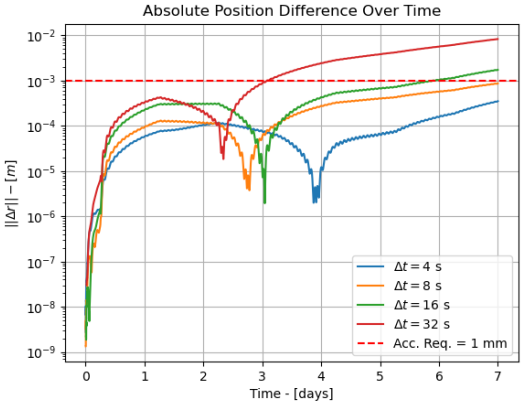


Figure E.5: Position Error Over Time for various Time-Steps - Zoomed

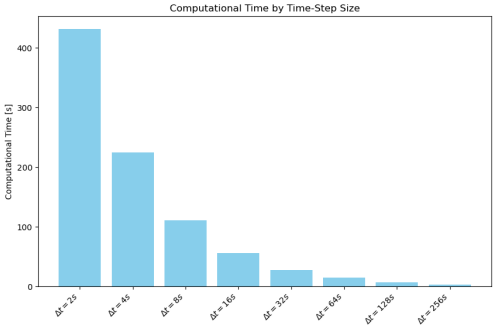


Figure E.6: Computational Time for different benchmark runs

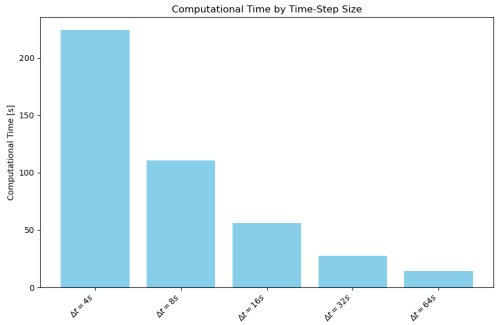


Figure E.7: Computational Time for different benchmark runs - zoomed

Table E.2: Different Integrators Tested

Type of Integrator	Name of Integrator	Symbol
Single-Stage	Runge-Kutta	RK4
Multi-Stage	Runge-Kutta-Fehlberg (RKF)	RKF4(5) - RKF5(6) - RKF7(8)
	Runge-Kutta-Dormand-Prince (RKDP)	RKDP8(7)
Multi-Step	Adam-Bashforth-Moulton	ABM
Extrapolation	Bulrisch-Stoer	BS

E.4. Integrator Choice

Once chosen a benchmark model, different integrators can be tested, to check which one offers the best performance in terms of computational time, while satisfying the required position accuracy, which is of 1500 m for the numerical model.

Variable step-size integrators were not considered for this use-case. The reason for this is that the orbit considered is a circular, sun-synchronous orbit. For such an orbit, the dynamics of the system stay constant throughout the simulation time, and thus there is no need to change the step-size dynamically, to accommodate for change in the dynamics of the system.

Thus, fixed-step size integrators will be compared. The step-size that will be used to compare the integrators is of 4 seconds, as from the benchmark selection, this was identified as a step-size that yields accurate results in acceptable computational times. The integrators that will be tested are indicated in Table E.2. Note that for ABM, the following settings were used: relative and absolute tolerance of 1e-12, minimum order of 6, maximum order of 11. For BS, the maximum number of steps was set to 6.

The performance of the different integrators in terms of position accuracy and computational time can be seen in Figures E.8 to E.11. From the first two Figures, it can be seen that the BS integrator achieves very high precision, but for a computational time that is one order of magnitude higher than all the other integrators. This means that the step size could be increased by a lot, keeping the error within the desired accuracy. Due to the constraint imposed by scheduling aspects, having very large time-steps is not desirable, and thus this is discarded.

Looking at the next two figures, it can be seen that the integrators that show the lowest computational time are RK4 and ABM. RK7(8) and RKDP8(7) perform almost identically, both in terms of accuracy and of computational time. RK4(5) and RK5(6) present mid-performances for computational time, as well as position difference.

When checking the position difference, it can be seen that RK4 is by far the largest, with one order of magnitude higher than the second least accurate one, which is RK4(5), followed by RK5(6) and ABM. With these results, the following reasoning was done:

- RK4 has x10 more error than RKF4(5), while having 1/2 of its computational time $\rightarrow$ RKF4(5) better;
- Improvement in accuracy when using higher order RKF methods is not needed, and comes at the cost of higher computational time $\rightarrow$ RKF4(5) is the best multi-stage integrator.
- ABM has smaller position error and smaller computational time of all $\rightarrow$ ABM looks promising

The reason for the good behaviour of the ABM can be found in the fact that multi-step methods use previous functions evaluations to calculate the next step. This means a rapid solve time for stable systems. In multi-stage methods, instead, function evaluations are discarded after a step, which means that no particular benefit is found for stable systems.

E.4.1. ABM vs RKF4(5)

The two best performing integrators will now be compared, with different time-step sizes, to see which one is more suited for this use case. As it can be seen from the position difference over time, for low values of Δt, the ABM integrator starts to show erratic behaviour, while the RKF4(5) shows a

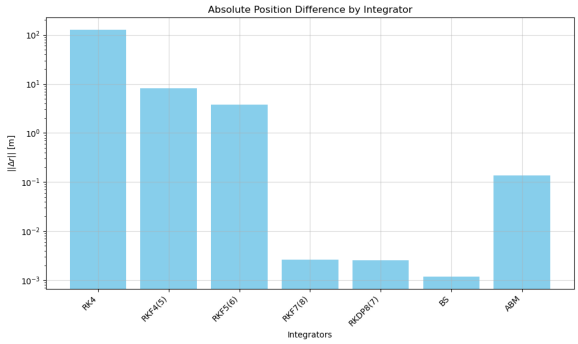


Figure E.8: Final Position Error for Different Integrators

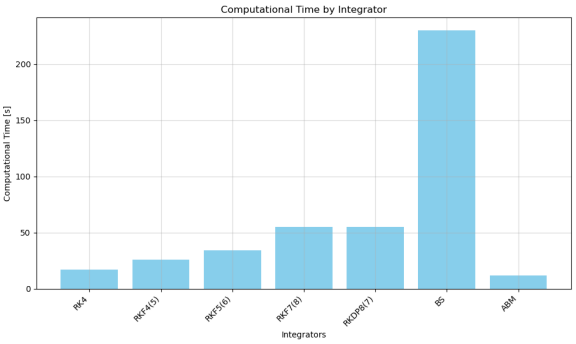


Figure E.9: Computational Time for Different Integrators

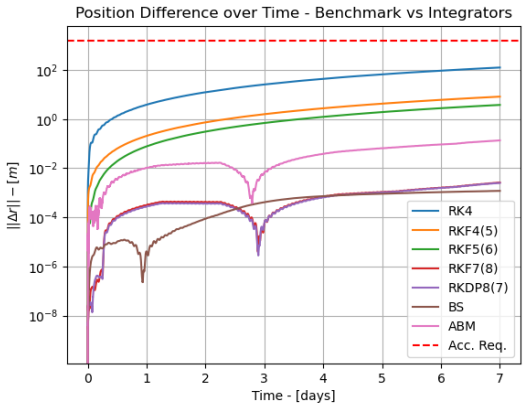


Figure E.10: Position Difference over Time for Different Integrators

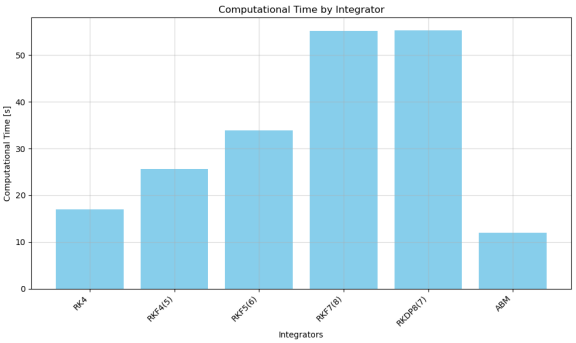


Figure E.11: Computational Time for Different Integrators without BS

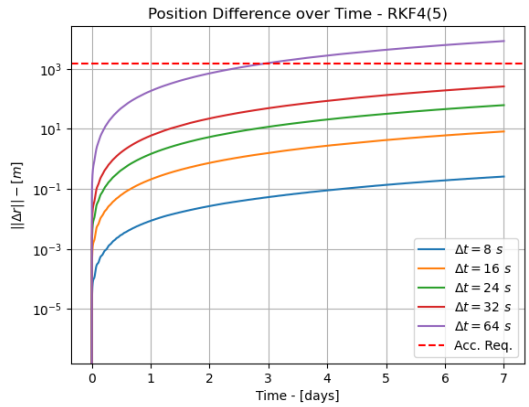


Figure E.12: Position Difference over Time for RKF4(5) with different time-step-sizes

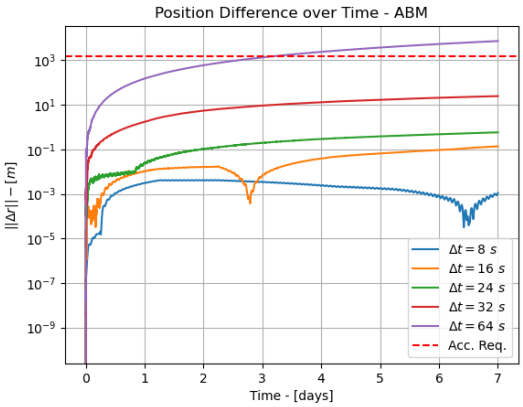


Figure E.13: Position Difference over Time for ABM with different time-step-sizes

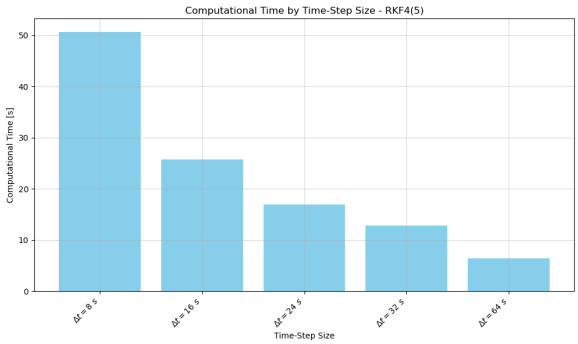


Figure E.14: Computational Time for RKF4(5) with different time-step-sizes

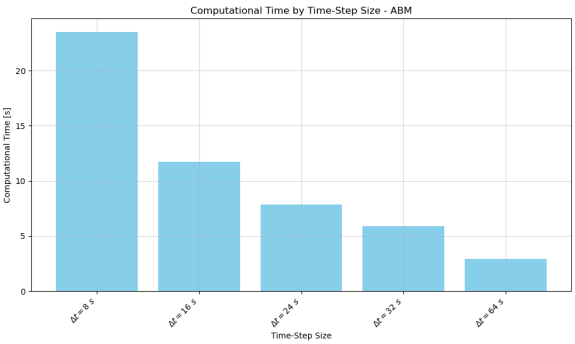


Figure E.15: Computational Time for ABM with different time-step-sizes

very good error behaviour down to a Δt of 8 seconds. It can be seen that the ABM integrator has a computational time x2 smaller than RKF4(5). Since for this work no iterative algorithms are used, and the computational performance of the integrator is not of paramount importance, the good error behaviour of the RKF4(5) integrator is preferred over the computational efficiency of the ABM. In case an iterative algorithm is used to tackle this problem, the ABM integrator would be a good choice.

E.5. Representation Choice

After a selection on which integrator to use has been made, it needs to be decided what representation of the state to use. The following representations are available in tudat:

- Unified State Models (USM);
- Cowell;
- Enke;
- Modified Equinoctial Elements (MEE);
- Kepler;

The results for the Kepler representation was not included, as it presented a singularity. The MEE representation showed erratic behaviour, and was thus also discarded. The computational time and position accuracy behaviour of the remaining representations were then compared.

The computational time was almost identical for all representations, as it can be seen from Figure E.17.

Concerning the position accuracy, the Encke propagator has an error 10 times larger than that of the

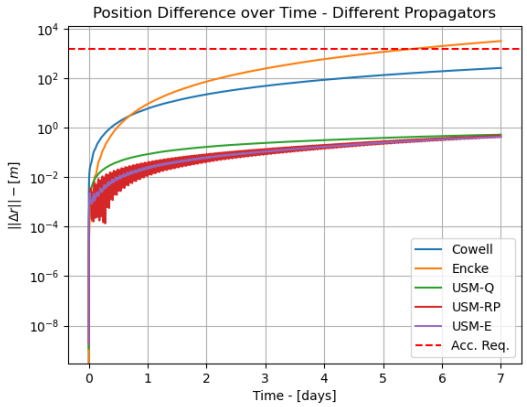


Figure E.16: Position Error over Time for different state representation.

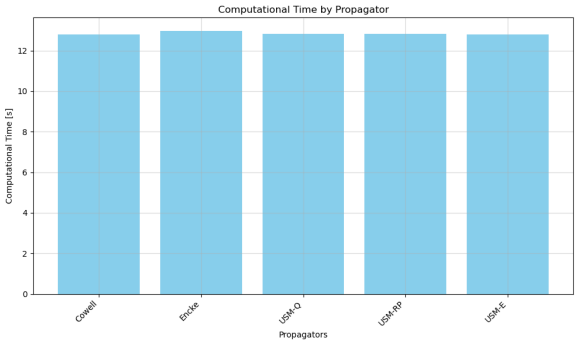


Figure E.17: Computational Time for Different Representations

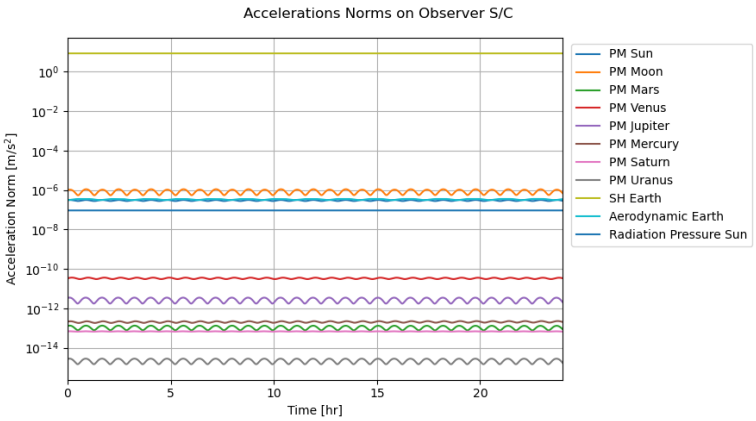


Figure E.18: Acceleration Norms of Accelerations on the Observer Satellite

Cowell propagator. Unified State Models representations show a good error behaviour, with a smaller error with respect to Cowell. Despite this, since the Cowell propagator satisfies the accuracy requirement, and is a more standard representation for astrodynamics problems, it was chosen to use it as a representation for the state of the observer satellite in this use-case.

As a result of the discussion made in this and the previous section, the integrator and representation chosen are: RKF4(5) integrator with time-step of 16 seconds, using Cowell representation.

E.6. Acceleration Selection

After selecting an integrator and propagator, it can be decided which accelerations to consider for the simulation. Based on the discussion made above, the accuracy requirement on the physical model is of 15,000 m.

The first step in selecting the acceleration is to include many accelerations that are readily available in tudat, and comparing their norms. This is shown in Figure E.18. It can be seen that the highest contribution is given by the Earth gravity, which makes sense since Earth is the central body of the orbit. The gravitational attraction of the Sun and the Moon and the aerodynamic drag of Earth atmosphere follow, with the same order of magnitude for their acceleration norms, at about $10^{-6}\ m/s^2$, followed shortly by the Solar Radiation Pressure (SRP) exerted by the Sun on the satellite. Then follows the accelerations due to the gravitational attraction of other bodies in the solar system.

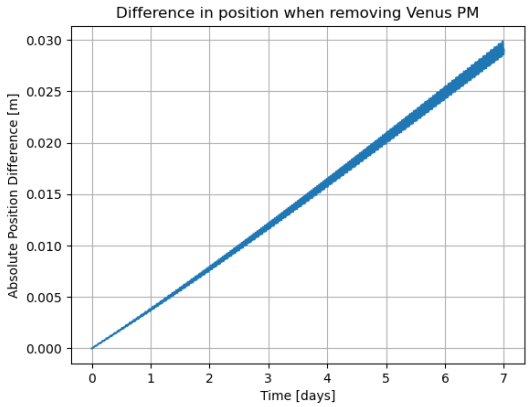


Figure E.19: Position Difference when removing Point Mass gravity acceleration from Venus

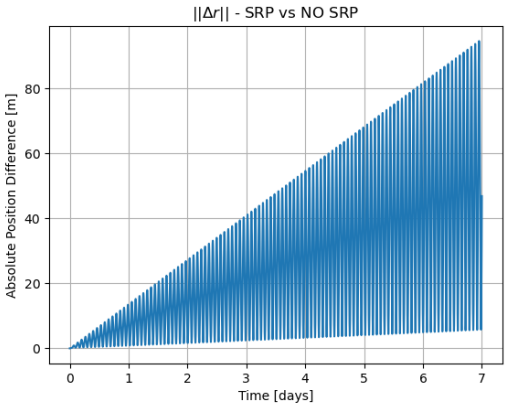


Figure E.20: Position Difference when removing the Solar Radiation Pressure acceleration. Reflectivity coefficient = 2.2

E.6.1. Venus Point Mass Gravity

Next, the effect on the position accuracy of removing the highest Point Mass (PM) gravity acceleration from an outer planet, which is that of Venus in this case, was investigated. This is shown in Figure E.19, which shows that the position error stays below 3 cm after 7 days of propagation. From this it can be inferred that all the accelerations that have a norm smaller than that of the Venus PM gravity pull can be disregarded.

E.6.2. SRP

Following this approach, the next acceleration to turn on and off is the SRP. This was done for a reflectivity coefficient of 2.0, to consider a case in which this acceleration should play an important role. The result is shown in Figure E.20, which shows how the position difference stays below 100 m after 7 days of propagation. From this result, it is safe to disregard the SRP.

E.6.3. Aerodynamic Drag

The next acceleration in terms of norm is that due to aerodynamic drag of Earth's atmosphere. Removing the aerodynamic drag results in a position error up to 100 km for an orbital altitude of 500 km, and of 600 m for an altitude of 1000 km, as shown in Figure E.21. This shows that the aerodynamic drag has too big of an impact on the position accuracy for some of the altitudes considered in this thesis, and thus needs to be considered.

E.6.4. Luni-Solar Perturbation

Since the acceleration norms resulting from the gravitational attraction of the Sun and the Moon on the observer satellite are higher than that of the aerodynamic drag, it can be concluded that removing these accelerations would result in a position error of more than 100 km, and that these needs to be included.

E.6.5. Earth Spherical Harmonics

Now, the effect of including more or less coefficients for the spherical harmonics of the Earth's gravity field has to be analysed. The idea behind this analysis is as follows: starting from a high degree and order (D/O) for the Spherical Harmonics coefficients, i.e. (100,100), the latter are progressively reduced. While doing so, the position error and computational time are recorded. As long as the position error stays below the position accuracy requirement of the physical model, which is of 15,000 m, the spherical harmonics coefficients can be reduced, without compromising the fidelity of the model.

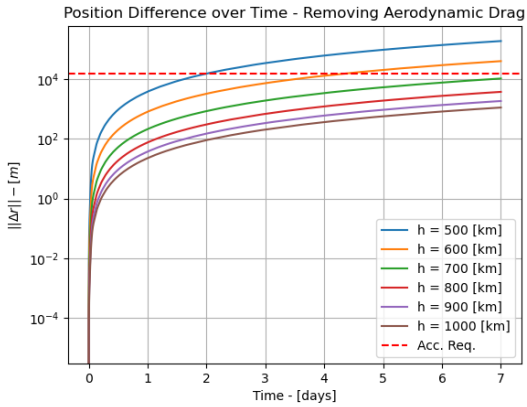


Figure E.21: Position Difference Over Time when removing the effect of aerodynamic drag. Altitudes going from 500 km to 1000 km, every 100 km.

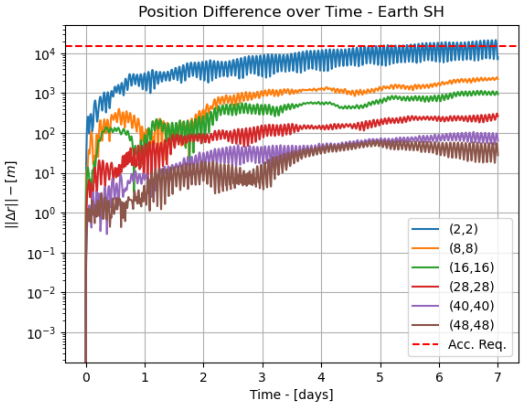


Figure E.22: Position error for increasing Spherical Harmonics Degrees and Order

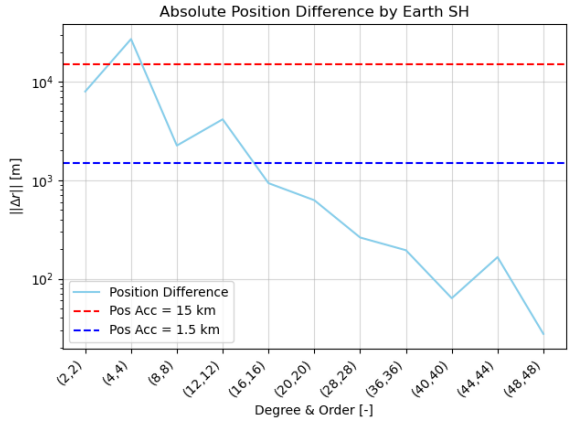


Figure E.23: Final Position Error for increasing Spherical Harmonics D/O

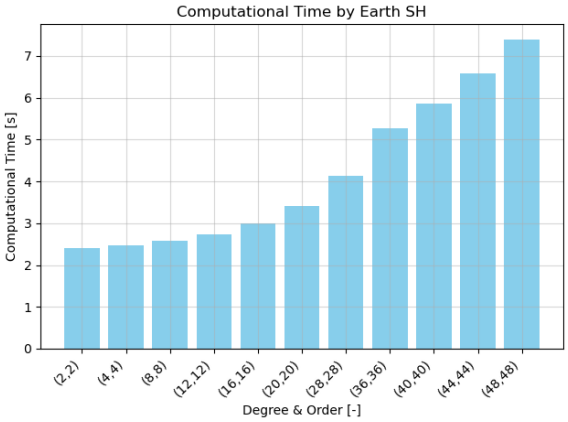


Figure E.24: Computational Time for increasing Spherical Harmonics D/O

When plotting the position error over time of simulation runs that use Earth spherical harmonics of degree and order smaller than (100,100), the result shown in Figure E.22 is obtained. From this graph two things can be noted: 1) the accuracy requirement is met by all the D/O considered, expect for (2,2) that oscillates around the threshold; 2) the accuracy increases as the D/O are increased.

The result can be better visualised by looking at Figures E.23 and E.24, which respectively depict the final position difference and computational time for different degrees and orders, going from (2,2) until (48,48). From the results shown above, it was chosen to use a D/O of (16,16), as it results in 25% more computational time than for (2,2), but reduces the error by an order of magnitude.

E.7. Environment Selection

Next, the influence of using different models on the position accuracy and computational time has to be investigated. Due to the nature of this problem, the atmospheric model is the one that is expected to have the biggest impact.

Atmospheric Model Considered

In tudat, there are different atmospheric models that can be used. These have an impact on the accuracy of the simulation, and on its run time.

Table E.3: Exponential Atmosphere Model Properties

Property	Value	Unit
Scale Height	7.2	km
Density at Zero Altitude	1.225	kg/m
Constant Temperature	246.0	K
Specific Gas Constant	287.0	J/kg/K
Ratio of Specific Heats	1.4	-

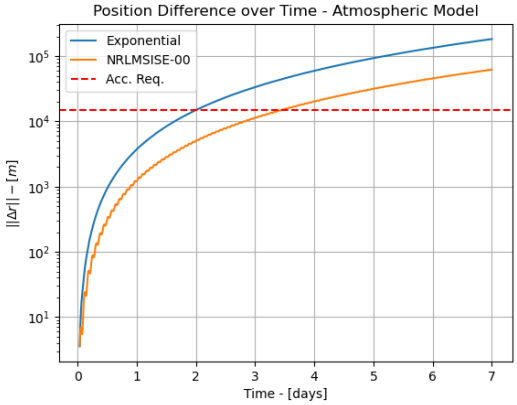


Figure E.25: Position Difference over time for different atmospheric models

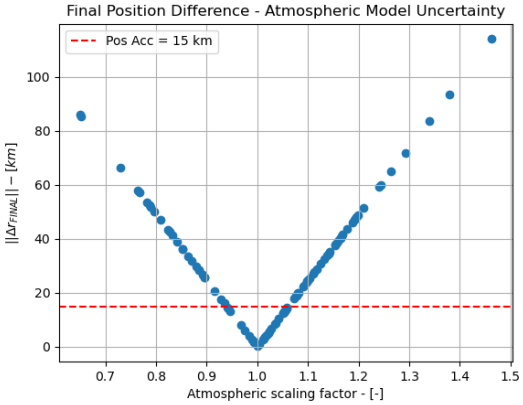


Figure E.26: Difference in final position for deviations due to NRLMSISE-00 atmospheric model uncertainty

The options available in tudat for the atmospheric models are: 1) US76, 2) Exponential Atmosphere, 3) NRLMSISE-00. Exponential atmosphere is expected to be more computationally efficient, but less accurate. The properties of the exponential atmosphere were set as indicated in Table E.3. NRLMSISE-00 is a model that includes seasonal and spatial variations of Earth's atmosphere, and it is thus expected to be more accurate but also more computationally intensive. US76 is a static atmospheric model, providing a fixed representation of how the Earth's atmospheric properties such as pressure, temperature, density, and viscosity vary across a range of altitudes.

From Figure E.25 it can be seen that the position error when changing atmospheric model is in the order of 100 km, which is above the required position accuracy for the physical model. Comparing the computational time, instead, results in the NRLMSISE-00 being approximately 10% slower than the exponential model and the US76, which perform almost identically. In the next section, a discussion on the uncertainty of the atmospheric model is made, before making a selection.

Uncertainty on Atmospheric Model

The NRLMSISE-00 model is the most accurate model available, and has an uncertainty of approximately 15%. Assume that this uncertainty is normally distributed. Thus, $\sigma = 15\%$, $\mu = 0$. To investigate the effect of this uncertainty, 100 simulation runs were performed, each using a different atmospheric scaling factor. The latter followed the normal distribution described above ($\mu = 0, \sigma = 15\%$). The effect that this has on the final position after 7 days of propagation can be seen in Figure E.26. The plot was produced using RKF4(5), with fixed-step-size of 32 seconds. It can be seen that for most of the runs that take into account the uncertainty of the atmospheric model, the difference in final position is above the threshold of 15 km. This means that, even when using the most accurate model, its uncertainties have an effect on the result higher than the accepted threshold. As a result of this, an atmospheric model with lower fidelity can be used, such as an exponential one or the US-76, as the result might still have a position error higher than the desired threshold, but for a smaller computational time. For this reason, it was chosen to use the default atmospheric model implemented in tudat, the US-76 model. This information will be relevant when validating the result of the discrete-time scheduler, by comparing it with GMAT.

F

Continuous-time Overview Plots

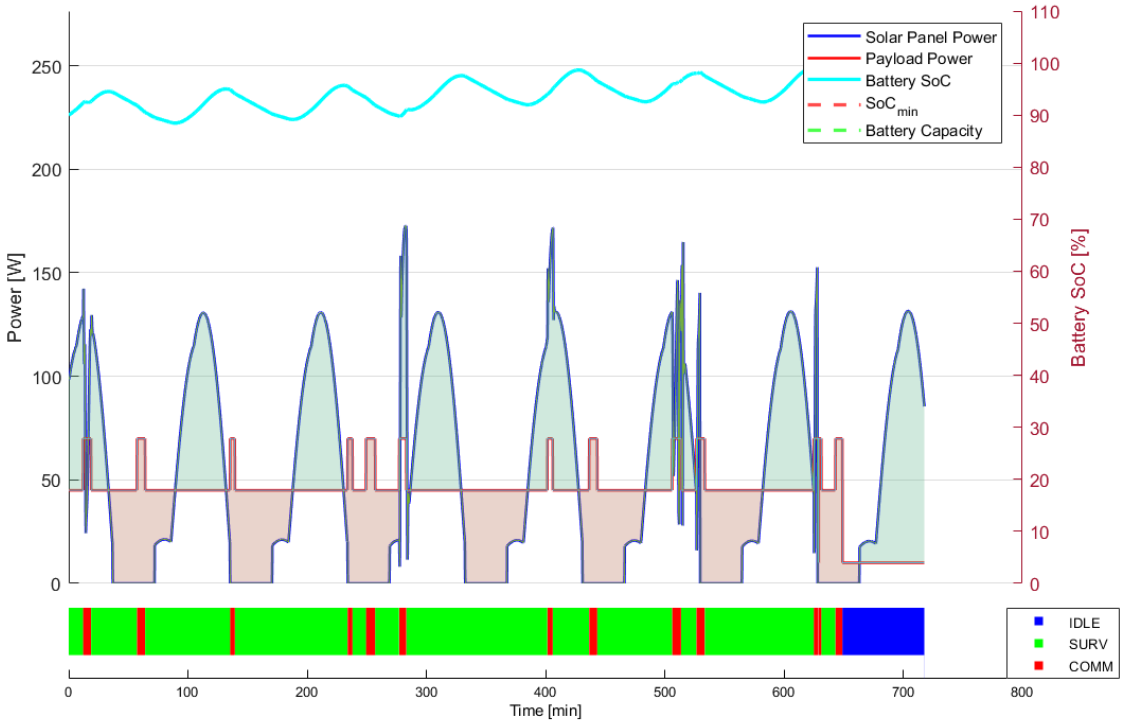


Figure F.1: Energy overview of a scheduling solution for an orbit-following surveillance pointing direction.

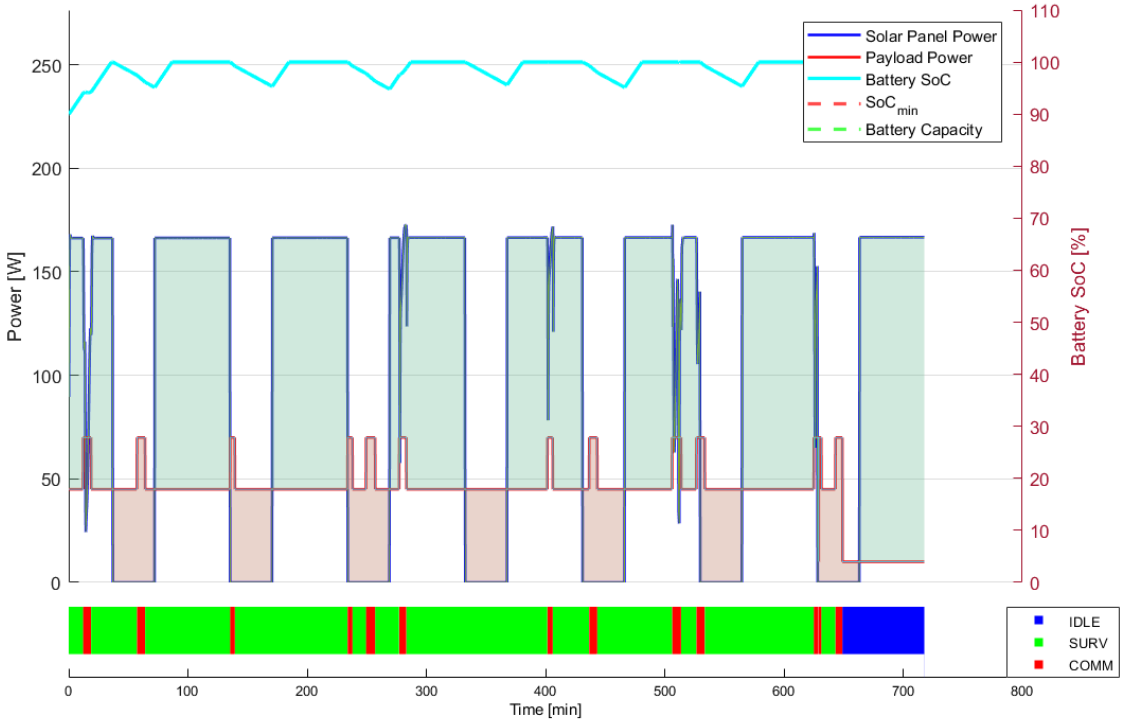


Figure F.2: Energy overview of a scheduling solution for a surveillance pointing direction away from the Sun.

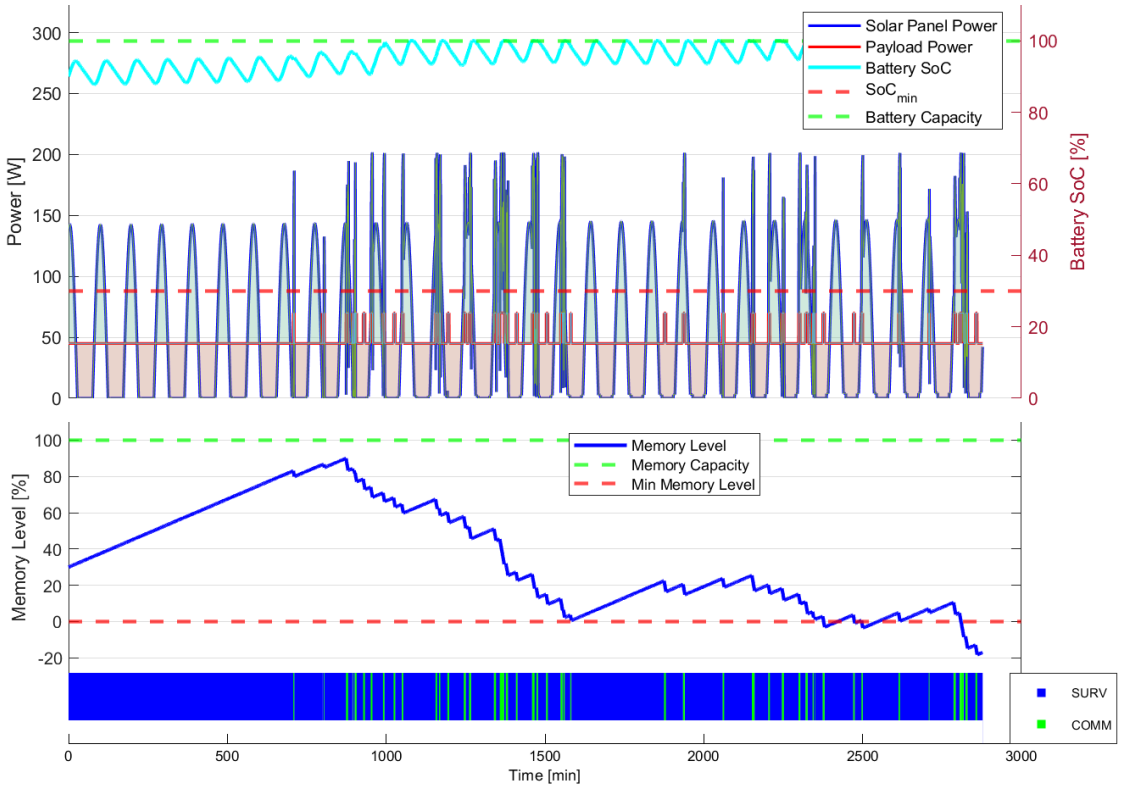


Figure F.3: Continuous-time solution overview for SURV-COMM framework. Nominal inputs, except scheduling horizon which is set to 48 hours.

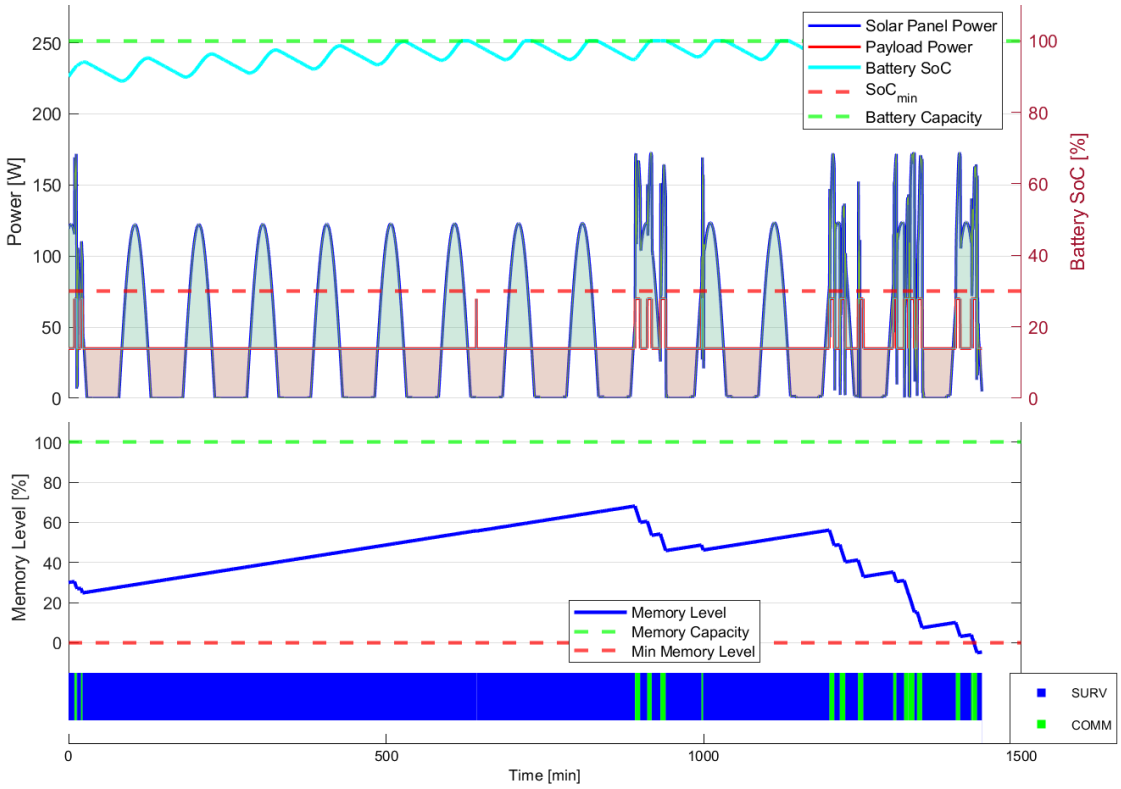


Figure F.4: Continuous-time simulator overview plot - PL 1 - solar panel size = 0.3m^2

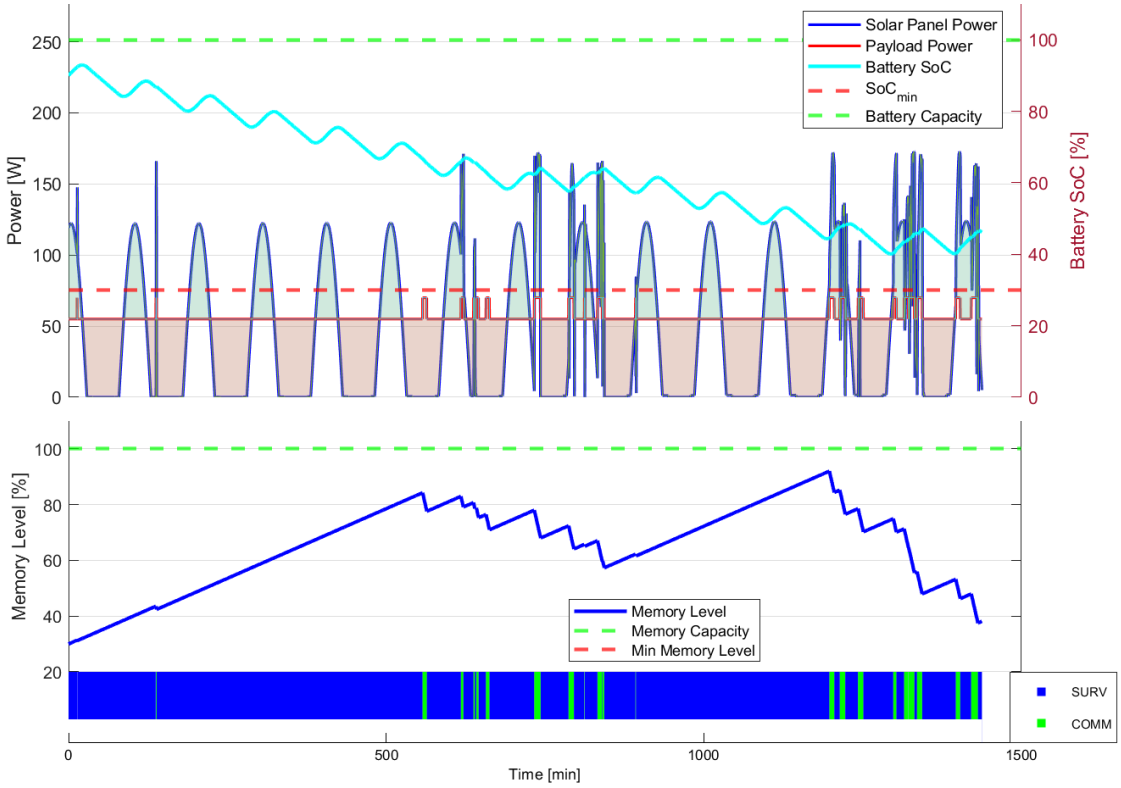


Figure F.5: Continuous-time simulator overview plot - PL 3 - solar panel size = 0.3m^2

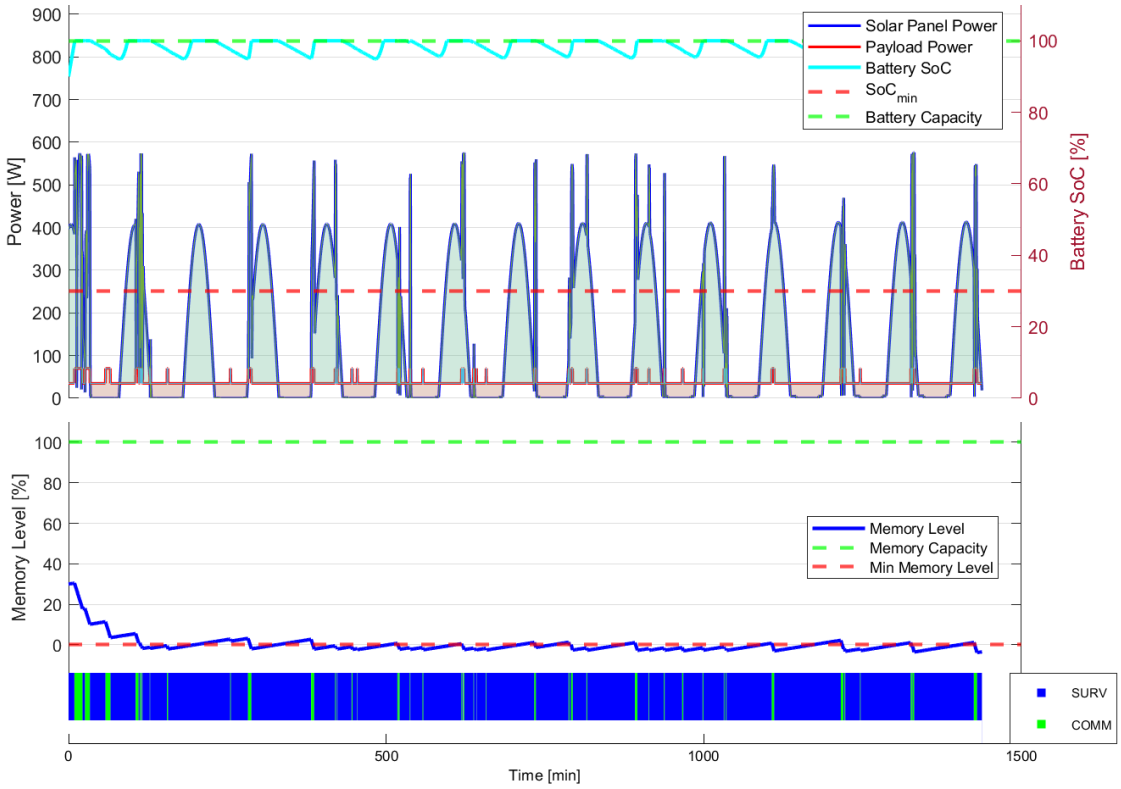


Figure F.6: Continuous-time simulator overview plot - PL 1 - solar panel size = 1.0m^2

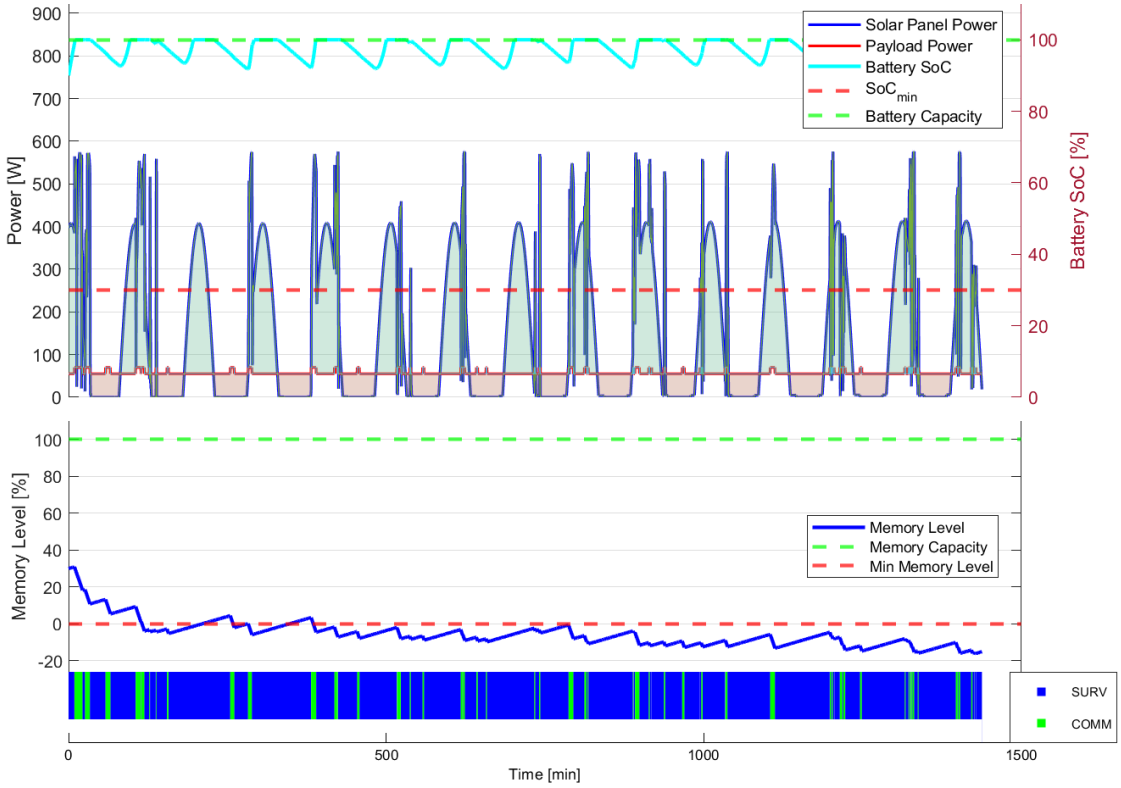


Figure F.7: Continuous-time simulator overview plot - PL 3 - solar panel size = 1.0m^2