Optimal Positioning of the Actuators of a Satellite in Low Earth Orbits

Diogo Melo e Silva

Dissertação para obtenção do Grau de Mestre em

Engenharia Aeronáutica

(Mestrado Integrado)

Orientadores: Professor Doutor Kouamana Bousson,

Doutor Pedro Filipe Neves Ferreira Lobão da Cruz

junho de 2025

Declaração de Integridade

Eu, Diogo Melo e Silva, que abaixo assino, estudante com o número de inscrição 44099 do Mestrado de Engenharia Aeronáutica da Faculdade de Engenharia, declaro ter desenvolvido o presente trabalho e elaborado o presente texto em total consonância com o **Código de Integridade da Universidade da Beira Interior.**

Mais concretamente afirmo não ter incorrido em qualquer das variedades de Fraude Académica, e que aqui declaro conhecer, que em particular atendi à exigida referenciação de frases, extratos, imagens e outras formas de trabalho intelectual, e assumindo assim na íntegra as responsabilidades da autoria.

Universidade da Beira Interior, Covilhã 09/06/2025

Assinado por: **DIOGO MELO E SILVA**
Num. de Identificação: 14326408
Data: 2025.06.09 05:37:04+01'00'

Dedicated to my family and friends.

Acknowledgments

I would like to express my sincere gratitude to Dr. Pedro Cruz and Dr. Kouamana Bousson for their guidance and support throughout all phases of this dissertation. The development and implementation of orbital propagation models, reference quaternion calculation, and particle swarm optimization techniques posed significant challenges, both mathematically and programmatically. Overcoming these obstacles provided me with technical and analytical skills that I might not have acquired through a different type of research project.

I am also deeply thankful to my father, mother, brother, and paternal grandparents for their unwavering academic and financial support throughout my years at the University of Beira Interior. Their encouragement in my potential made it possible for me to pursue my studies and ultimately fulfill my dream of becoming an Aeronautical Engineer. I consider myself fortunate to have had them by my side, guiding and supporting me every step of the way.

Moreover, I extend my heartfelt appreciation to all the friends I was privileged to meet during this master's degree. The long nights of study that stretched into morning, the joyful moments we shared in the snow city of Covilhã, and the stories and experiences we accumulated over the years are memories I will never forget.

A special acknowledgment is reserved for my friends from the "S Índice" group, Hugo Gonçalves, João Barateiro, João Charuto, and Nuno Souza, for their companionship, support, and the countless humorous and unforgettable moments we shared. I also thank all those I met during the "praxe" tradition, whose camaraderie enriched my university experience. Finally, I am especially thankful to my girlfriend, Teresa Almeida, for her continuous assistance during the writing of this dissertation. To all of you, thank you.

Resumo

Esta tese tem como foco a otimização da configuração das rodas de inércia para o controlo de atitude de satélites em órbita heliossíncrona. O controlo de satélites em órbita terrestre baixa é crucial para a missão de um satélite de observação terrestre. As rodas de inércia são essenciais para executar manobras de atitude precisas, sendo importante minimizar o consumo energético de forma a otimizar a utilização de recursos do satélite.

A abordagem proposta centra-se inicialmente na simulação de um período orbital completo, considerando perturbações como o arrasto atmosférico, o potencial gravítico da Terra e a pressão de radiação solar. A fase de simulação orbital adota um modelo de propagação Cowell. Esta escolha equilibra realismo e eficiência computacional. Após a simulação orbital das posições do satélite ao longo do seu período orbital são realizadas manobras de atitude recorrendo a um controlador ótimo linear quadrático. O consumo energético das rodas de inércia é então calculado de forma aproximada.

Posteriormente, recorre-se a um algoritmo evolutivo, baseado na otimização por enxame de partículas, e variando o número, orientação e posição das rodas de inércia. O objetivo é minimizar o consumo energético, respeitando as restrições temporais pelo plano de operação, para garantir uma viabilidade computacional do processo iterativo, assume-se massa constante e emprega-se um modelo simplificado de propagação orbital. Os resultados obtidos identificam as configurações de rodas de inércia que tiveram baixo consumo e o desempenho de controlo exigido, contribuindo para missões mais sustentáveis e eficientes, em órbita terrestre baixa.

O contexto do paradigma “New Space” tem promovido uma crescente democratização do acesso ao espaço, com empresas privadas a desenvolverem soluções inovadoras e mais económicas. Neste cenário, a otimização de subsistemas críticos, como o controlo de atitude, torna-se um fator decisivo para a competitividade e sustentabilidade das missões. Recorrendo a esta abordagem é possível contribuir para a flexibilidade e rapidez da análise de desempenho e desenvolvimento dos sistemas de controlo de atitude na fase inicial de projeto.

Palavras-Chave

LEO – Controlo Ótimo e Robusto – Algoritmo Genético – Controlo da Atitude – Quaterniões

Abstract

This dissertation focuses on optimizing the configuration of reaction wheels for satellite attitude control in a sun-synchronous orbit.

The control of a satellite in low Earth orbit is crucial throughout the mission period of an Earth observation satellite. Reaction wheels are essential for executing precise attitude maneuvers in space, but their energy consumption must be minimized to extend mission life.

The proposed approach involves simulating a full orbital period, considering key orbital disturbances such as atmospheric drag, J2, and solar radiation pressure. During this period, a series of attitude maneuvers is performed using control strategies based on Linear Quadratic Regulator controllers. The energy consumed by the reaction wheels for these maneuvers is calculated, approximately.

Subsequently, the proposed approach is to carry out simulations, using a genetic algorithm, by varying the number, orientation, and position of the reaction wheels, with the objective of minimizing energy consumption while respecting the time constraints required to perform the maneuvers defined in the reference concept of operations.

The outcome of this work provides an efficient reaction wheel configuration that balances energy usage and performance, contributing to more sustainable and efficient satellite missions in low Earth orbit.

Similarly, the mission development and design phase is optimized, as this approach provides greater flexibility and accelerates the analysis and performance evaluation of attitude control systems during the early stages of the design process.

Keywords

LEO – Optimal and Robust Control – Genetic Algorithm – Attitude Control – Quaternions

Contents

List of Figures

List of Tables

List of Acronyms

ACS	Attitude Control System
API	Application Programming Interface
CONOPS	Concept of Operations
ECI	Earth-Centered Inertia
GA	Genetic Algorithm
IR	Infrared
LEO	Low Earth Orbit
LOS	Line of Sight
LQR	Linear Quadratic Regulator
MIMO	Multi-Input Multi-Output
PPC	Peak Power Consumption
PSO	Particle Swarm Optimization
RAAN	Right Ascension of the Ascending Node
RK4	Fourth-Order Runge-Kutta Integrator
RW	Reaction Wheel
RWCM	Rection Wheel Consumption Model
RWTD	RW Torque Distributor
SAR	Synthetic Aperture Radar
SLSQP	Sequential Least Squares Programing
SSO	Sun-Synchronous Orbit

List of Symbols

Symbols	Comments	Unit
$\hat{I}, \hat{J}, \hat{K}$	right-handed triad	$[-]$
h	specific angular momentum	$[m^2/s]$
i	inclination	$[deg]$
Ω	right ascension of the ascending node	$[deg]$
e	eccentricity	$[-]$
ω_p	argument of perigee	$[deg]$
θ	true anomaly	$[deg]$
a	semimajor axis	$[m]$
M	mean anomaly	$[deg]$
$\vec{r}$	position vector	$[m]$
$\vec{v}$	velocity vector	$[m/s]$
X, Y, Z	relative position Components	$[m]$
v_X, v_Y, v_Z	relative velocity Components	$[m/s]$
$\vec{x}$	state vector of the orbital model	$[-]$
μ	standard gravitational parameter	$[m^3/s^2]$
t	time	$[s]$
a_d	disturbances acceleration	$[m/s^2]$
$\vec{F}$	aerodynamic force	$[N]$
ρ	atmospheric density	$[kg/m^3]$

C_D	drag coefficient	$[-]$
A_p	projected area	$[m^2]$
α_1	angle between the surface normal and the velocity vector	$[deg]$
A_{area}	area of the surface	$[m^2]$
β	angle between the surface and the velocity vector	$[deg]$
s_{solar}	solar pressure	$[N]$
p	power of the sunlight	$[W]$
c_{light}	speed of light	$[m/s^2]$
ρ_a	absorbed fraction of photons	$[-]$
ρ_s	specularly reflected fraction of photons	$[-]$
ρ_d	diffusely reflected fraction of photons	$[-]$
ρ_t	transmitted fraction of photons	$[-]$
$\vec{s}$	sun vector	$[m]$
$\vec{n}$	unit normal to the surface	$[-]$
S	solar flux	[N/m2]
$\overrightarrow{c_{LOS}}(\tau)$	LOS algebraic function	$[-]$
$\overrightarrow{a_{SAT}}$	satellite vector	$[m]$
$\overrightarrow{b_{SUN}}$	Sun vector	$[m]$

$R_{\oplus}$	Earth radius	$[m]$
τ_{min}	value of τ where the function $\lvert\overrightarrow{c_{LOS}}(\tau)\rvert$ reaches its minimum	$[-]$
q	quaternion	$[-]$
q_0, q_1, q_2, q_3	4 quaternion elements	$[-]$
ϕ	quaternion rotation angle	$[deg]$
ω	angular velocity	$[rad/s]$
I	inertia matrix	$[kg \cdot m^2]$
m_k	mass	$[kg]$
x_k, y_k, z_k	position elements	$[m]$
c_{mass}	Centre of mass	$[m]$
r_k	position	$[m]$
B_k	transformation matrix	$[-]$
$x_{box}, y_{box}, z_{box}$	box dimensions	$[m]$
H	total angular momentum for the system in the inertial frame	$[kg \cdot m^2/s]$
h_b	angular momentum in the body frame	$[kg \cdot m^2/s]$
A_t	transformation matrix	$[-]$
T	torque defined in the body frame	$[N \cdot m]$
Ω_k	angular rate of the *kth* wheel about its spin axis	$[rad/s]$
ω	body rates	$[rad/s]$
J_k	spin-axis moment of inertia	$[kg \cdot m^2]$

u_k	unit vector of the RW spin axis measured in the body frame	$[-]$
T	body-fixed torque on the spacecraft	$[N \cdot m]$
T_k	scalar torque on each RW	$[N \cdot m]$
$x_{initial}$	initial Primary alignment	$[-]$
x_{final}	final primary alignment	$[-]$
$z_{initial}$	initial secondary alignment	$[-]$
z_{final}	final secondary alignment	$[-]$
$\vec{v}_{sup}$	support vector	$[-]$
$\vec{z}_{initial_{unit}}$	initial secondary alignment unit vector	$[-]$
b_1	body primary alignment vector	$[-]$
b_2	body secondary alignment vector	$[-]$
u_1	target inertial primary alignment vector	$[-]$
u_2	target inertial secondary alignment vector	$[-]$
q_{ROT1}	quaternion that rotates u_1 to b_1	$[-]$
s	scalar element of q_{ROT1}	$[-]$
d_x, d_y, d_z	elements of q_{ROT1} Gibbs vector	$[-]$
q_{ROT2}	quaternion that rotates about vector b_1	$[-]$
q_{EB}	full ECI-to-body quaternion	$[-]$
a, b, c	body frame three-unit vectors	$[-]$
α_2	angle of rotation of b about axis a	$[deg]$

$\vec{b}^*$	b after rotation	$[-]$
$\vec{c}^*$	c after rotation	$[-]$
θ	angle of rotation of q_{ROT2}	$[deg]$
$x(t)$	state vector	$[-]$
$u(t)$	vector of control input variables	$[-]$
t	time variable	$[s]$
$y(t)$	output variable	$[-]$
g_{fun}	g real function	$[-]$
f_{fun}	f real function	$[-]$
A	system matrix	$[-]$
B	input matrix	$[-]$
C	output matrix	$[-]$
D	feedforward matrix	$[-]$
λ	eigenvalues	$[-]$
C_0	controllability matrix	$[-]$
n_{dim}	dimension of the state-space	$[-]$
Δt	observation time step	$[-]$
Θ	observability matrix	$[-]$
$u(t)$	control vector	$[-]$
$J(u)$	performance criterion function	$[-]$
$L(u, x)$	quadratic function	$[-]$

u_c	control vector	$[-]$
m	number of rows	$[-]$
n	number of columns	$[-]$
K	control matrix	$[-]$
Q	state vector controller's performance	$[-]$
R	control vector controller's performance	$[-]$
η_k	Q matrix constants	$[-]$
λ_k	R matrix constants	$[-]$
$V(x)$	Lyapunov function	$[-]$
P	Riccati matrix solution	$[-]$
$E_c(\overline{\omega}, T, h_{step})$	energy consumed function	$[-]$
$\overline{\omega}$	rotor's average angular velocity	$[rad/s]$
h_{step}	time-step	$[s]$
a, b, c, d, e, f, g	reaction wheel consumption model coefficients	$[-]$
$P_{C,T}$	instantaneous power	$[W]$
H	angular momentum	$[kg \cdot m^2/s]$
J	rotor's moment of inertia	$[kg \cdot m^2]$
K	torque consumption constant	$[-]$
$(\%)_{Torque}$	torque percentage, relative to the max torque	$[-]$
$Max\ (\omega)$ $Steady\ State\ Consumpt$	Steady State consumption at max angular velocity	$[J]$

PPC	peak power consumption	$[J]$
T_{max}	max torque output	$[N \cdot m]$
P	state-space matrix	$[-]$
$x(t)$	state vector	$[-]$
$u(t)$	control vector	$[-]$
$y(t)$	output vector	$[-]$
A_1, A_2	A matrix constituent parts	$[-]$
g	Gibbs vector	$[-]$
$J_f(\omega_1, \omega_2, \omega_3)$	jacobians in order to ω	$[-]$
$\Omega_{k,max}$	RW max angular speed	$[rad/s]$
$\Omega_{k,min}$	RW minimum angular speed	$[rad/s]$
$\Omega_{k,0}$	RW initial angular speed	$[rad/s]$
$T_{control}$	control Torque	$[N \cdot m]$

Chapter 1

1. Introduction

The demand for increasingly precise satellite attitude control has grown significantly with the development of advanced space missions, especially for Low Earth Orbit (LEO) satellites. Attitude control systems (ACS) play a critical role in maintaining the correct orientation of spacecraft to ensure successful mission execution, whether for Earth observation, communication, or scientific exploration. Among the various methods for attitude control, reaction wheels (RWs) are one of the most widely used due to their strong heritage, precision, and efficiency in performing small, continuous adjustments without expending propellant.

This dissertation is framed within the context of satellite development under the "New Space" paradigm. "New Space" represents the democratization of space exploration, characterized by the increasing involvement of private sector companies and the growing commercialization of space activities. This shift is driving a fundamental change in how satellites are designed and developed.

In this evolving landscape, satellite design must become more cost-effective and highly adaptable, allowing for greater flexibility in conceptualizing new space missions. As development cycles accelerate, the space industry is expected to adopt a more serialized approach to satellite production. Within this new perspective, this dissertation aims to contribute by developing a tool that enhances adaptability and accelerates the early stages of satellite development.

However, designing an optimal configuration of reaction wheels involves complex trade-offs between power consumption, control accuracy, and system robustness. Typically, satellites are subject to disturbances from gravitational forces, solar radiation pressure, and atmospheric drag, all of which require careful compensation to maintain proper attitude, namely for very high-resolution payloads. The challenge lies in minimizing energy consumption while ensuring the spacecraft can meet all maneuvering requirements throughout the entire lifetime of the mission.

This dissertation explores the optimization of RW configurations for a satellite in a sun-synchronous orbit, focusing on minimizing energy consumption during attitude maneuvers while maintaining compliance with mission constraints. By leveraging control strategies such as the Linear Quadratic Regulator (LQR) controllers, combined with genetic algorithms, this work seeks to identify the optimal number, orientation, and position of RWs to maximize mission performance with the fewest resources.

1.1. State of LEO Space Exploration

Before discussing the *State of LEO Space Exploration*, it is necessary to first define Low Earth Orbit (LEO). LEO is typically defined as the region of space extending from approximately 160 km to 2,000 km in altitude above Earth's surface. Satellites in LEO complete an orbit in roughly 90 to 120 minutes, depending on their altitude. This region is characterized by high orbital velocities requirements compared to higher orbits, due to the stronger gravitational pull at lower altitudes, and LEO is the most densely populated orbital regime for satellites.

Figure 1.1 shows a LEO visualization:

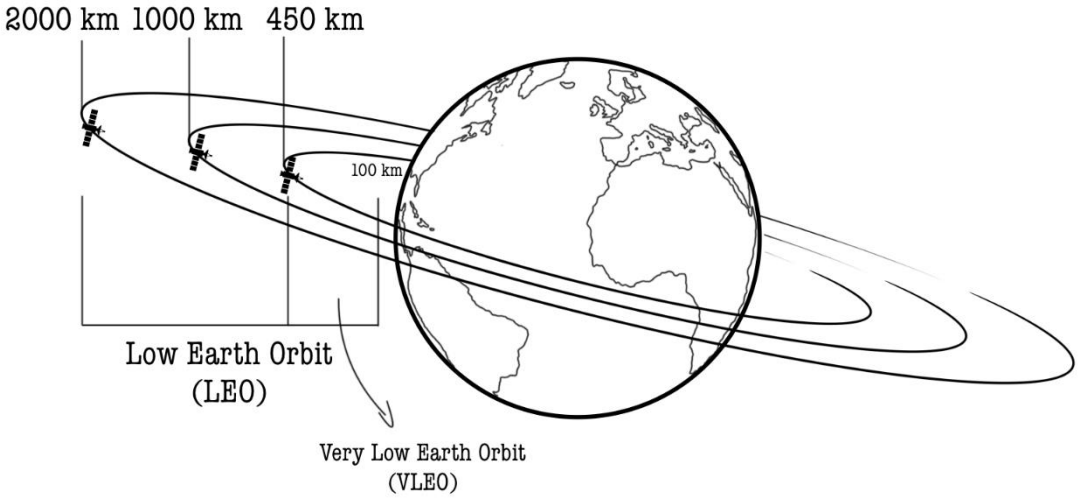

Figure 1.1: LEO Orbit Diagram [1]

When placing a satellite in LEO, certain challenges must be considered. The presence of the upper atmosphere causes gradual orbital decay, requiring station-keeping maneuvers or reboosts to maintain orbit. Although LEO is partially shielded from high-energy cosmic radiation by the Earth's magnetic field, satellites in this region are still subject to atmospheric drag, which can reduce their operational lifespan. Additionally, as previously mentioned, the growing number of satellites in LEO increases the likelihood of collisions with space debris, making active debris mitigation strategies essential.

However, placing satellites in LEO is not only challenging; it also offers several advantages. LEO has lower launch costs due to its proximity to Earth, requiring significantly less energy to reach compared to higher orbits such as Medium Earth Orbit (MEO) or Geostationary Orbit (GEO). Likewise, at lower altitudes, satellites in LEO can capture high-resolution images of Earth's surface, provide low-latency communications, and frequently revisit specific locations.

Various types of missions are conducted in LEO to take advantage of the benefits of operating in this type of orbit, including:

- Earth observation and surveillance missions, these missions leverage LEO's capability to capture high-resolution images of Earth's surface and provide frequent revisit times. Some of the most prominent Earth observation missions in LEO include the Sentinel satellites of the European Space Agency's (ESA) Copernicus program and the Landsat program, jointly operated by NASA and the United States Geological Survey (USGS) [2] [3]. Additionally, commercial constellations such as PlanetScope (Planet Labs) and ICEYE have transformed the field by providing high-frequency, high-resolution imagery through small satellite constellations.

- Communication missions, such missions were primarily conducted, historically, in GEO. However, in recent years, LEO has become the preferred orbit for communication missions utilizing large satellite constellations. These constellations are increasingly used for broadband internet coverage due to lower launch costs and reduced signal latency. Notable examples include SpaceX's Starlink and Amazon's Kuiper project.
- Scientific research and technology demonstration missions, any scientific experiments and technology demonstrations are conducted in LEO, which is also the primary domain for human spaceflight activities. Examples include the International Space Station (ISS) and CubeSat technology demonstration missions.

As previously mentioned, the number of satellites in LEO has been steadily increasing. In 2021, there were approximately 6,200 satellites in LEO, a number that has risen exponentially, reaching around 8,700 by 2023. Given these trends, it is projected that by 2034, about 30,000 satellites will be orbiting in LEO. As seen in Figure 1.2, most LEO satellites weigh around 400 kg [4]. This rapid increase highlights the growing importance of developing more efficient ACS to support the operational needs of this expanding satellite population.

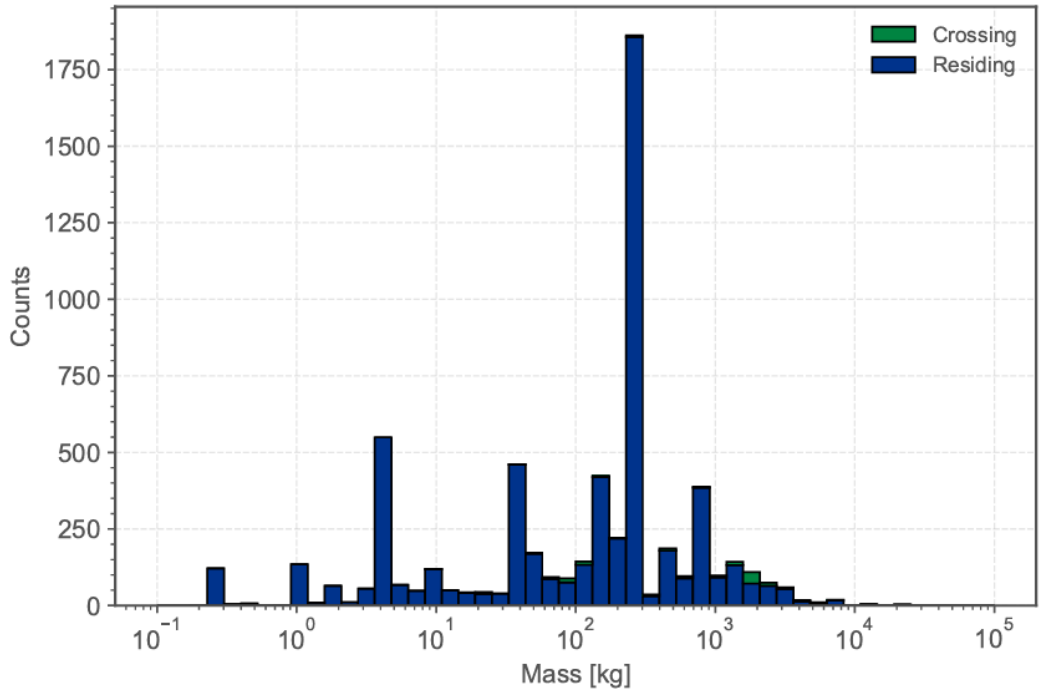

Figure 1.2: Mass Distribution of Payloads Intersecting LEO [4]

The exponential growth in the number of satellites in LEO can be attributed to several factors, with two standing out most prominently.

First, the increasing demand for global connectivity and high-resolution Earth observation data has accelerated the deployment of large-scale satellite constellations. Projects like SpaceX's Starlink and OneWeb aim to provide global broadband coverage, while Earth observation companies such as Planet Labs and Maxar Technologies offer high-resolution imagery for applications ranging from environmental monitoring to disaster response.

Secondly, the significant reduction in launch costs over the past decade, driven by technological advancements, increased competition, and the advent of reusable launch systems, has made space missions more accessible to a wider range of stakeholders, including private companies and academic institutions. Companies such as SpaceX, Blue Origin, etc. have played a pivotal role in this transformation, facilitating the cost-effective deployment of satellite constellations.

In the mid-20th century, space missions were predominantly government-funded, with costs often exceeding $100,000 per kilogram to LEO. For instance, NASA's Space Shuttle program, operational from 1981 to 2011, had an estimated cost of approximately $54,500 per kilogram to LEO [5].

The early 21st century marked a transformative period in the aerospace industry with the emergence of private companies introducing more cost-effective launch solutions. SpaceX's Falcon 9 rocket, first launched in 2010, significantly disrupted the market by offering launch services at substantially lower prices. As of 2020, the Falcon 9 advertised a cost of $62 million per launch, translating to approximately $2,720 per kilogram to LEO [6].

This evolution in launch costs to LEO can be observed in Figure 1.3 below:

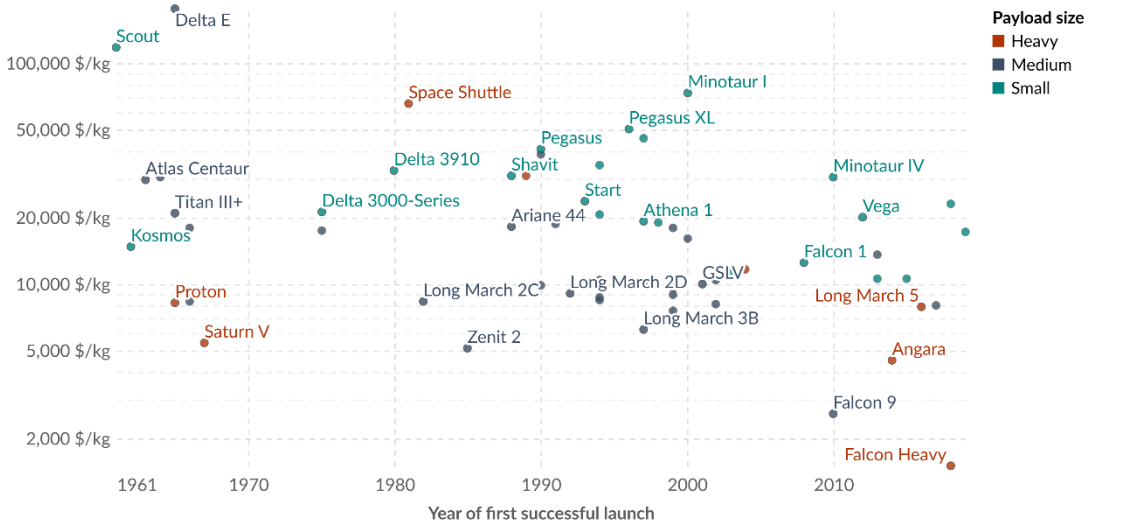

Figure 1.3: Cost of Space Launches per Kilogram to Low Earth Orbit [7]

The focus of this dissertation on Earth observation missions, particularly in the context of reduced launch costs to LEO, which have contributed to a significant decrease in the cost of satellite imagery. The cost of satellite imagery has experienced significant variations over time, influenced by several technological advancements, market dynamics and the economies of small satellite constellations. Historically, acquiring high-resolution satellite images was a costly endeavor, often limiting access to large governmental agencies and well-funded organizations. However, recent trends indicate a democratization of access due to decreasing costs.

According to data from 2022, the cost of acquiring satellite imagery varied depending on the satellite source. For instance, images from the GeoEye-1 satellite were priced at approximately $25 per square kilometer. In contrast, imagery from other commercial providers was available at different price points, reflecting a range of options in the market. The commercial satellite imaging market has also seen substantial growth.

Valued at $3.3 billion in 2022, it is projected to reach $9.6 billion by 2032, growing at a compound annual growth rate of 11.5% [8] [9]. The figure 1.4, presents a relation between spatial resolution and price, in LEO.

These developments underscore the importance of efficient ACS in satellites. As the number of satellites in LEO increases, maintaining precise orientation becomes critical to ensure the quality and reliability of the imagery captured, thereby supporting the expanding applications of satellite data across industries.

These developments underscore the importance of efficient ACS in satellites. As the number of satellites in LEO increases, maintaining precise orientation becomes critical to ensure the quality and reliability of the imagery captured. This challenge is further amplified by the growing use of high-resolution sensors, which require greater stability and pointing accuracy to avoid image degradation, thereby supporting the expanding applications of satellite data across industries.

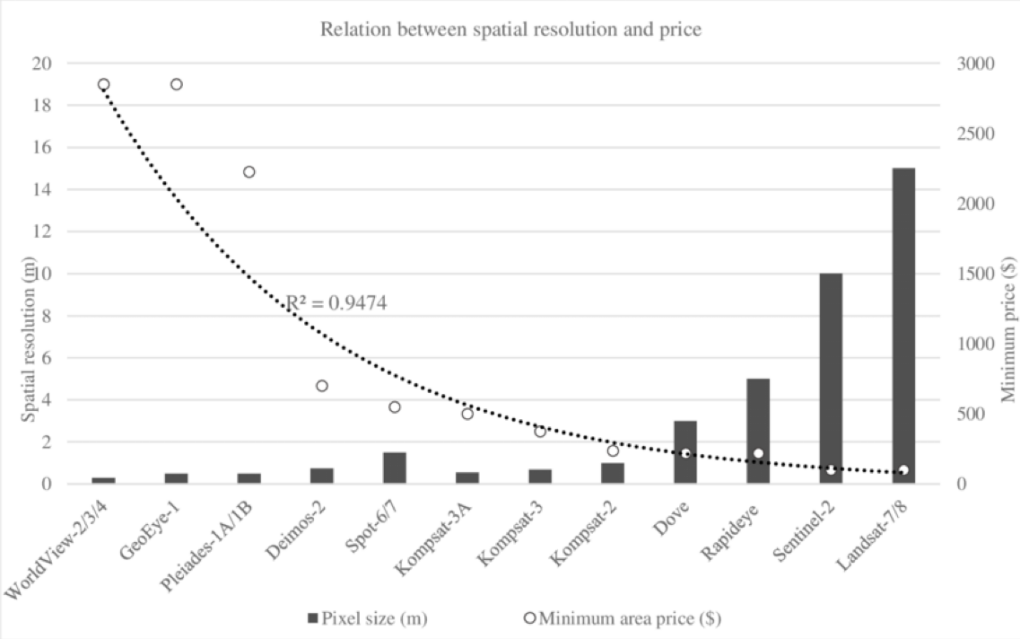

Figure 1.4: Relation between spatial resolution and price [10]

The space exploration industry has undergone a transformative shift over the past decade. The growing demand for satellite services has driven the deployment of large-scale satellite constellations aimed at providing global internet coverage, Earth observation, and other critical applications. Private space companies and governments are now launching hundreds of satellites annually, making LEO a key sector for innovation, research, and economic development.

In the future, concerns regarding space traffic management and debris mitigation will need to be addressed more seriously. Advancements in space sustainability, including active debris removal and collision avoidance systems, will become essential. As LEO becomes increasingly commercialized, regulatory frameworks must evolve to effectively manage space traffic and implement sustainability measures.

This evolving scenario has created a growing demand for satellites with precise attitude control capabilities to ensure optimal performance for communication links, imaging accuracy, and other mission-critical tasks. These developments require highly dynamic and precise attitude control systems, increasing the operational demands on ACS.

Consequently, the design and optimization of ACS have become crucial to ensuring mission success while minimizing resource consumption.

1.2. How an Earth Observation Mission is Developed

The success of an Earth observation mission to LEO relies on careful planning. With the proliferation of satellites in this orbital region, as previously mentioned, designing an efficient and reliable mission has become critical for the competitiveness of many space businesses. Their success requires structured engineering processes.

The development process for Earth observation missions follows a series of phases, as outlined by frameworks such as NASA's Systems Engineering Lifecycle [11] and ESA's mission selection guidelines [12]. These phases ensure that every aspect of the mission, from concept studies to closeout, is aligned with the mission's goals and the constraints imposed by LEO's dynamic environment. A description of each of these phases is given next.

1.2.1. Pre-Phase A: Concept Studies

This phase aims to generate a broad range of mission concepts and alternatives to facilitate the selection of new projects. It involves assessing the feasibility of the desired system, formulating mission concepts, and evaluating performance, cost, and schedule feasibility. Additionally, this phase identifies potential technological needs and defines the overall scope of the mission.

To develop an Earth observation mission, it is essential to first define the mission's objectives, based on the intended applications, such as environmental monitoring, disaster response, or urban planning. The selection of appropriate payloads is crucial, from optical sensors to SAR being considered based on the mission's requirements. Given the crowded nature of LEO, single-satellite missions may be preferred for targeted studies, while satellite constellations are employed for global coverage and reduced revisit times. The unique advantages of LEO, including proximity to Earth and high-frequency imaging capabilities, shape the mission's preliminary design considerations.

1.2.2. Phase A: Concept and Technology Development

This phase focuses on evaluating the feasibility and desirability of a proposed system and establishing an initial baseline. It includes the development of the final mission concept, identification of necessary technological advancements, and formulation of project technical management plans.

With the Earth observation mission objectives established, a feasibility study is conducted, and the selection of an appropriate orbit is made. Sun-synchronous orbits (SSO) within LEO are often preferred for Earth observation due to their ability to provide consistent lighting conditions, enabling precise imaging. This phase also involves preliminary assessments of power requirements, data transmission capabilities, and mitigation strategies for LEO-specific disturbances, such as atmospheric drag.

1.2.3. Phase B: Preliminary Design and Technology Completion

The primary goal of this phase is to refine the project to a level of detail sufficient for establishing an initial baseline that meets mission needs. This involves developing the structural design requirements and generating preliminary designs for each subsystem system structure end product.

During this phase, for an Earth observation mission, the spacecraft's architecture is conceptualized to accommodate its payload and subsystems. The spacecraft must also be designed to handle perturbative effects such as solar radiation pressure and aerodynamic drag. Ensuring sufficient energy generation, especially for the observation instruments that have high power consumption.

1.2.4. Phase C: Final Design and Fabrication

This phase entails finalizing the detailed design of the system and its associated subsystems, including operational components. It also involves the fabrication of hardware, and the development of software required for mission execution.

In this phase, for an Earth observation mission, the detailed engineering design is completed, integrating subsystems such as communication, thermal management, power distribution, and attitude control. Special attention is given to the ACS which is particularly critical for ensuring the stability and precision required for Earth observation. Given the increasing number of satellites in LEO, collision avoidance measures and compliance with debris mitigation guidelines are incorporated into the design.

1.2.5. Phase D: System Assembly, Integration, Testing, and Launch

During this phase, all system components, hardware, software, and human elements, are assembled and integrated. Testing procedures are conducted to ensure the system meets predefined requirements. The system is then launched and prepared for operations.

The spacecraft, used in an Earth observation mission, undergoes extensive testing to validate its performance under the harsh conditions of space. Thermal vacuum testing simulates temperature fluctuations in LEO, while vibration testing ensures the satellite can withstand launch conditions. The ACS is rigorously evaluated to confirm its ability to maintain precise orientation despite LEO-specific disturbances. Once testing is completed, a suitable launch vehicle is selected based on the satellite's mass, dimensions, and target orbit. After launch, the satellite enters the commissioning phase, during which its systems are activated, payloads are calibrated, and orbital corrections are performed if necessary. Given the dense satellite population in LEO, collision avoidance protocols are implemented during this phase.

1.2.6. Phase E: Operations and Sustainment

This phase focuses on executing the mission in accordance with the predefined operational plan while ensuring sustained functionality and support throughout its lifecycle. Following commissioning, in an Earth observation mission, the satellite begins its operational phase, collecting data for applications such as climate monitoring, deforestation assessment, and urban development.

Given the increasing reliance on space-based data, commercial satellite constellations have become integral to Earth observation. As a result, mission operations emphasize efficient ACS performance to ensure continuous data acquisition.

1.2.7. Phase F: Closeout

The objective of this final phase involves implementing the system's decommissioning plan. It also includes conducting post-mission analyses of returned data ensuring a thorough assessment of mission outcomes.

Upon the completion of an Earth observation mission, the satellite undergoes deorbiting procedures in compliance with space debris mitigation policies. In the case of large constellations, individual satellites may be deorbited in a staggered manner to minimize risks to active spacecraft. Data collected throughout the mission is analyzed and stored.

1.3. State of Space Technology for Earth Observation Satellites

The exponential growth of satellites in LEO, as discussed in Section 1.1, has paved the way for rapid advancements in space technology. This surge, driven by reduced launch costs and increasing demand for data, has placed Earth observation satellites at the forefront of addressing critical global challenges. The technologies employed in these satellites have become increasingly sophisticated, enabling the collection of precise and comprehensive data that supports a variety of applications.

The strategic importance of these technologies is evident in applications such as maritime surveillance, disaster management, and urban planning. For instance, Figure 1.5 illustrates how Earth observation satellites contribute to monitoring exclusive economic zones, ensuring compliance with international regulations, and supporting efforts against illegal activities such as unregulated fishing.

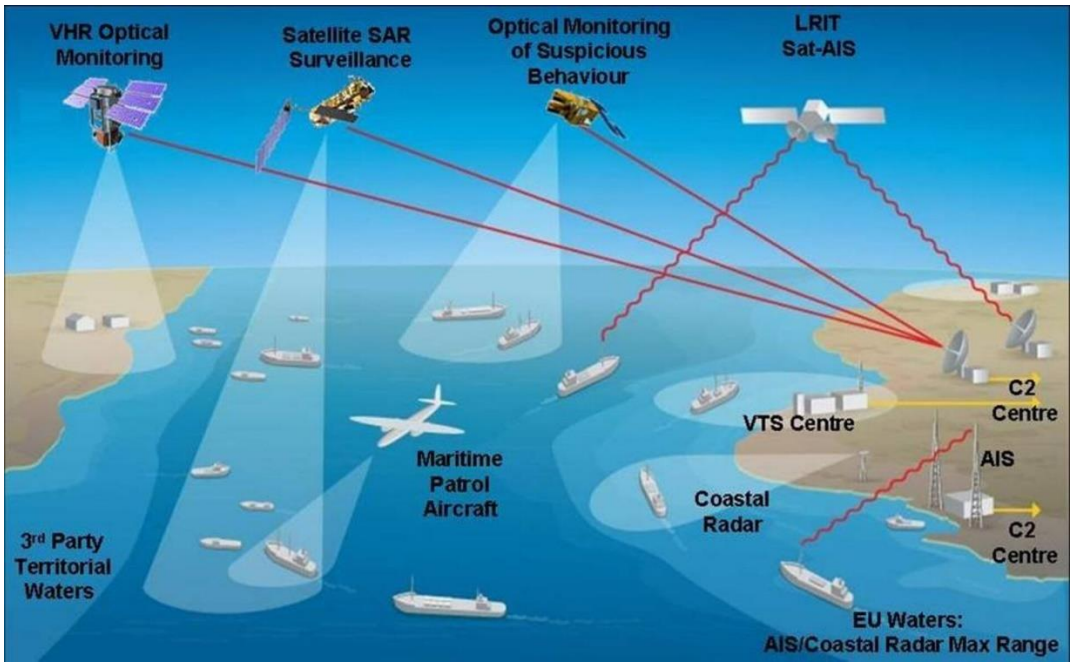

Figure 1.5: Earth Observation Mission Diagram [13]

Despite having a very precise objectives, each Earth observation mission relies on a set of fundamental technologies, common to most LEO missions, that ensure their operational functionality and longevity. These core technologies play a crucial role in spacecraft performance, as described next:

- Power systems: most LEO satellites rely on solar panels to generate electrical power from sunlight. These panels continuously recharge onboard batteries, which supply energy during orbital periods when the satellite is in Earth's shadow.
- Communication systems: efficient communication is essential for data transmission between the satellite and ground stations. Satellites are equipped with high-gain antennas to send high-volume data, such as Earth observation images, while low-gain antennas facilitate basic telemetry and command functions.
- Structural components: the body frame or structure of a satellite provides mechanical support for all onboard subsystems and ensures resistance to harsh space conditions.
- Propulsion systems: many LEO satellites incorporate propulsion systems for orbital adjustments and collision avoidance. Depending on the mission, these systems can include chemical thrusters, electric propulsion, or even cold gas thrusters for fine maneuvers.
- Attitude control technologies: one of the most critical technologies for any LEO mission, ensures the satellite maintains the correct orientation. This is especially vital for Earth observation satellites, where precise pointing accuracy is required for imaging instruments.

In the case of attitude control, one of the key components are the RWs, used for fine attitude adjustments by leveraging angular momentum; magnetorquers, employ the Earth's magnetic field to generate torque; star trackers and sun sensors, provide orientation data by tracking celestial bodies; and gyroscopes, that measure angular velocity and acceleration for navigation and stability.

While these technologies are universally used in LEO missions, their specific design and implementation vary depending on mission requirements. For instance, a small CubeSat can rely solely on magnetorquers for attitude control, while a high-resolution Earth observation satellite requires a combination of reaction wheels and advanced control algorithms for precision pointing.

Among the most critical payload technologies are optical and multispectral cameras, which provide high-resolution imagery across multiple wavelengths, enabling detailed surface analysis. Additionally, hyperspectral sensors extend this capability by capturing data across hundreds of spectral bands, allowing precise element identification and land-use classification. Another essential technology is Synthetic Aperture Radar (SAR), which operates independently of weather conditions and daylight, offering continuous monitoring of Earth's surface through radar imaging. Meanwhile, Light Detection and Ranging (LiDAR) technology provides highly accurate 3D mapping capabilities by using laser pulses to measure surface elevation and terrain structure.

To ensure the successful operation of these sophisticated payloads, Earth observation satellites integrate specialized data processing and storage systems, capable of handling large volumes of high-resolution imagery before transmitting the data to ground stations.

The increasing demand for actionable data, driven by policymakers, scientists, and industry leaders, has catalyzed advancements in satellite payloads. While it remains a subject of debate whether this demand is a consequence of technological progress or a driving force behind it, the relationship is undoubtedly synergistic. The deployment of advanced sensors has expanded the scope of satellite applications, enabling data-driven decision-making for climate action, sustainable development, and urban resilience.

1.4. Current State of Attitude Control Systems

Attitude control systems are fundamental to satellite operations, ensuring that spacecraft maintain the correct orientation for their mission objectives. This is particularly crucial for Earth observation satellites, where precise pointing accuracy is required for imaging instruments. This section provides an in-depth discussion of these technologies and their relevance which justify the exclusive use of RWs in this dissertation. Additionally, it explores the maximum torque capacity, the interaction between RWs and magnetorquers for desaturation and presents an overview of the ACS configurations employed in well-known satellites.

Reaction wheels are among the most widely used actuators for precise attitude control. They consist of spinning disks that, when accelerated or decelerated, create torque that changes the spacecraft's orientation without expelling mass, making them useful for long-duration missions. Some examples of RWs are illustrated in Figure 1.6.

Figure 1.6: NewSpace Systems' Reaction Wheels [14]

The torque generated by a reaction wheel depends on its size, mass, and angular velocity limits. Table 1.1 below provides examples of maximum torque capabilities for commonly used RWs in LEO missions:

Table 1.1: Reaction Wheel Specifications in LEO missions

Reaction Wheel Model	**Max Torque (mNm)**	**Max Angular Momentum (Nms)**	**Typical Application**
Honeywell HR16	125	6.8	Small Satellites
Rockwell Collins RWA-1000	250	20	Medium-Class Satellites
Sinclair RW-4	25	0.1	CubeSats, SmallSat
Blue Canyon XACT-50	100	5.0	Small-Medium Satellites

In addition to RWs, other hardware components like magnetorquers and control moment gyroscopes may be utilized to enhance the robustness and efficiency of attitude control. Magnetorquers exploit the Earth's magnetic field to produce torque, while gyroscopes leverage gyroscopic precession to control orientation, providing complementary capabilities to reaction wheels.

Figure 1.7 shows a representation of a magnetorquer integrated within the interior of a satellite:

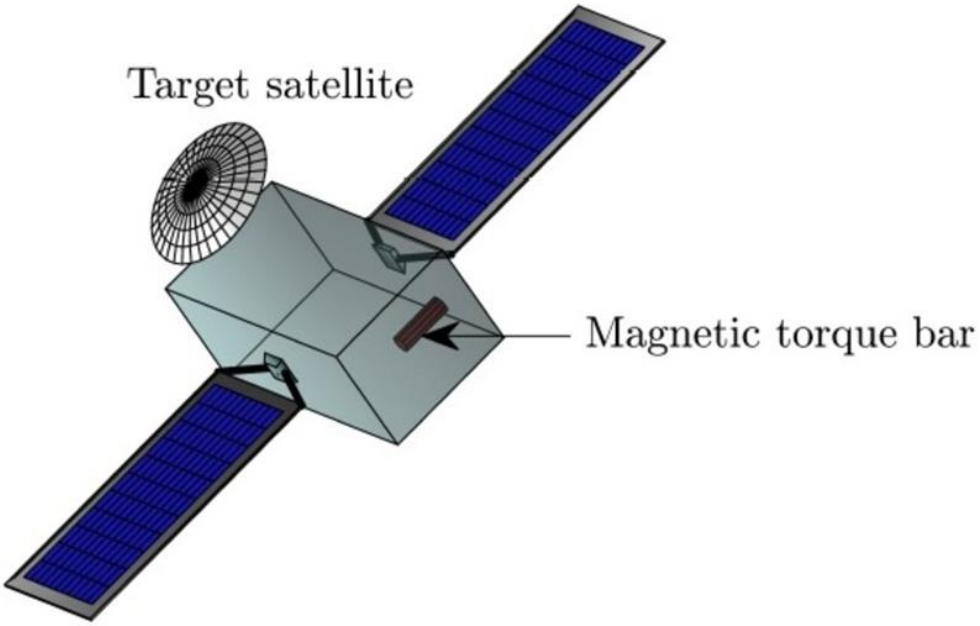

Figure 1.7: Magnetic Torque Bar Representation in a Satellite [15]

Control moment gyroscopes (CMGs) operate similarly to RWs but introduce an additional degree of freedom by tilting the spinning wheel, producing larger torques. They are typically used in high-agility spacecraft requiring rapid maneuvering. CMGs generate significantly higher torques than RWs, making them suitable for large satellites and space stations, not commonly used in small or medium-sized Earth observation satellites due to their complexity and Table 1.2 presents examples of CMGs used in LEO missions:

Table 1.2: Control Moment Gyroscopes Specifications in LEO missions

CMG System	**Max Torque (Nm)**	**Application**
ISS CMG	300	ISS
Large CMG (ESA)	100–500	Large Satellites

Magnetorquers (MTQs) consist of electromagnetic coils that produce a magnetic moment, which interacts with the planet’s magnetic field to create torque. While MTQs provide a reliable and low power means of attitude control, they generate much lower torque compared to RWs or CMGs, and their effectiveness depends on the satellite's position relative to Earth's magnetic field.

Table 1.3 presents examples of MTQs used in LEO missions:

Table 1.3: Magnetorquers Specifications in LEO missions

Magnetorquer Model	Max Torque (mNm)	Application
ZARM MTQ-15	15	Small Satellites
ISIS Magnetorquer Board	10	CubeSats
Rockwell Collins MTQ	50	Medium Satellites

Reaction wheels are not without limitations. While RWs provide precise attitude control, their stored angular momentum must be managed to prevent saturation. Over time, external disturbances such as atmospheric drag and solar radiation pressure cause unwanted momentum buildup in the wheels. To counteract this, magnetorquers are often used in a desaturation process, where they apply small torques to gradually release excess momentum from the RWs. This hybrid approach allows for prolonged use of RWs without overloading them, extending the mission lifespan and reducing the need for thrusters or other propellant-based actuators.

Given the comparative analysis of attitude control technologies, the exclusive use of RWs in this dissertation is justified by their crucial role in satellite attitude control, particularly for Earth observation missions, where precision and stability are paramount. Their ability to provide fine-tuned control makes them essential for achieving high-resolution imaging and accurate data collection. Unlike thrusters, RWs do not consume propellant, making them highly suitable for long-duration missions.

Additionally, RWs feature a compact and scalable design, allowing them to be effectively integrated into small and medium-sized satellites without imposing excessive power or mass requirements. The reliability of RWs is well-established, their proven effectiveness in real-world applications highlights their significance in modern space missions.

By focusing on RWs as the primary attitude control technology, this dissertation aims to optimize their configuration, number, and placement to achieve maximum energy efficiency while fulfilling mission constraints.

Different Earth observation satellites employ various configurations of RWs, MTQs, and other ACS technologies. Table 1.4 shows examples of well-known satellites and their ACS configurations, which highlight the widespread reliance on RWs for precise control in Earth observation satellites, with magnetorquers playing a key role in desaturation:

Table 1.4: ACS Setups

Satellite	RW Configuration	MTQ Usage	Additional ACS Components
Sentinel-2	4 RWs (Redundant)	Yes	Star Trackers, IMUs
Landsat-8	4 RWs	Yes	Sun Sensors, Gyroscopes
Terra (EOS)	4 RWs	Yes	Magnetometers, Star Cameras
SPOT-6	4 RWs	Yes	GPS-based attitude determination

From a control methodology perspective, advanced algorithms like the LQR are commonly implemented to optimize attitude control. LQR excels at minimizing a cost function related to energy consumption and attitude deviation, allowing for optimal performance under nominal conditions.

These control strategies enable satellites to perform complex maneuvers and remain stable despite disturbances such as atmospheric drag, solar radiation pressure, etc. As satellite missions become more sophisticated, the need for advanced ACS continues to grow, driving ongoing research into innovative hardware and control methodologies.

This section has outlined the principal technologies employed for satellite attitude control in Low Earth Orbit, with particular emphasis on the advantages of RWs and their interaction with magnetorquers for desaturation. While these technologies provide a robust hardware foundation, their performance is inherently dependent on how they are configured and controlled. This leads directly to the next section, which delves into the state of the art in modeling orbital disturbances, implementing control algorithms such as the LQR, and applying advanced optimization techniques, including Genetic Algorithms and Particle Swarm Optimization, to determine the optimal number, positioning, and orientation of reaction wheels. These strategies are aimed at minimizing energy consumption and maximizing the efficiency of satellite attitude maneuvers.

1.5. Orbital Disturbances and Control Optimization

This section highlights the state of the art in key areas relevant to this dissertation. Namely, in the modeling of orbital disturbances in LEO, the use of RWs for satellite attitude control, the implementation of LQR control methods for maneuver execution, and the application of genetic algorithms (GAs) for optimizing RW configurations.

Each of these domains provides both the theoretical and practical foundation necessary for the successful completion of the research objective, which is, finding the optimal number, orientation, and position of RWs that minimizes the energy consumption while achieving all required satellite maneuvers within a representative period of the mission nominal operations.

1.5.1. Disturbances in Low Earth Orbit

Accurately determining a satellite’s position over time is a fundamental requirement for attitude control analysis and optimization. Since the satellite's trajectory is influenced by various perturbative forces, an orbital simulation that accounts for these effects is necessary. In LEO, disturbances such as atmospheric drag, Earth's oblateness (J2 effect), and solar radiation pressure significantly impact the satellite’s motion, necessitating precise modeling to achieve realistic simulations.

To accurately apply the acceleration or force generated by these disturbances, it is essential to review state-of-the-art methodologies for incorporating these effects into an orbital propagator. This dissertation employs Cowell’s method [16], a numerical propagation technique that directly integrates the equations of motion by summing all perturbative accelerations. Given its ability to handle complex force models, Cowell’s propagator is well-suited for simulating the dynamic environment of LEO.

The studies on orbital disturbances by Paluszek [17] and Curtis [18] provide a comprehensive framework for understanding these external forces. Accounting for such disturbances ensures the accuracy of the satellite's simulated trajectory and the efficiency of attitude maneuvers controlled by the RWs. These advancements ensure that the propagated orbit closely matches real-world observations.

1.5.2. RWs in Satellite Attitude Control

The RWs have proven to be essential components for precise satellite attitude control. Their ability to adjust a spacecraft's orientation without expending fuel makes them particularly advantageous for long-term missions, especially in LEO. The fundamental operating principle of RWs is based on the conservation of angular momentum, where changes in the wheel's angular velocity generate a corresponding torque, adjusting the satellite's orientation accordingly.

As the satellite propagates along its orbit, its position relative to Earth and mission targets continuously changes. Consequently, the vector connecting the satellite to its intended target undergoes constant variation over time. This dynamic shift necessitates precise attitude control to ensure that the satellite maintains accurate alignment with its target throughout its orbit. To achieve this, a robust ACS is required.

Given the dynamic nature of LEO, the inclusion of a well-designed RW system is crucial for ensuring reliable and precise attitude control. The studies reviewed in the literature provide a strong foundation for understanding the role of RWs in satellite attitude control, reinforcing their exclusive use in this dissertation.

Additionally, the research by Wertz [19] explores various control strategies, such as LQR-based methods, to efficiently stabilize and maneuver satellites using RWs. These control techniques play a critical role in achieving the required attitude accuracy while optimizing energy consumption, one of the key objectives of this study.

Furthermore, Wie [20] provides an extensive review of spacecraft attitude dynamics and control systems, with a detailed discussion on the application of RWs in different configurations to meet diverse mission requirements. In particular, Wie emphasizes the importance of selecting the optimal number and orientation of RWs to balance performance, redundancy, and power consumption.

1.5.3. Control Algorithms for Attitude Maneuvers

Attitude control algorithms are essential for the precise orientation of spacecraft, ensuring that satellites maintain their desired positioning relative to Earth or other celestial targets. These algorithms manage the spacecraft's actuators, such as RWs, CMGs, or magnetorquers, to achieve and sustain accurate attitude adjustments. Various methodologies have been developed for attitude control, each tailored to optimize performance under specific mission parameters and environmental disturbances.

The Linear Quadratic Regulator is a widely adopted control strategy, particularly effective in linear systems where dynamics can be approximated linearly. LQR aims to minimize a cost function that balances control effort and deviation from the desired attitude, providing an optimal solution for satellite orientation. Research has demonstrated the application of LQR in spacecraft attitude control, highlighting its capability to achieve precise control while minimizing energy consumption [21].

For instance, the paper “Attitude Control of a 2U Cubesat by Magnetic and Air Drag Torques” describes the development of a magnetic attitude control subsystem for a 2U CubeSat that employed a LQR to maintain desired orientation despite environmental disturbances, showcasing its practical utility in LEO missions [22].

In addition to LQR, several other control methods have been explored for satellite attitude control. One such method is H∞ control, which seeks to minimize the worst-case performance of the system under uncertainty.

This approach provides robustness against disturbances and model inaccuracies, making it particularly useful for missions exposed to unpredictable external forces or parameter variations. H∞ control has been successfully applied in scenarios requiring enhanced robustness, though it introduces a higher level of complexity compared to LQR.

For example, studies have applied H-Infinity controllers to spacecraft attitude control [23], demonstrating their robustness towards space disturbances and their ability to maintain stability under adverse conditions.

Another method worth mentioning is Model Predictive Control (MPC), which is gaining traction in complex missions requiring the management of multiple constraints, such as fuel limitations and actuator saturation. MPC optimizes control inputs by predicting future system behavior over a specified horizon, making it adept at handling multivariable control problems. Despite its higher computational demand, MPC has been applied in real-time satellite systems, thanks to advancements in computational power.

Recent studies have explored MPC for spacecraft attitude control [24], particularly in missions necessitating precise maneuvering and high-level optimization.

Despite H∞ control and MPC being viable alternatives for specific mission profiles requiring robustness or sophisticated trajectory management, they were not employed in this dissertation. Instead, the LQR is employed as the control methods due to its balance of computational efficiency and control performance, aligning with the objectives of minimizing energy consumption while maintaining high control precision.

1.5.4. Optimization of RW Configuration for Orbit Satellite Control in LEO

The field of satellite attitude control has seen significant advancements, particularly in the optimization of actuator systems such as RWs to ensure efficient satellite operation in LEO.

This work focuses on optimizing RW configurations, specifically, determining the optimal number, orientation, and positioning of the RWs, which can significantly improve control efficiency and extend the operational mission lifetimes. Consequently, the proper configuration of RWs is essential to optimize their overall performance.

Given the high degrees of freedom involved in configuring RW architectures and placements, the use of GAs has been explored due to their capacity to efficiently search vast solution spaces and identify near-optimal configurations. These algorithms are particularly beneficial for balancing energy efficiency with control precision. Consequently, the proper configuration of RWs is essential to optimize their overall performance.

This integrated approach of control theory and optimization algorithms establishes a robust foundation for the enhancement of satellite ACS, thereby supporting the objectives of this dissertation.

A paper, titled “Optimization of Pyramidal Reaction Wheel Configuration for Minimizing Angular Momentum” [25] develops a study on the arrangement of four RWs in a pyramidal setup to provide three-axis control with redundancy. Research has investigated the optimal tilt angles in such configurations to minimize angular momentum during maneuvers, thereby reducing energy consumption, highlighting the importance of geometric optimization in RW placement.

Another paper, titled “Optimum Placement of Satellite Components with Genetic Algorithms” [26] presents a GA-based approach for the optimal placement of satellite components, demonstrating that GA-based optimization effectively balances constraints such as mass distribution, center of mass location, and system inertia. The research introduces a discrete optimization problem where satellite components, including actuators, are placed within a three-dimensional grid. This method can be extended to optimize RW placement in attitude control systems, where proper positioning is critical for minimizing energy use and achieving precise attitude control.

Another relevant paper, titled "Genetic Programming Method for Satellite System Topology and Parameter Optimization" [27] introduces a genetic programming method for satellite system design to simultaneously optimize the topology and parameters of a satellite system. In this research, the satellite's internal space is discretized into a three-dimensional grid, with each cell representing a unit volume. The genetic algorithm is employed to determine the optimal placement of components, including actuators like reaction wheels, by optimizing the center of mass locations while adhering to constraints such as mass balance, center of mass positioning, and moment of inertia requirements. This methodology effectively balances energy efficiency with control precision, enhancing overall satellite performance.

The use of GA for optimizing reaction wheel configurations has been demonstrated to significantly improve satellite attitude control by reducing energy consumption and increasing robustness. Studies have shown the effectiveness of GA-based optimization in balancing multiple constraints. By leveraging GA-based techniques, spacecraft designers can achieve locally optimal RW configurations, reducing the duration of the mission planning and development phase.

1.5.5. Optimization Techniques: Genetic Algorithms

Genetic algorithms are widely recognized for their efficacy in solving complex optimization problems with many variables, making them ideal for determining the optimal configuration of RWs in satellite attitude control. By simulating the process of natural selection, GAs iteratively explore the solution space, balancing the exploration of new configurations with the exploitation of promising solutions. This iterative process enables GAs to identify configurations that minimize energy consumption while satisfying mission-specific constraints.

Goldberg's work [28] discusses the successful application of GAs to various engineering optimization problems, emphasizing their effectiveness in managing multi-objective tasks with intricate constraints. Engelbrecht [29] further explores the potential of GAs in satellite attitude control, specifically highlighting their ability to optimize RW orientation and number to reduce energy consumption and enhance mission performance.

In this dissertation, a modified GA and a Particle Swarm Optimization (PSO) approach were employed, evolution strategies such as PSO, are optimization algorithms that model social behaviors observed in nature, such as the coordinated movement of birds or fish. In PSO, each "particle" represents a candidate solution and adjusts its position within the search space based on both its own best experience and the best experience of its neighboring particles. Through iterative velocity and position updates, the particles converge towards optimal solutions, effectively optimizing the objective function [30], this concept will be further explored throughout this document.

Despite the widespread use of GAs, several alternative optimization techniques have been applied to similar problems in satellite attitude control, offering complementary approaches.

One such technique is Simulated Annealing (SA) [31], which draws inspiration from the physical process of annealing in metallurgy. In SA, the solution space is explored probabilistically by occasionally accepting suboptimal solutions to escape local minima. This approach is particularly useful for problems with complex, multimodal landscapes. While SA provides a simpler and more straightforward alternative to GAs, it tends to be slower and less efficient in large solution spaces.

Another technique that has gained attention is Ant Colony Optimization (ACO) [32]. Inspired by the foraging behavior of ants, ACO is effective in discrete optimization tasks, such as actuator placement, where solutions are sought in a combinatorial space. ACO has been applied to satellite optimization problems, offering an advantage in scenarios where a balance between exploration and exploitation is required. It can outperform GAs in certain applications where fine-tuned search behavior is crucial.

Furthermore, Differential Evolution (DE) [33] is another optimization technique used in satellite attitude control. DE relies on the mutation and recombination of solution candidates to evolve a population toward optimal solutions. It is particularly efficient for continuous optimization problems and has been applied to optimize satellite configurations, demonstrating their robustness in high-dimensional search spaces.

Although GAs are the primary focus of this dissertation due to their versatility and proven performance in complex, multi-variable optimization tasks, these alternative techniques, such as SA, ACO, and DE, offer valuable insights and could be considered in scenarios where GAs may face challenges, particularly with respect to speed or solution space dimensionality.

1.6. Methods Limitations

The modeling of orbital disturbances was carried out in a simplified manner, using average values for the satellite's exposed surface areas. This simplification was adopted to reduce the complexity of the code and minimize computational power consumption. Orbital propagation and attitude control were not simulated simultaneously. This decoupling further reduces the computational burden and allows for a faster evaluation of different configurations. As a result, the influence of attitude variation on the satellite's orbital trajectory was not considered.

Additionally, a constant average mass was assumed for the satellite throughout the orbital simulation. This assumption is justified by the structure of the genetic algorithm, which uses a precomputed time series of satellite positions as input. Since the number of reaction wheels, and hence the satellite's mass, may vary across generations in the PSO process, incorporating a time-varying mass would be computationally unfeasible within the iterative optimization framework.

The principal limitations underlying these methodological choices arise from the balance between control performance and computational efficiency. While the LQR lacks the robustness of more advanced techniques like H∞ control, it offers a significant advantage in simplicity. Advanced controllers such as H∞ require solving complex formulations (e.g., linear matrix inequalities) and often rely on iterative convex optimization solvers, which significantly increase implementation and runtime complexity. This overhead is particularly problematic in the context of PSO, where CONOPS based attitude maneuvers must be simulated repeatedly across many generations.

As for MPC, although it is capable of handling additional constraints and performing optimization over a prediction horizon, LQR provides a more computationally efficient solution by explicitly minimizing a quadratic cost function that balances control effort and attitude deviation. Moreover, the LEO environment is characterized by relatively benign and well-understood disturbances. In this context, LQR offers sufficient robustness and disturbance rejection, making the marginal benefits of more complex controllers unjustified for this application.

The PSO algorithm was selected as the optimization method due to its enhanced convergence behavior, particularly compared to other evolutionary algorithms. Its structure, where each particle retains a personal best (pBest) and the swarm is guided by a global best (gBest), representing the best-performing individual in the entire population. This dual mechanism promotes convergence toward high-quality solutions. Such convergence behavior is especially advantageous when optimizing configurations involving a large number of interdependent elements within a single individual, as is the case with reaction wheel configurations. In this context, PSO proves to be the most suitable evolutionary algorithm, offering an effective balance between exploration and convergence speed, where repeated simulations are required within a time-constrained optimization process, even when compared to other methods that may provide higher precision but at the cost of increased complexity or slower convergence.

1.7. Objective

The primary objective of this dissertation is to determine the optimal number, orientation, and position of RWs in a satellite's ACS. A genetic algorithm, specifically a PSO, is employed to optimize the RW configuration, aiming to minimize energy consumption while adhering to temporal constraints. This method is chosen due to its superior convergence capability, which is necessary for the successful execution of all required maneuvers, in a useful time window.

To evaluate the energy consumption of a given configuration for RW, a complete orbital period in a SSO is propagated, using a simplified orbital propagator, considering orbital disturbances such as atmospheric drag, J2, and solar radiation pressure. Additional assumptions include the use of a constant average satellite mass. Then, a set of representative attitude maneuvers, reflecting nominal mission operations, is executed using a controller based on LQR algorithms to quantify the energy consumed by the RWs.

This approach also reduces the computational load during the algorithm's execution in Python.

1.8. Document Structure

This dissertation follows the formal structure of an academic thesis and is organized as follows: The first chapter presents the orbital model used in the study. Then, the concepts of dynamics and kinematics related to satellite attitude are explored in depth, with particular emphasis on the use of quaternions. The subsequent chapter focuses on the fundamentals of control systems, providing a detailed explanation of the LQR control algorithm. Next, the Concept of Operations (CONOPS) for the mission is outlined.

Afterward, a chapter is dedicated to the reaction wheel consumption model (RWCM), followed by a section on the implementation of the ACS. Finally, the dissertation includes a chapter about the GA employed for optimization, another chapter for simulations and results, and a concluding chapter that summarizes the key findings and contributions of the work.

Chapter 2

2. Orbital Model

To propagate the trajectory of our satellite over time, an orbital model is required.

The orbital model for an Earth observation satellite in LEO is defined by several fundamental components. First, the model establishes an inertial coordinate frame, which serves as a fixed reference for describing the satellite's position and motion. The orbit itself is characterized using classical orbital elements that provide a complete description of the satellite's trajectory within its orbital plane.

The motion of the satellite is governed by equations of motion derived from Newton's laws of motion and gravitation. These equations are crucial for predicting the satellite's position and velocity over time. Additionally, the model can incorporate various orbital disturbances. Together, these components form a comprehensive framework for understanding and predicting the satellite's behavior in orbit.

The objective of this chapter is to explain the Orbital Model employed in this study. It begins by defining the Inertial Coordinate Frame, followed by an introduction to the concept of orbit using classical orbital elements. Additionally, the Equations of Motion utilized in the orbital propagator are presented. Finally, the chapter describes the key orbital disturbances considered to ensure the model closely reflects real-world conditions.

2.1. Inertial Coordinate Frame

A commonly used nonrotating right-handed cartesian coordinate system is the geocentric equatorial frame shown in Figure 2.1:

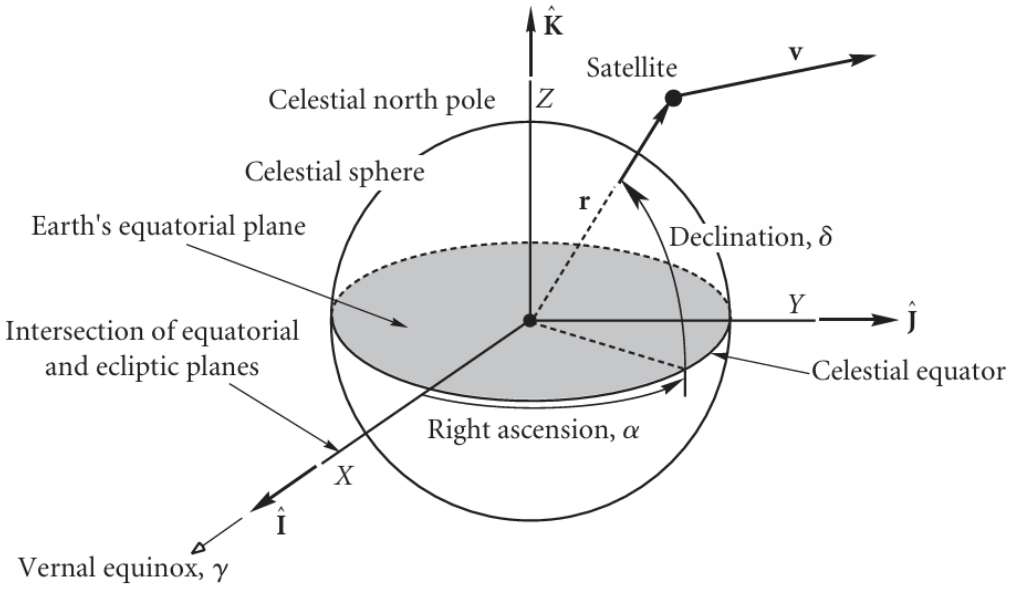

Figure 2.1: Motion of a Satellite Orbiting Earth in the Spherical Coordinate System

The X axis points in the vernal equinox direction. The XY plane is the earth's equatorial plane, and the Z axis coincides with the earth's axis of rotation and points northward. The unit vectors $\hat{I}$, $\hat{J}$ and $\hat{K}$ form a right-handed triad. The non-rotating geocentric equatorial frame serves as an inertial frame for the two-body earth satellite problem [18].

2.2. Definition of Orbit (Classical Orbital Elements)

To define an orbit in the plane requires two parameters: eccentricity and angular momentum. Other parameters, such as the semimajor axis, the specific energy, and (for an ellipse) the period, are obtained from these two. To locate a point on the orbit requires a third parameter, the true anomaly, which leads us to the time since perigee. Describing the orientation of an orbit in three dimensions requires three additional parameters, called the Euler angles, which are illustrated in Figure 2.2. [18].

In summary, the six orbital elements are:

-h, specific angular momentum

- i, inclination

- Ω, right ascension of the ascending node (RAAN)

- e, eccentricity

- ω_p, argument of perigee

- θ, true anomaly

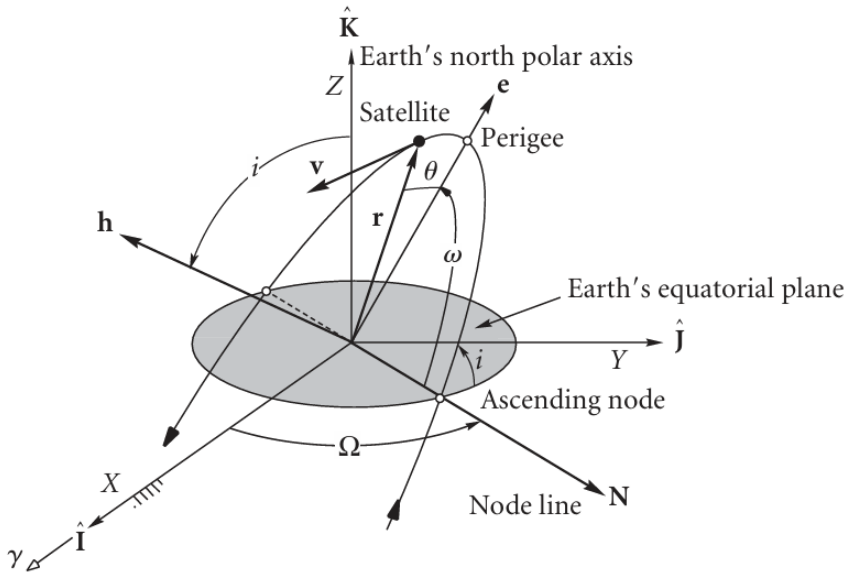

Figure 2.2: Orbital Elements

The angular momentum h and true anomaly θ are frequently replaced by the semimajor axis a and the mean anomaly M, respectively. Given the orbital elements in the geocentric equatorial frame, we can obtain the position $\vec{r}$ and velocity $\vec{v}$ of a satellite and vice versa.

2.3. Equations of motion

The orbital model considered herein follows a set of equations of motion which are numerically integrated to obtain the spacecraft position and velocity at each instant of time. In the geocentric equatorial frame, the orbital state vector is given by:

$$\vec{r} = X\hat{I} + Y\hat{J} + Z\hat{K} \tag{2.1}$$

$$\vec{v} = v_X\hat{I} + v_Y\hat{J} + v_Z\hat{K} \tag{2.2}$$

And:

$$\vec{x} = \begin{bmatrix}\vec{r}\\ \vec{v}\end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{bmatrix} \tag{2.3}$$

To arrive to a system of equations of the form [16]:

$$\frac{dx}{dt} = \begin{bmatrix}\dot{r}\\ \dot{v}\end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ -\mu\dfrac{x}{r^3} + a_{dx} \\ -\mu\dfrac{y}{r^3} + a_{dy} \\ -\mu\dfrac{z}{r^3} + a_{dz} \end{bmatrix} = f(t, x, \mu, a_d) \tag{2.4}$$

Where $a_d = a_d(t, x, \mu)$ are the orbital disturbances.

2.4. Orbital disturbances

The space environment encompasses all external influences and phenomena that affect spacecraft operations. This environment generates various forces and torques that disturb the spacecraft's dynamics, including gravitational gradients, solar radiation pressure, and atmospheric drag in LEO. Additionally, it contains multiple forms of radiation that can degrade electronic components through both cumulative dose effects and single-event phenomena [17].

In a LEO orbit, the most significant orbital disturbances arise from the Earth's oblateness, and from the atmospheric drag, which reduces the altitude over time [18]. Solar radiation pressure is another disturbance that cumulatively induces changes to the orbit's shape and orientation [34], and thus, it has also been considered.

In particular, the Earth's oblateness can be exploited in specific trajectories, such as Sun-synchronous orbits, to maintain alignment with the Sun [35]. However, the extent and utility of this effect depend on the satellite's mission objectives and the chosen orbital configuration.

There are other types of disturbances which affect spacecraft in LEO, such as the third-body disturbances, for instance the lunar and solar gravitational forces, and the Earth's gravity anomalies (higher-order harmonics), among others.

Table 2.1 represents the magnitude of each disturbance's contribution to acceleration, in LEO, arranged from the greatest to the lowest [34] [35]:

Table 2.1: Approximate Magnitude of Each Disturbance's Contribution to Acceleration in LEO

Disturbance	Acceleration Magnitude (in m/s²)	Comments
Atmospheric Drag	10^{-4} to 10^{-6}	Dominates long-term disturbances in LEO
J2 (Earth's Oblateness)	10^{-6} to 10^{-7}	Depends on altitude, atmospheric density, and satellite cross-sectional area.
Solar Radiation Pressure	10^{-7} to 10^{-9}	Affected by satellite reflectivity, surface area, and solar activity
Higher-Order Harmonics	10^{-8} to 10^{-10}	Higher-order harmonics of Earth's gravity field
Third-Body Disturbances	10^{-9} to 10^{-10}	Gravitational forces from the Moon and Sun

The exact values of the disturbance magnitudes depend on factors such as the satellite's altitude, design, and specific orbital parameters. In LEO, atmospheric drag and the J2 effect are the most influential disturbances affecting the satellite's trajectory. A

It should be noted that although solar radiation pressure is weaker compared to atmospheric drag and J2, it has periodic effects over time, which, in the case of eclipses, can gradually alter the orbit in a manner that is analytically complex to model. Third-body disturbances, such as the lunar and solar gravitational forces, along with Earth's gravity anomalies, are secondary disturbances in LEO. While their impact is generally minimal in comparison to the primary disturbances, they can become relevant for precise orbit modeling and more accurate predictions.

Only the atmospheric drag, the J2 and the solar radiation pressure were considered in order to reduce the complexity of the orbital model, recognizing that the three implemented ones have the greatest influence on the satellite's orbit, and their respective orders of magnitude are much higher than the third above.

2.4.1. Atmospheric Disturbances (Drag)

A primary source of disturbances to the satellite dynamics is the Earth's atmosphere, which generates drag on the spacecraft. Aerodynamic disturbances are due to the interaction between a planetary atmosphere and the spacecraft's surfaces. It is assumed that the interaction is by a stream of particles that do not interact with each other [17].

The simplest model for the aerodynamic force is:

$$\vec{F} = \frac{1}{2}\rho C_D A_p \vec{v}|\vec{v}| \tag{2.5}$$

Where ρ is the atmospheric density, C_D is the drag coefficient, A_p is the perpendicular projected area with the air flow, and $\vec{v}$ is the velocity vector. Lift is perpendicular to the velocity vector.

The projected area for a flat plate is:

$$A_p = A_{area} \cdot \cos(\alpha_1) \tag{2.6}$$

Where α_1 is the angle between the surface normal and the velocity vector.

In this dissertation analysis, the area is assumed to be constant to simplify the algorithm, specifically the lateral area of the spacecraft, assuming α_1 to be zero. The atmospheric drag was implemented using the "Poliastro API" in Python [36].

To achieve more accurate values for the atmospheric density affecting the satellite along its trajectory, the MSISE-00 atmospheric model was used. This model provides detailed information about atmospheric conditions based on the altitude and other parameters, allowing for precise calculations of the drag experienced by satellites in orbit. The MSISE-00 (Mass Spectrometer and Incoherent Scatter Extended 2000) is an empirical atmospheric model used to describe the Earth's atmosphere from the surface up to the exosphere, around 1,000 km altitude. It provides predictions of atmospheric temperature, density, and the composition of several atmospheric constituents [37].

To implement this model the python "PyNRLMSISE-00 API" [38] was used. When an incoming particle strikes the surface, considering a free molecular flow regime, it can be absorbed, reflected specularly or reflected diffusely [17]. The resulting actuation of incoming particles is shown in Figure 2.3:

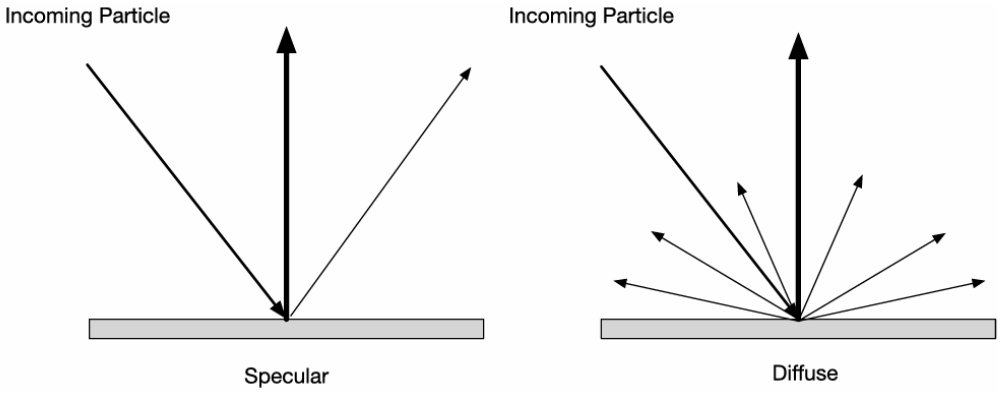

Figure 2.3: Specular and Diffuse Reflection Diagram

Figure 2.4 below presents a diagram illustrating the lift and drag forces acting on a plate:

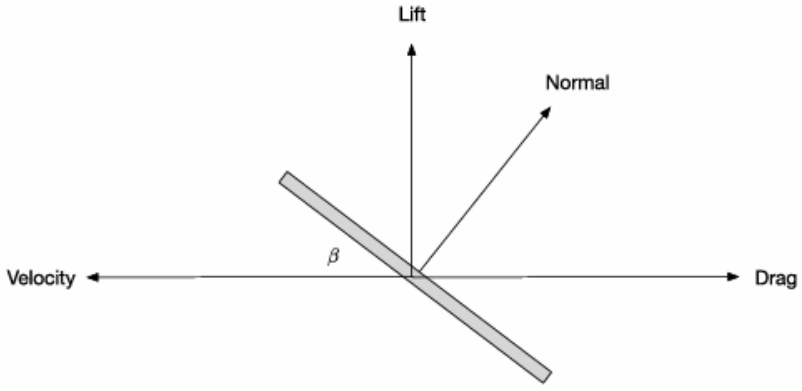

Figure 2.4: Lift and Drag on a Plate

A constant value for β was assumed to reduce the complexity of the algorithm. The β angle is the angle between the surface and the velocity vector, assuming a pure specular case with β =π/2, C_D will be equal to 4 [17]. The formulation containing the equations that lead to this conclusion is provided in the Appendix G.

2.4.2. Earth's oblateness (J2 Disturbances)

Earth is more accurately modeled as an oblate spheroid because it lacks the perfect symmetry of a sphere [18].

This lack of symmetry means that the force of gravity on an orbiting body is not directed towards the center of the earth. Whereas the gravitational field of a perfectly spherical planet depends only on the distance from its center, oblateness causes a variation also with latitude, that is, the angular distance from the equator (or pole). This is called a zonal variation. The dimensionless parameter which quantifies the major effects of oblateness on orbits is J2, the second zonal harmonic [18]. Earth J2 is equal to 1.08263×10^−3 [18].

The J2 disturbance causes the RAAN to gradually drift over time. In a SSO, this natural precession is leveraged to maintain a consistent angle between the satellite's orbital plane and the Sun. This alignment ensures that the satellite passes over the same region on Earth at the same local solar time on each orbit.

The differential equation that represents the variation of the RAAN from the J2 disturbance is given as [18]:

$$\dot{\Omega} = -\left[\frac{3}{2}\frac{\sqrt{\mu}J_2R_\oplus^2}{(1-e^2)^2a^{\frac{7}{2}}}\right]\cos i \tag{2.7}$$

The J2 effect also causes the argument of perigee to rotate within the orbital plane.

The differential equation that represents the variation of the perigee is given as [18]:

$$\dot{\omega}_p = -\left[\frac{3}{2}\frac{\sqrt{\mu}J_2R_\oplus^2}{(1-e^2)^2a^{\frac{7}{2}}}\right]\left(\frac{5}{2}sin^2i - 2\right) \tag{2.8}$$

Where R and μ are the radius and gravitational parameter of the planet, a and e are the semimajor axis and eccentricity of the orbit, and i is the orbit's inclination.

2.4.3. Solar radiation pressure disturbance

Solar pressure results from the momentum transferred by photons when impacting the surfaces of the spacecraft [17].

The solar pressure is expressed as:

$$s_{solar} = \frac{p}{c_{light}} \tag{2.9}$$

Where p is the power of the sunlight and c_{light} is the speed of light. As can be seen, a 1-N force would require 3×108 W.

In terms of fractions of the incoming photons, the following is true for a surface [17]:

$$1 = \rho_a + \rho_s + \rho_d + \rho_t \tag{2.10}$$

Where ρ stands for the fraction of photons that are absorbed (ρ_a), specularly reflected (ρ_s), diffusely reflected (ρ_d), or transmitted (ρ_t).

The solar pressure force is [17]:

$$F = -SA_{area}\vec{s}^T\vec{n}(2(\rho_s\vec{s}^T\vec{n} + \rho_d/3)\vec{n} + (\rho_a + \rho_d)\vec{s}) \tag{2.11}$$

Where $\vec{s}$ is the Sun vector, $\vec{n}$ is the unit vector normal to the surface, A is the area of the surface, and S is the solar flux in N/m².

Again, we consider a constant area and incidence angle ($\pi/2$), similar to the approach taken for atmospheric disturbance (drag), with the same objective of reducing the complexity of the algorithm.

The material used was Kapton Film (Aluminium-Aluminium Oxide Overcoating) 1 mil, with an absorptivity (ρ_a) of 0.12 and a reflectivity (ρ_s) of 0.8 [39].

In this dissertation, only the fractions of photons that are absorbed and specularly reflected are considered because the transmitted fraction is significantly reduced (approaching 0%) when the material is opaque [39]. Additionally, calculating the fraction of diffusely reflected photons is more complex, requiring laboratory experiments to confirm values, leading to disregarding diffusely reflected photons. The solar system ephemeris utilized is DE432S [40], which was implemented using the Python "Astropy API" [41].

A LIGHT algorithm [35] was used to determine whether the satellite is within the line of sight of the Sun (LOS). This algebraic approach involves defining a function dependent on the parameter τ, where $\tau \in [0\ ,1.0]$, such as:

$$\begin{cases} \overrightarrow{c_{LOS}}(0) = \overrightarrow{a_{SAT}} \\ \overrightarrow{c_{LOS}}(1.0) = \overrightarrow{b_{SUN}} \end{cases} \tag{2.12}$$

Where $\overrightarrow{a_{SAT}}$ is the satellite vector (from the center of the earth to the center of mass of the satellite), and $\overrightarrow{b_{SUN}}$ is the Sun vector (from the center of the Earth to the center of the Sun).

The goal is to determine τ_{min}, the value of τ where the function $|\overrightarrow{c_{LOS}}(\tau)|$ reaches its minimum. Consequently, this also gives the value of $|\overrightarrow{c_{LOS}}(\tau_{min})|$. However, in the specific case of this dissertation, will consider $|\overrightarrow{c_{LOS}}(\tau_{min})|^2$.

These values can be calculated as follows [35]:

$$\tau_{min} = \frac{|\overrightarrow{a_{SAT}}|^2 - \overrightarrow{a_{SAT}} \cdot \overrightarrow{b_{SUN}}}{|\overrightarrow{a_{SAT}}|^2 + \left|\overrightarrow{b_{SUN}}\right|^2 - 2 \cdot \overrightarrow{a_{SAT}} \cdot \overrightarrow{b_{SUN}}} \tag{2.13}$$

$$\frac{|\overrightarrow{c_{LOS}}(\tau_{min})|^2}{{R_\oplus}^2} = \frac{(1 - \tau_{min}) \cdot |\overrightarrow{a_{SAT}}|^2 + \left(\overrightarrow{a_{SAT}} \cdot \overrightarrow{b_{SUN}}\right)\tau_{min}}{{R_\oplus}^2} = \overrightarrow{c_{LOS}}(\tau_{min})_{R_\oplus} \tag{2.14}$$

Where for equation (2.14):

$$R_{\oplus} = 6378.137\ km \tag{2.15}$$

Defining the logical framework for the LIGHT algorithm:

LOS = FALSE

IF ($\tau_{min} < 0\ or\ \tau_{min} > 1$)

LOS = TRUE

ELSE

IF ($\overrightarrow{c_{LOS}}(\tau_{min})_{R_{\oplus}} \geq 1$)

LOS = TRUE

After having described the orbital model, which is used to propagate the satellite's orbital position and velocity, the Attitude model is considered next. Combining both models allows to represent the locations where the satellite is pointing, which will define the attitude profile for an Earth observation mission.

Chapter 3

3. Attitude

A satellite's attitude refers to its orientation in space, specifically how it is aligned relative to a reference frame such as the Earth or the stars. The attitude for the satellite operation, and for the success of the mission. Controlling and adjusting a satellite's attitude allows it to maintain proper alignment for operations such as communication, imaging, and maintaining a stable orbit.

The attitude model in this chapter considers the geocentric equatorial frame which was defined in Section 2.1 as the inertial coordinate frame.

This chapter begins by analyzing the attitude dynamics, kinematics and dynamics of a satellite equipped with RWs. Following this, the alignment method is explained, an essential element for determining the target quaternion based on the desired final attitude.

3.1. Attitude Representation

The orientation of one reference frame relative to another can be defined through various methods, with quaternions, Euler angles, and transformation matrices being among the most commonly used approaches. Figure 3.1 illustrates a representation of the body and inertial frames. In this dissertation, quaternions were chosen to represent the attitude due to their efficiency and ease of implementation.

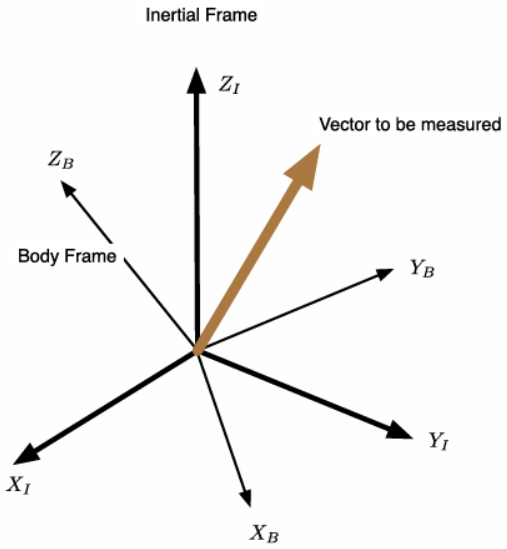

Figure 3.1: Body and Inertial Frames

A quaternion is a four-element vector that incorporates an axis and an angle of rotation. The vector orientation and length do not change, but it's x, y, and z components change depending on which frame they are measured [17].

3.1.1. Quaternions

Quaternions are among the most efficient methods for representing the orientation of one frame relative to another. Using quaternions to propagate an object's orientation is computationally efficient since only four components are propagated, without requiring trigonometric functions. Additionally, quaternions are free from singularities, making them particularly suited for numerical integration. Essentially, a quaternion is a four-dimensional parameter set that accurately describes the orientation of one reference frame with respect to a second reference frame.

The quaternion is represented by a four row-vector:

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} s \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} cos\frac{\phi}{2} \\ a_1 sin\frac{\phi}{2} \\ a_2 sin\frac{\phi}{2} \\ a_3 sin\frac{\phi}{2} \end{bmatrix} = q \tag{3.1}$$

The Gibbs vector of a quaternion is only the last three elements of the quaternion, excluding the scalar of the quaternion, as presented below:

$$\boldsymbol{g} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \tag{3.2}$$

3.1.2. Quaternion nomenclature

Quaternions can be written as:

$$q_{ab} \tag{3.3}$$

Which is defined as the quaternion that rotates a vector from frame b to frame a.

Thus:

$$u_a = q_{ab} \otimes u_b \tag{3.4}$$

And quaternion multiplication is defined as:

$$q_{ac} = q_{ab} \otimes q_{bc} \tag{3.5}$$

A more detailed explanation of quaternion operations is provided in the Appendix C.

3.2. Kinematics

Kinematics in a spacecraft refers to the expression and propagation of the satellite attitude, without considering any accelerations [17].

3.2.1. Quaternion Kinematics

The derivative of a quaternion is defined as:

$$\dot{q} = \lim_{\Delta t \to 0} \frac{\Delta q}{\Delta t} = \lim_{\Delta t \to 0} \frac{q(t + \Delta t) - q(t)}{\Delta t} \tag{3.6}$$

And as stated in [17], the same equation, using the angular velocity vector, can be written as:

$$\dot{q} = \frac{1}{2} \begin{bmatrix} 0 & -\vec{\omega}^T \\ \vec{\omega} & -\vec{\omega}^{\times} \end{bmatrix} q \tag{3.7}$$

3.3. Attitude Dynamics

The motion being discussed is the rigid-body rotation, a type of motion that is generally nonlinear across all three axes coupled. The rigid body under study is the gyrostat, with any number of symmetric rotors (RWs).

3.3.1. Inertia Matrix

The inertia matrix is a positive-definite-symmetric matrix that represents the resistance to the rotation of a spacecraft. The elements of the inertia matrix are in Appendix A and Appendix D.

It is useful to decompose a spacecraft into a set of point masses [17]. The inertia matrix for N point masses is:

$$I = -\sum_{k=1}^{N} m_k r_k^{\times} r_k^{\times} \tag{3.8}$$

Where:

$$r_k^{\times} = \begin{bmatrix} 0 & -z_k & y_k \\ z_k & 0 & -x_k \\ -y_k & x_k & 0 \end{bmatrix} \tag{3.9}$$

For point masses:

$$c_{mass} = \frac{\sum_{k=1}^{N} m_k r_k}{\sum_{k=1}^{N} m_k} \tag{3.10}$$

This method can be expanded to include rigid bodies instead of point masses:

$$I = \sum_{k=1}^{N} \left(B_k I_k B_k^T - m_k r_k^{\times} r_k^{\times} \right) \tag{3.11}$$

Where B_k transforms from the frame in which the inertia matrix is calculated to the spacecraft body frame.

3.3.2. Solid Box Inertia Matrix

In calculating the inertia matrix, it was assumed that all satellite components are solid boxes with uniformly distributed mass.

For a solid box the inertia matrix can be represented as follows:

$$I_k = \frac{m}{12}\begin{bmatrix} y_{box}^2 + z_{box}^2 & 0 & 0 \\ 0 & x_{box}^2 + z_{box}^2 & 0 \\ 0 & 0 & x_{box}^2 + y_{box}^2 \end{bmatrix} \tag{3.12}$$

For this equation, m is the mass, x_{box}, y_{box}, and z_{box} are box dimensions. The satellite bus is treated as a solid since it is full of all the bus components.

3.3.3. Rigid Body

The dynamical equations for a spacecraft can be derived by first writing the total angular momentum for the system in the inertial frame:

$$H = A_t h_b \tag{3.13}$$

Where h_b is the angular momentum in the body frame and the transformation matrix A_t transforms from the body-fixed coordinate frame to the inertial reference frame.

The torque on the spacecraft is:

$$A_t T = \dot{H} = \dot{A}_t h_b + A_t \dot{h}_b \tag{3.14}$$

$$\dot{A}_t = A_t \omega^{\times} \tag{3.15}$$

$$T = \dot{h}_b + \omega^{\times} h_b \tag{3.16}$$

Where T is defined in the body frame and ω is the angular rate of the body with respect to the inertial frame measured in the body frame. For a rigid body $h_b = I\omega$ so:

$$T = I\dot{\omega} + \omega^{\times} I\omega \tag{3.17}$$

Which is Euler's equation. I does not have a time derivative when the spacecraft is rigid.

Thus:

$$I\dot{\omega} = T - \omega^{\times} I\omega \tag{3.18}$$

These equations are written about the spacecraft's center-of-mass, which decouples the angular motion from the translational motion.

3.3.4. Gyrostat

A gyrostat is a spacecraft that is rigid except for symmetric rotors that spin about axes fixed to the body frame. Many spacecrafts are gyrostats since RWs can control the spacecraft's attitude without the need to expend propellant. A gyrostat is used to represent spacecraft with reaction wheels [17].

The momentum for a gyrostat is:

$$H = A_t\left(I\omega + \sum_k u_k J_k\left(\Omega_k + u_k^T\omega\right)\right) \tag{3.19}$$

Where Ω_k is the angular rate of the *kth* wheel about its spin axis, ω are the body rates, J_k is the spin-axis moment of inertia, u_k is the unit vector of the RW spin axis measured in the body frame, and I is the inertia of the rigid body excluding the RWs. A is a transformation matrix that is normally represented as a quaternion.

The equations of motion are then:

$$T = I\dot{\omega} + \omega^{\times}\left(I\omega + \sum_k u_k J_k\left(\Omega_k + u_k^T\omega\right)\right) + \sum_k u_k J_k\left(\dot{\Omega}_k + u_k^T\dot{\omega}\right) \tag{3.20}$$

$$T_k = J_k\left(\dot{\Omega}_k + u_k^T\dot{\omega}\right) \tag{3.21}$$

The transverse axes' inertias are lumped into the core body inertia. T is the body-fixed torque on the spacecraft (that in our case is null) and T_k is the scalar torque on each RW.

These equations give $3 + n$ equations of motion. To integrate these equations, solve for the angular acceleration of the core first and then substitute the angular-acceleration vector into the RW equations [17]:

$$\dot{\omega} = I^{-1}\left(T - \omega^{\times}\left(I\omega + \sum_k u_k J_k\left(\Omega_k + u_k^T\omega\right)\right) - \sum_k u_k T_k\right) \tag{3.22}$$

$$\dot{\Omega}_k = \frac{T_k}{J_k} - u_k^T\dot{\omega} \tag{3.23}$$

3.4. Alignment method

The alignment method is used to calculate the target quaternion, which is the desired final quaternion value that aligns the satellite according to the intended orientation.

Two important vectors are involved in the alignment method, the x and z vectors, that together represent the primary and secondary alignments, respectively.

Figure 3.2 presents a technical drawing of the satellite, illustrating the three vectors that define the satellite's body reference frame. Specifically, the $\vec{x}$ vector represents the primary alignment vector, and the $\vec{z}$ vector corresponds to the secondary alignment vector. The $\vec{y}$ vector is the remaining axis required to complete the orthogonal triad, defined according to the right-hand rule.

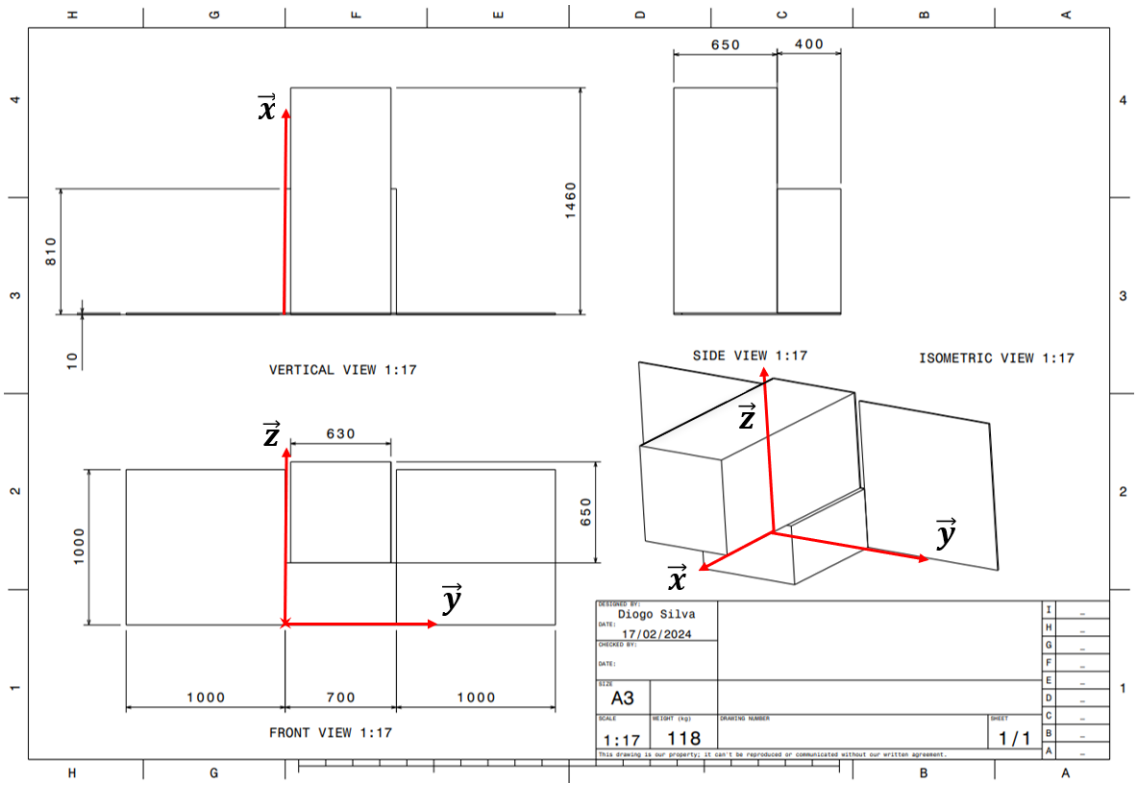

Figure 3.2: Technical Drawing of the Satellite

3.4.1. Calculation of the final auxiliary vector

The z_{final} vector is calculated from the initial x and z vectors, along with the x_{final}vector.

First, $x_{initial}$, x_{final} and $z_{initial}$ are converted into their respective unit vectors. Next, it is necessary to define a support vector to which z_{final} should align as closely as possible. By default, the support vector is equal to z_{initial}, though this can be adjusted if needed.

$$\vec{v}_{sup} = \vec{z}_{initial_{unit}} \tag{3.24}$$

Finally, an optimization problem is solved to determine z_{final}.

This minimization problem is defined as follows:

fun:

$$-\left(\vec{z}_{final} \cdot \vec{v}_{sup}\right) \tag{3.25}$$

$cons$:

$$\vec{z}_{final} \cdot \vec{x}_{final_{unit}} = 0 \tag{3.26}$$

$$\left\|\vec{z}_{final}\right\| = 1 \tag{3.27}$$

To solve this optimization problem, the "SciPy API" [42] was used and the SLSQP (Sequential Least Squares Programming) method was employed.

3.4.2. Calculation of the objective quaternion

For both the primary and secondary alignments, a body vector on the spacecraft is chosen to be aligned with an inertial vector. Let b_1 and b_2 be the body vectors for the primary and secondary alignments, respectively. Similarly, let u_1 and u_2 be the target inertial vectors.

The desired inertial to body quaternion for the primary alignment is found by computing the quaternion that rotates u_1 to b_1.

This quaternion is defined as:

$$q_{ROT1} = \begin{bmatrix} s/2 \\ d_x/s \\ d_y/s \\ d_z/s \end{bmatrix} \tag{3.28}$$

Where:

$$\vec{d} = \frac{\vec{u}_1 \times \vec{b}_1}{|\vec{u}_1||\vec{b}_1|} \tag{3.29}$$

And:

$$s = \sqrt{2\left(1 + \frac{\vec{u}_1 \cdot \vec{b}_1}{|\vec{u}_1||\vec{b}_1|}\right)} \tag{3.30}$$

The secondary alignment is achieved by rotating about vector b_1. Let this rotation be denoted by quaternion q_{ROT2}. The full ECI-to-body quaternion is therefore:

$$q_{EB} = q_{ROT1} q_{ROT2} \tag{3.31}$$

Where standard quaternion multiplication is used. The derivation of q_{ROT2} is provided in the next section.

Three-unit vectors, a, b, c, all defined in the body frame of the spacecraft, are considered. From the previous section, $a = b_1$ is defined as the axis about which rotation may occur. The vector $b = b_2$ is identified as the body-fixed vector to be aligned with vector c. Finally, c is defined as the secondary inertial target in the body frame by rotating u_2 with q_{EB}. The b vector is rotated about axis a so that b is as closely aligned with c as possible.

To closely align b with c, the separation angle between the two vectors needs to be minimized. The separation angle α is found from the dot product definition:

$$\cos(\alpha_2) = \vec{b}^* \cdot \vec{c}^* \tag{3.32}$$

Where b^* is the new value of b after rotation. Both vectors are expressed in the xyz coordinate system. The geometry is illustrated in Figure 3.3. A new coordinate system xyz is defined with x along a, $z = a \times b$, and y completes the right-hand system with $y = z \times x$. The b vector lies in the xy-plane. This plane is rotated through θ to place b^* in the same plane with a and c.

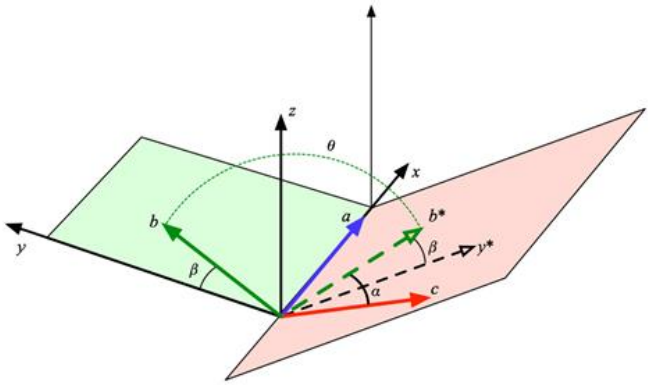

Figure 3.3: Plane Rotation Representation

The desired vector b^* expressed in the xyz frame is:

$$\vec{b}^* = \begin{bmatrix} x^T\vec{b} \\ y^T\vec{b}\cos(\theta) \\ z^T\vec{b}\sin(\theta) \end{bmatrix} \tag{3.33}$$

The target vector c^* expressed in the xyz frame is:

$$\vec{c}^* = \begin{bmatrix} x^T\vec{c} \\ y^T\vec{c} \\ z^T\vec{c} \end{bmatrix} \tag{3.34}$$

Combining equation (3.32) and equation (3.33) into equation (3.31), the following is obtained:

$$\cos(\alpha_2) = \left(x^T\vec{b}\right)(x^T\vec{c}) + \left(y^T\vec{b}\right)(y^T\vec{c})\cos(\theta) + \left(y^T\vec{b}\right)(z^T\vec{c})\sin(\theta) \tag{3.35}$$

To minimize α, the goal is to maximize $cos(\alpha)$. The maximum occurs when the derivative of cos(α) with respect to θ becomes zero.

$$\frac{d(\cos(\alpha_2))}{d\theta} = y^T\vec{b}(-\sin(\theta)\, y^T\vec{c} + \cos(\theta)\, z^T\vec{c}) = 0 \tag{3.36}$$

This leads to the simple solution of equation (3.35):

$$\tan(\theta) = \frac{z^T\vec{c}}{y^T\vec{c}} \tag{3.37}$$

With the axis and angle of rotation now defined, the quaternion to perform this rotation is:

$$q_{ROT2} = \begin{bmatrix} \cos(\theta/2) \\ -a_x\sin(\theta/2) \\ -a_y\sin(\theta/2) \\ -a_z\sin(\theta/2) \end{bmatrix} \tag{3.38}$$

The next chapter will focus on explaining the control techniques used in the context of attitude control, specifically the LQR.

Chapter 4

4. Fundamentals of Control Systems

Humans frequently engage with dynamic systems while performing tasks that may be routine or repetitive. As a result, there emerged a need to develop control techniques to execute specific actions to solve practical problems governed by mathematical equations.

Therefore, a dynamic system can be controlled through inputs, outputs, actuators, sensors, and a controller.

This dissertation focuses on controlling the spacecraft's attitude to ensure precise and efficient management. This chapter introduces control theory fundamentals and details the process for defining robust LQR controllers, which are central to the evaluation of the energy consumption of each configuration of reaction wheels.

4.1. State-Space

The state-space representation is a mathematical framework used to model and analyze dynamic systems by describing their behavior in terms of a set of state variables. These variables encapsulate the essential information about the system's past behavior, allowing for the prediction of future states. The state-space approach uses matrices to represent the relationships between inputs, outputs, and internal states of the system. This model is particularly well-suited for both continuous and discrete-time systems, especially when dealing with multi-input multi-output (MIMO) systems [43] [44].

This representation has become fundamental in control theory, allowing engineers and scientists to design, simulate, and control complex dynamic systems, including aerospace, robotics, and process control applications [45]. The model's ability to handle both linear and nonlinear systems, as well as time-varying systems, makes it a versatile tool in system analysis and design [46].

4.2. System Dynamics

Most dynamic systems can be mathematically represented through ordinary differential equations, where time is defined as the independent variable. These systems are represented in the form:

$$\dot{x} = f_{fun}[x(t), u(t), t] \tag{4.1}$$

Where f_{fun} represents the real function, t denotes the time variable, $x(t) \in R^n$ represents the state vector of the system, and $u(t) \in R^m$ represents the vector of control input variables to the system.

This system is dependent on the physical structure it represents and the physical equations to which it is subjected, arising from the inherent nature of the system. The output variable of the system is represented as $y(t) \in R^q$ and corresponds to the results generated by the system in response to the input data. The output variable $y(t)$ is represented in the form:

$$y(t) = g_{fun}[x(t), u(t), t] \tag{4.2}$$

Where g_{fun} represents a real function that can be defined around each equilibrium state. Thus, the system (state-space) defined by the previous equations is represented by:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \tag{4.3}$$

Where $\mathrm{A} \in R^{\mathrm{n}\times\mathrm{n}}$ is the system matrix, $\mathrm{B} \in R^{\mathrm{n}\times\mathrm{m}}$ the input matrix, $\mathrm{C} \in R^{\mathrm{q}\times\mathrm{n}}$ is the output matrix and $\mathrm{D} \in R^{\mathrm{q}\times\mathrm{m}}$ is the feedforward matrix.

4.3. Stability, controllability, and observability

It is essential to study stability, controllability, and observability of any dynamic system to assess its potential applicability. Various categories for dynamic systems are based on these three properties. Stability is crucial, as an unstable system requires significant resources for implementation, rendering it unviable. Conversely, a highly stable system may be undesirable due to limited controllability by the user. Controllability is closely related to the actuators implemented, which are responsible for introducing changes to guide the system towards a desired outcome. Observability pertains to the sensors implemented in the system to gather information about its evolution.

4.3.1. Stability

Assuming a linear system, it is a property of significant importance for a dynamic system. The usual method for verifying the stability of a system is through the determination of the eigenvalues of matrix A. A dynamic system described by $\dot{x} = \mathrm{A}x$ is stable if all the real parts of the eigenvalues of matrix A are less than zero, $Re(\lambda) < 0$. If the real parts of the eigenvalues are negative or equal to zero, the system is marginally stable.

4.3.2. Controllability

Controllability studies the potential of a system to be governed by input variables, representing the capacity for external intervention in the system's evolution. A system is controllable if it is possible to define an input that alters the variables from a specified initial state to a specified final state within a finite time interval, allowing the system to be guided from any state to another within a finite timeframe. A dynamic system defined in the aforementioned state-space is controllable if the rank of the controllability matrix equals n.

The controllability matrix is defined as follows:

$$C_0 = [B \quad AB \quad A^2B \quad \cdots \quad A^{n-1}B] \tag{4.4}$$

Where n_{dim} is the dimension of the state-space (the number of elements in the state vector x).

4.3.3. Observability

A system is considered observable if the values of the observation vector (output) at times $t^*, t^* + n\Delta t$ (with $n \geq 2$, where Δt is the observation time step) allow for a reliable estimation of the state vector at time t^*. In other words, a dynamic system is observable if the initial state can be determined from the various inputs and outputs over time. A dynamic system defined in the aforementioned state-space is observable if the rank of the observability matrix equals n.

The observability matrix is defined by:

$$\Theta = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{4.5}$$

Where n_{dim} represents the dimension of the state-space (the number of elements in the state vector x).

4.4. LQR

The optimal control of a dynamic system involves stabilizing the state, regulating, and controlling the output. Therefore, the goal of the controller is to determine the control actions that maximize or minimize the criterion constituting the optimization problem for the system under study. This section presents a method for designing optimal linear controllers based on quadratic performance criteria for multivariable systems with time-invariant parameters that remain constant around specific operating modes.

Thus, the following linearized dynamic system is considered:

$$\dot{x} = Ax + Bu \tag{4.6}$$

Where $x \in R^n$, $u \in R^m$, $A \in R^{n \times n}$, $B \in R^{n \times m}$, that is, the state vector, the control vector, the state matrix, and the control matrix.

To design a quadratic controller, it is necessary to select a control vector $u(t)$ such that a performance criterion J is minimized, where J is defined by:

$$J(u) = \int_0^\infty L(x, u)dt \tag{4.7}$$

Where $L(u, x)$ is a quadratic function of x and u. Since the intended controller is designed for linear systems, the resulting controller in this case is an LQR.

In LQR controllers, it is important to parameterize the control vector as a linear function of the state vector x, that is:

$$u_c = -Kx \tag{4.8}$$

Where K is a matrix with m rows and n columns ($K \in R^{m \times n}$).

Thus, the primary objective of the controller is to define the matrix K that minimizes the performance criterion (J) through the control function defined in equation (4.8).

The expression $L(x,u)$ must be quadratic in x and u and can be written in the following form:

$$L(x,u) = x^T Q x + u^T R u \tag{4.9}$$

Where $Q \in R^{n\times n}$ is a symmetric and positive semi-definite matrix (i.e., $\forall x = 0, Q(x) \geq 0$ and $Q(x) = 0$ for $x = 0$), and the matrix $R \in R^{m\times m}$ is positive definite (i.e., $\forall x = 0, R(x) > 0$ and $R(x) = 0$ for $x = 0$). Both matrices Q and R will affect the controller's performance. Therefore, their values should not be chosen arbitrarily but rather determined through various iterations and simulations.

The matrix Q is a diagonal matrix that depends on the number of elements in the state vector x.

$$Q = \begin{bmatrix} q_1 & 0 & \cdots & 0 \\ 0 & q_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & q_n \end{bmatrix} \tag{4.10}$$

For the present optimization problem, the matrix Q will be a 6×6 matrix, which can be defined as follows:

$$Q = \begin{bmatrix} q_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & q_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & q_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & q_6 \end{bmatrix} \tag{4.11}$$

Where its elements q_1, q_2, ..., q_6 can be defined in various ways. One example is the Modified Bryson Method [47]:

$$q_1 = \frac{\eta_1}{x_{1_{max}}^2} \tag{4.12}$$

$$q_2 = \frac{\eta_2}{x_{2_{max}}^2} \tag{4.13}$$

$$q_3 = \frac{\eta_3}{x_{3_{max}}^2} \tag{4.14}$$

$$q_4 = \frac{\eta_4}{x_{4_{max}}^2} \tag{4.15}$$

$$q_5 = \frac{\eta_5}{x_{5_{max}}^2} \tag{4.16}$$

$$q_6 = \frac{\eta_6}{x_{6_{max}}^2} \tag{4.17}$$

Where $x_{1_{max}}$, $x_{2_{max}}$, ..., $x_{6_{max}}$ are the maximum acceptable values for each state, and the values of $\eta_1, \eta_2, ..., \eta_6$ must be equal to or greater than 1 and can take different values.

The matrix R, similar to matrix Q, is also a diagonal and square matrix, with the number of rows and columns depending on the number of elements in the control vector u.

$$R = \begin{bmatrix} r_1 & 0 & \cdots & 0 \\ 0 & r_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & r_m \end{bmatrix} \tag{4.18}$$

For the dynamic model under study, the matrix R will be a 3×3 matrix and is defined as follows:

$$R = \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix} \tag{4.19}$$

Where r_1, r_2 and r_3 can be defined in various ways.

For the Modified Bryson Method, it is given that:

$$r_1 = \frac{\lambda_1}{u_{1_{max}}^2} \tag{4.20}$$

$$r_2 = \frac{\lambda_2}{u_{2_{max}}^2} \tag{4.21}$$

$$r_3 = \frac{\lambda_3}{u_{3_{max}}^2} \tag{4.22}$$

Where $u_{1_{max}}$, $u_{2_{max}}$ and $u_{3_{max}}$ are the maximum acceptable values for control, and the values of λ_1, λ_2 and λ_3 are defined between 0 and 1, taking any value within this range, but never reaching 0, while being able to equal 1.

It is important to note that the way matrices Q and R were defined above is merely one possible example for optimizing the controller.

Once $L(x,u)$ is defined, the equation (4.7), which defines J, becomes:

$$J(u) = \int_0^\infty (x^T Q x + u^T R u) dt \tag{4.23}$$

Thus, substituting equation (4.8) into equation (4.6) yields:

$$\dot{x} = Ax + B(-Kx) = Ax - BKx = (A - BK)x \tag{4.24}$$

Similarly, substituting equation (4.24) into equation (4.23):

$$J(u) = \int_0^\infty (x^T (Q + K^T R K) x) dt \tag{4.25}$$

Equation (4.25) corresponds to the closed-loop system equation, which describes how the controller operates.

To determine the control law, it is necessary to find the Lyapunov scalar function V, for the closed-loop system in the form $V(x) \equiv x^T P x$, where P is a symmetric positive definite matrix.

The Lyapunov function is used to analyze the stability of equilibrium points in dynamical systems. Intuitively, it is like an "energy-like" function that decreases over time for a stable system. The time derivative of the Lyapunov function must equal the negative of the function to be integrated in the equation, that is:

$$\dot{V}(x) \equiv \frac{\partial}{\partial t}(x^T P x) = -x^T (Q + K^T R K) x \tag{4.26}$$

$$\frac{\partial}{\partial t}(x^T P x) = \dot{x}^T P x + x^T P \dot{x} \tag{4.27}$$

Thus, it is obtained:

$$x^T[(A - BK)^T P + P(A - BK)]x = -x^T (Q + K^T R K) x \tag{4.28}$$

For the previous differential equation to be stable, the matrix K must satisfy the following Lyapunov equation:

$$(A - BK)^T P + P(A - BK) = -(Q + K^T R K) \tag{4.29}$$

Where it is assumed that the matrix K is unknown. Thus, the solution obtained is:

$$K = R^{-1} B^T P \tag{4.30}$$

Where the matrix P is obtained from the Riccati equation. Therefore, the control function obtained is:

$$u_c = -Kx = -R^{-1} B^T P x \tag{4.31}$$

To obtain the solution to the Riccati equation, it is necessary to ensure that the matrices in question meet the stability and controllability conditions mentioned earlier. Thus, the matrix P used in equation (4.31) can be derived from the Riccati equation:

$$A^T P + PA - PBR^{-1} B^T P + Q = 0 \tag{4.32}$$

The LQR described above is used to control the attitude of the spacecraft such that it points to desired targets. These targets are determined by the set of tasks that the satellite must perform throughout its mission. These tasks collectively form the CONOPS, which will be explored in greater detail in the following chapter.

Chapter 5

5. Mission Control Profile

The mission control profile comprises a series of orientation maneuvers or “pointing modes” that the satellite is expected to successfully execute during its nominal operation. This mission profile is represented in a CONOPS.

This chapter describes the mission profile, considered for assessing the energy efficiency of each ACS configuration. The spacecraft is assumed to be in nominal operations mode, which is dominant mode throughout the lifetime of the mission. In the sequel, each task, the respective assumptions, and their distribution within a representative orbital period are given in more detail.

5.1. CONOPS

A CONOPS is a document describing the characteristics of a proposed system from the viewpoint of an individual who will use that system [48].

This is a high-level document that outlines the intended operational framework of a system or project. It provides a clear description of how the system is expected to function in real-world scenarios, serving as a bridge between technical specifications and practical usage [49].

This dissertation considers an Earth observation mission whose CONOPS can be divided into three main parts: the launch and early operations, the nominal operations, and the end-of-life operations. Since nominal operations are planned to represent the largest portion of the CONOPS, the simulation considers the tasks executed in such phase of the mission, through a predefined time expand, one orbital period.

5.2. Tasks

The tasks that the satellite will perform are as follows.

5.2.1. Sun Pointing Maneuver

The Sun Pointing maneuver aligns the satellite’s solar panels directly toward the Sun. This orientation maximizes solar energy collection, ensuring optimal power generation for the satellite. For this maneuver, the principal alignment vector is the direction vector $\overrightarrow{[SUN][SAT]}$, and the secondary alignment is the satellite’s orbital velocity vector, $\vec{v}$.

5.2.2. Imagery Acquisition Maneuver

The Imagery Acquisition maneuver positions the satellite's optical sensors (telescope, etc.) on a specific target area on Earth. This is crucial for our Earth observation mission, where high-quality data collection over designated regions is necessary and mandatory. In this maneuver, the satellite's optical sensors must remain pointed at their targets for a specified duration or instantaneously, depending on the characteristics of the payload's sensors. In this dissertation, to simplify the Imagery Acquisition maneuver, it was chosen to point toward the target instantaneously.

For this maneuver, the principal alignment vector is the $\overrightarrow{[SAT][target]}$ and the secondary alignment corresponds to the satellite's tangential orbital velocity vector, $\vec{v}_{\perp}$. The vector $\overrightarrow{[SAT][target]}$ connects the satellite to a point located in the plane perpendicular to the tangential orbital velocity. Therefore, the primary alignment is always associated with a rotation angle, with $\vec{v}_{\perp}$ serving as the axis of rotation.

5.2.3. Nadir Pointing Maneuver

The Nadir Pointing maneuver involves orienting the satellite so that its optical sensors will point directly downward toward the Earth's Nadir Point.

For this maneuver, the principal alignment vector is the inverse of the satellite's orbital position vector, $-\vec{r}$, and the secondary alignment corresponds to the satellite's tangential orbital velocity vector, $\vec{v}_{\perp}$.

5.3. Mission Profile

The simulation time will be one orbital period: 94.616 minutes. This is based on the orbital parameters used, which will be discussed in greater detail in the following chapter.

To reduce the time required for each simulation, we opted to simulate only a select set of maneuvers during one orbital period. This approach minimizes the need for higher computational power, allowing us to obtain simulation results more swiftly.

Below, Table 5.1 summarizes all the tasks that constitute the mission profile. The table includes a column with numerical IDs for each task, additional columns indicating the duration of each task (both in absolute terms and relative to one orbital period), and finally, a column for comments.

Table 5.1: Mission Profile Tasks Constitution

Sequence ID	Task Name	Duration (min)	Total/Duration	Comment
1	Sun Pointing	11.83	0.125	30%
2	Imagery Acquisition	3.94	0.042	15 degrees
3	Sun Pointing	7.88	0.083	20%
4	Imagery Acquisition	3.94	0.042	15 degrees
5	Sun Pointing	19.71	0.208	50%
6	Nadir Pointing	47.31	0.500	Half Orbit

Figure 5.1 shows a boxplot that represents each task chronologically which facilitates the understanding of the temporal sequence of events in the mission and provides insights into the timing and duration of each task. The shaded area in Figure 5.1 is intended to represent the portions of the orbit where the satellite is not illuminated by the Sun, and is therefore in the Earth's shadow.

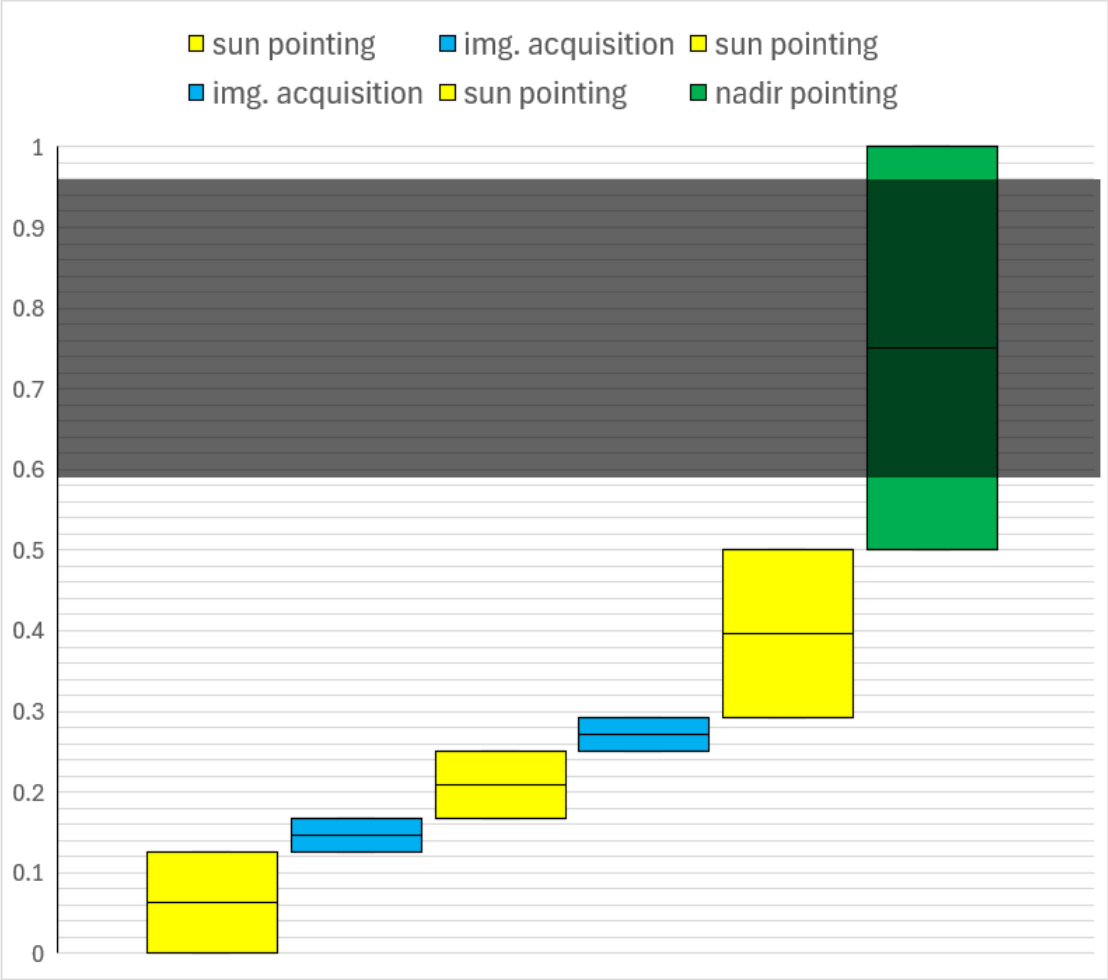

Figure 5.1: Boxplot Representing Each Task Duration Chronologically

Additionally, Figure 5.2 shows a pie chart with the distribution of tasks with respect to the percentage of time they are executed in each orbit. This provides an intuitive and straightforward visualization of the predominance of specific tasks. In the context of this dissertation, nadir pointing is identified as the most dominant task.

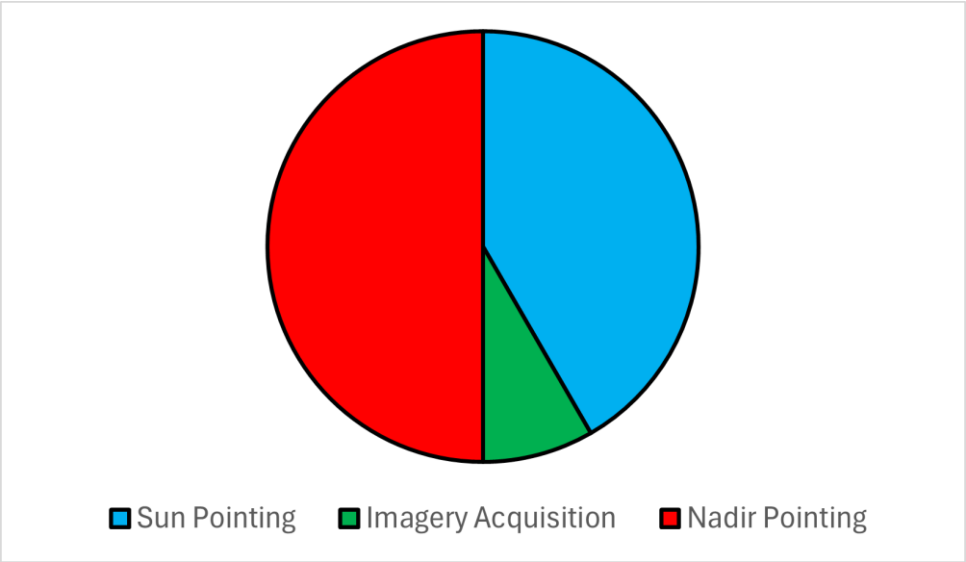

Figure 5.2: CONOPS Tasks Composition

Now that the CONOPS of the mission to be simulated in this dissertation has been presented, the focus of the following chapters will be to explain the various components of the system that must be implemented to simulate the mission, starting with the chapter on the "Reaction Wheel Consumption Model."

Chapter 6

6. Reaction Wheel Consumption Model

The purpose of this chapter is to detail the development and implementation of the RWCM.

The cost function of the genetic algorithm involves summing the total energy consumed by the RWs over a complete orbit. For this purpose, it is necessary to develop a RWCM to accurately estimate the energy usage, the RWCM implemented calculates the energy consumption in Joules.

The data utilized for the calculation of our model was sourced from a datasheet provided by NewSpace Systems [14].

6.1. Consumption model

This is the model that is used to calculate the energy consumed over a defined time interval:

$$E_c(\overline{\omega}, T, h) = \left((a * \overline{\omega} + b) + (c * \overline{\omega} + d) * (e * \overline{\omega} + f) * \left(\frac{T}{g}\right) \right) * h_{step} \qquad (6.1)$$

The energy consumption function depends solely on the average speed of the reaction wheel over the defined time interval and the torque applied throughout the entire time interval. Here, $\overline{\omega}$ represents the average speed during the integration step, denoted as h.

6.2. Data Analysis

By analyzing the consumption data of the RWs along with their operational constraints, sufficient information is obtained to perform the necessary calculations and determine the coefficients for the consumption model equation.

6.3. Calculation method

To simplify the calculations, the average speed is used for computing the energy consumption, knowing that the time step should be reduced.

Additionally, an instantaneous torque output was considered, which further simplifies the algorithm by eliminating the need for a State-Space model and the associated complexities.

The time step is considered as the interval for calculating the average speed.

6.3.1. LIBRA-065 and LIBRA-2

To calculate the relationship between angular velocity and consumption in steady state, a linear regression was performed using the consumption data in quiescent and steady states at a fixed angular velocity.

Next, to account for the effects of torque on consumption, the equation for the instantaneous power of a body undergoing angular acceleration was used.

$$\omega * T = P_{C,T} \tag{6.2}$$

Assuming there are no energy losses, as the datasheet does not provide any reference to the mechanical efficiency of the actuators.

The RWCM is specific to these two reaction wheels:

$$E_c\left(\overline{\omega}, T, h_{step}\right) = (a * \overline{\omega} + b + \overline{\omega} * T) * h_{step} \tag{6.3}$$

6.3.2. LIBRA-6

To calculate the relationship between angular velocity and steady-state consumption, a linear regression was performed using the consumption data in quiescent and steady-state conditions with a constant angular momentum.

To determine the angular velocity of the reaction wheel for a given angular momentum, considering the rotor's moment of inertia, the following formula was used:

$$H = J * \omega \tag{6.4}$$

Similarly to what was done for the smaller inertia wheels, a linear regression was calculated to establish the relationship between steady-state consumption and angular velocity.

Then, considering the Peak Power Consumption (PPC) (which occurs at maximum speed and maximum torque), it is possible to calculate the torque consumption constant, K.

The torque consumption constant, K, is calculated using the formula:

$$\mathrm{K} = \frac{(\%)_{Torque}}{1 - (\%)_{Torque}} \tag{6.5}$$

At 100% of the maximum torque, where:

$$(\%)_{Torque} = 1 - \frac{Max\ (\omega) Steady\ State\ Consumption}{PPC} \tag{6.6}$$

The RWCM for this reaction wheel is:

$$E_c\left(\overline{\omega}, T, h_{step}\right) = \left((a * \overline{\omega} + b) + (a * \overline{\omega} + b) * c * \left(\frac{T}{d}\right)\right) * h_{step} \tag{6.7}$$

Knowing equation (6.5), equation (6.7) c and d is obtained as follows:

$$c = K \tag{6.8}$$

$$d = T_{max} \tag{6.9}$$

6.3.3. LIBRA-80

For this last reaction wheel, all calculations were performed similarly to those for the LIBRA-6, except for the calculation of the Torque consumption constant, K. For this reaction wheel, K depends on the angular velocity of the RW.

Thus, it is necessary to calculate two constants K: one for the PPC situation and another for Peak Torque at a given angular momentum.

Both constants are calculated similarly to the LIBRA-6, and finally, a linear regression is performed to express K as a function of the angular velocity.

The RWCM for this reaction wheel is:

$$E_c\left(\overline{\omega}, T, h_{step}\right) = \left((a * \overline{\omega} + b) + (a * \overline{\omega} + b) * (e * \overline{\omega} + f) * \left(\frac{T}{d}\right)\right) * h_{step} \qquad (6.10)$$

Where for equation (6.10):

$$d = T_{max} \qquad (6.11)$$

6.4. Used Model

Finally, Table 6.1 presents all the coefficients used in equation (6.1) for each of the RWs. The calculations for these coefficients are in Appendix H.

Table 6.1: Reaction Wheel Consumption Model Coefficients

X	LIBRA-065	LIBRA-2	LIBRA-6	LIBRA-80
a	0.0049	0.0122	0.0582	0.0583
b	2.6	1.7	5.1	4.7
c	1	1	0.0582	0.0583
d	0	0	5.1	4.7
e	0	0	0	-0.0017
f	1	1	2.92428	4.8956
g	1	1	0.31	0.3

Figure 6.1 presents a diagram that shows a simplified version of the RWCM used:

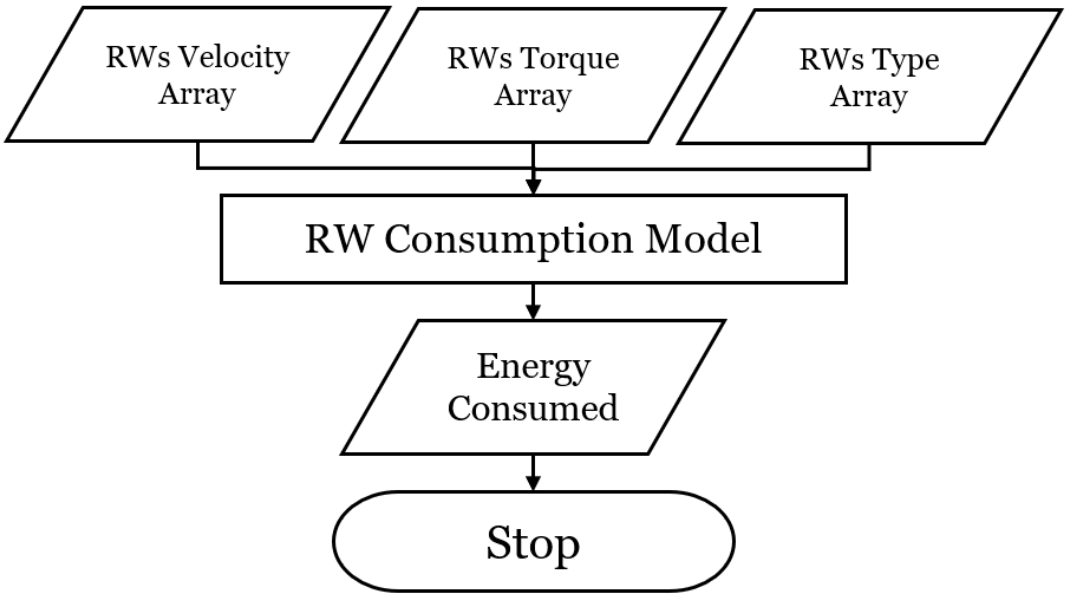

Figure 6.1: Reaction Wheel Consumption Model Diagram

Other key elements of the system, in addition to the RWCM, will be explained in the next chapter, which focuses on the simulation of the control system as a whole.

Chapter 7

7. Control System Implementation

This chapter begins by explaining the key aspects of the reference vector calculation, also denoted as objective vector, where the computation of the satellite's position vector and the quaternion reference vector over time will be discussed sequentially.

Subsequently, the "Dynamic Model" employed in this study is introduced. The construction of the matrices that form the state-space representation is then detailed. Furthermore, key considerations regarding the implementation of the Dynamic Model within the developed algorithm are examined in depth. Additionally, the design and implementation of the RW Torque Distributor algorithm are described. Finally, the discretization process is explained, followed by the presentation of a diagram that graphically represents the control system.

7.1. Reference Vector Calculation

The output vector used is the Gibbs vector of the state vector, which is the sole vector considered in the calculation of the error vector for the LQR controller. Therefore, the objective vector is the Gibbs vector corresponding to the desired quaternion vector for the satellite, referred to as the quaternion objective vector.

7.1.1. Position Vector Calculation

To compute the reference quaternion vector, the position vector time series is required which in turn requires an orbital propagator. The orbital model described in Chapter 2 was implemented numerically to obtain such time-series.

Figure 7.1 presents the orbital propagator used:

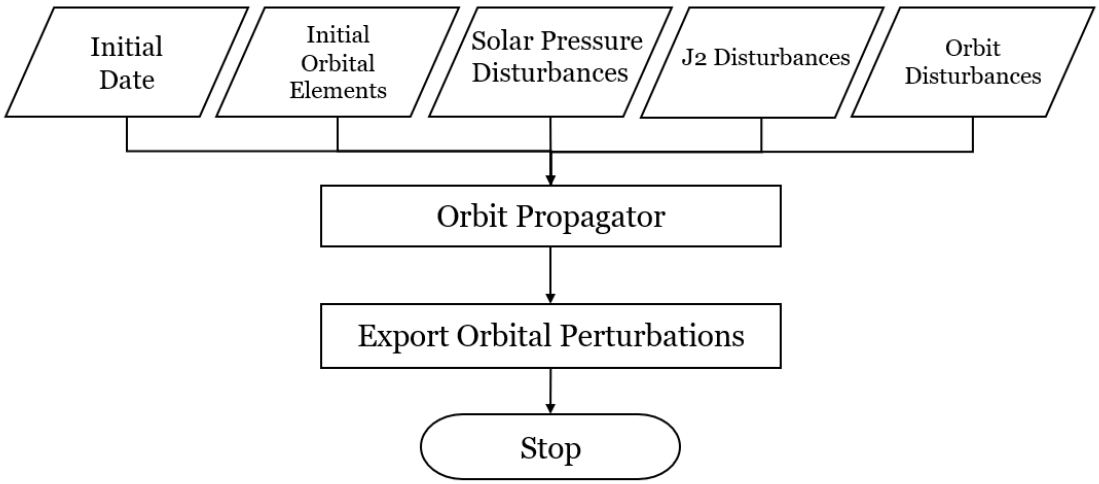

Figure 7.1: Orbital Propagator Diagram

7.1.2. Reference Quaternion Vector Calculation

After calculating the position vector time series, the objective vector must be determined by combining the mission CONOPS with the position vector time series. The tasks outlined in the mission CONOPS influence the alignment of both the primary and secondary vectors, which must be achieved in different positions of the satellite, the celestial bodies, and the observation targets. Finally, a reference quaternion vector calculator was employed, which is presented in Figure 7.2 below:

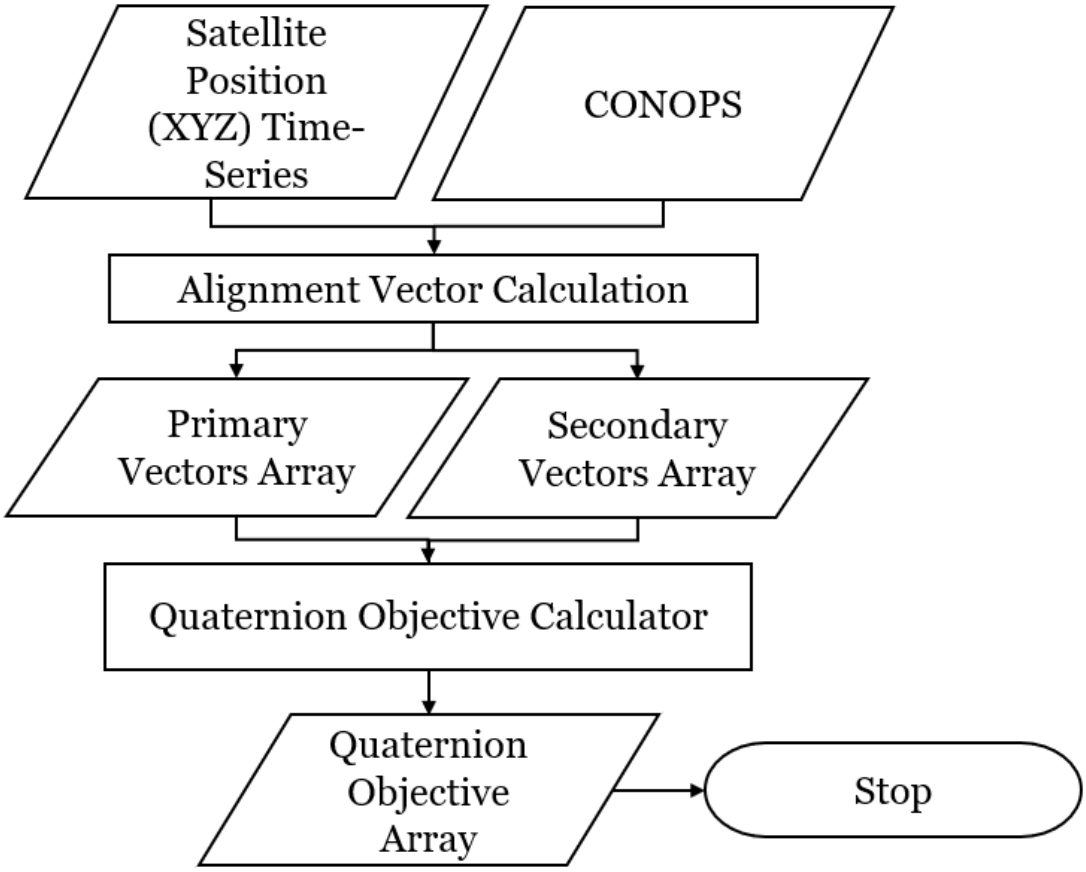

Figure 7.2: Quaternion Objective Vector Calculator Diagram

Due to the simplified method used for calculating the reference quaternion (Section 3.), the reference quaternions are computed based on the initial values of the primary and secondary alignment vectors, [1, 0, 0] and [0, 0, 1], respectively. This approach is adopted to minimize the required computational cost and to avoid the accumulation of errors that may arise from repeatedly performing operations with high associated numerical uncertainty, such as matrix cross products, which are frequently required when working with quaternions. Due to the method used, it is possible that the reference quaternion calculator returns, for the reference quaternion, the representation with the smallest rotation angle around the quaternion axis, which may result in discontinuities in the reference quaternion graph.

7.2. Dynamic Model

As previously mentioned, the representation adopted for the dynamic model is the state-space. The main components of the implemented state-space are now more concretely defined. The dynamic system's objective is to propagate the differential equations of attitude and angular velocity over time (in a discretized manner). The step size utilized remains constant throughout the simulation.

$$P = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \tag{7.1}$$

In equation (7.1), P is the state-space matrix (Plant matrix).

7.2.1. State Vector

The state vector is:

$$x(t) = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{7.2}$$

Thus, the state vector combines the elements of the quaternion, excluding the scalar part, as well as the three components of angular velocity. The scalar component of the quaternion was excluded, as it does not form part of the Gibbs vector. In fact, the quaternion elements included in the state vector are those corresponding to the Gibbs vector. A justification for this choice is provided later in this chapter.

7.2.2. Control Vector

The control vector is:

$$u(t) = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} \tag{7.3}$$

Consequently, the control vector comprises the three components of the total torque provided by all the RWs.

7.2.3. Output Vector

The output vector is:

$$y(t) = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{7.4}$$

Therefore, the output vector consists of the three elements of the quaternion, excluding the scalar part of the quaternion.

7.3. State-Space Matrices

7.3.1 Matrix A

The matrix A used in equation (7.1) is:

$$A = \begin{bmatrix} 0_{3\times3} & A_1 \\ 0_{3\times3} & A_2 \end{bmatrix} \tag{7.5}$$

To calculate the A_1 matrix the quaternion differential equation must be linearized. As stated in [50], this equation can be linearized as follows:

$$\frac{d}{dt} g = \frac{1}{2} I_{3\times3} \omega \tag{7.6}$$

Where:

$$g = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{7.7}$$

Thus A_1 used in equation (7.5) is:

$$A_1 = \frac{1}{2} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7.8}$$

To determine the appropriate time step for the integrator, various step sizes were tested. It was ultimately decided to use a step size of 0.1 seconds, as this value provides sufficient accuracy while not demanding excessive computational power.

Thereby ensuring that the simulations are not overly time-consuming. The results of these tests are detailed in the Appendix I. To calculate the A_2 matrix, it is necessary to compute the Jacobians of the following gyrostat differential equation:

$$\dot{\omega} = I^{-1} \left(T - \omega^{\times} \left(I\omega + \sum_k u_k J_k (\Omega_k + u_k^T \omega) \right) - \sum_k u_k T_k \right) \tag{7.9}$$

$$T_{k,Total} = \sum_k u_k T_k \tag{7.10}$$

The cross product, in this context, is a nonlinear operation; therefore, the Jacobians of the expression inside the parentheses, which is multiplied by I^{-1}. In the model used, T is assumed to be null.

$$f = \omega^{\times} \left(I\omega + \sum_k u_k J_k (\Omega_k + u_k^T \omega) \right) \tag{7.11}$$

The Jacobians in order to ω are, as stated in Appendix B:

$$J_f(\omega_1, \omega_2, \omega_3) = \begin{bmatrix} \frac{\partial f_1}{\partial \omega_1} & \frac{\partial f_1}{\partial \omega_2} & \frac{\partial f_1}{\partial \omega_3} \\ \frac{\partial f_2}{\partial \omega_1} & \frac{\partial f_2}{\partial \omega_2} & \frac{\partial f_2}{\partial \omega_3} \\ \frac{\partial f_3}{\partial \omega_1} & \frac{\partial f_3}{\partial \omega_2} & \frac{\partial f_3}{\partial \omega_3} \end{bmatrix} = \begin{bmatrix} A_{2,11} & A_{2,12} & A_{2,13} \\ A_{2,21} & A_{2,22} & A_{2,23} \\ A_{2,31} & A_{2,32} & A_{2,33} \end{bmatrix} \tag{7.12}$$

The results for $J_f(\omega_1, \omega_2, \omega_3)$ are in the Appendix F.

Finally, A_2 used in equation (7.5) is:

$$A_2 = -I^{-1} \begin{bmatrix} A_{2,11} & A_{2,12} & A_{2,13} \\ A_{2,21} & A_{2,22} & A_{2,23} \\ A_{2,31} & A_{2,32} & A_{2,33} \end{bmatrix} \tag{7.13}$$

7.3.2. Matrix B

This structure results in the inverse inertia matrix placed inside a larger matrix of zeros, multiplied by minus one.

The B matrix used inequation (7.1) is:

$$B = \begin{bmatrix} 0_{3\times3} \\ -I^{-1} \end{bmatrix} \tag{7.14}$$

7.3.3. Matrix C

Ideal sensors are assumed; thus, an identity matrix is implemented within a larger matrix of zeros.

The C matrix used in equation (7.1) is:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{7.15}$$

7.3.4. Matrix D

It was assumed, for the system used, that the control vector would have no influence on the calculation of the output vector.

The D matrix used equation (7.1) is:

$$D = [0_{3\times3}] \tag{7.16}$$

7.4. Considerations

White noise was added to the input data to introduce uncertainty into the system, thereby simulating noise measurement. Such noise approximates real-world conditions due to sensor inaccuracies and environmental interference. Adding this element enhances the model's realism, aligning it with typical measurement scenarios.

For the simulations, the fourth-order Runge-Kutta integrator (RK4) was utilized. Its implementation was carried out using the "Scipy API" [42], with a time step of 0.1 seconds, as previously mentioned.

7.5. Reaction Wheel Torque Distributor

As defined in the control vector, the controller only determines the total control torque to be applied by the actuators.

Therefore, a RWTD must be developed and implemented to minimize the imbalance in the RWs' speeds and reduce the risk of RW saturation (reaching maximum angular speed), which is detrimental as it increases energy consumption and reduces system control capability.

To address this, a RWTD was designed to distribute the required torque across all RWs, minimizing the sum of their angular velocities while respecting their maximum angular speeds and torque outputs.

Developing the optimal torque distribution system is not the primary focus of this dissertation. As a result, a simpler torque distribution method was designed and implemented for our specific case.

Aside from the inputs related to the RWs and the system state, the RWTD consists of three main elements:

1. Solver 1

2. Solver 2 & Observer 1

3. Solver 3

These three elements are executed sequentially, and if any of them succeed, the RWTD is completed and interrupted. The outputs of the RWTD include the RW speed array, RW torque array, and the system state elements. In the event of failure in all three elements, the maneuver simulation is interrupted, leading to the cancellation of the mission simulation.

7.5.1. Solver 1

The operation of "Solver 1" is based on minimizing the sum of the magnitudes of the torque from the RWs in order to match the control torque requested by the LQR controller, while always respecting the operational limits of the RWs.

Below is the logical framework of Solver 1:

SOLVER 1

min:

$$\sum {T_k}^2$$

eq:

$$\sum (u_k T_k) - T_{control} = 0$$

$ineq$:

$$\Omega_{k,max} - \left(\left(\frac{T_k}{J_k} - u_k^T \dot{\omega} \right) \cdot h_{step} + \Omega_{k,0} \right) > 0$$

$$\left(\left(\frac{T_k}{J_k} - u_k^T \dot{\omega} \right) \cdot h_{step} + \Omega_{k,0} \right) - \Omega_{k,min} > 0$$

IF $Solver\ 1_{success} = TRUE$:

$$\Omega_{k,1} = \left(\frac{T_k}{J_k} - u_k^T \dot{\omega} \right) \cdot h_{step} + \Omega_{k,0}$$

ELSE:

$$Execute\ Solver\ 2\ \&\ Observer\ 1;$$

Figure 7.3, presents the Sover 1 used:

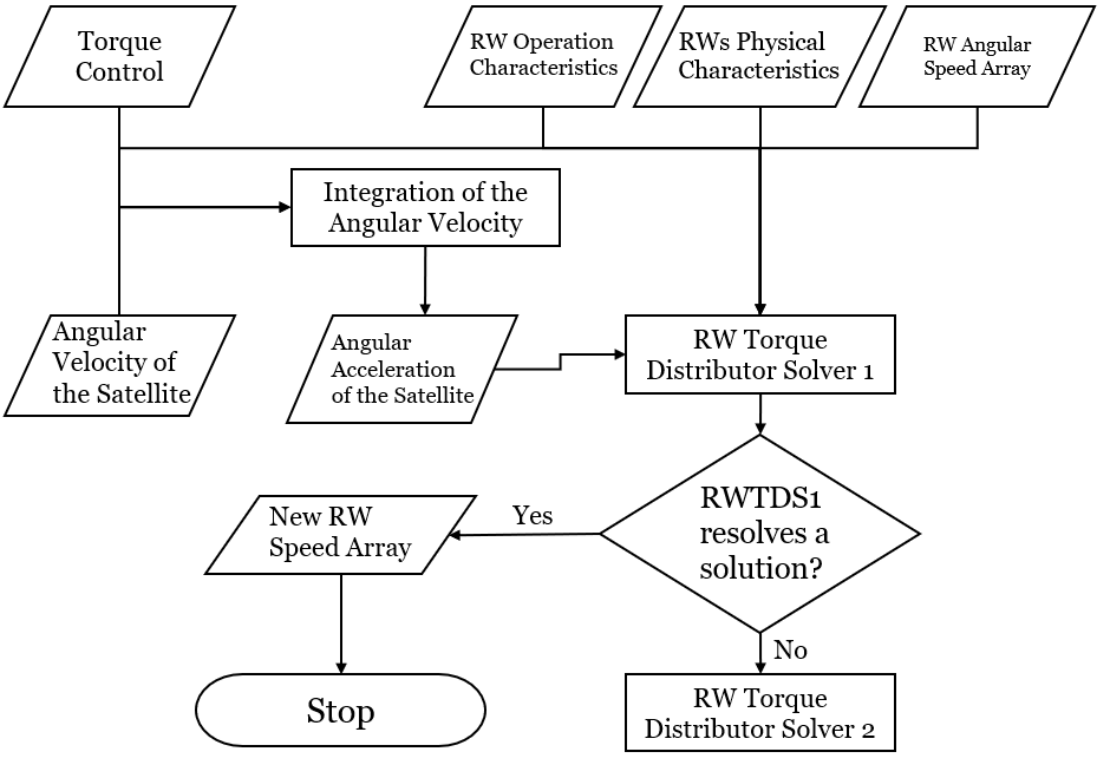

Figure 7.3: Reaction Wheel Torque Distributor Solver 1

The RW physical characteristics include the max. RW Torque array, the max. RW speed torque array and the min. RW speed array.

7.5.2. Solver 2 & Observer 1

The operation of "Solver 2" is based on maximizing the norm of the sum of the torques from the RWs, while respecting the direction of the initial control torque. This results in a new control torque, requiring the angular velocity of the satellite to be integrated once again, thereby obtaining a new angular acceleration. "Observer 1" is then used to verify whether the operational constraints of the RWs are being met.

Below is the logical framework of Solver 2 & Observer 1:

SOLVER 2

max:

$$\sum \|u_k T_k\|$$

eq:

$$dot\left(\frac{\sum u_k T_k}{\sum \|u_k T_k\|}, \frac{T_{control}}{\|T_{control}\|}\right) - 1 = 0$$

$ineq$:

$$\sum \|u_k T_k\| > 0$$

$$NEW\_T_{control} = \sum \|u_k T_k\|;$$

$$Integrate\ to\ calculate\ the\ NEW\_\dot{\omega};$$

OBSERVER 1

IF $\left[\Omega_{k,max} - \left(\left(\frac{T_k}{J_k} - u_k^T \dot{\omega}\right) \cdot h_{step} + \Omega_{k,0}\right) > 0\right]$ &&

$\left[\left(\left(\frac{T_k}{J_k} - u_k^T \dot{\omega}\right) \cdot h_{step} + \Omega_{k,0}\right) - \Omega_{k,min} > 0\right]$:

$$Observer\ 1_{success} = TRUE;$$

$$\Omega_{k,1} = \left(\frac{T_k}{J_k} - u_k^T \dot{\omega}\right) \cdot h_{step} + \Omega_{k,0}$$

ELSE:

$$Observer\ 1_{success} = FALSE;$$

$$Execute\ Solver\ 3;$$

Figure 7.4, presents the Sover 2 & Observer 1 used:

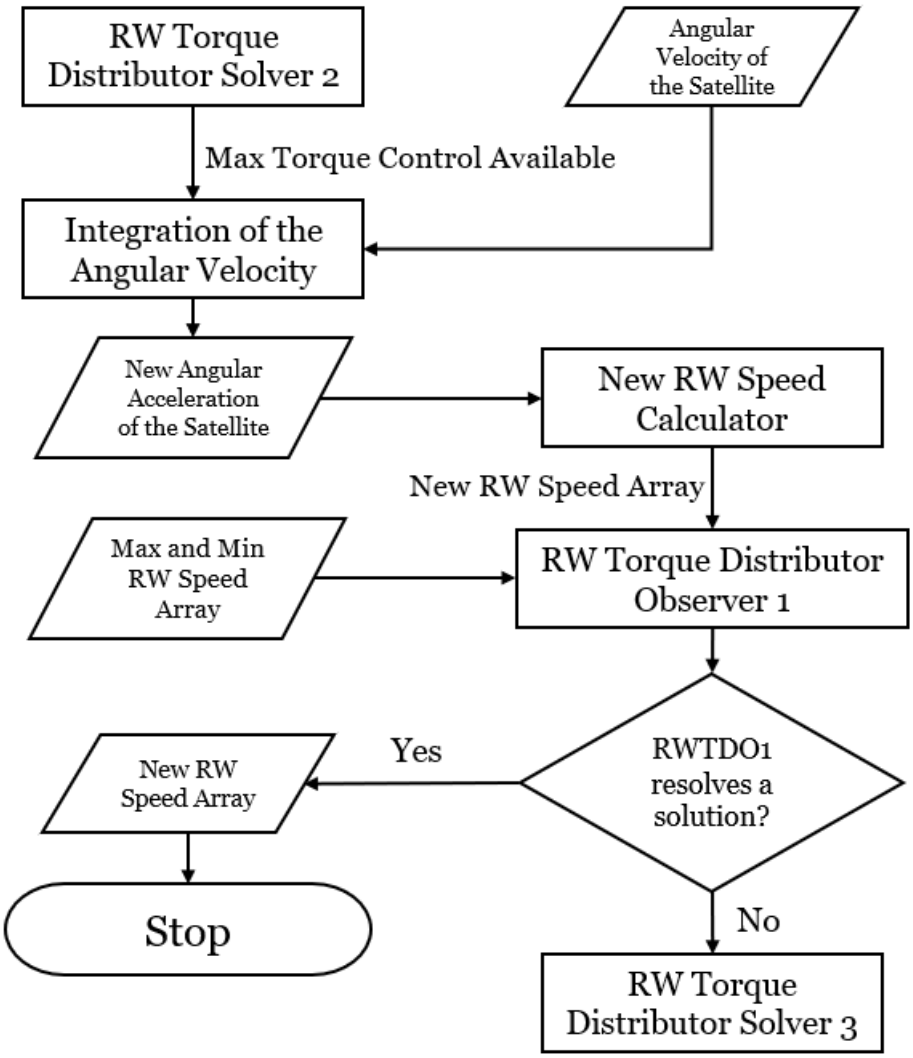

Figure 7.4: Reaction Wheel Torque Distributor Solver & Observer 1

7.5.3. Solver 3

The operation of "Solver 3" is based on the assumption that the satellite's angular acceleration is zero, with the control torque also being zero. Given this, the goal is to maximize the norm of the sum of the torques from the RWs, while maintaining the direction of the initial control torque, and ensuring that the operational constraints of the RWs are respected.

Below is the logical framework of Solver 3:

SOLVER 3

$\dot{\omega}_{null} = [0,0,0]$;

max:

$$\sum \|u_k T_k\|$$

eq:

$$dot\left(\frac{\sum u_k T_k}{\sum \|u_k T_k\|}, \frac{T_{control}}{\|T_{control}\|}\right) - 1 = 0$$

$ineq$:

$$\sum \|u_k T_k\| > 0$$

$$\frac{\left(\Omega_{k,max} - \Omega_{k,0}\right)}{h_{step}} - \frac{T_k}{J_k} + u_k^T \dot{\omega}_{null} > 0$$

$$\frac{T_k}{J_k} - u_k^T \dot{\omega}_{null} - \frac{\left(\Omega_{k,min} - \Omega_{k,0}\right)}{h_{step}} > 0$$

IF $Solver\ 3_{success} = TRUE$:

$$\Omega_{k,1} = \left(\frac{T_k}{J_k} - u_k^T \dot{\omega}\right) \cdot h_{step} + \Omega_{k,0}$$

$$NEW_{T_{control}} = [0,0,0];$$

ELSE:

$$BREAK; \#\ Cancel\ Misson\ Simulation$$

Figure 7.5, presents the Sover 3 used:

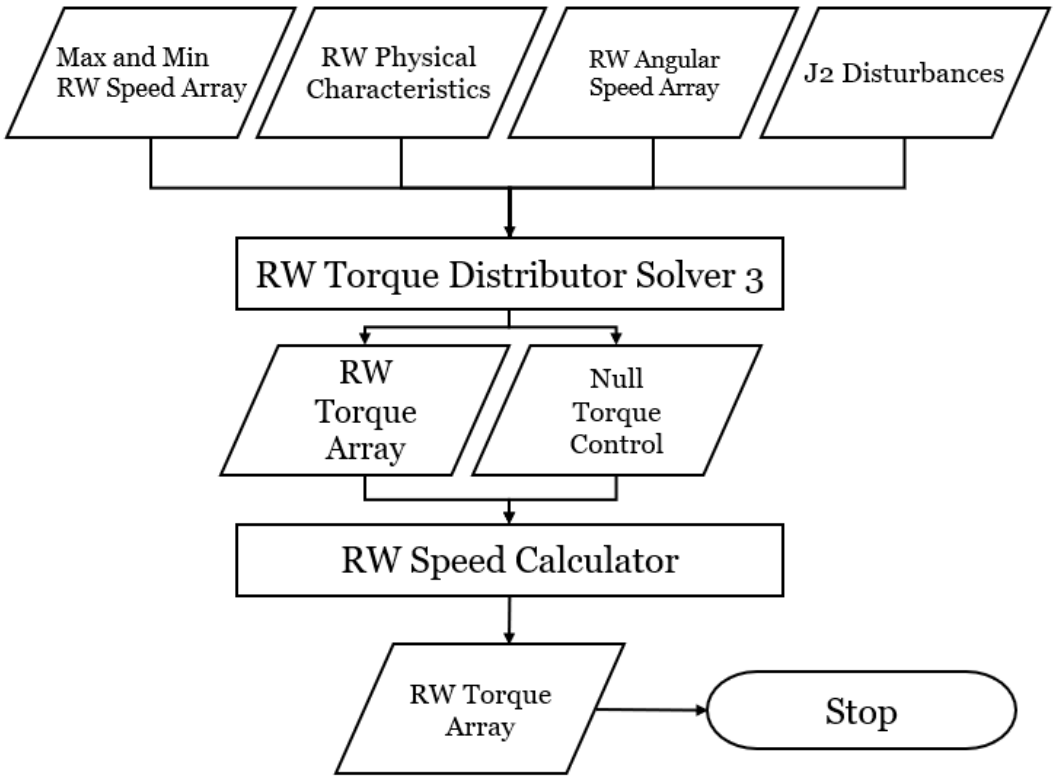

Figure 7.5: Reaction Wheel Torque Distributor Solver 3

7.6. Discretization

Dynamic models in state-space representation are often modeled in a discrete manner due to the nature of digital control systems, which operate with sampled data rather than continuous signals.

Digital processors in control applications typically execute at fixed intervals, requiring the system's continuous dynamics to be discretized to maintain alignment with this sampling rate. Discrete modeling allows for practical implementation of state control algorithms. This ensures stability and accuracy in computations when interacting with digital hardware or software controllers.

To ensure accuracy in the calculations, a time step of 0.1 seconds was chosen. The rationale behind selecting this value will be discussed in more detail later in this chapter.

To discretize the state-space model, was used the "control.ss()" function from the Python Control API [51].

7.7. Control system diagram

Figure 7.6 illustrates the control system designed to manage the satellite's attitude throughout the entire duration of the mission.

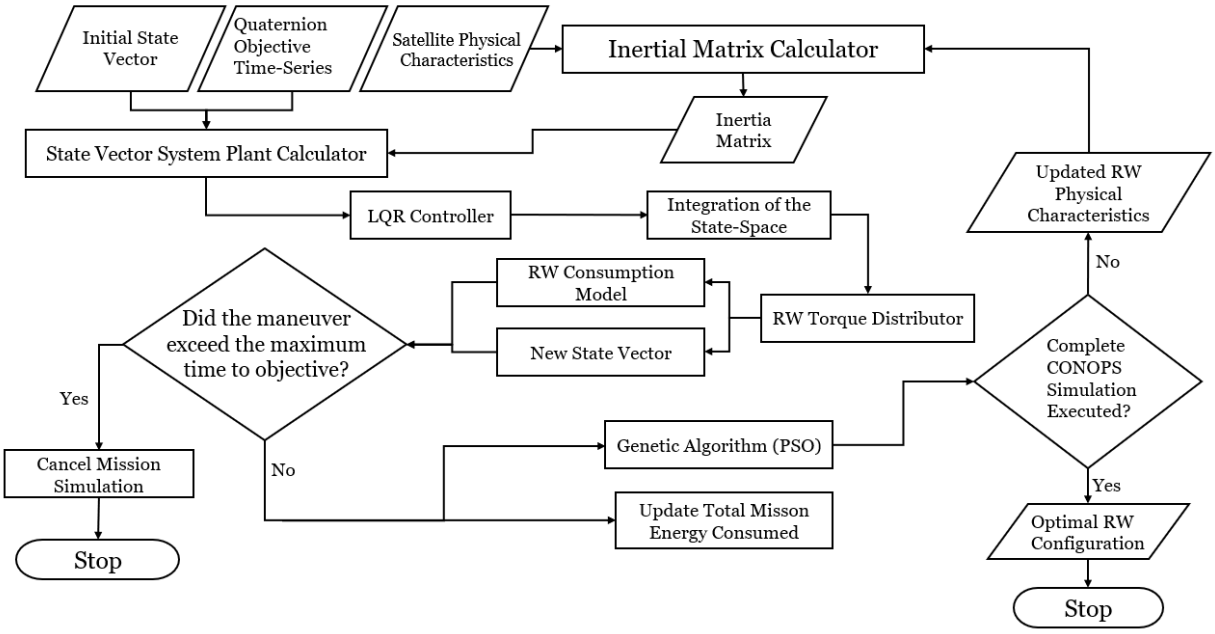

Figure 7.6: Control System Diagram

Starting from the initial state vector of the satellite and the physical characteristics of both the RWs and the satellite, the state-space system controls the state vector using an LQR controller, based on the quaternion objective time series. After integrating the state vector, the torque is distributed among all the RWs within the utilized RW configuration, and the RWCM calculates the energy consumed during each time step, which is used for the state-space integration.

Additionally, an algorithm is employed to verify whether the system has achieved the desired attitude orientation. If not, the control system checks whether there is sufficient time to achieve the objective; the cycle then repeats until the CONOPS is completed. If this is not the case, the mission simulation is interrupted, and the simulation is not excluded from the considered results.

Upon successful execution of the mission CONOPS, the control system collects data on the total energy consumed by the satellite and the time to reach the objective for each task in the CONOPS. This data is then utilized by the GA, which is further developed in the next chapter.

Now that the control system has been implemented, it must be integrated into the framework of the genetic algorithm used, which is based on the concept of a Particle Swarm Optimization (PSO). The general concept of genetic algorithms, and more specifically PSO, will be explored in greater detail in the next chapter, along with the models of the RW configurations used.

Chapter 8

8. Genetic Algorithm

GAs are valuable for addressing problems with numerous variables as they efficiently explore complex solution spaces by simulating the process of natural selection, enabling the identification of optimal or near-optimal solutions without exhaustive evaluation of all possibilities.

A GA works by mimicking the process of natural evolution to find optimal solutions. It starts with a population of potential solutions, represented as chromosomes. Through iterative processes of selection, crossover (recombination), and mutation, better solutions evolve over generations. The algorithm selects the fittest individuals to breed and generate offspring, gradually improving the solution set [28].

This chapter aims to present the key elements that constitute a Genetic Algorithm, namely the “Individual”, the “Population”, and the “Fitness” function, as detailed in the subsection "Genetic Algorithm Core Elements". Subsequently, the evolutionary strategy to be employed, the PSO, will be explained in the subsection "Genetic Algorithm Strategy". Finally, the chapter concludes with a discussion of the RW configurations used in this dissertation: the "Cube Configuration" and the "Pyramid Configuration."

8.1. Genetic Algorithm Core Elements

8.1.1. Individual (particle)

Simply by thinking about the different flavors of evolutionary algorithms, we notice that an extremely large variety of individuals are possible [52].

In the context of a GA, an individual represents a single candidate solution to the optimization problem being solved [28]. Each individual is typically encoded as a chromosome, consisting of genes that represent the variables or parameters of the solution [52].

A particle is a special type of individual as it usually has a speed and generally stores its best position [52].

This type of individual was chosen because the aim is to leverage the Particle Swarm Optimization (PSO) to solve the problem.

8.1.2. Population (swarm)

Populations are much like individuals. Instead of being initialized with attributes, they are filled with individuals, strategies or particles [52].

A swarm is used in PSO. It is different in the sense that it contains a communication network. The simplest network is the completely connected one, where each particle knows the best position that has ever been visited by any particle. This is generally implemented by copying that global best position to a g_{best} attribute and the global best fitness to a $g_{best\,fit}$ attribute [52].

8.1.3. Fitness

Fitness is a measure of quality of a solution [52].

This is typically defined by the user; in our case, it refers to the total energy consumption by the actuators of the ACS during all maneuvers executed throughout a trajectory. Our objective is to minimize this total (minimization problem), while being aware that there is a maximum time for the maneuver to be successfully executed. Thus, our cost function will be influenced by the total maneuver time and the energy consumed in the aggregate of the maneuvers.

In a GA, the fitness of an individual quantifies its quality in solving the optimization problem. The fitness function evaluates how well an individual meets the problem's objectives, with higher scores indicating better solutions [28]. Fitness values are essential for the selection process, as individuals with higher fitness are more likely to reproduce, guiding the evolutionary process toward optimal solutions [53].

8.2. Genetic Algorithm Strategy

A GA evolution strategy is an optimization approach that focuses on evolving a population of candidate solutions using mechanisms inspired by natural evolution. It involves selection, where the best individuals are chosen to reproduce; crossover, where genetic material is exchanged between individuals; and mutation, where small changes are introduced to maintain diversity in the population [28].

Over generations, this process leads to improved solutions through “survival” of the fittest [52].

8.2.1. Particle Swarm Optimization

The PSO offers several advantages over traditional GAs. Firstly, PSO is generally easier to implement, as it does not require complex genetic operators like crossover and mutation, which are essential in GAs [30]. Additionally, PSO tends to converge more quickly compared to GAs; particles share information directly with one another, reducing the time needed to explore the solution space [54].

Furthermore, PSO requires fewer parameters to tune than GAs, primarily focusing on particle velocity and position update increments, making it simpler to control and adjust [55]. This optimization approach efficiently balances exploration, searching new areas, and exploitation, refining known good solutions, by guiding particles with both personal and collective experience [56]. Finally, unlike GAs, PSO does not rely on fitness sorting or selection processes, which can be computationally expensive [29].

Overall, these attributes make PSO a compelling alternative for solving optimization problems.

8.3. Genetic Algorithm Diagram

A visual representation of the GA utilized is presented in Figure 8.1 below:

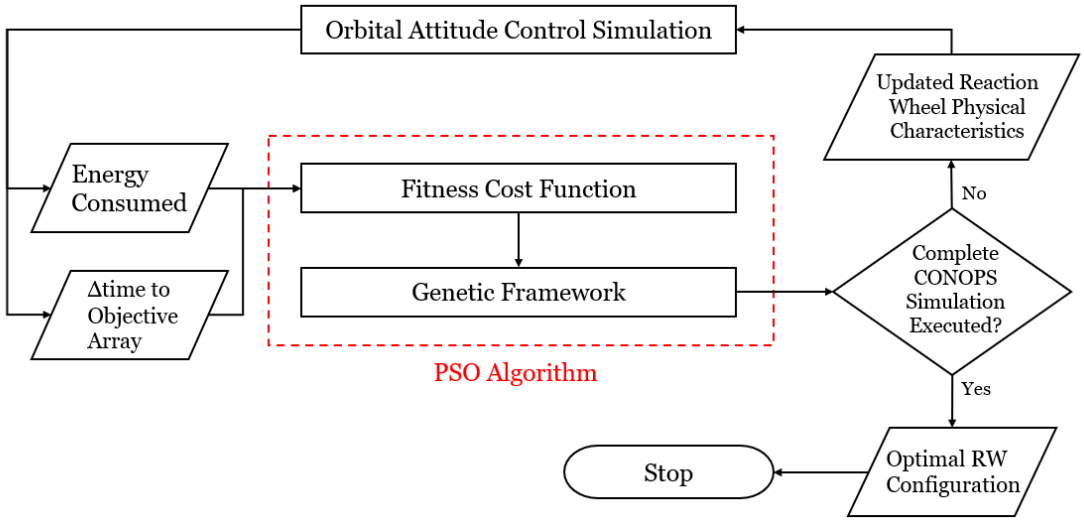

Figure 8.1: Genetic Algorithm Diagram

To provide a deeper explanation of the genetic algorithm diagram, we must begin with the GA framework, which creates and updates the population of individuals. This population consists of a set of RW configurations that must be simulated using the control system algorithm. The output of these simulations is the total energy consumed by each simulated RW configuration, which, along with the time required to complete each maneuver, is analyzed by the GA's fitness function. Returning to the GA framework, this process occurs cyclically until all generations of GA are simulated, resulting in the final product of the created tool: the optimal RW configuration.

8.4. Reaction Wheel Architecture

The actuators responsible for the crucial function of attitude control must be positioned within the satellite's structure. These actuators must adhere to the satellite's dimensions and spatial constraints imposed by other critical components already installed, such as the payload. The Reaction Wheel Architecture aims to reduce the complexity of allocating reaction wheels, as implied by its name, while also providing the user with a greater understanding of the cause-effect relationship regarding the actuators' positions and the efficiency of the maneuvers performed by the satellite.

8.4.1. Cube Configuration

The cube configuration involves positioning the RWs on the faces of a cube. This setup allows for balanced distribution of torque and enhances control over the satellite's orientation along multiple axes.

This setup allows for robust control of the spacecraft's pitch, yaw, and roll axes. By utilizing fewer RWs, the cube architecture ensures redundancy and enhances the system's reliability, as it can maintain functionality even if one RW fails. The cube configuration also facilitates a good distribution of the generated torques.

The Individual used to represent the Cube Configuration will be defined as follows:

$$Individual\text{: } [\ conf\_type\ , rw\_type\ , alfa\ , x\ , y\ , z\ , r\ ]$$

The definition of each parameter constituting the Individual for the Cube Configuration is provided in Table 8.1:

Table 8.1: Cube Configuration Individual Constants Definition

Cube Conf. Constant	Constant Values	Unit	Comments
conf_type	{0,1,2,3}	N/A	0: Four RWs are positioned on the four lateral faces of the cube; 1: Five RWs, with four on the lateral faces of the cube and one on the top face; 2: Five RWs, with four on the lateral faces of the cube and one on the bottom face; 3: Six RWs, with one placed on each face of the cube.
rw_type	{0,1,2,3}	N/A	0: The RWs used are "LIBRA-065"; 1: The RWs used are "LIBRA-2"; 2: The RWs used are "LIBRA-6"; 3: The RWs used are "LIBRA-80".
alfa	[0,45]	degree	Rotation around the Oz axis.
x	$\mathbb{R}$	mm	Variation in the x-coordinate of the cube's CG relative to the reference CG.
y	$\mathbb{R}$	mm	Variation in the y-coordinate of the cube's CG relative to the reference CG.
z	$\mathbb{R}$	mm	Variation in the z-coordinate of the cube's CG relative to the reference CG.
r	$\mathbb{R}$	mm	The "cube radius", defined as half the length of one side of the cube.

Note: The rotation axis of each RW is aligned with the vector connecting the center of the configuration to each RW.

Note: The reference CG is assumed to correspond to the CG of the "Instruments Box".

Figure 8.2 provides a graphical representation of the Cube Configuration, illustrating the six possible positions for the RWs, assuming zero rotation around the Oz axis.

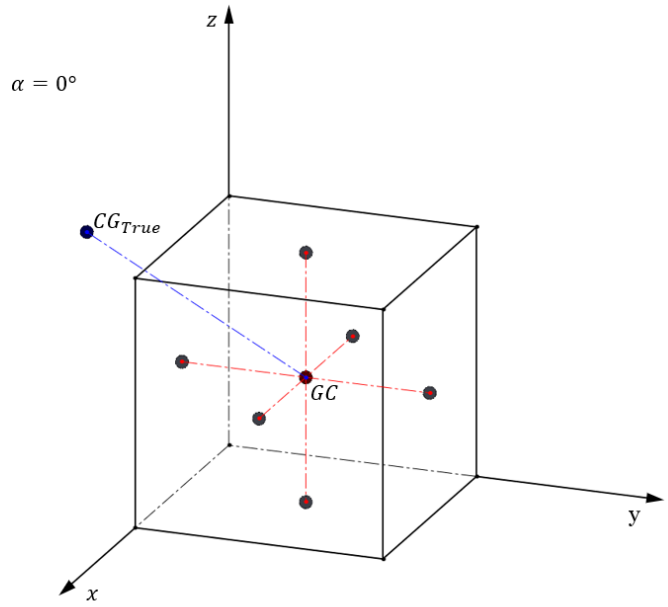

Figure 8.2: Cube Configuration Graphical Representation

Figure 8.3 demonstrates how rotation around the Oz axis influences the Cube Configuration.

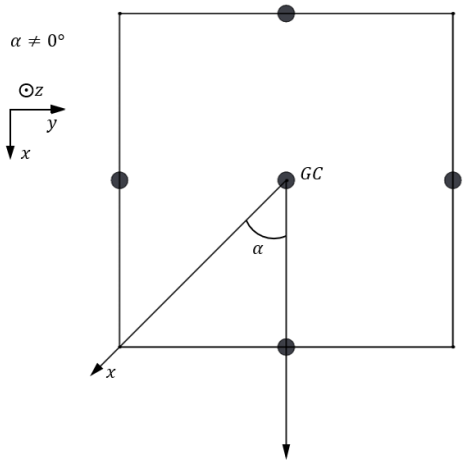

Figure 8.3: Cube Configuration Rotation About OZ Diagram

8.4.2. Pyramid Configuration

The pyramid architecture arranges the RWs at the centroids of the faces of one or two pyramids, which are connected by their common base. This pyramid(s) can take various forms, such as triangular or quadrangular, depending on the design.

This configuration allows for efficient control of the spacecraft's attitude by enabling effective torque generation in multiple directions while minimizing the complexity of the control system. The pyramid(s) arrangement reduces the number of degrees of freedom needed for attitude control, providing a more straightforward and efficient solution.

The Individual used to represent the Pyramid Configuration will be defined as follows:

$$Individual: [\ base\_type\ ,conf\_type\ ,up\_rw\_type\ ,down\_rw\_type\ ,alfa\ ,x\ ,y\ ,z\ ,r\ ,h\ ]$$

The description of each parameter that forms the Individual for the Pyramid Configuration is outlined in Table 8.2:

Table 8.2: Pyramid Configuraton Individual Constants Definition

Pyramid Conf. Constant	Constant Values	Unit	Comments
base_type	{0,1,2,3}	N/A	0: Triangular base; 1: Square base; 2: Pentagonal base; 3: Hexagonal base.
conf_type	{0,1,2}	N/A	0: A single pyramid positioned above the central base; 1: A single pyramid positioned below the central base; 2: Two pyramids, one above and one below the central base.
up_rw_type	{0,1,2,3}	N/A	0: The RWs used for the upper pyramid are "LIBRA-065"; 1: The RWs used for the upper pyramid are "LIBRA-2"; 2: The RWs used for the upper pyramid are "LIBRA-6"; 3: The RWs used for the upper pyramid are "LIBRA-80".
down_rw_type	{0,1,2,3}	N/A	0: The RWs used for the lower pyramid are "LIBRA-065"; 1: The RWs used for the lower pyramid are "LIBRA-2"; 2: The RWs used for the lower pyramid are "LIBRA-6"; 3: The RWs used for the lower pyramid are "LIBRA-80".
alfa	[0,120]	degree	Rotation around the Oz axis.
x	$\mathbb{R}$	mm	Variation in the x-coordinate of the cube's CG relative to the reference CG.
y	$\mathbb{R}$	mm	Variation in the y-coordinate of the cube's CG relative to the reference CG.
z	$\mathbb{R}$	mm	Variation in the z-coordinate of the cube's CG relative to the reference CG.
r	$\mathbb{R}$	mm	The "Base radius of the pyramid", which is the distance from the center of the base to its farthest point.
h	$\mathbb{R}$	mm	Height of the pyramid(s).

Note: The rotation axis of each RW is aligned with the vector connecting the center of the configuration to each RW.

Note: The reference CG is assumed to correspond to the CG of the "Instruments Box".

Figure 8.4 presents a graphical representation of the Pyramid Configuration, showing the eight possible positions for the RWs on a square base, with zero rotation around the Oz axis.

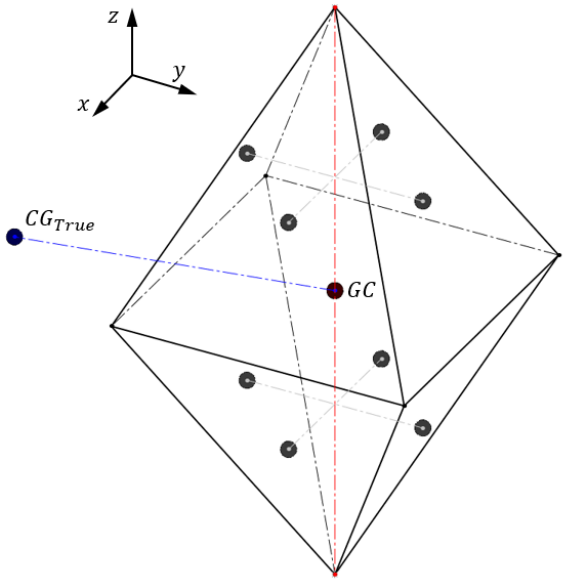

Figure 8.4: Pyramid Configuration Graphical Representation

Figure 8.5 illustrates the impact of rotation around the Oz axis on the Pyramid Configuration.

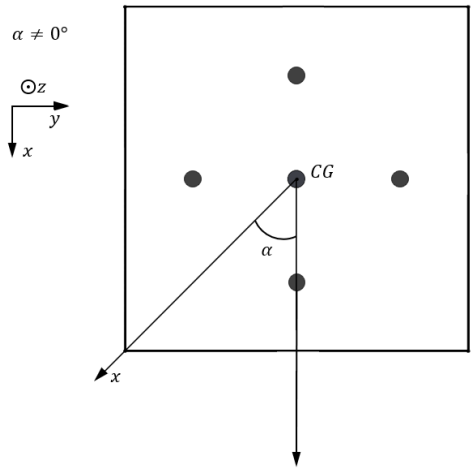

Figure 8.5: Pyramid Configuration Rotation about Oz Diagram

With the framework of the genetic algorithm explained and a deeper understanding of the concepts forming the foundation of the tool to be used for obtaining simulations of the most energy-efficient RW configurations, the following chapter can be thoroughly analyzed. This chapter shows the process of preparation of the simulations, as well as the results of the best simulations performed.

8.5. Individual Transformation Function

In the setup of the implemented PSO simulation, it is necessary to establish the maximum, minimum, and variation values for each component of the individual. To standardize the process for different sets of variation the PSO was configured to generate and update individuals composed of positive integer values (0^+).

These positive integers are then converted to their corresponding true values, which are used in the evaluation of the fitness function, through the individual transformation function.
For example, Table 8.3 presents a possible set of maximum, minimum, and variation (delta) values for a generic individual:

Table 8.3: Generic Elements Variation Values

Element	**max**	**min**	**delta**
1st	30	0	10
2nd	50	50	0
3rd	600	100	100

For the element's values presented in Table 8.3, the number of possible values for each element of the individual is: [3 0 5], respectively, noting that the count starts from the origin, i.e., from 0.

Selecting a generic individual that could be used by the PSO, based on the data from Table 8.3, is exemplified by [2 0 3]. Using the proposed individual transformation function, the corresponding true value of [2 0 3] is [20 50 400]. This transformed individual is then used in the evaluation of the fitness function.

Using a function similar to the individual transformation function is particularly useful, as it first reduces the number of possible combinations for the elements that compose each individual, by setting a number of possibilities per individual's elements. This increases the user's control over the code's inputs and reduces the need to re-optimize the PSO parameters whenever the maximum, minimum, and variation values for the individual's elements are changed.

Chapter 9

9. Simulation

For the simulation of the algorithm, the controller (LQR) is integrated into the PSO so that the optimal positioning and type of actuators (RWs) can be evaluated and the cost function minimized. The two RW architectures defined in Section 8.3 are considered independently: the pyramid configuration and the cube configuration.

First, the PSO simulation sets are executed using only the cube configuration as the RW configuration. Subsequently, a second group is simulated using exclusively the pyramid configuration. This approach is useful for the analysis of results, as it provides an alternative perspective on performance depending on the RW configuration employed.

Therefore, we can conclude that the final positions determined by the algorithm will represent the optimal locations for the RWs. It is essential to ensure that any of the final architectures adhere to the maximum time constraints for each task outlined in the CONOPS.

9.1. Orbital Simulation

The satellite's orbital propagation was performed independently of the attitude control simulation. The classical orbital parameters used to initialize the orbital propagator are given in Table 9.1:

Table 9.1: Initial Orbital Parameters

Epoch ISO	2023-04-17 00:00:00.000
Time scale	TAI
Inertial frame	ECI
θ	0°
e	0.001075637
i	90°
Ω	351.320169°
ω	41.172731°
a	6878 km

Additionally, the total duration of the orbital propagation is the duration of a single orbital period, which also defines the duration of a mission cycle as outlined in the CONOPS. The orbital propagation results were provided with a time step of 0.25 seconds. The selection for such value is related with the linearization error of the quaternion kinematics. The complete justification details can be found in Appendix I.

Despite each RW configuration resulting in a slightly different system mass, the spacecraft mass is assumed to be constant across all simulations with a value of is 169.6 kg. The impact of such assumption in the orbital propagation results is small as analyzed in Appendix J. Figure 9.1 shows the evolution of the orbital position vector over the course of one orbital period of the satellite. As previously discussed, the propagated satellite positions (XYZ) were then input into the Quaternion Objective Time Series Calculator.

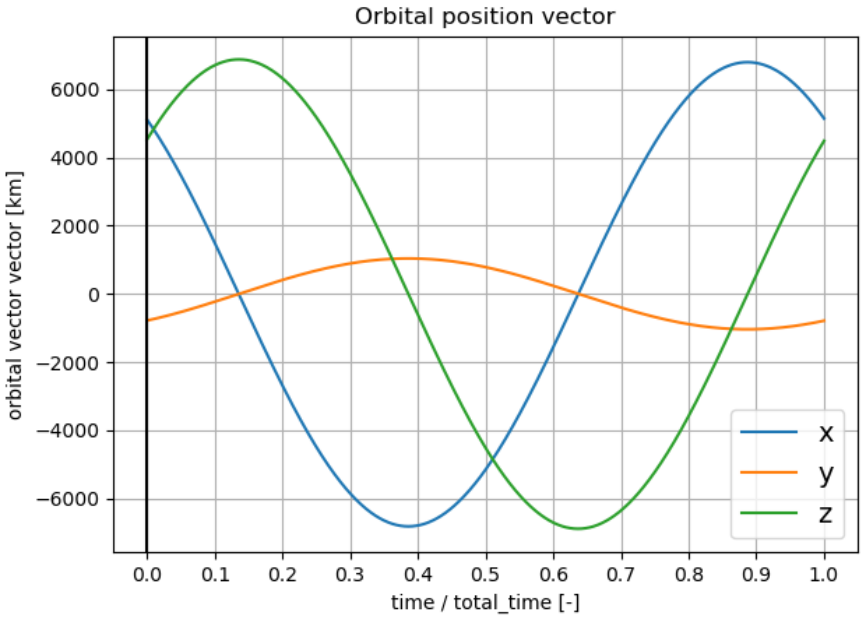

Figure 9.1: Orbital Position Vector Vs Time/Time orbit

9.2. Reference Quaternion Calculation

Following the satellite's orbital propagation, the reference quaternion time series must be computed. Similar to the orbital propagation process, this calculation is conducted independently from the attitude control simulation. Table 5.1 presents the CONOPS maneuver sequence, which serves as the basis for generating the reference quaternion time series.

Figure 9.2 illustrates the evolution of the Gibbs vector, derived from the reference quaternion time series, over the course of one orbital period of the satellite.

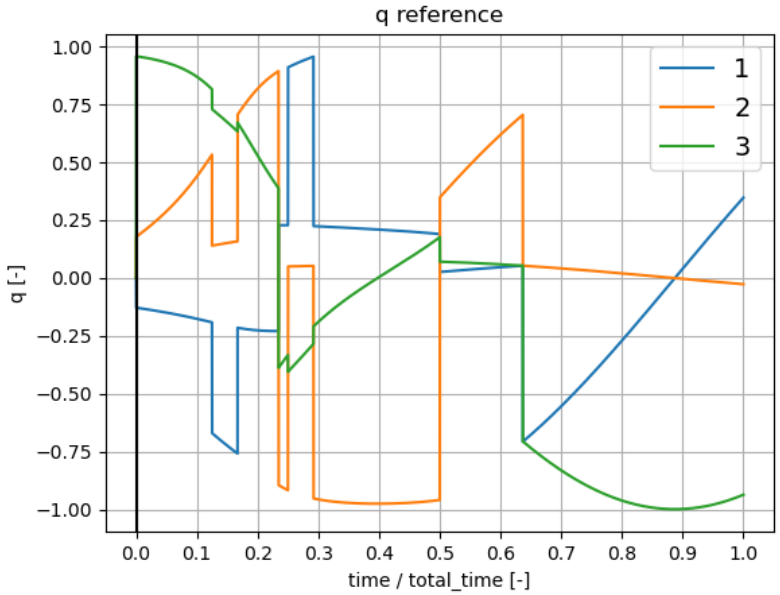

Figure 9.2: Reference Gibbs Vector Vs Time/Time orbit

A discontinuity point can be observed at approximately 0.634 of the normalized time fraction. The possibility of such a discontinuity occurring was previously addressed in Section 7.1.2 and is attributed to the use of a simplified algorithm for calculating the reference quaternion, which calculates the reference quaternion with the smallest rotation angle from the reference vectors. However, since the objective is not to optimize the satellite's attitude control but rather to compute the total energy consumption throughout the CONOPS defined maneuvers, the quaternion profile shown in Figure 9.2 is used despite the presence of this discontinuity.

9.3. Cube Strategy Simulation

For the cube strategy simulation, the individuals used are those presented in Section 8.3.1. Two groups of simulation sets were performed. In the first group, all elements of the individual vary except for those related to the position of the geometric center of the RW configuration. In contrast, the second group of sets varies only the elements that remain static in the first group.

The matrix Q and R used during the cube strategy simulation for the LQR control are:

$$Q = \begin{bmatrix} 60 & 0 & 0 & 0 & 0 & 0 \\ 0 & 60 & 0 & 0 & 0 & 0 \\ 0 & 0 & 60 & 0 & 0 & 0 \\ 0 & 0 & 0 & 40 & 0 & 0 \\ 0 & 0 & 0 & 0 & 40 & 0 \\ 0 & 0 & 0 & 0 & 0 & 40 \end{bmatrix} \tag{12.1}$$

$$R = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix} \tag{12.2}$$

The diagonal values used were chosen arbitrarily (always greater than zero), as the optimization of these diagonal elements for both matrices Q and R lies outside the scope of this dissertation.

It was assumed that the RWs must be contained within the interior of the satellite's instruments box. The physical characteristics of the instruments box are detailed in Appendix A. For the simulations, large population sizes were considered relative to the number of generations. This approach was adopted to attempt to cover the maximum number of points within the solution space.

9.3.1. Cube Strategy First Simulation Set

In this initial set of simulations, the population size, the number of generations max/min element speed and position update increment (μ_{PSO}) are presented in Table 9.2:

Table 9.2: Cube First Simulation Set Pop. and Gen Size/Number

Simulation	Population size	Number of generations	Max element speed	Min element speed	μ_{PSO}
1	30	3	5	-5	5
2	30	3	5	-5	5
3	30	3	5	-5	5

Table 9.3 shows the maximum, minimum and variation values (delta) for the elements of the individuals that are simulated.

Table 9.3: Cube First Simulation Set Elements Variation Values

Cube Individual Elements	**max**	**min**	**delta**	**units**
conf_type	3	0	1	[-]
rw_type	3	0	1	[-]
alfa	45	0	5	[deg]
x	800	800	0	[mm]
y	700	700	0	[mm]
z	400	400	0	[mm]
r	100	0	25	[mm]

In this set, 148 different individuals were simulated, which is far from the maximum number of possibilities, however, this is precisely the purpose of using a genetic algorithm, to avoid simulating every possible solution.

Table 9.4 shows the five best solutions, that is, the individuals with the lowest energy consumption:

Table 9.4: Cube First Simulation Set Best Five Individuals

[-]	[-]	[-]	[deg]	[mm]	[mm]	[mm]	[mm]	[Wh]
Position	**conf_type**	**rw_type**	**alfa**	**x**	**y**	**z**	**r**	**consumed energy**
1	1	3	5	800	700	400	100	43.34837
2	2	3	45	800	700	400	100	43.38284
3	1	3	5	800	700	400	75	43.3852
4	2	3	40	800	700	400	75	43.63634
5	1	3	25	800	700	400	100	43.81395

Analyzing the data from Table 9.4, we can conclude that for the cube configuration, the PSO algorithm tends to compute solutions in which the best individual consists of a smaller number of RWs with higher torque output, rather than a larger number of RWs with lower torque capacity. Furthermore, the best individuals identified by the PSO also tend to feature RWs arranged in configurations that are not aligned with the orthogonal reference axes and are positioned farther from the geometric center of the configuration.

Another analysis I performed was to compare the sum of the unit vectors of the RWs' rotation axes with the three diagonal elements of the inertia matrix of the satellite equipped with reaction wheels. The data from this analysis for the five best individuals in this set of simulations are presented in Table 9.5:

Table 9.5: Cube First Simulation Set Best Five Individuals I matrix comparation

[-]	[J]	[-]	[-]	[-]	$[kg \cdot m^2]$	$[kg \cdot m^2]$	$[kg \cdot m^2]$
Position	**consumed energy**	**x rw_sum**	**y rw_sum**	**z rw_sum**	I_{xx}	I_{yy}	I_{zz}
1	156054	0	0	-1	0.916781	0.916781	1.454379
2	156178.1	0	0	1	0.916781	0.916781	1.454379
3	156186.6	0	0	-1	0.816331	0.816331	1.310879
4	157090.7	0	0	1	0.816331	0.816331	1.310879
5	157730.1	0	0	-1	0.916781	0.916781	1.454379

It can be concluded that the ratio of the sum of the unit vectors corresponding to the reference axes of the RW configurations tends to resemble the ratio of the diagonal elements of the satellite's inertia matrix, including the contribution of the reaction wheels.

9.3.2. Cube Strategy Second Simulation Set

In this initial set of simulations, the population size, the number of generations max/min element speed and position update increment (μ_{PSO}) are presented in Table 9.6:

Table 9.6: Cube Second Simulation Set Pop. and Gen. Size/Number

Simulation	**Population size**	**Number of generations**	**Max element speed**	**Min element speed**	μ_{PSO}
1	40	3	5	-5	5
2	40	5	5	-5	5
3	40	3	5	-5	5

Table 9.7 shows the maximum, minimum and variation values (delta) for the elements of the simulated individuals.

Table 9.7: Cube Second Simulation Set Elements Variation Values

Cube Individual Elements	**max**	**min**	**delta**	**units**
conf_type	1	1	0	[-]
rw_type	3	3	0	[-]
alfa	5	5	0	[deg]
x	1000	600	50	[mm]
y	875	525	25	[mm]
z	500	300	25	[mm]
r	100	100	0	[mm]

In this set, 231 different individuals were simulated.

Table 9.8 shows the five best solutions, that is, the individuals with the lowest energy consumption:

Table 9.8: Cube Second Simulation Set Best Five Individuals

[-]	[-]	[-]	[deg]	[mm]	[mm]	[mm]	[mm]	[Wh]
Position	**conf_type**	**rw_type**	**alfa**	**x**	**y**	**z**	**r**	**consumed energy**
1	1	3	5	750	700	400	100	43.22023
2	1	3	5	750	750	300	100	43.30137
3	1	3	5	800	750	300	100	43.30256
4	1	3	5	750	675	325	100	43.30351
5	1	3	5	800	775	300	100	43.30712

Another analysis I performed was to compare the distance between the CG of the RW configuration and the CG of the satellite without reaction wheels. The data from this analysis for the five best individuals in this set of simulations are presented in Table 9.9:

Table 9.9: Cube Second Simulation Set Best Five Individuals CG Distance

[-]	[J]	[mm]	[mm]	[mm]	[mm]
Position	**consumed energy**	x_{cg} **rw**	y_{cg} **rw**	z_{cg} **rw**	**Distance**
1	155592.7	750	700	420	414.128
2	155884.8	750	750	320	476.9717
3	155889.1	800	750	320	501.7236
4	155892.5	750	675	345	408.1997
5	155905.5	800	775	320	521.8731

Comparing the distance between the CG of the RW configuration and the CG of the satellite without reaction wheels, it can be concluded that these values tend to be directly proportional. This is attributed to the reduction in the magnitude of the constituent elements of the inertia matrix.

9.3.3. Best RW Cube Configuration Graphs

Analyzing the results from the previous section, it is possible to state that the best reaction wheel cube configuration is:

$$best\ RW\ cube\ configuration = [1 \quad 3 \quad 5 \quad 750 \quad 700 \quad 400 \quad 100]$$

Below, several plots related to the satellite attitude control simulation are presented.

Figure 9.3 shows a graph of the total energy consumption by the satellite's reaction wheels over the course of one orbital period.

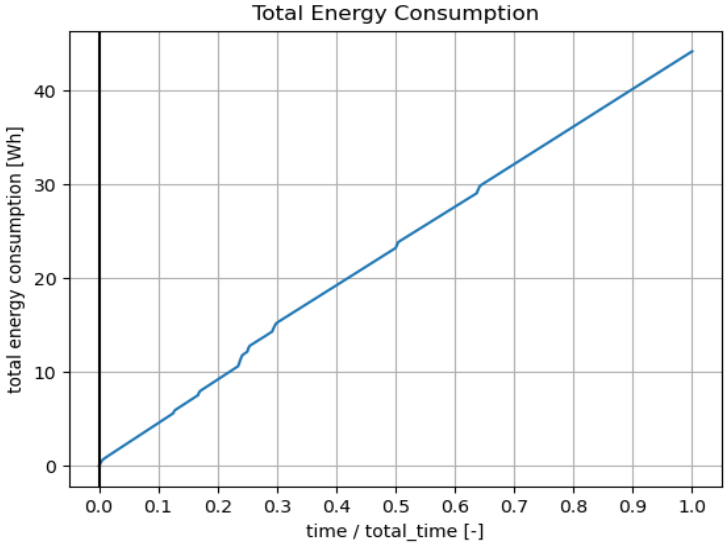

Figure 9.3: Best Cube Conf. Total Energy Consumption Vs Time/Time orbit

Energy consumption occurs in a largely linear manner throughout most of the period, with distinct peaks corresponding to points where the slope of the consumption graph increases significantly. These peaks are due to the reaction wheels generating the necessary torque to execute control maneuvers. After the maneuvers, the energy consumption stabilizes, as can be observed by analyzing the total energy consumption graph.

Figure 9.4 and Figure 9.5 show the graphs of the satellite's angular velocity and angular acceleration, respectively, over time.

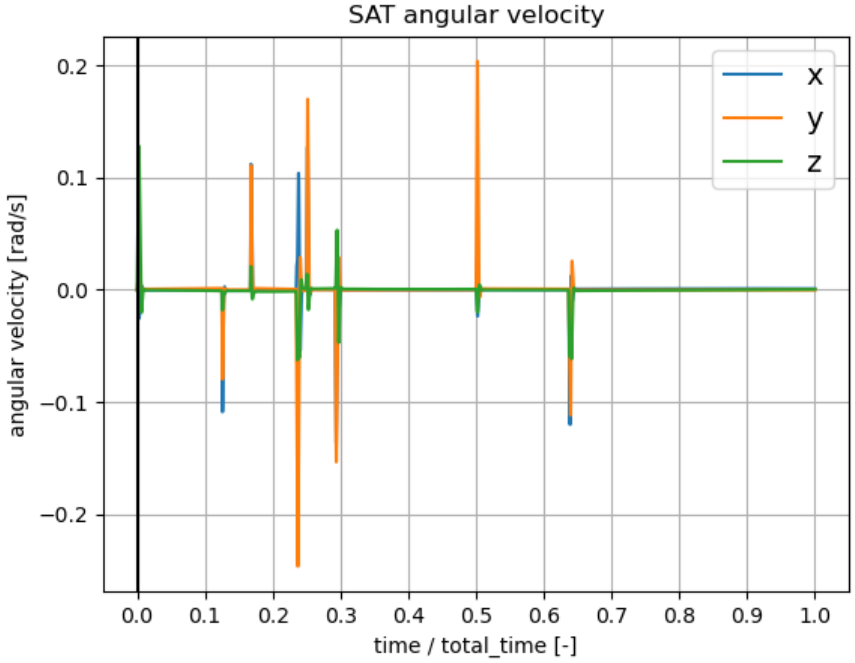

Figure 9.4: Best Cube Conf. SAT Angular Velocity Vs Time/Time orbit

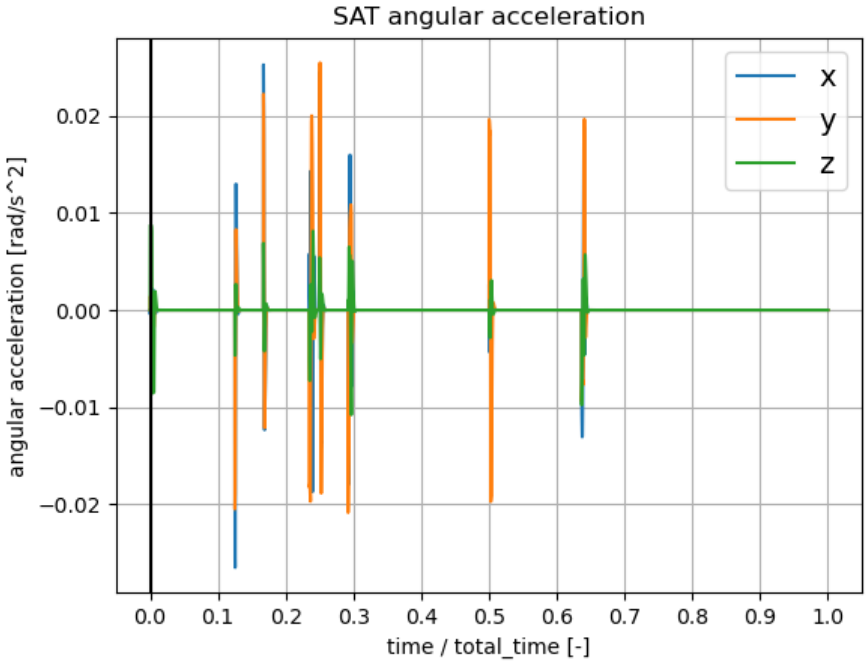

Figure 9.5: Best Cube Conf. SAT Angular Acceleration Vs Time/Time orbit

These last two figures confirm that the largest variations in angular velocity and angular acceleration occur during the satellite's maneuvering phases, with rotations around the y-axis being the most critical, followed by rotations around the x-axis, for the successful execution of the maneuvers.

Figure 9.6 and Figure 9.7 show the graphs of the total torque requested by the controller (LQR) and the total torque executed, respectively, over time.

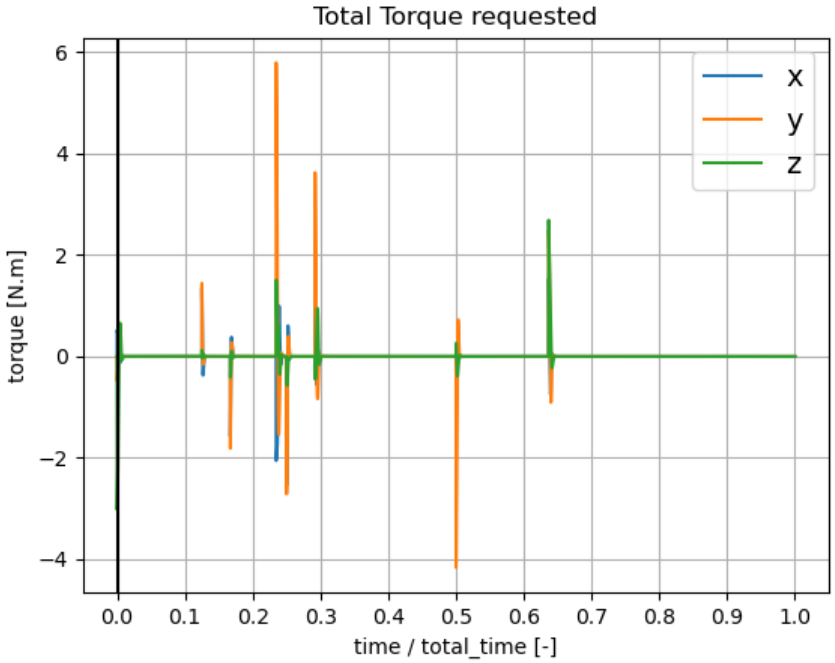

Figure 9.6: Best Cube Conf. Total Torque Requested Vs Time/Time orbit

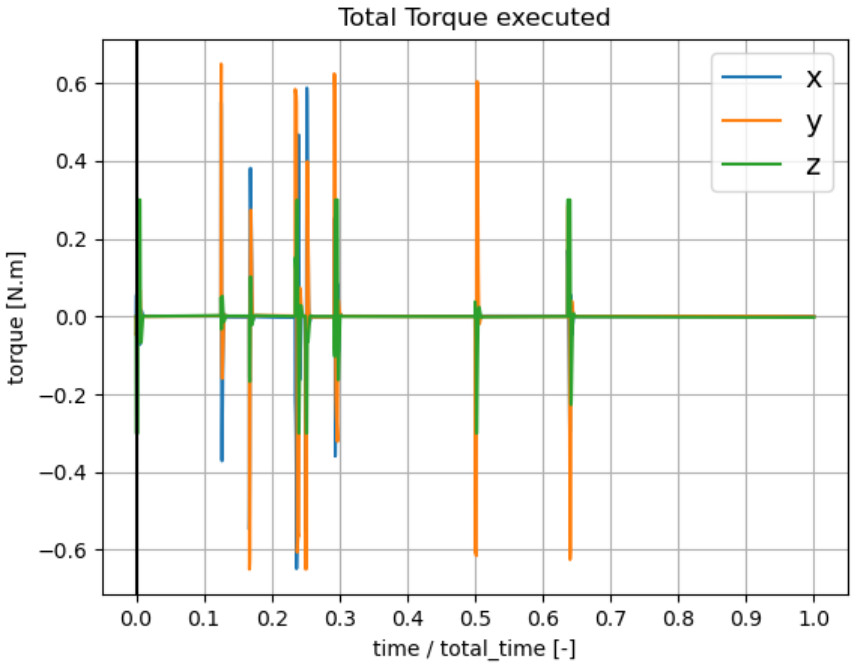

Figure 9.7: Best Cube Conf. Total Torque Requested Vs Time/Time orbit

Figure 9.6 and Figure 9.7 coincide at the moments when the controller requests the highest torque, indicating peak demand. Another important observation is that the LQR controller exhibits an aggressive response, characteristic of this control method, which leads to prolonged saturation periods in the RWs. This can have consequences for the effectiveness of the satellite's attitude control.

Figure 9.8 shows the angular velocity values for all the RWs in the simulated configuration. As expected, the highest velocity peaks occur during periods of increased torque demand from the controller. It is also evident that the torque distributor (as described in Section 7.5) is functioning as intended. There is no significant discrepancy between the velocities of the RWs, demonstrating that the requested torque is being effectively distributed among them to prevent prolonged saturation of any single wheel.

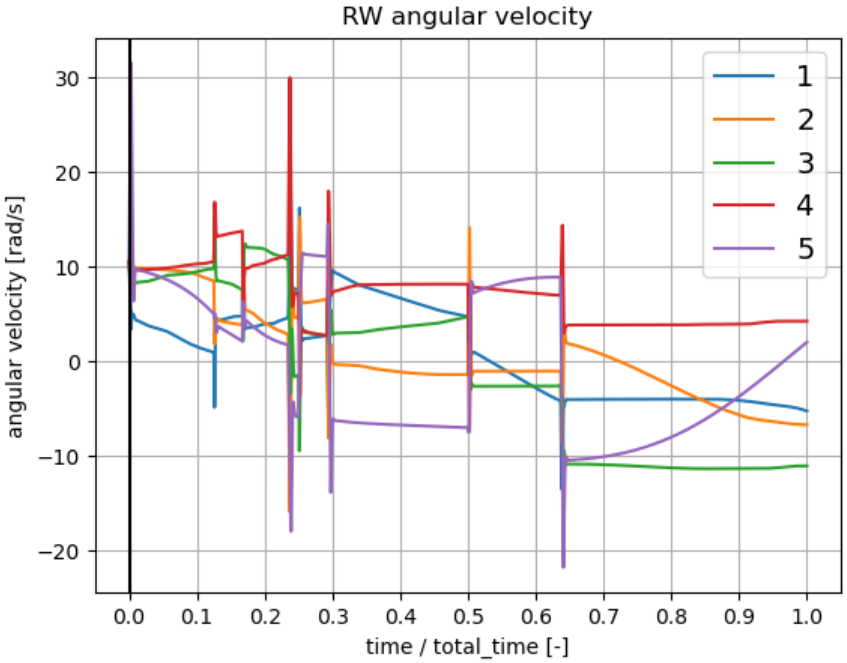

Figure 9.8: Best Cube Conf. RW Angular Velocity Vs Time/Time orbit

Figure 9.9 shows the instantaneous torque output of the reaction wheels over time. Analyzing the graph, it can be concluded that during the execution of the attitude control maneuvers, the RWs tend to exhibit torque output saturation, something that aligns with the observations made in Figure 9.7.

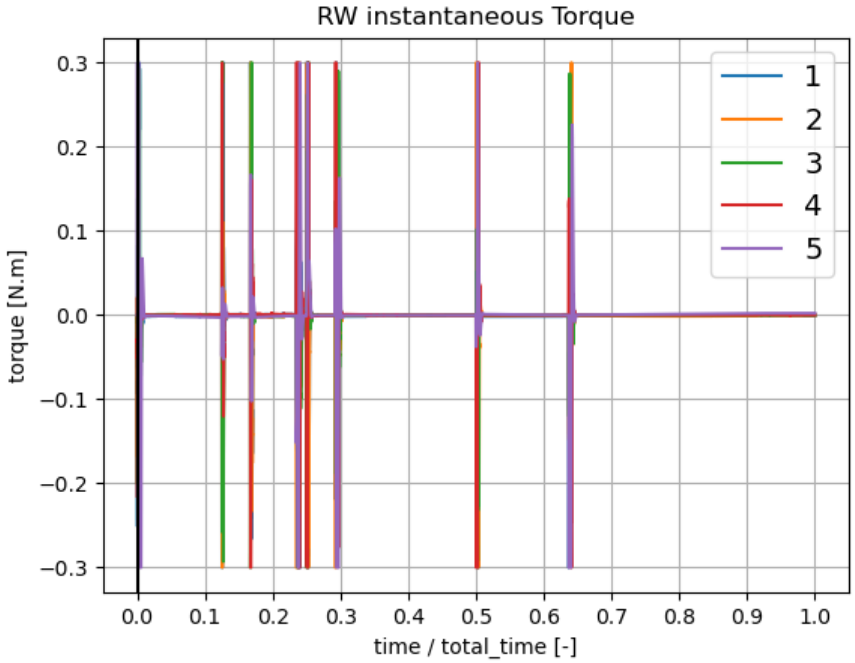

Figure 9.9: Best Cube Conf. RW Instantaneous Torque Vs Time/Time orbit

Figure 9.10 shows the instantaneous Gibbs vector of the satellite over time. During the periods in which the satellite performs maneuvers, it can be observed that the controller's response is, in some cases, excessive, as evidenced by the clear overshoot in the graph of the Gibbs vector components. This type of behavior is undesirable, as it reduces control precision and increases instability in the satellite's attitude control, which, in turn, leads to higher energy consumption across a variety of common operational scenarios.

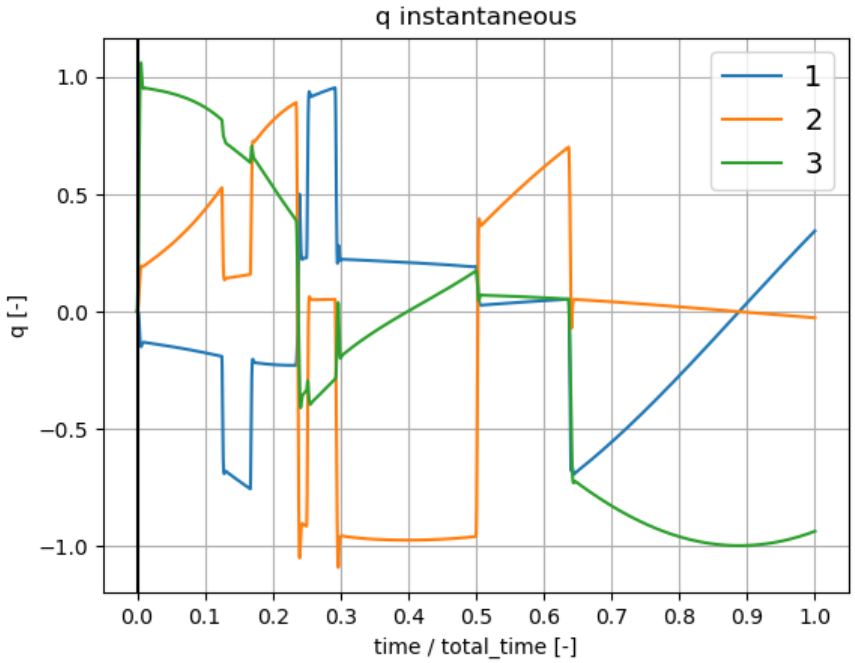

Figure 9.10: Best Cube Conf. q Instantaneous Vs Time/Time orbit

Figure 9.11 shows the error of the Gibbs vector, calculated as the difference between the instantaneous Gibbs vector and the previously determined reference quaternion (Section 9.2). The highest peaks in this error occur during the same time intervals in which the controller demands the most torque. As a result of this high torque demand, and due to the saturation state experienced by the reaction wheels during the most demanding maneuvers, where the error is also higher, it is expected that the controller would have greater difficulty stabilizing the system and bringing the state vector to its reference, where the error would ideally be zero. This behavior is indeed observed, as shown in the previous Figure 9.10.

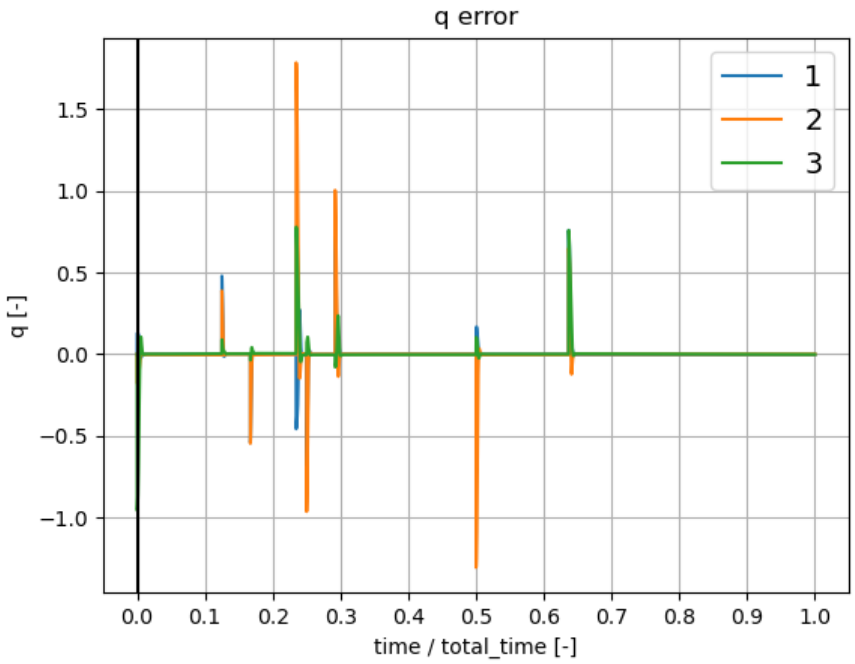

Figure 9.11: Best Cube Conf. q Error Vs Time/Time orbit

Figure 9.12 shows the solver used by the torque distributor over time. When compared with the previous figure, it can be observed that the moments in which the torque distributor encounters greater difficulty in allocating the required torque among the reaction wheels, resorting to the use of Solver 2 (Section 7.5.2), coincide with the periods where the Gibbs vector error is highest, a correlation that is logically consistent. Another conclusion that can be drawn is that the torque distributor is functioning successfully as designed.

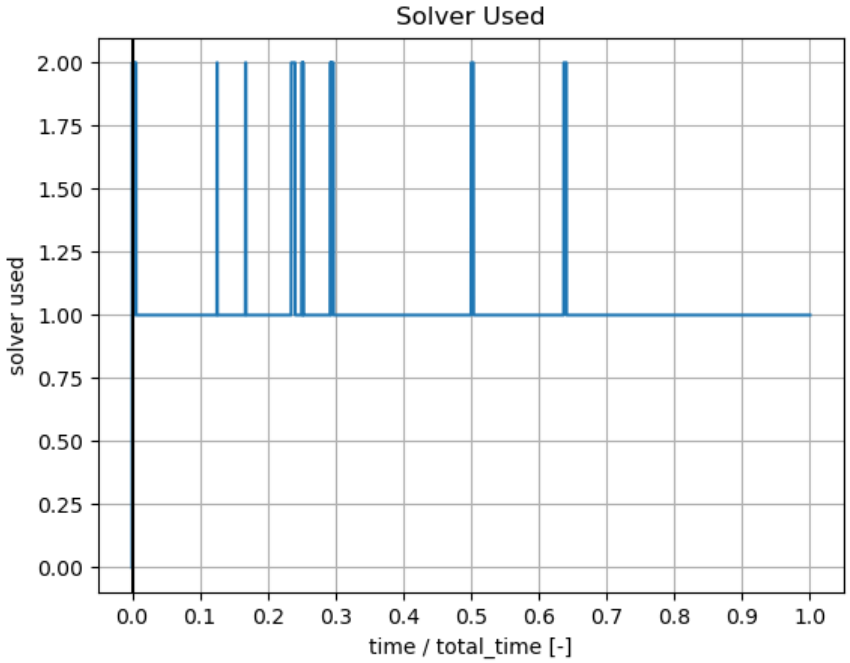

Figure 9.12: Best Cube Conf Solver Used Vs Time/Time orbit

9.4. Pyramid Strategy Simulation

For the pyramid strategy simulation, the individuals used are those presented in Section 8.3.2. Similarly to the Cube strategy configuration two groups of simulation sets were performed. In the first group, all elements of the individual vary except for those related to the position of the geometric center of the RW configuration. In contrast, the second group of sets varies only the elements that remain static in the first group.

The matrix Q and R used during the pyramid strategy simulation for the LQR control are the same used for the cube strategy simulation. Similarly to what was considered for the cube strategy simulation, the RWs are required to be contained within the interior of the satellite's instruments box. For the simulations, large population sizes were also considered relative to the number of generations. This approach was adopted to cover as many points as possible within the solution space.

9.4.1. Pyramid Strategy First Simulation Set

In this initial set of simulations, the population size, the number of generations max/min element speed and position update increment (μ_{PSO}) are presented in Table 9.10:

Table 9.10: Pyramid First Simulation Set Pop. and Gen. Size/Number

Simulation	Population size	Number of generations	Max element speed	Min element speed	μ_{PSO}
1	30	3	5	-5	5
2	40	3	5	-5	5
3	40	3	5	-5	5

The population size for the pyramid strategy is larger in the second and third simulations due to a greater number of elements in the individual formulation corresponding to the pyramid reaction wheel configuration.

Table 9.11 presents the maximum, minimum, and variation values (delta) for the elements of the simulated individuals.

Table 9.11: Pyramid First Simulation Set Elements Variation Values

Pyramid Individual Coeficients	**max**	**min**	**delta**	**unit**
base_type	3	0	1	[-]
conf_type	2	0	1	[-]
up_rw_type	3	0	1	[-]
down_rw_type	3	0	1	[-]
alfa	60	0	10	[deg]
x	800	800	0	[mm]
y	700	700	0	[mm]
z	400	400	0	[mm]
r	175	0	25	[mm]
h	300	0	25	[mm]

In this set, 286 different individuals were simulated.

Table 9.12 shows the five best solutions, that is, the individuals with the lowest energy consumption:

Table 9.12: Pyramid First Simulation Set Best Five Individuals

[-]	[-]	[-]	[-]	[-]	[deg]	[mm]			[mm]		[Wh]
Position	**base_ type**	**conf_ type**	**up_rw _type**	**down_ rw_type**	**alfa**	**x**	**y**	**z**	**r**	**h**	**consumed energy**
1	0	0	3	3	10	800	700	400	50	50	28.61638
2	0	0	3	3	10	800	700	400	100	100	28.61711
3	0	0	3	3	10	800	700	400	125	125	28.61802
4	0	0	3	3	10	800	700	400	175	175	28.62077
5	0	0	3	3	60	800	700	400	100	100	28.62125

Analyzing the data from Table 9.12, we can once again conclude that, for the cube configuration, the PSO algorithm tends to compute best individual solutions consisting of a smaller number of RWs with higher torque output, rather than a larger number of RWs with lower torque capacity. Additionally, it can also be confirmed that the best

individuals identified by the PSO tend to have RWs arranged in configurations that are not aligned with the orthogonal reference axes and are positioned farther from the geometric center of the configuration.

I repeated the analysis that compares the sum of the unit vectors of the reaction wheel rotation axes with the three diagonal elements of the inertia matrix of the satellite including the reaction wheels. The data from this analysis for the five best individuals in this set of simulations are presented in Table 9.13:

Table 9.13: Pyramid First Simulation Set Best Five Individuals I Matrix Comparation

[-]	[J]	[-]	[-]	[-]	$[kg \cdot m^2]$	$[kg \cdot m^2]$	$[kg \cdot m^2]$
Position	**consumed energy**	**x rw_sum**	**y rw_sum**	**z rw_sum**	I_{xx}	I_{yy}	I_{zz}
1	103018.9	0	0	-1.34164	0.417229	0.417229	0.685668
2	103021.5	0	0	-1.34164	0.431989	0.431989	0.715188
3	103024.8	0	0	-1.34164	0.443059	0.443059	0.737328
4	103034.7	0	0	-1.34164	0.472579	0.472579	0.796368
5	103036.4	0	0	-1.34164	0.431989	0.431989	0.715188

A less assertive conclusion can be drawn from the analysis of Table 9.5, is that the ratio of the sum of the unit vectors corresponding to the reference axes of the RW configurations tends to resemble the ratio of the diagonal elements of the inertia matrix of the satellite including the reaction wheels.

9.4.2. Pyramid Strategy Second Simulation Set

In this initial set of simulations, the population size, the number of generations max/min element speed and position update increment (μ_{PSO}) are presented in Table 9.14:

Table 9.14: Pyramid Second Simulation Set Pop. and Gen. Size/Number

Simulation	**Population size**	**Number of generations**	**Max element speed**	**Min element speed**	μ_{PSO}
1	40	3	5	-5	5
2	40	5	5	-5	5
3	40	3	5	-5	5

Table 9.15 presents the maximum, minimum, and variation values (delta) for the elements of the simulated individuals.

Table 9.15: Pyramid Second Set Elements Variation Values

Pyramid Individual Elements	**max**	**min**	**delta**	**unit**
base_type	0	0	0	[-]
conf_type	0	0	0	[-]
up_rw_type	3	3	0	[-]
down_rw_type	3	3	0	[-]
alfa	10	10	0	[deg]
x	1000	600	50	[mm]
y	875	525	25	[mm]
z	500	300	25	[mm]
r	50	50	0	[mm]
h	50	50	0	[mm]

In this set, 213 different individuals were simulated.

Table 9.16 shows the five best solutions, that is, the individuals with the lowest energy consumption:

Table 9.16: Pyramid Second Simulation Set Best Five Individuals

[-]	[-]	[-]	[-]	[-]	[deg]	[mm]			[mm]		[Wh]
Position	**base_type**	**conf_type**	**up_ rw_type**	**down_ rw_type**	**alfa**	**x**	**y**	**z**	**r**	**h**	**consumed energy**
1	0	0	3	3	10	650	550	400	50	50	28.45402
2	0	0	3	3	10	650	575	400	50	50	28.46444
3	0	0	3	3	10	650	550	350	50	50	28.46594
4	0	0	3	3	10	650	575	375	50	50	28.46949
5	0	0	3	3	10	700	550	400	50	50	28.47102

Once again, I performed an analysis that compares the distance between the CG of the RW configuration and the CG of the satellite without reaction wheels. The data from this analysis for the five best individuals in this set of simulations are presented in Table 9.17:

Table 9.17: Pyramid Second Simulation Set Best Five Individuals CG Distance

[-]	[J]	[mm]	[mm]	[mm]	[mm]
Position	**consumed energy**	x_{cg} **rw**	y_{cg} **rw**	z_{cg} **rw**	$Distance_{cg}$
1	102434.4	650	550	410	237.7032
2	102471.9	650	575	410	259.0904
3	102477.3	650	550	360	253.4814
4	102490.1	650	575	385	265.2882
5	102495.6	700	550	410	265.9462

Comparing the distance between the CG of the RW configuration and the CG of the satellite without reaction wheels, we can again conclude that these values tend to be directly proportional.

9.4.3. Best RW Pyramid Configuration Graphs

Analyzing the results from the previous section, it can be stated that the best reaction wheel pyramid configuration is:

$$best\ RW\ pyramid\ configuration = [0 \quad 0 \quad 3 \quad 3 \quad 10 \quad 650 \quad 550 \quad 400 \quad 50 \quad 50]$$

Below, several plots related to the satellite attitude control simulation are presented. The analysis to which these figures will be subjected will focus on comparing the results between the plots obtained for the best RW cube configuration and the best RW pyramid configuration, rather than repeating the same analysis conducted for the best RW cube configuration, as that would lead to redundant conclusions.

Figure 9.13 shows that the best RW pyramid configuration exhibits higher peaks in energy consumption compared to the best RW cube configuration.

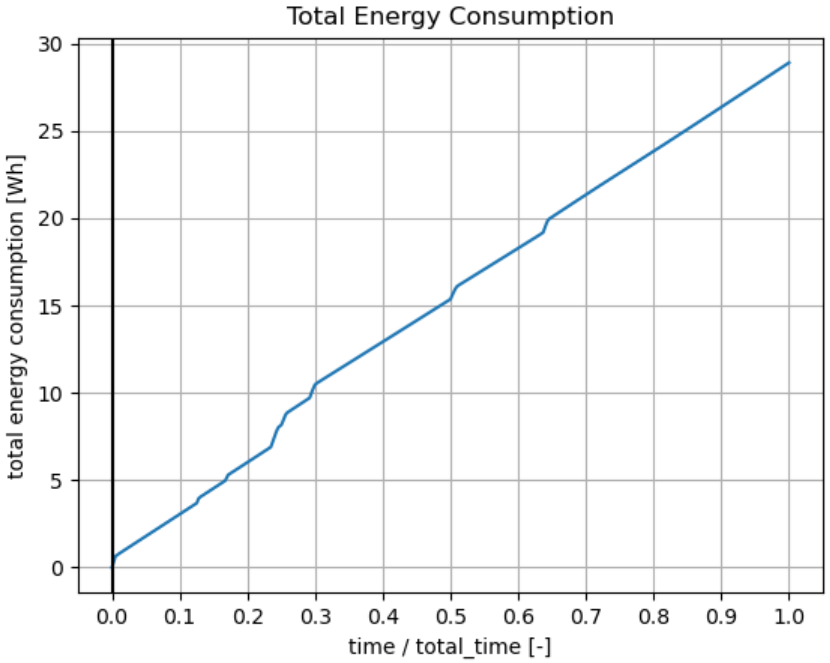

Figure 9.13: Best Pyra. Conf. Total Energy Consumption Vs Time/Time orbit

In Figure 9.14 and Figure 9.15, it is possible to observe that, over extended periods, the satellite is subjected to continuous variations in angular velocity and angular acceleration, which may indicate that the controller is experiencing some difficulties in maintaining the satellite's attitude control.

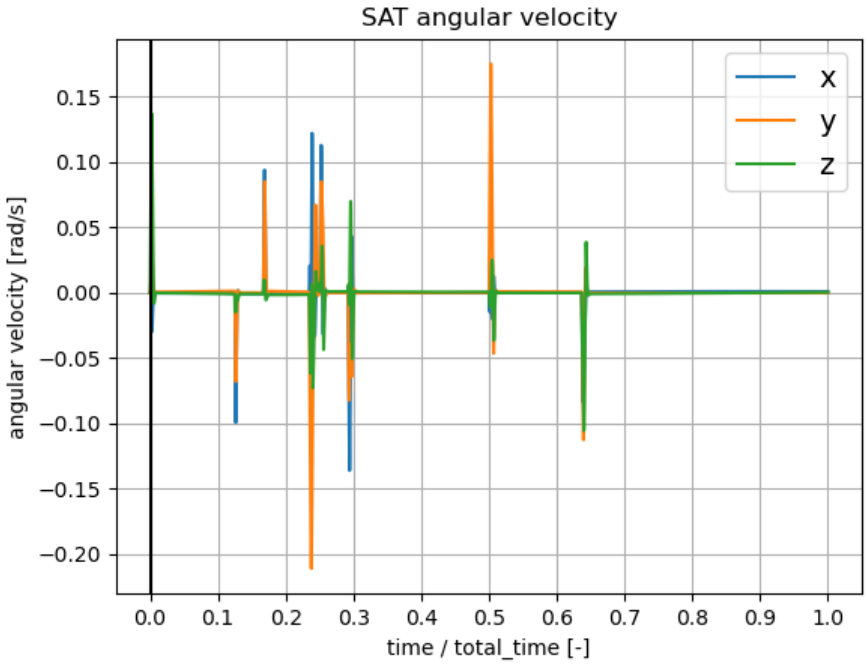

Figure 9.14: Best Pyra. Conf. SAT Angular Velocity Vs Time/Time orbit

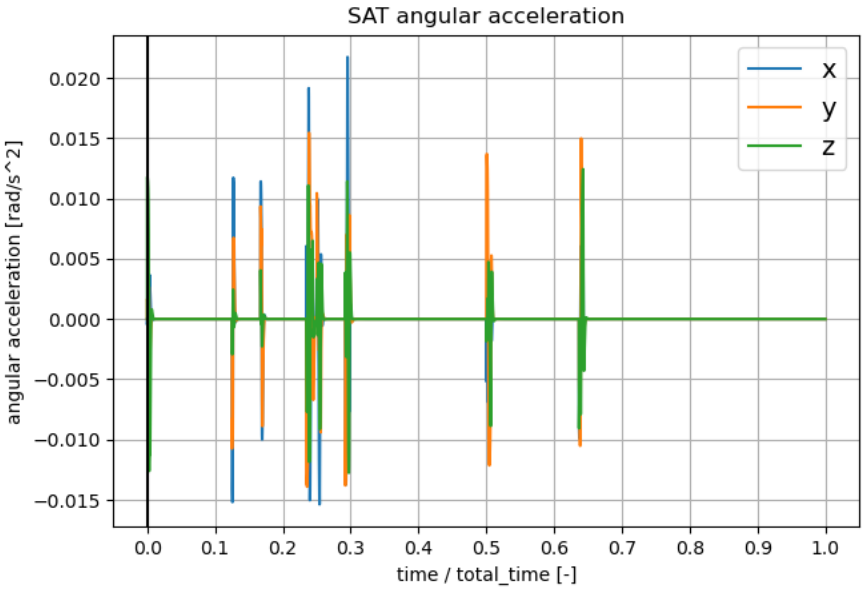

Figure 9.15: Best Pyra. Conf. SAT Angular Acceleration Vs Time/Time orbit

Analyzing Figure 9.16 and Figure 9.17, it is possible to observe that the simulated configuration exhibits a lower maximum torque output compared to the best RW cube configuration, despite having a similar total torque requested profile. Additionally, the previous graph again shows that the total torque executed maintains significant magnitude values over extended periods of time.

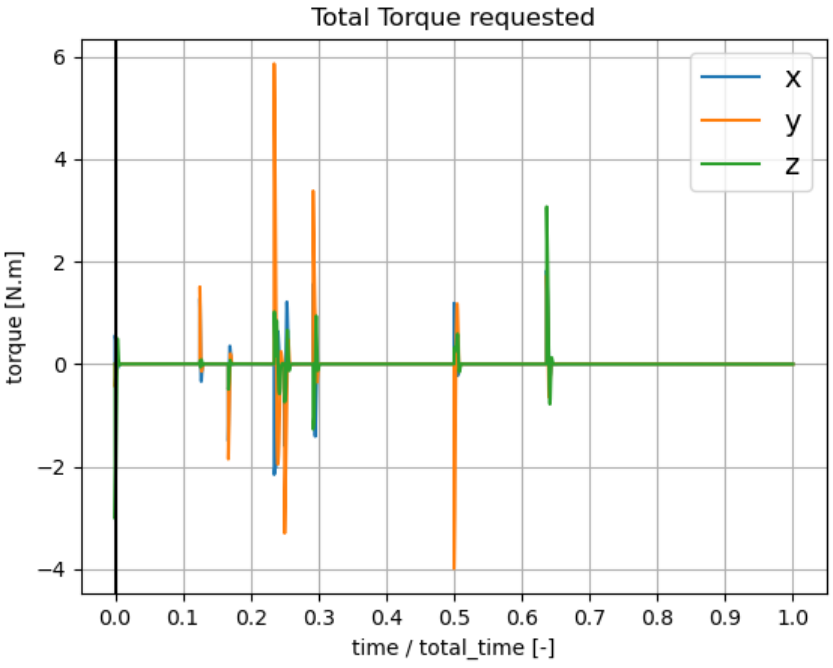

Figure 9.16: Best Pyra. Conf. Total Torque Requested Vs Time/Time orbit

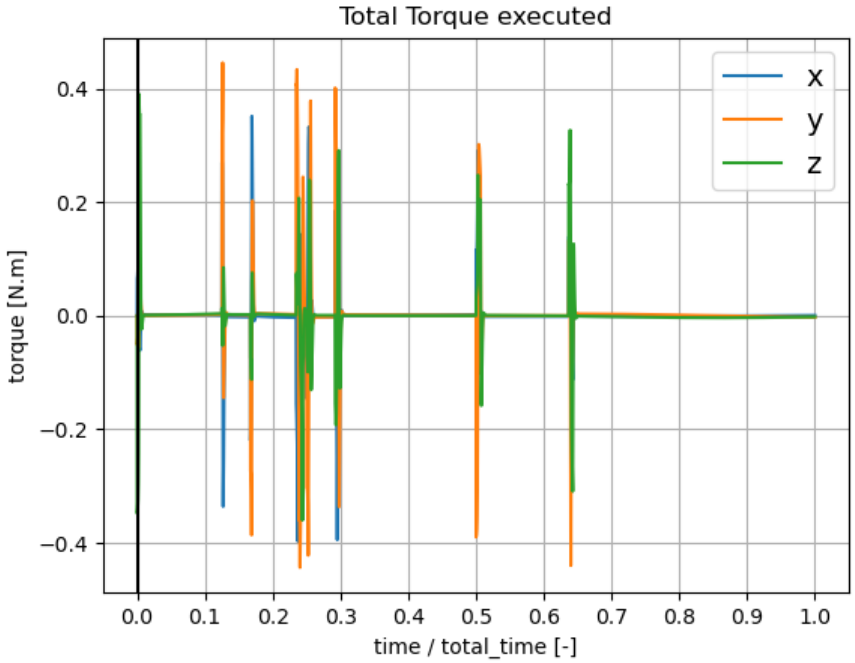

Figure 9.17: Best Pyra. Conf. Total Torque Executed Vs Time/Time orbit

Examining Figure 9.18 and Figure 9.19, it is evident that the simulated configuration operates under suboptimal conditions, exhibiting continuous variations in the angular velocity of the reaction wheels and experiencing prolonged periods of saturation with respect to torque output.

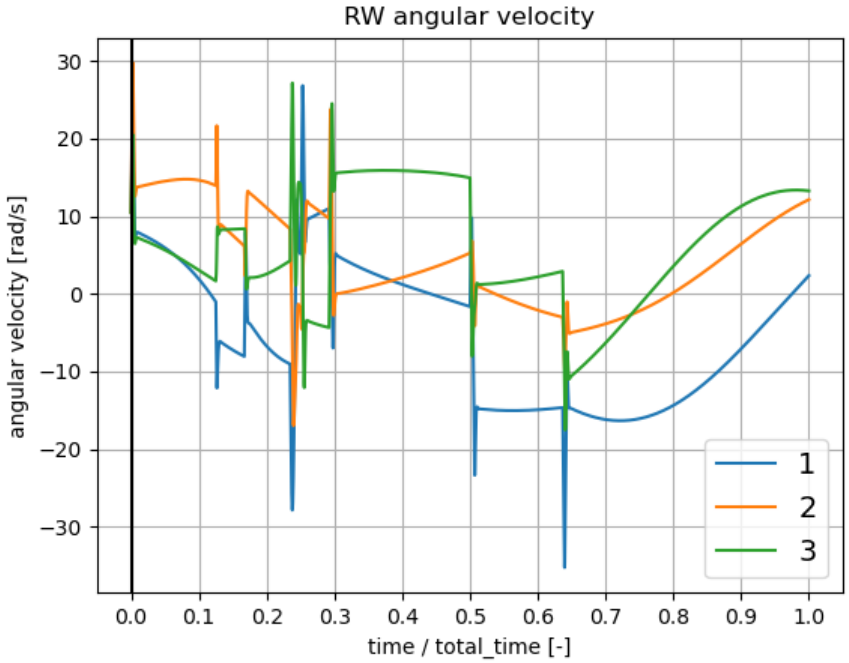

Figure 9.18: Best Pyra. Conf. RW Angular Velocity Vs Time/Time orbit

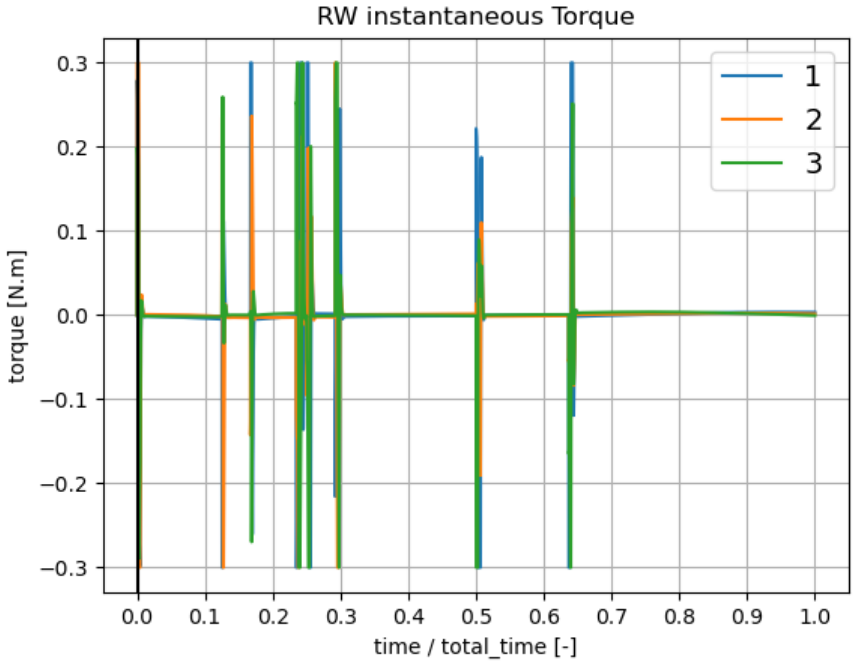

Figure 9.19: Best Pyra. Conf. RW Instantaneous Torque Vs Time/Time orbit

Comparing Figure 9.20, Figure 9.21 and Figure 9.22 with their corresponding counterparts, it is evident that the controller experiences greater difficulty in reducing the attitude error. In Figure 9.21, the error persists over extended continuous periods, only stabilizing after a considerable amount of time. This reinforces the idea that the controller has limited capability to achieve stable points, as it frequently encounters overshoot situations, as seen in Figure 9.20, and needs to resort to executing Solver 2 more often, as shown in Figure 9.22.

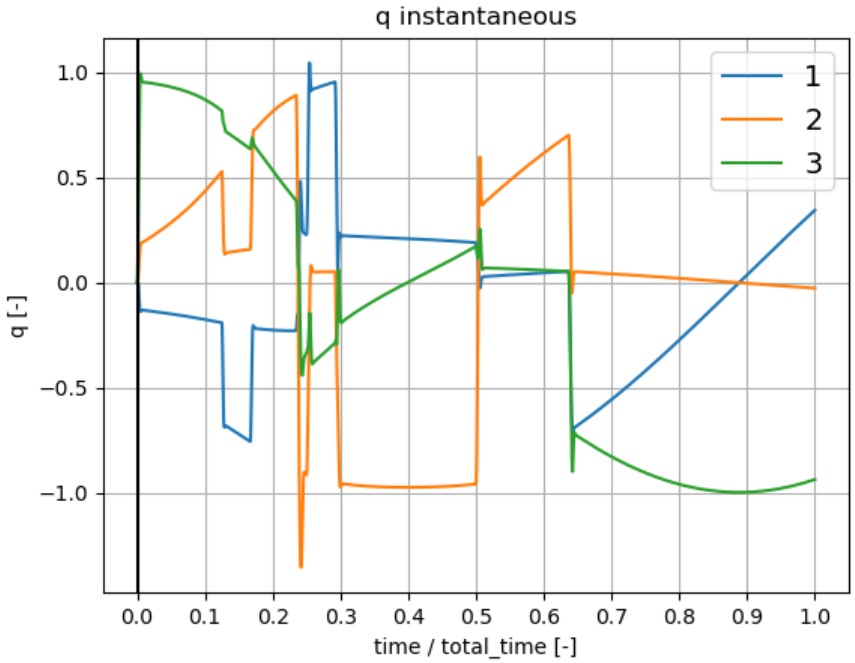

Figure 9.20: Best Pyra. Conf. q Instantaneous Vs Time/Time orbit

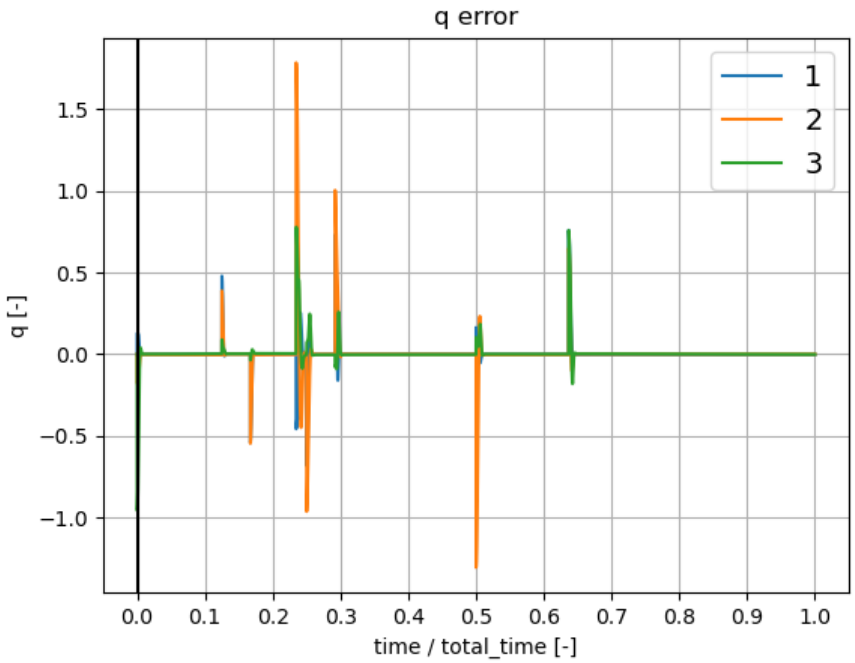

Figure 9.21: Best Pyra. Conf. q Error Vs Time/Time orbit

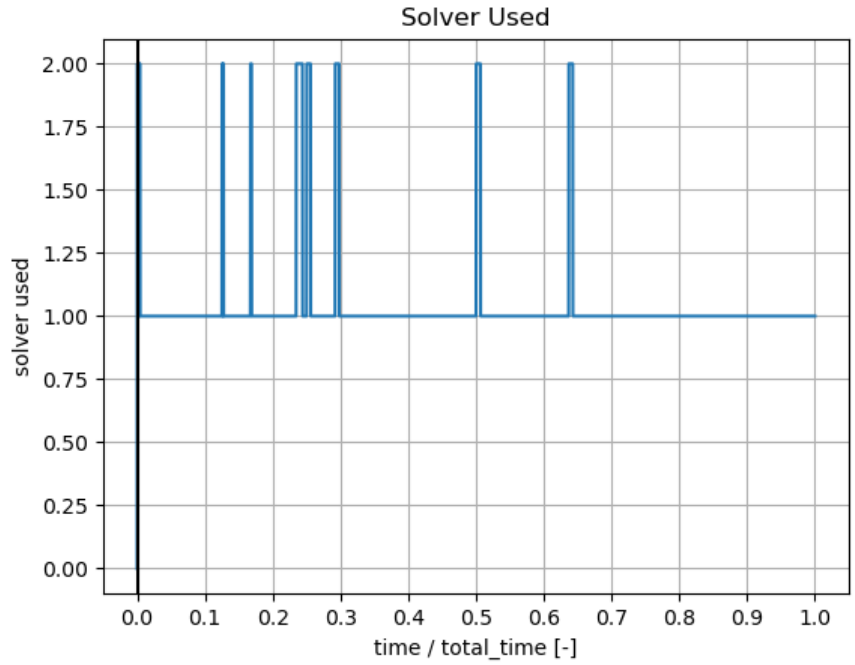

Figure 9.22: Best Pyra. Conf. Solver Used Vs Time/Time orbit

9.5. Discussion of results

Looking at the results more broadly, it is possible to verify that the implemented PSO algorithm demonstrates convergence capability in identifying the best individuals, i.e., those that result in the lowest energy consumption. This phenomenon can be observed in greater detail in Appendix K. It is possible to conclude that the four groups of simulated sets achieved plausible outcomes, allowing for valid conclusions about which parameters tend to generate more efficient RW configurations.

Between the two strategies used, cube and pyramid, each has its own tendencies and capabilities. The pyramid strategy offers a wider range of possibilities and, consequently, a broader spectrum of results.

Although the best RW pyramid configuration exhibits lower total energy consumption compared to the best RW cube configuration, it also shows significantly reduced control capability, which is expected given the smaller number of RWs in the best pyramid configuration. However, it is important to understand that in a real design scenario, the configuration with the lowest total energy consumption would never be chosen in isolation. Other factors are considered, such as the agility required by the control system, redundancy methods, and additional devices that can be integrated to support the RWs, such as magnetorquers, for example.

After a thorough analysis of the results obtained, the final chapter of this dissertation, the conclusion, will be presented. This chapter will assess the work carried out throughout the dissertation, examining both its future implications and its impact on contemporary aerospace engineering. Additionally, suggestions for the continuation and further development of the research conducted in this dissertation will be provided.

Chapter 10

10. Conclusion

10.1. Conclusion

The primary objective of this dissertation was to determine the optimal number, orientation, and positioning of RWs within a satellite's ACS. A PSO algorithm was employed to optimize RW configurations, aiming to minimize energy consumption while meeting temporal constraints essential for timely maneuver execution. To evaluate energy consumption, a complete orbital period in a Sun-synchronous orbit was simulated using a simplified propagator accounting for key perturbations such as atmospheric drag, J2, and solar radiation pressure. Representative attitude maneuvers, reflecting nominal mission operations, were performed under an LQR-based controller to quantify RW energy use, while also maintaining computational efficiency during the optimization process.

A detailed orbital model was developed to accurately simulate satellite dynamics, incorporating key disturbances such as atmospheric drag, the J2 effect, and solar radiation pressure. To ensure precise and stable attitude maneuvers, a LQR controller was implemented, executing a series of representative operations. Additionally, a RWCM was introduced to estimate the energy usage of RWs during nominal operations. To optimize RW configuration while balancing energy efficiency and temporal constraints, a GA (based on PSO theory) was applied. These methodologies were integrated into a CONOPS framework and validated through simulations.

Moreover, based on the obtained simulation results, the developed tool proves to be particularly useful in the early stages of satellite design. By leveraging this tool, preliminary solutions for RW placement can be obtained, aiding initial mission planning. However, for a more precise assessment of controller performance and the system energy consumption, a more advanced simulation tool would be required.

This dissertation presents a systematic approach to optimizing ACS configurations, contributing to the development of more energy-efficient satellite missions. The methodologies and findings can be adapted to other orbital regimes or expanded to incorporate additional actuators, such as magnetorquers, further enhancing satellite control capabilities.

Considering that the primary objective was to develop a tool capable of determining, from a range of possible inputs, the RW configuration with the lowest energy consumption for a defined CONOPS, this goal was successfully achieved by successfully addressing the challenge of optimizing RW configurations for satellite attitude control in LEO. By employing two distinct RW arrangement strategies, cube and pyramid configurations, the convergence trends of these configurations were analyzed using a PSO algorithm. These results contribute to the thesis objective of developing a computational tool that calculates a set of optimal RW configurations, providing highly valuable information during the early stages of satellite design.

The outcomes pave the way for more efficient and sustainable satellite missions, supporting the growing demands of Earth observation and other space applications. The results and methodologies established in this work serve as a foundation for further research and practical applications in satellite mission design.

10.2. Future Work

Several areas for future research and development could further enhance the effectiveness of the proposed approach. First, enhancing the precision of orbital disturbance calculations would increase the overall system accuracy. Incorporating supplementary actuators, such as magnetorquers, could further enhance system redundancy and robustness, providing greater flexibility in attitude control strategies.

Another key area for refinement is the optimization of computational performance, along with increasing the efficiency of the GA and the quaternion reference calculation algorithm used. Further tuning of the controller's cost matrices could enhance system responsiveness and stability, leading to more effective maneuver execution. Moreover, the implementation of an H∞ controller would allow for a deeper analysis of the system's robustness under uncertainties and external disturbances, providing valuable insights into its real-world performance.

Additionally, refining the estimation of actuator energy consumption, particularly for RWs, would improve energy management and mission longevity. Extending the control framework by incorporating classical orbital elements into the control algorithm, which would require integrating thruster positioning and orientation into the GA, could lead to a more comprehensive control solution.

Finally, the development and testing of a more efficient RW configuration generator could significantly enhance the optimization process, resulting in improved system performance. By addressing these areas, future research could build upon the findings of this dissertation, further advancing the field of satellite attitude control and actuators configuration optimization.

11. References

[1] A. Grillo, "Low Earth Orbit (LEO): An Overview," Deep InSecurity, 16 4 2024. [Online]. Available: https://www.deepinsecurity.com/low-earth-orbit/. [Accessed 1 6 2025].

[2] European Space Agency, "SentiWiki Home - Copernicus," European Space Agency, 2025. [Online]. Available: https://sentiwiki.copernicus.eu/web/. [Accessed 1 6 2025].

[3] S. N. Goward, D. L. Williams, T. Arvidson, L. E. P. Rocchio, J. R. Irons, C. A. Russell and S. S. Johnston, Landsat's Enduring Legacy: Pioneering Global Land Observations from Space, Bethesda, Maryland, USA: American Society for Photogrammetry and Remote Sensing, 2017.

[4] ESA, "ESA Space Environment Report, Issue 6, Revision 0," ESA, Paris, France, 2022.

[5] F. Corporation, "Space Transportation Costs: Trends in Price Per Pound to Orbit 1990-2000," Fultron Corporation, Bethesda, Maryland, USA, 2002.

[6] H. W. Jones, "The Recent Large Reduction in Space Launch Cost," in *Proc. 48th International Conference on Environmental Systems (ICES)*, Albuquerque, NM, USA, 2018.

[7] C. S. I. S. A. S. Prog., "Cost of space launches to low Earth orbit," 2022. [Online]. Available: https://ourworldindata.org/grapher/cost-space-launches-low-earth-orbit?time=earliest..2019. [Accessed 1 6 2025].

[8] L. S. Vailshery, "Statista," 3 2022. [Online]. Available: https://www.statista.com/statistics/1293877/commercial-satellite-imagery-cost-worldwide/. [Accessed 1 6 2025].

[9] S. I. Barad and L. J. Katare, "Commercial Satellite Imaging Market Size | Forecast 2023–2032," Allied Market Research, UK, 2024.

[10] SatSummit Landscape Contributors, "Resolution Considerations," SatSummit , 2025. [Online]. Available: https://landscape.satsummit.io/capture/resolution-considerations.html. [Accessed 1 6 2025].

[11] NASA, "Space Environment Phases," 1 2 2019. [Online]. Available: https://www.nasa.gov/wp-content/uploads/2019/02/seh_figure_2-2_1_se_phases.png. [Accessed 1 6 2025].

[12] ESA, "How a mission is chosen," ESA, 2025. [Online]. Available: https://www.esa.int/Science_Exploration/Space_Science/How_a_mission_is_chosen. [Accessed 1 6 2025].

[13] ESA, "Satellites for maritime surveillance," 11 7 2011. [Online]. Available: https://www.esa.int/ESA_Multimedia/Images/2011/07/Satellites_for_maritime_surveillance. [Accessed 1 6 2025].

[14] NewSpace Systems, "LIBRA Reaction Wheel Datasheet," NewSpace Systems, Somerset West, Western Cape, South Africa, 2024.

[15] ESA, "Magnetotorquer," 15 6 2017. [Online]. Available: https://www.esa.int/ESA_Multimedia/Images/2017/06/Magnetotorquer. [Accessed 1 6 2025].

[16] J. L. C. Rodríguez, "Study of Analytical Solutions for Low-Thrust Trajectories," Escuela Técnica Superior de Ingenieros Aeronáuticos, Universidad Politécnica de Madrid, Madrid, Spain, 2017.

[17] M. Paluszek, Spacecraft Attitude Determination and Control, 1st ed., Amsterdam, Netherlands: Elsevier, 2023.

[18] H. D. Curtis, Orbital Mechanics for Engineering Students, 4th ed., Oxford, U.K.: Butterworth-Heinemann, 2020.

[19] J. R. Wertz, Spacecraft Attitude Determination and Control, Dordrecht, Netherlands: Springer, 1978.

[20] B. Wie, Space Vehicle Dynamics and Control, 2nd ed., Reston, VA, USA: AIAA Education Series, 2008.

[21] Y. Yang, "Spacecraft Attitude and Reaction Wheel Desaturation Combined," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 53, no. 1, pp. 286 - 295, 2017.

[22] R. Sutherland, I. Kolmanovsky and A. Girard, "Attitude Control of a 2U Cubesat by Magnetic and Air Drag Torques," *IEEE Transactions on Control Systems Technology,* vol. 27, no. 3, pp. 1047-1059, 2019.

[23] X. C. M. Cubillos and L. C. G. Souza, "Using of H-infinity control method in attitude control system of rigid-flexible satellite," *Mathematical Problems in Engineering,* vol. 2009, 2009.

[24] R. Rodrigues, L. C. G. S. Souza, A. Murilo and R. V. Lopes, "Hardware in the Loop Simulation for Model Predictive Control Applied to Satellite Attitude Control," *IEEE Access,* vol. 7, pp. 157401-157416, 2019.

[25] A. Kasiri, F. F. Saberi and M. Kashkul, "Optimization of Pyramidal Reaction Wheel Configuration for Minimizing Angular Momentum," in *24th International Conference on Electrical Machines and Systems (ICEMS)*, Gyeongju, Korea, 2021.

[26] A. M. Kassem, "Optimum Placement of Satellite Components with Genetic Algorithms," in *19th Annual AIAA/USU Conference on Small Satellites*, Dhahran, Saudi Arabia, 2005.

[27] J. Li, Y. Dong, M. Xu and H. Li, "Genetic Programming Method for Satellite System Topology and Parameter Optimization," *International Journal of Aerospace Engineering,* vol. 2020, pp. 1-14, 2020.

[28] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, 1st ed., Boston, USA: Addison-Wesley, 1989.

[29] A. Engelbrecht, "Computational Intelligence: An Introduction, 2nd ed.," Chichester: J. Wiley & Sons, Hoboken, New Jersey, USA, 2007.

[30] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks*, 1995.

[31] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by Simulated Annealing," *Science,* vol. 220, no. 4598, pp. 671-680, 1983.

[32] D. Marco and T. Stützle, "Handbook of Metaheuristics," in *The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances*, Springer, Boston, MA, USA, Springer, 2001, p. 250–285.

[33] R. Storn and K. Price, "Differential Evolution– A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization,* vol. 11, p. 341–359, 1997.

[34] O. Montenbruck and E. Gill, Satellite Orbits: Models, Methods, and Applications, Berlin, Germany: Springer, 2000.

[35] D. A. Vallado, Fundamentals of Astrodynamics and Applications, 4th ed., Hawthorne, CA: Microcosm Press, 2013.

[36] J. d. l. Torre, "Poliastro: Python Library for Orbital Mechanics," Poliastro, 2023. [Online]. Available: https://docs.poliastro.space/en/stable/index.html. [Accessed 1 6 2025].

[37] A. E. Hedin, J. M. Picone, D. P. Drob and A. C. Aikin, "NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues," *JGR space physics,* vol. 107, no. A12, pp. SIA 15-1-SIA 15-16, 2002.

[38] S. Bender, "PyNRLMSISE-00: A Python Implementation of the NRLMSISE-00 Atmosphere Model," 2020. [Online]. Available: https://pynrlmsise00.readthedocs.io/en/latest/index.html. [Accessed 1 6 2025].

[39] L. Kauder, "Spacecraft Thermal Control Coatings References," NASA/Goddard Space Flight Center, Greenbelt, Maryland, USA, 2005.

[40] NAIF, "SPICE: Generic Kernels – Planetary Ephemeris SPK Files," 2023. [Online]. Available: https://naif.jpl.nasa.gov/pub/naif/generic_kernels/spk/planets/. [Accessed 1 6 2025].

[41] Astropy Collaboration, "Astropy: Python Tools for Astronomy," Astropy, 2023. [Online]. Available: https://docs.astropy.org/en/latest/index.html. [Accessed 1 6 2025].

[42] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt and SciPy 1.0 Contributors, "SciPy 1.0:

Fundamental Algorithms for Scientific Computing in Python," *Python. Nat Methods,* vol. 17, p. 261–272, 2020.

[43] K. Ogata, Modern Control Engineering, 5th ed., Upper Saddle River, NJ, USA: Prentice Hall, 2010.

[44] B. Friedland, Control System Design: An Introduction to State-Space Methods, Mineola, NY, USA: Dover Publications, 2012.

[45] G. F. Franklin, J. D. Powell and A. Emami-Naeini, Feedback Control of Dynamic Systems, 7th ed., Boston, MA, USA: Pearson, 2014.

[46] C. T. Chen, Linear System Theory and Design, 4th ed., New York, NY, USA: Oxford University Press, 2012.

[47] A. E. Bryson, Applied Optimal Control: Optimization, Estimation, and Control, 1st ed., London, UK: Routledge, 1975.

[48] D. o. H. &. H. Services, "Concept of Operations (ConOps)," 2008. [Online]. Available: https://web.archive.org/web/20090611232934/http://www3.cms.hhs.gov/SystemLifecycleFramework/Downloads/ConOps.pdf. [Accessed 1 6 2025].

[49] L. B. Rainey and A. Tolk, "Modeling and Simulation Support for System of Systems Engineering Applications," Wiley, New Jersey, USA , 2014.

[50] M. Blanke and M. B. Larsen, Satellite Dynamics and Control in a Quaternion Formulation, 2nd edition, Kongens Lyngby, Denmark: Technical University of Denmark, Department of Electrical Engineering, 2010.

[51] Python Control Systems Library, "Control Python API Documentation," 2023. [Online]. Available: https://python-control.readthedocs.io/en/latest/index.html. [Accessed 1 6 2025].

[52] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau and C. Gagné, "DEAP: Distributed Evolutionary Algorithms in Python," 2023. [Online]. Available: https://deap.readthedocs.io/en/master/index.html. [Accessed 1 6 2025].

[53] M. Mitchell, "An Introduction to Genetic Algorithms," MIT Press, Boston, USA, 1996.

[54] M. Clerc, Particle Swarm Optimization, London, UK: ISTE Ltd., 2006.

[55] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence*, Anchorage, AK, USA, 1998.

[56] J. Kennedy, R. C. Eberhart and Y. Shi, Swarm Intelligence (The Morgan Kaufmann Series in Evolutionary Computation), 1st ed., Burlington, Massachusetts, USA: Morgan Kaufmann, 2001.

[57] R. M. Fredo, A Numerical Procedure for Calculating the Aerodynamic Coefficients for Complex Spacecraft Configurations in Free-Molecular Flow, Pennsylvania, USA: The Pennsylvania State University, 1980.

[58] J. A. Storch, "Aerodynamic Disturbances on Spacecraft in Free Molecular Flow," The Aerospace Corporation, California, USA, 2002.

12. Appendix

Appendix A Satellite Properties

Figure 12.1 shows a technical drawing of the satellite considered for the simulations, where the dimensions and positions of the satellite's components are specified.

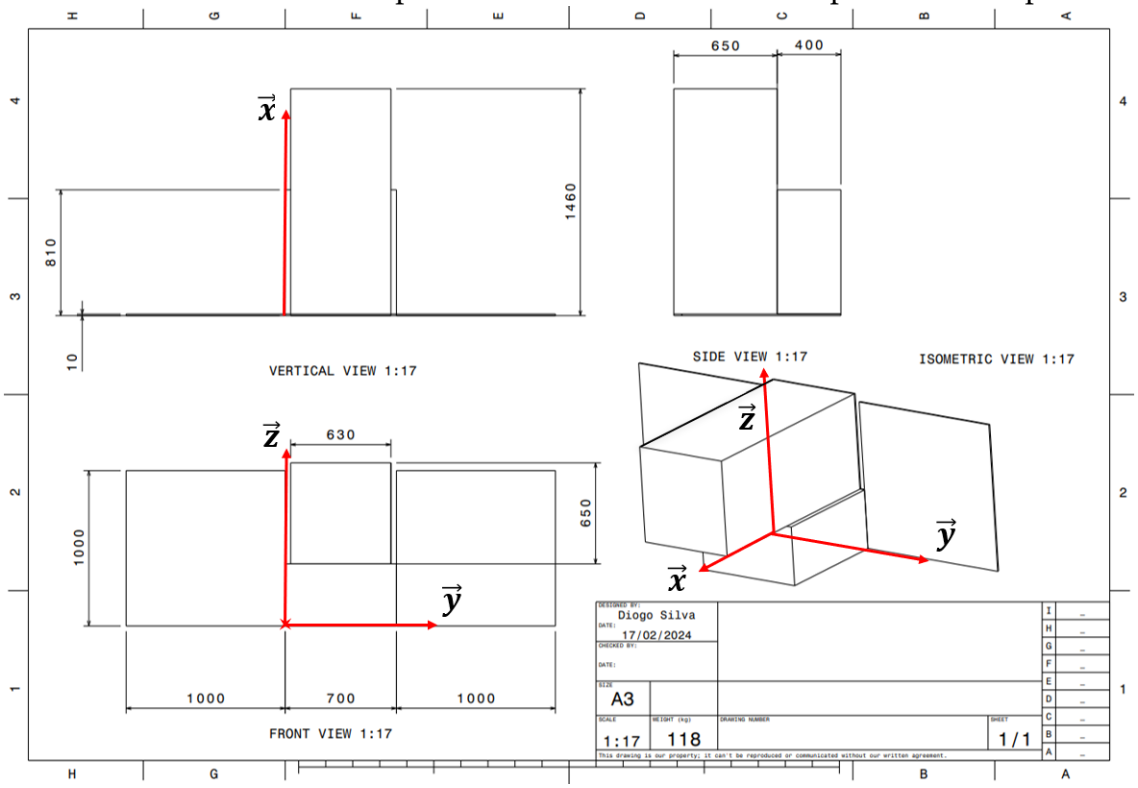

Figure 12.1: Technical Drawing of the Satellite

Table 12.1 shows the physical characteristics of the satellite, where the dimensions and positions of the satellite's components are specified in detail:

Table 12.1: Sat. Physical Characteristics

[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[kg]	[-]
x_size	**y_size**	**z_size**	**x**	**y**	**z**	**mass**	**name**
10	1000	1000	5	-500	500	3.5	solar panel 1
10	1000	1000	5	1200	500	3.5	solar panel 2
810	700	400	405	350	200	56	instrument box
1460	630	650	730	350	725	55	telescope

Appendix B Jacobian Matrix

The Jacobian matrix is a fundamental concept in multivariable calculus and numerical analysis, particularly relevant in systems involving vector-valued functions. It represents the first-order partial derivatives of a vector function with respect to a set of input variables and plays a critical role in sensitivity analysis, optimization, and nonlinear system modeling.

$$\boldsymbol{J_f} = \begin{bmatrix} \dfrac{\partial \boldsymbol{f}}{\partial x_1} & \cdots & \dfrac{\partial \boldsymbol{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \nabla^T f_1 \\ \vdots \\ \nabla^T f_m \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial f_m}{\partial x_1} & \cdots & \dfrac{\partial f_m}{\partial x_n} \end{bmatrix} \tag{12.1}$$

Appendix C Quarternion Operations

Quaternions are one of the most practical methods for representing the attitude of a body over time. In order to utilize quaternions effectively, a set of operations, outlined below, must be applied.

Transpose of a quaternion:

$$q^T = \begin{bmatrix} q_0 \\ -q_1 \\ -q_2 \\ -q_3 \end{bmatrix} = \begin{bmatrix} s \\ -v_1 \\ -v_2 \\ -v_3 \end{bmatrix} = \begin{bmatrix} cos\dfrac{\phi}{2} \\ -a_1 sin\dfrac{\phi}{2} \\ -a_2 sin\dfrac{\phi}{2} \\ -a_3 sin\dfrac{\phi}{2} \end{bmatrix} \tag{12.2}$$

Property of quaternion multiplication by its transpose:

$$qq^T = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{12.3}$$

Quaternion multiplication:

$$q_1 \otimes q_2 = \begin{bmatrix} s_1 s_2 - \vec{v}_1 \vec{v}_2 \\ s_1 \vec{v}_1 + s_2 \vec{v}_2 + \vec{v}_1 \times \vec{v}_2 \end{bmatrix} \tag{12.4}$$

where s1 is the scalar part of the first quaternion and s2 is the scalar part of the second quaternion, v1 and v2 are the corresponding vector parts. The arrows are used to clarify that they are vectors. Quaternion multiplication can be expressed in matrix form in two ways:

Expanding quaternion multiplication:

$$q_1 \otimes q_2 = \begin{bmatrix} q_1(1) & -q_1(2) & -q_1(3) & -q_1(4) \\ q_1(2) & q_1(1) & -q_1(4) & q_1(3) \\ q_1(3) & q_1(4) & q_1(1) & -q_1(2) \\ q_1(4) & -q_1(3) & q_1(2) & q_1(1) \end{bmatrix} q_2 \tag{12.5}$$

Or:

$$q_1 \otimes q_2 = \begin{bmatrix} q_2(1) & -q_2(2) & -q_2(3) & -q_2(4) \\ q_2(2) & q_2(1) & q_2(4) & -q_2(3) \\ q_2(3) & -q_2(4) & q_2(1) & q_2(2) \\ q_2(4) & q_2(3) & -q_2(2) & q_2(1) \end{bmatrix} q_1 \tag{12.6}$$

Quaternions transform vectors using the following operation:

$$x_b = q_{ba} \otimes x_a \otimes q_{ba}^T \tag{12.7}$$

using quaternion multiplication with the vectors defined as quaternions with a scalar part equal to zero or:

$$x_a = \begin{bmatrix} 0 \\ x_a(1) \\ x_a(2) \\ x_a(3) \end{bmatrix} \tag{12.8}$$

Writing this out in full:

$$x_b = \begin{bmatrix} q_{ba}(1) & -q_{ba}(2) & -q_{ba}(3) & -q_{ba}(4) \\ q_{ba}(2) & q_{ba}(1) & q_{ba}(4) & -q_{ba}(3) \\ q_{ba}(3) & -q_{ba}(4) & q_{ba}(1) & q_{ba}(2) \\ q_{ba}(4) & q_{ba}(3) & -q_{ba}(2) & q_{ba}(1) \end{bmatrix} \begin{bmatrix} 0 & -x_a(1) & -x_a(2) & -x_a(3) \\ x_a(1) & 0 & x_a(3) & -x_a(2) \\ x_a(2) & -x_a(3) & 0 & x_a(1) \\ x_a(3) & x_a(2) & -x_a(1) & 0 \end{bmatrix} \begin{bmatrix} q_{ba}(1) \\ -q_{ba}(2) \\ -q_{ba}(3) \\ -q_{ba}(4) \end{bmatrix} \tag{12.9}$$

The middle matrix is the skew-symmetric matrix of Equation 12.8.

Appendix D Inertial Matrix

The inertial matrix is essential in modeling the rotational dynamics of rigid bodies in aerospace systems. It defines how mass is distributed relative to rotational axes and how the body reacts to applied torques. This appendix presents its mathematical definition, key properties, and typical structures used in satellite attitude dynamics.

The elements of the inertia matrix are:

$$I_{xx} = \iiint_V^0 \rho(x,y,z)(y^2+z^2)\,dxdydz \tag{12.10}$$

$$I_{yy} = \iiint_V^0 \rho(x,y,z)(x^2+z^2)\,dxdydz \tag{12.11}$$

$$I_{zz} = \iiint_V^0 \rho(x,y,z)(x^2+y^2)\,dxdydz \tag{12.12}$$

$$I_{xy} = -\iiint_V^0 \rho(x,y,z)xy\,dxdydz \tag{12.13}$$

$$I_{xz} = -\iiint_V^0 \rho(x,y,z)xz\,dxdydz \tag{12.14}$$

$$I_{yz} = -\iiint_V^0 \rho(x,y,z)yz\,dxdydz \tag{12.15}$$

The center-of-mass is found from:

$$c_x = \frac{1}{M}\iiint_V^0 \rho(x,y,z)x\,dxdydz \tag{12.16}$$

$$c_y = \frac{1}{M}\iiint_V^0 \rho(x,y,z)y\,dxdydz \tag{12.17}$$

$$c_z = \frac{1}{M}\iiint_V^0 \rho(x,y,z)z\,dxdydz \tag{12.18}$$

Where c is the vector to the center-of-mass and:

$$M = \iiint_V^0 \rho(x,y,z)\,dxdydz \tag{12.19}$$

Appendix E Quaternion differential equation Linearization, time-step tests

The kinematics of the satellite, in described using the attitude quaternion $q = [q_1, q_2, q_3, q_4]^T$ to represent a rotation. The orientation of the satellite is obtained rotating from the inertial frame to the satellite frame. The kinematics of the satellite is [50]:

$$\frac{d}{dt}\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix}\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{12.20}$$

In a compact notation:

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{\Omega}(\omega)\boldsymbol{q} \tag{12.21}$$

The kinematics equation can be written using $q = [\boldsymbol{g}^T, q_4]$ where $\boldsymbol{g}$ in the first three components of the quaternion:

$$\frac{d}{dt}\begin{bmatrix} \boldsymbol{g} \\ q_4 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -\boldsymbol{S}(\omega) \\ -\omega^T \end{bmatrix}\boldsymbol{g} + \frac{1}{2}q_4\begin{bmatrix} \boldsymbol{I}_{3\times 3} \\ 0 \end{bmatrix}\omega \tag{12.22}$$

The unit length of the quaternion is preserved in this operation.

This non-linear equation of motion is now linearized in an arbitrary point of operation $(\bar{\omega}, \bar{\boldsymbol{g}}, \bar{q}_a, \bar{h})$ in order to arrive at a set of linear state-space equations. This is needed to enable stringent stability and performance analysis for the linear control systems. The deviation from steady state (point of linearization) is denoted by a tilde above the variables, $\omega = \bar{\omega} + \tilde{\omega}$, but the quaternion representation of attitude poses a specific problem. With $d\boldsymbol{g}$ denoting the orientation at time $t + dt$ relative to the attitude at time t, then, since [50]:

$$\begin{bmatrix} \tilde{g} \\ \tilde{q}_4 \end{bmatrix} = \begin{bmatrix} dg_1 \\ dg_2 \\ dg_3 \\ dq_4 \end{bmatrix} = \begin{bmatrix} e_1 \sin(1/2\omega dt) \\ e_2 \sin(1/2\omega dt) \\ e_3 \sin(1/2\omega dt) \\ \cos(1/2\omega dt) \end{bmatrix} \simeq \begin{bmatrix} 1/2\omega_1 dt \\ 1/2\omega_2 dt \\ 1/2\omega_3 dt \\ 1 \end{bmatrix} \tag{12.23}$$

Where e_1, e_2 and e_3 are the three vectorial elements of the rotation angle of the quaternion.

Then $\frac{d}{dt}q_4 = 0$ and $\boldsymbol{S}(\omega)d\boldsymbol{g} = 0$ hence:

$$\frac{d}{dt}\tilde{\boldsymbol{g}} = -\frac{1}{2}\boldsymbol{S}(\omega)\tilde{\boldsymbol{g}} + \frac{1}{2}\tilde{q}_4\boldsymbol{I}_{3\times 3}\omega = \frac{1}{2}\boldsymbol{I}_{3\times 3}\omega \tag{12.24}$$

Appendix F A_2 Jacobians results

This appendix presents the Jacobian matrices for the A_2. These results support the dynamic analysis and control design discussed:

$$J_f(\omega_1, \omega_2, \omega_3) = \begin{bmatrix} \frac{\partial f_1}{\partial \omega_1} & \frac{\partial f_1}{\partial \omega_2} & \frac{\partial f_1}{\partial \omega_3} \\ \frac{\partial f_2}{\partial \omega_1} & \frac{\partial f_2}{\partial \omega_2} & \frac{\partial f_2}{\partial \omega_3} \\ \frac{\partial f_3}{\partial \omega_1} & \frac{\partial f_3}{\partial \omega_2} & \frac{\partial f_3}{\partial \omega_3} \end{bmatrix} = \begin{bmatrix} A_{2,11} & A_{2,12} & A_{2,13} \\ A_{2,21} & A_{2,22} & A_{2,23} \\ A_{2,31} & A_{2,32} & A_{2,33} \end{bmatrix} \tag{12.25}$$

$$A_{2,11} = D_{31}\omega_2 + I_{31}\omega_2 - D_{21}\omega_3 - I_{21}\omega_3 \tag{12.26}$$

$$A_{2,21} = -2D_{31}\omega_1 - 2I_{31}\omega_1 - S_3 - D_{32}\omega_2 - I_{32}\omega_2 + D_{11}\omega_3 - D_{33}\omega_3 + I_{11}\omega_3 - I_{33}\omega_3 \tag{12.27}$$

$$A_{2,31} = 2D_{21}\omega_1 + 2I_{21}\omega_1 + S_2 - D_{11}\omega_2 + D_{22}\omega_2 - I_{11}\omega_2 + I_{22}\omega_2 + D_{23}\omega_3 + I_{23}\omega_3 \tag{12.28}$$

$$A_{2,12} = 2D_{32}\omega_2 + 2I_{32}\omega_2 + S_3 + D_{31}\omega_1 + I_{31}\omega_1 + D_{33}\omega_3 - D_{22}\omega_3 + I_{33}\omega_3 - I_{22}\omega_3 \tag{12.29}$$

$$A_{2,22} = -D_{32}\omega_1 - I_{32}\omega_1 + D_{12}\omega_3 + I_{12}\omega_3 \tag{12.30}$$

$$A_{2,32} = -2D_{12}\omega_2 - 2I_{12}\omega_2 - S_1 - D_{11}\omega_1 + D_{22}\omega_1 - I_{11}\omega_1 + I_{22}\omega_1 - D_{13}\omega_3 - I_{13}\omega_3 \tag{12.31}$$

$$A_{2,13} = -2D_{23}\omega_3 - 2I_{23}\omega_3 - S_2 - D_{21}\omega_1 - I_{21}\omega_1 + D_{33}\omega_2 - D_{22}\omega_2 + I_{33}\omega_2 - I_{22}\omega_2 \tag{12.32}$$

$$A_{2,23} = 2D_{13}\omega_3 + 2I_{13}\omega_3 + S_1 + D_{11}\omega_1 - D_{33}\omega_1 + I_{11}\omega_1 - I_{33}\omega_1 + D_{12}\omega_2 + I_{12}\omega_2 \tag{12.33}$$

$$A_{2,33} = D_{23}\omega_1 + I_{23}\omega_1 - D_{13}\omega_2 - I_{13}\omega_2 \tag{12.34}$$

Where:

$$\omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{12.35}$$

$$I = \begin{bmatrix} I_{11} & I_{12} & I_{13} \\ I_{21} & I_{22} & I_{23} \\ I_{31} & I_{32} & I_{33} \end{bmatrix} \tag{12.36}$$

$$S = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix} = \sum_k u_k J_k \Omega_k \tag{12.37}$$

$$D = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix} = \sum_k u_k J_k u_k^T \tag{12.38}$$

Appendix G Free Molecular Flow Equation

This appendix presents the fundamental equations governing free molecular flow, which describes the interaction between gas molecules and a satellite's surface in rarefied atmospheric conditions. These formulations are essential for understanding drag force modeling in low-density environments. In particular, the derivation included here demonstrates that when the angle $\beta = \pi/2$, the drag coefficient C_D reaches a value of 4.

At altitudes above 120 km, the atmospheric density is sufficiently low that continuum flow does not apply. The Knudsen number shows the flow regime; less than 0.01 is continuum flow, greater than 10 is a free-molecular flow, the Knudsen number is given by:

$$Kn = \frac{\lambda}{L} \tag{12.39}$$

Where λ is the mean free path and L is a characteristic dimension of the spacecraft. L for a 1U CubeSat is 10 cm. For the International Space Station (ISS), it ranges from 73 m to 109 m. The mean free path for a Boltzmann gas is:

$$\lambda = \frac{\mu}{\rho}\sqrt{\frac{\pi m}{2k_B T}} \tag{12.40}$$

Where μ is the dynamic viscosity, k_B is the $Boltzmann\ constant\ = 1.38 \times 10^{-23}\ J/K$, m is the molecular weight, and T is the temperature. The Knudsen number then becomes:

$$Kn = \frac{\mu}{\rho L}\sqrt{\frac{\pi m}{2k_B T}} \tag{12.41}$$

An alternative formulation is:

$$Kn = \frac{k_B T}{\sqrt{2}\pi d^2 p L} \tag{12.42}$$

Where d is the hard-shell diameter of the particles and p is the total pressure. The hard-shell diameters of some molecules are given in the table below:

Table 12.2: Hard-Shell or Kinetic Diameters

Name	Formula	Molecular Weight	Diameter (pm)
Hydrogen	H	2	289
Water	HO	18	265
Nitrogen	N	28	364
Carbon Dioxide	CO	44	330

For a free-molecular flow, an incoming particle strikes the surface, where it can be absorbed, reflected specularly, or reflected diffusely. Surface-accommodation coefficients [57] [58]incorporate these effects.

The surface-accommodation coefficients are defined as:

$$\sigma_n = \frac{p_i - p_r}{p_i - p_w} \tag{12.43}$$

$$\sigma_t = \frac{\tau_i - \tau_r}{\tau_i - \tau_w} \tag{12.44}$$

Where p_i and p_r are the incident and reflected normal momentum flux, τ_i and τ_r are the incident and reflected tangential momentum flux, p_w and τ_w are the normal and tangential components of momentum carried away by diffusively emitted momentum, and τ_w is, by definition, zero. For purely specular reflection with, $p_i = p_r$ and $\tau_i = \tau_r$, it follows that $\sigma_n = \sigma_t = 0$.

There are two approaches to computing aerodynamic forces. One assumes that the only contribution is the motion of the spacecraft, while the other assumes that the particles in the atmosphere have a Maxwellian velocity distribution. The average normal velocity of diffusely reflected molecules is given by:

$$V_w = \sqrt{\frac{\pi R T_w}{2\mathfrak{M}}} \tag{12.45}$$

Where R is the ideal gas constant (8314.5 J/kg-mole K), $\mathfrak{M}$ is the mean molecular weight, and T is the gas temperature, which can be obtained from the atmospheric model. The surface-accommodation coefficients are also a function of the incidence angle. Experimental results obtained by bombarding an aluminum surface with nitrogen [k3]:

$$\sigma_n = 0.93 - 1.48 \times 10^{-3}\theta - 7 \times 10^{-5}\theta^2 \tag{12.46}$$

$$\sigma_t = 0.63\left(1 - e^{-3.38\times10^{-2\theta}}\right) \tag{12.47}$$

For the case of zero atmospheric motion, the following expressions are obtained:

$$C_L = \left[\sigma_n \frac{V_w}{V} + (2 - \sigma_n - \sigma_t) sin\beta\right] sin2\beta \tag{12.48}$$

$$C_D = 2\left[\sigma_t + \sigma_n \frac{V_w}{V} sin\beta + (2 - \sigma_n - \sigma_t) sin^2\beta\right] sin\beta \tag{12.49}$$

For a pure specular case with β =π/2, C_D will be equal to 4.

$$C_D = 4 \tag{12.50}$$

Appendix H Reaction Wheels Consumption Model Coefficients Calculations

This appendix details the calculation of the coefficients used in the reaction wheel energy consumption model. These coefficients are derived based on the physical properties and operating characteristics of the actuators and are essential for accurately estimating power usage during attitude maneuvers.

PERFORMANCE	LIBRA-065 [NRWA-T065]	LIBRA-2 [NRWA-T2]	LIBRA-6 [NRWA-T6]	LIBRA-80
FUNCTIONAL CHARACTERISTICS				
Torque	20 mNm (Nominal) 24 mNm (Peak)	84 mNm (Nominal) 90 mNm (Peak)	150 mNm (Nominal) 310 mNm (Peak)	175 mNm (Nominal) 300 mNm (Peak)
Momentum	0.65 Nms @ 6500 RPM (Nominal) 0.94 Nms @ 9000 RPM (Peak)	1 Nms @ 1850 RPM (Nominal) 1.47 Nms @ 2700 RPM (Peak)	6 Nms @ 3845 RPM (Nominal) 7.83 Nms @ 5000 RPM (Peak)	80 Nms @ 3600RPM (Nominal) 100 Nms @ 4500RPM (Peak)
Speed range	±9000 rpm (@24V-34V)	±2600 RPM (Speed/Torque mode) ±2700 RPM (Current mode) (@24V-34V)	±5000 RPM (@20V-36V)	±3100 RPM (@20V) ±4500 RPM (@28V) ±5550 RPM (@34V)
Rotor moment of inertia	1.0 x 10-3 kg.m2	5.2 x 10-3 kg.m2	14.9 x 10-3 kg.m2	212.5 x 10-3 kg.m2
Speed control accuracy	±1 RPM @ >100 RPM	±1 RPM @ >100 RPM	±1 RPM @ >10 RPM ±0.14RPM @ > 100RPM	±1 RPM @ >10 RPM ±0.14RPM @ > 100RPM
PHYSICAL CHARACTERISTICS				
Dimensions (wheel)	102 mm x 102 mm x 105 mm	150 mm x 150 mm x 75 mm	206 mm x 206 mm x 100 mm	460 mm x 190.5mm
Dimensions (electronics)	Internal	Internal	Internal	198 mm x 182 mm x 36.5 mm
Mass	1.6 kg	2.8 kg	4.7 kg	8.2kg (wheel) 1.2kg (electronics)
INTERFACES				
Power consumption	2.6W (Quiescent) 4W (Steady state at 3000RPM) 6W (Steady state at 6500RPM)	1.7W (Quiescent) 4W (Steady state at 1500RPM) 4.8W (Steady state at 2600RPM)	5.1W (Quiescent) 15.3W (Steady state at 3Nms) 29.3W (Steady state at 6Nms) 139.6W Peak power consumption	4.7W (Quiescent) 17.3W (steady state at 44Nms) 28W (steady state at 86Nms) 76W (Peak torque at 32Nms) 188W (Peak power consumption)

Figure 12.2: LIBRA Reaction Wheel Properties

Recalling the equation that represents the model used to calculate the energy consumed over a defined time interval:

$$E_c(\overline{\omega}, T, h) = \left((a * \overline{\omega} + b) + (c * \overline{\omega} + d) * (e * \overline{\omega} + f) * \left(\frac{T}{g}\right) \right) * h_{step} \qquad (12.51)$$

Where, $\overline{\omega}$ represents the average speed during the integration step, denoted as h.

For the LIBRA-065, it is necessary to calculate the corresponding coefficients a and b. Figure 12.3 presents the linear regression of the Steady-State Consumption data as a function of angular velocity:

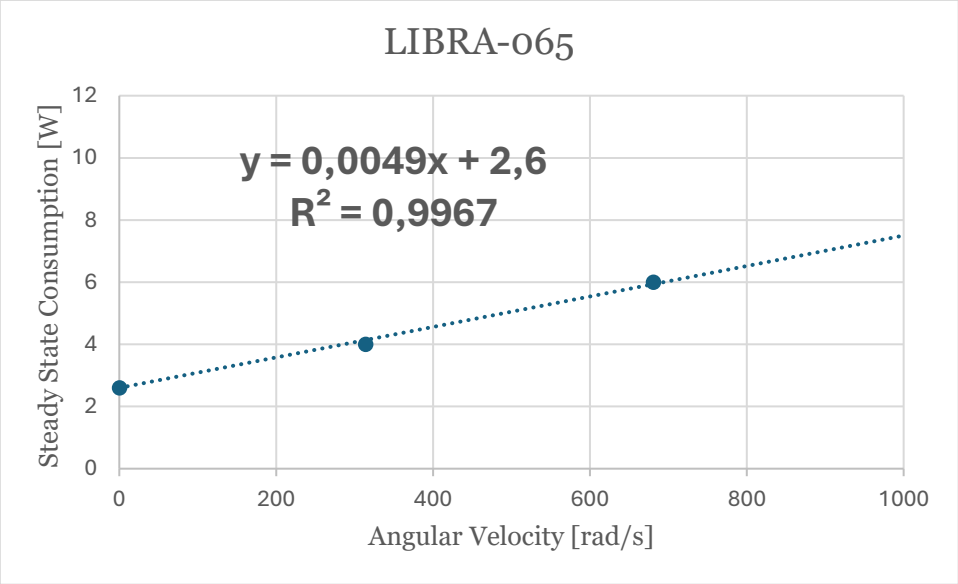

Figure 12.3: LIBRA-065 linear regression

Where:

$$a = 0.0049 \qquad (12.52)$$

$$b = 2.6 \qquad (12.53)$$

For the LIBRA-2, the corresponding coefficients a and b must be calculated. Figure 12.4 presents the linear regression of the Steady-State Consumption data as a function of angular velocity:

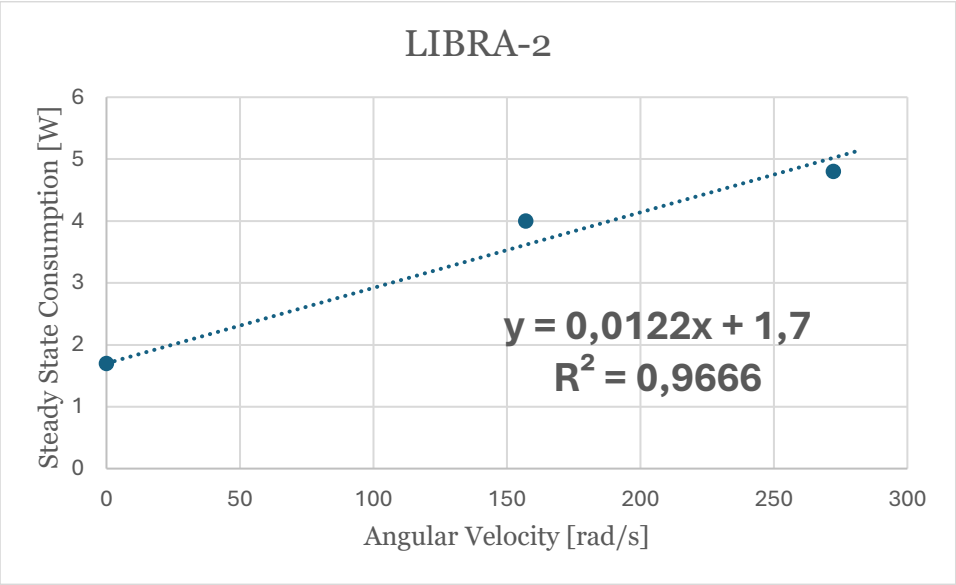

Figure 12.4: LIBRA-2 linear regression

Where:

$$a = 0.0122 \tag{12.54}$$

$$b = 1.7 \tag{12.55}$$

For the LIBRA-6, the corresponding coefficients a, b, c, d and f must be calculated. Figure 12.5 presents the linear regression of the Steady-State Consumption data as a function of angular velocity:

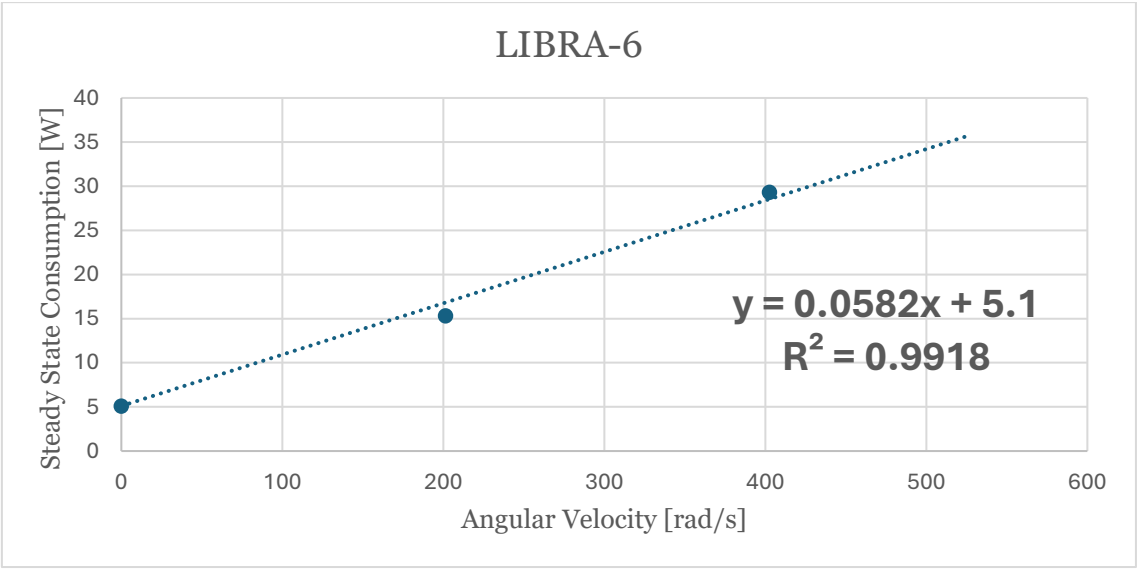

Figure 12.5: LIBRA-6 linear regression

Given a peak torque consumption of 139.6 W at a velocity of 523.6 rad/s, the steady-state consumption is 35.5734 W. Therefore, the torque consumption percentage is 0.74518%, and the torque consumption constant (k) is 2.92428 at 100% torque.

Where:

$$a = c = 0.0582 \tag{12.56}$$

$$b = d = 5.1 \tag{12.57}$$

$$f = 2.92428 \tag{12.58}$$

For the LIBRA-80, the corresponding coefficients a, b, c, d, e and f must be calculated. Figure 12.6 presents the linear regression of the Steady-State Consumption data as a function of angular velocity:

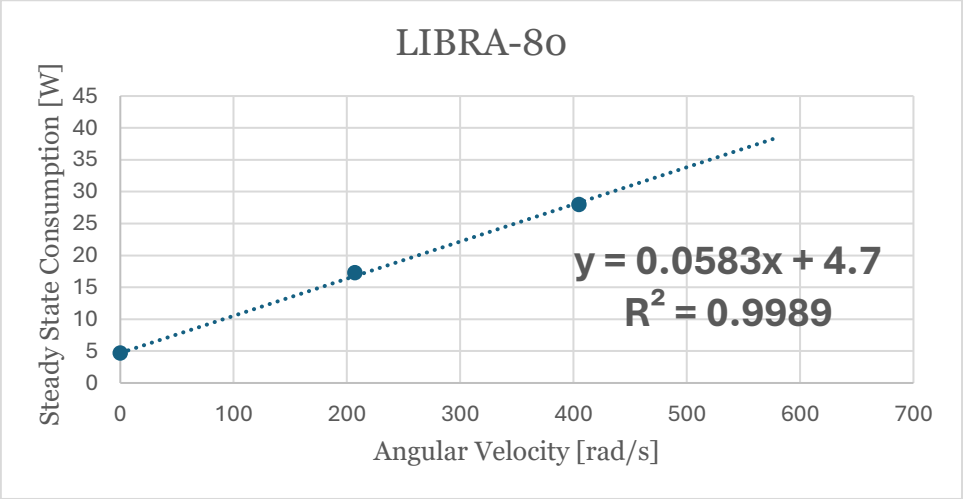

Figure 12.6: LIBRA-80 first linear regression

Since there are two Peak Torque Consumption values for the LIBRA-80, two corresponding Torque Consumption Constants are defined. Their values are presented in Table 12.3:

Table 12.3: k constants calculations

	Peak Torque 1	Peak Torque 2	Unit
Angular Velocity	150.5882353	575.9586532	Rad/s
Peak True Consumption	76	188	W
Consumption in Steady State	13.47929412	38.27838948	W
Torque Consumption percentage	82.2640867	79.6391545	%
Torque Consumption Constant (k)	4.638277445	3.911387406	-

Figure 12.7 presents the linear regression of the Torque Consumption Constant data as a function of angular velocity:

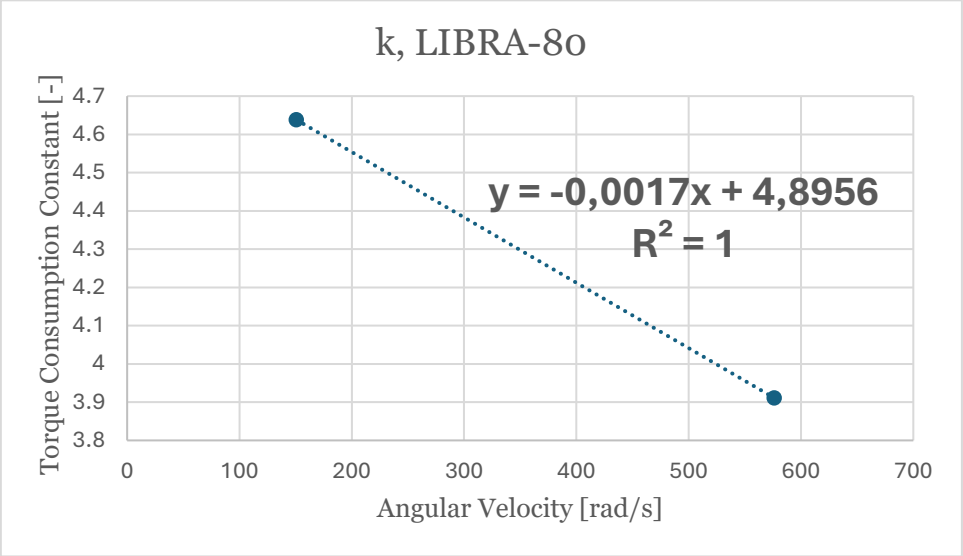

Figure 12.7: LIBRA-80 second linear regression

Where:

$$a = c = 0.0583 \tag{12.59}$$

$$b = d = 4.7 \tag{12.60}$$

$$e = -0.0017 \tag{12.61}$$

$$f = 4.8956 \tag{12.62}$$

Appendix I Quaternion Differential Linearization Error

To determine the appropriate time step, the maximum absolute error values of the propagated Gibbs vector were calculated using both the quaternion differential equation with respect to angular velocity and its corresponding linearized differential equation. Table 12.4 presents the maximum absolute error for various time steps (in seconds) across different rotation rates (in degrees per second) along all rotational axes.

Table 12.4: Absolute Error of the Linearization

	1 seg	**0.5 seg**	**0.25 seg**	**0.1 seg**	**0.05 seg**
0 deg/seg	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
10 deg/seg	3.32E-04	4.15E-05	5.19E-06	3.32E-07	4.15E-08
20 deg/seg	2.65E-03	3.32E-04	4.15E-05	2.66E-06	3.32E-07
30 deg/seg	8.88E-03	1.12E-03	1.40E-04	8.97E-06	1.12E-06
40 deg/seg	2.09E-02	2.65E-03	3.32E-04	2.13E-05	2.66E-06
50 deg/seg	4.04E-02	5.16E-03	6.48E-04	4.15E-05	5.19E-06
60 deg/seg	6.89E-02	8.88E-03	1.12E-03	7.17E-05	8.97E-06
70 deg/seg	1.08E-01	1.40E-02	1.77E-03	1.14E-04	1.42E-05
80 deg/seg	1.58E-01	2.09E-02	2.65E-03	1.70E-04	2.13E-05
90 deg/seg	2.21E-01	2.96E-02	3.76E-03	2.42E-04	3.03E-05
100 deg/seg	2.96E-01	4.04E-02	5.16E-03	3.32E-04	4.15E-05
110 deg/seg	3.85E-01	5.34E-02	6.85E-03	4.42E-04	5.53E-05
120 deg/seg	4.87E-01	6.89E-02	8.88E-03	5.73E-04	7.17E-05
130 deg/seg	6.01E-01	8.69E-02	1.13E-02	7.29E-04	9.12E-05
140 deg/seg	7.28E-01	1.08E-01	1.40E-02	9.10E-04	1.14E-04
150 deg/seg	8.66E-01	1.31E-01	1.72E-02	1.12E-03	1.40E-04
160 deg/seg	1.01E+00	1.58E-01	2.09E-02	1.36E-03	1.70E-04
170 deg/seg	1.17E+00	1.88E-01	2.50E-02	1.63E-03	2.04E-04
180 deg/seg	1.34E+00	2.21E-01	2.96E-02	1.93E-03	2.42E-04

Considering a maximum allowable error of 10^{-4}, and given that it is unlikely for the satellite to undergo a rotation greater than 5 degrees within 0.25 seconds, the time step of 0.25 seconds was selected. This choice also contributes to reducing the computational effort required to perform the simulations.

Appendix J Constant Mass Justification in the Orbital Simulation

Three simulations are conducted based on the initial orbital parameters presented in Table 12.5. Each simulation uses a different satellite mass: the minimum mass corresponds to the base satellite mass (with components detailed in Appendix A) plus the mass of three LIBRA-065 reaction wheels, 122.8 kg. The maximum mass is defined as the base satellite mass plus the mass of twelve LIBRA-80 reaction wheels, 216.4 kg. The third case considers the average between the minimum and maximum masses, 169.5 kg.

Table 12.5: Initial Orbital Parameters

Epoch ISO	2023-04-17 00:00:00.000
Time scale	TAI
Inertial frame	ECI
θ	0°
e	0.001075637
i	90°
Ω	351.320169°
ω	41.172731°
a	6878 km

Subsequently, for each component of the propagated position vector across the three simulations, the standard deviation of the corresponding orthogonal elements is computed. Equation 12.62 shows the formula for the standard deviation:

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)}{n}} \tag{12.63}$$

Where x_i is the individual population value, μ the men population value and n the total population size.

The resulting standard deviation is less than 10^{-5}, indicating a very small variation in the satellite's position throughout its orbit (the CONOPS comprises only one orbital period). Therefore, it was decided to adopt a constant mass for the orbital propagation, specifically the average between the maximum and minimum masses, 169.6 kg.

Appendix K PSO Convergence Demonstration

This appendix aims to demonstrate that the PSO used in the simulations of Chapter 9 possesses adequate convergence capability, making it possible to achieve the simulation objectives within a reasonable time frame. Specifically, it shows that the results of the simulations for the individuals composing the populations converge in terms of final energy consumption values, allowing the identification of a minimum value.

Table 12.6 shows the elements variation values for the individuals used, based on the cube RW configuration:

Table 12.6: PSO Convergence Demo. Elements Variation Values

Cube Individual Elements	**max**	**min**	**delta**	**units**
conf_type	3	0	1	[-]
rw_type	3	0	1	[-]
alfa	45	0	5	[deg]
x	1000	600	50	[mm]
y	875	525	25	[mm]
z	400	300	25	[mm]
r	100	25	25	[mm]

For the PSO simulation parameters, the population consists of five individuals evolved over ten generations. The maximum and minimum element velocities are set to 5 and −5, respectively, with a position update increment of 5.

Table 12.7 shows the results obtained in each generation simulated:

Table 12.7: Convergence Demonstration Simulation Results

[-]	[-]	[Wh]	[Wh]
Generation	**Valid Ind. Nº**	**Best fitness value**	**Mean fitness value**
1	1	43.79118	43.79118
2	2	43.63785	45.28
3	1	43.55704	43.55704
4	2	43.55704	43.87269
5	2	43.55704	43.60116
6	3	43.55704	44.09429
7	2	43.55704	43.58569
8	0	43.55704	n/a
9	3	43.55704	43.70737
10	3	43.55704	43.62544

Analyzing Table 12.7, it is possible to affirm that even for low numbers of simulated generations, there is convergence towards the minimum value of total energy consumed by the ACS.