Attitude Control of a Satellite based on a Decentralized Architecture

Manuel Varela Gomes de Azevedo

Master's degree dissertation

Orientation: Prof. Sérgio Cunha PhD

Master's in Mechanical Engineering

June 2019

Abstract

Firstly, this dissertation presents a solution for the Attitude Control System (ACS) of satellites up to 150 Kg, using reaction wheels and magnetorquers as actuators. This solution is made up of a collection of autonomous control algorithms that are able to detumble the spacecraft upon orbit injection, maintain a reference attitude according to the mission phase, desaturate the reaction wheels and keep them at an ideal velocity. The case study of this work is the portuguese satellite *Infante* that is currently under development.

Secondly, this dissertation presents a new concept of control architecture for small satellites, called decentralized architecture. In this architecture, each reaction wheel has its own local, independent ACS, which has no knowledge regarding the existence of the others. This independence allows for a greater modularity and development flexibility, since reaction wheels can be modified, removed or added, without having to modify the others. This total system independence has as a consequence differences between the control action computed by the controller, and the action that is effectively performed by the reaction wheel assembly. This difference is formally described, showing that for the typical cases, this has reduced influence in system behavior and stability.

The developed algorithms are validated by simulation results using *Simulink*, in a model developed specifically for this work.

Resumo

Em primeiro lugar, esta dissertação propõe uma solução para o Sistema de Controlo de Atitude (ACS) de satélites até 150 Kg, que utilizem rodas de inércia e magnetorquers como meio de atuação. Esta solução engloba o desenvolvimento de algoritmos de controlo autónomos, capazes de estabilizar o satélite após injeção em órbita, manter uma atitude de referência de acordo com a fase de missão, dessaturar as rodas de inércia e mantê-las num regime de velocidade ideal. O caso de estudo deste trabalho é o satélite português *Infante* que se encontra em fase de desenvolvimento.

Em segundo lugar, esta dissertação propõe uma arquitetura de controlo inovadora para pequenos satélites, denominada arquitetura descentralizada. Nesta arquitetura, cada roda de inércia tem o seu próprio ACS local e independente, que é alheio à existência das restantes. Esta total independência permite uma maior modularidade e flexibilidade de desenvolvimento, uma vez que as rodas de inércia podem ser modificadas, removidas ou acrescentadas, sem ter de modificar as restantes. A independência total dos sistemas tem como consequência diferenças entre a ação de controlo calculada pelo controlador, e aquela que é efetivamente realizada pelo conjunto das rodas inércia. Esta diferença é formalmente descrita, mostrando-se que para o casos mais típicos, esta tem uma influência reduzida no comportamento e estabilidade do sistema.

Os algoritmos desenvolvidos são validados através de simulações em *Simulink*, num modelo desenvolvido especificamente para esta dissertação.

Agradecimentos

Gostaria de agradecer muito ao meu orientador, o professor Sérgio Cunha, por inúmeras razões. Em primeiro lugar, pela confiança que depositou em mim ao atribuir-me este tema tão complexo, mesmo não sendo seu aluno. Esta tese aconteceu simplesmente porque lhe disse que gostava do espaço e de sistemas de controlo, e que gostaria de trabalhar com ele. Em segundo lugar, agradeço-lhe por ter abdicado do seu tempo livre, até aos feriados, para me ensinar sobre os mais variados temas, tendo por vezes que os explicar mais que uma vez. Também lhe agradeço por me ter colocado em contacto com os restantes membros da *Tekever*, ter ouvido as minhas opinões e soluções, e por ter tratado o meu trabalho como o de um engenheiro experiente. Por último, é importante mencionar que apesar de ter sido eu a fazer toda a escrita e formalização da arquitetura descentralizada, a ideia partiu do Sérgio, devendo ele receber o merecido crédito.

Gostaria de deixar um abraço especial ao Bruno Conceição da *Tekever*, sem o qual este trabalho não teria andado para a frente como andou em todos os aspetos. A sua colaboração foi essencial para tirar qualquer dúvida, confirmar os resultados, emendar alguns erros e encontrar o melhor caminho mais depressa. Tenho que lhe agradecer pelas vezes que ele interrompeu o seu trabalho para me ajudar.

Agradeço também ao resto da equipa da *Tekever*, onde fui muito bem recebido, em especial ao João e ao Tiago, que também contribuiram para este trabalho.

Gostaria de dedicar este trabalho à minha Mãe, que o tornou possível ao fazer tudo para que eu tivesse o mínimo de preocupações com os restantes aspetos da minha vida.

Um obrigado muito grande ao meu Pai por ter lido a tese toda, ter feito contribuições para que ela fosse escrita da forma mais clara, e pelos almoços de terça-feira.

Tenho muito que agradecer à Lisa, que ao final de cada dia me aturou, enquanto eu lhe explicava os meus problemas com os filtros de Kalman e o espaço nulo das rodas de inércia.

Um último abraço para a Maria, o Vítor, a Célia, a Teresa, o Xico, para toda a minha família, para a malta dos Feios Porcos e Maus, para o Matos, o Miguel, o Estêvão, a Clara, o Ken, o pessoal de Automação, aos Vips, os Solteiros contra Casados, os Adesivos, todo o pessoal da FEUP e da Valentim e todos os outros que me acompanharam nestes últimos 5 anos.

Contents

1 Introduction **1**
1.1 Motivation . . . 1
1.2 The *Infante* Project . . . 1
1.3 Thesis Objectives . . . 1
1.4 Structure of the thesis . . . 2

2 State of the Art **3**
2.1 State of the Satellite Industry . . . 3
2.2 State of the art in Attitude Control . . . 4

3 Mission Requirements **7**

4 Spacecraft Attitude Modeling **11**
4.1 Attitude Parameterizations . . . 11
4.2 Attitude Kinematics . . . 13
4.3 Attitude Dynamics . . . 13
4.4 Reference Frames . . . 15

5 Spacecraft Orbital Dynamics **17**
5.1 Keplerian Motion and Orbital Propagation . . . 17
5.2 Sun-syncrhonous Orbit . . . 20
5.3 *Epoch* Calculations . . . 20
5.4 Sun and Earth Position . . . 21
5.5 Latitude and Longitude . . . 22
5.6 Eclipse . . . 24

6 Environment Modeling and Disturbances **25**
6.1 Gravity Gradient Torque . . . 25
6.2 Aerodynamic Torque and Drag . . . 26
6.3 Atmosphere Model . . . 31
6.4 Optimal Attitude and Stability . . . 31
6.5 Earth's Magnetic Field . . . 35
6.6 Earth's Albedo . . . 35

7 Sensors and Actuators **37**
7.1 Gyroscopes . . . 37
7.2 Coarse Sun Sensor . . . 41
7.3 Magnetometer . . . 42
7.4 Star Tracker . . . 43
7.5 Magnetorquers . . . 44
7.6 Reaction Wheels . . . 45

8 Attitude Estimation and Filtering Algorithms **49**

9 Decentralized Architecture **55**

10 Attitude Control Laws **63**
10.1 Regulation . . . 63
10.2 Influence of torque distortion in control . . . 71
10.3 Detumbling . . . 74
10.4 Desaturation of the Reaction Wheels . . . 77
10.5 Momentum Management of the Reaction Wheels . . . 79

11 ADCS Modes and Programming **82**

12 Simulation Results **92**

13 Conclusions **96**
13.1 Contributions . . . 96
13.2 Future Works and Recommendations . . . 97

Glossary

ACS Attitude Control System

ADS Attitude Determination System

ADCS Attitude Determination and Control System

ARW Angular Random Walk

AW Angular White Noise

BI Bias Instability

CT Correlation Time

DOY Day of Year

ECEF Earth-Centered/Earth-Fixed

GSIM Gas-Surface Interaction Model

GM Gauss-Markov

GCI Geocentric Inertial

GPS Global Positioning System

GMST Greenwich Mean Sideral Time

JD Julian Date

LVLH Local-Vertical/Local-Horizontal

LEO Low Earth Orbit

MEKF Multiplicative Extended Kalman Filter

NRLMSISE Naval Research Laboratory Mass Spectrometer and Incoherent Scatter Radar Exosphere

NGA National Geospatial-Intelligence Agency

NED North-East-Up

OBC On-Board Computer

P-POD Poly-Picosatellite Orbital Deployer

RRW Rate Random Walk

RTP Ray Tracing Panel

RW Reaction Wheel

SAR Synthetic Aperture Radar

UT Universal Time

WGS84 World Geodetic System 84

WMM2015 World Magnetic Model 2015

List of Figures

1 Sun Synchronous Orbit . . . 7
2 Dimensions of the *Infante* Satellite . . . 8
3 Deployment of the Solar Panels . . . 8
4 SAR Scanning Procedure . . . 9
5 Reaction Wheel Schematics . . . 14
6 Block Diagram of Spacecraft Attitude Model . . . 14
7 Body Frame of *Infante* . . . 15
8 GCI, ECEF and LVLH Frames . . . 16
9 Perifocal Frame . . . 18
10 Ellipse Dimensions . . . 18
11 Ecliptic Plane . . . 21
12 Simulation results for a few orbits of *Infante* plotted on a world map . . . 23
13 Eclipsed Region . . . 24
14 Satellite under Gravity Gradient Torque . . . 25
15 Reflection of a particle . . . 27
16 Schamberg's scattering model . . . 28
17 Panel Model of Infante . . . 29
18 Tangent Direction of Air Speed . . . 30
19 Ray Tracing of the Wings . . . 30
20 Different Attitude Pitch and Disturbance Torques . . . 32
21 Disturbance Torques with Pitch Angle, $\Delta x = 11$mm, h = 500 km . . . 33
22 Disturbance Torques with Pitch Angle, $\Delta x = 100$mm, h = 500 km . . . 33
23 Disturbance Torques with Pitch Angle, $\Delta x = 11$mm, h = 400 km . . . 33
24 Disturbance Torques with Pitch Angle, $\Delta x = 100$mm, h = 400 km . . . 34
25 Albedo Model . . . 35
26 Block Diagram of the Gyroscope model . . . 38
27 Allan Standard Deviation of an Hypothetical Gyro Measurement [36] . . . 39
28 Root Allan Variance of *BMG*250 from Simulink Data . . . 40
29 Root Allan Variance taken from Datasheet [38] . . . 41
30 Root Allan Variance generated from Simulink Data . . . 41
31 Block Diagram of the Sun Sensor . . . 42
32 Block Diagram of a Magnetometer . . . 42
33 Angle of Exclusion in a Star Tracker Camera . . . 43
34 Simulink Model of the Star Tracker . . . 44
35 Block Diagram of the Reaction Wheels . . . 45
36 Torque Trajectory of Modeled Reaction Wheel . . . 47
37 Speed Trajectory of Modeled Reaction Wheel . . . 47
38 Reaction Wheel Pyramid Configuration . . . 48
39 Cross product illustration . . . 51
40 Kalman Filter Quaternion . . . 54
41 Kalman Filter Error in Euler Angles . . . 54
42 Sensors Used in the Kalman Filter . . . 54
43 Example of a common satellite architecture . . . 55
44 Another Example of a Common Satellite Architecture . . . 56
45 Proposed Decentralized architecture . . . 57
46 Four Reaction Wheel Assembly in a Pyramid . . . 58
47 Dot and Pseudoinverse Distributions for a pyramid assembly with four wheels . . . 59
48 Six Reaction Wheel Assembly in a Pyramid . . . 60
49 Dot and pseudoinverse distributions for a pyramid assembly with six wheels . . . 61
50 Controller Step Response for 30^o in roll, $\omega_n = 0.1$ rad/s and $\zeta = 1$ . . . 65
51 Control Error in Euler Angles for 30^o in roll, $\omega_n = 0.1$ rad/s and $\zeta = 1$ . . . 66
52 Controller Step Response for 60^o in all axis, $\omega_n = 0.1$ rad/s and $\zeta = 1$ . . . 66
53 Control Error in Euler Angles for 60^o in all axis, $\omega_n = 0.1$ rad/s and $\zeta = 1$ . . . 67
54 Controller Step Response for 60^o in all axis, $\omega_n = 0.1$ rad/s and $\zeta = 1.5$ . . . 67
55 Control Error in Euler Angles for 60^o in all axis, $\omega_n = 0.1$ rad/s and $\zeta = 1.5$ . . . 68
56 Attitude Error when Aligned with local frame, $\omega_n = 0.1$ rad/s and $\zeta = 1.5$ . . . 68
57 Attitude Error of the Spacecraft when rolled 30^o, $\omega_n = 0.1$ rad/s and $\zeta = 1.5$ . . . 69
58 Estimation Error of the Spacecraft in cruise mode $\omega_n = 0.01$ rad/s and $\zeta = 3$ . . . 70
59 Control Error of the Spacecraft in cruise mode $\omega_n = 0.01$ rad/s and $\zeta = 3$ . . . 70

60 Total Attitude Error of the Spacecraft in cruise mode $\omega_n = 0.01$ rad/s and $\zeta = 3$ 70
61 Reaction Wheel Control Effort with High and Low Gains 71
62 Performance comparison between assemblies and distributions 73
63 Absolute Angular Velocity $||\omega||$ during Detumbling for various controllers 76
64 Energy used during Detumbling for various controllers 76
65 Total momentum of the Reaction Wheels during desaturation 78
66 Energy used during desaturation for various control gains 78
67 Total momentum of the Reaction Wheels during desaturation 80
68 Total momentum of the Reaction Wheels during desaturation 80
69 State Machine for Main Attitude Modes (Parent) 84
70 State Machine for Main Attitude Modes (Child) 84
71 Estimation error of *Infante* during regulation control 85
72 Variable "ε" and its correlation with the estimation error 85
73 State Machine for Mission Handling 86
74 State Machine for the Star Tracker 87
75 State Machine for the Precision Gyroscope 87
76 State Machine For Main Attitude Modes 88
77 State Machine Handling Magnetometers and Magnetorquers 89
78 State Machine for Inner and Outer Magnetometers 90
79 State Machine for Desaturation of the Reaction Wheels 90
80 Attitude of the Spacecraft in the Local Frame, states of internal variables 92
81 Control and Estimation Errors represented in 321 Euler Angles 93
82 Time Diagram of Control laws and uncertainty measure "ε" 94
83 Time Diagram of Attitude Sensors 94
84 Time Diagram of Gyroscopes and the Occurrence of Eclipse 94
85 Reaction Wheel Momentum on Body and Wheel Space 95
86 World Map with SAR targets and Orbits 95

List of Tables

1 Classification of Satellites with examples, taken from [6] . . . 3
2 Dimensions of *Infante* in millimeters . . . 8
3 Noise characteristics of the *Bosch BMG*250 gyro . . . 39
4 Noise characteristics of the *STIMM*210 gyro . . . 40
5 Information regarding the used magnetometers [39] [40] . . . 42
6 Information regarding the KU Leuven Star Tracker [41] . . . 43
7 Reaction Wheel Specifications [43] . . . 46
8 Chosen gains and simulation results for five different detumbling control laws . . . 76
9 Output Variables of the ADCS . . . 82
10 Internal Variables of the ADCS . . . 82
11 Input Variables of the ADCS . . . 83

1 Introduction

1.1 Motivation

Space exploration is one of the youngest branches in science and engineering, a field that has only just started to become accessible to universities and small companies. Because this is such a recent and competitive area, the satellite industry suffers from a notable lack of standardization and documentation about its methods. Decentralized architecture fits here as an idea for reaction wheel manufacturers to make ACS development easier for small companies and universities. Reaction wheels could be sold individually with an ACS and connected to a standardized "attitude bus", automatically producing a stable, working system, without needing any expert knowledge in attitude systems.

1.2 The *Infante* Project

The *Infante* Project is an initiative led by the Portuguese company *Tekever*. Its goal is to launch the *Infante* Satellite in Low Earth Orbit (LEO) in 2020, the first of a constellation of small satellites focused on Earth observation. This observation, aimed at the Atlantic Ocean, can be separated in the following objectives:

- Maritime Surveillance - *Infante* will identify ships in the Atlantic and report alerts to users.
- Environmental Monitoring - *Infante* will monitor slow evolving environmental phenomena (such as oil spills or algal blooms) and report data to users.
- Extreme Event Detection and Monitoring - *Infante* will detect and monitor fast evolving phenomena (e.g. fires, floods, extreme meteorological event or other) and report data to users.

To accomplish these objetives the *Infante* is equiped with the following features:

- A Multispectral Camera and a Synthetic Aperture Radar (SAR) for Earth observation
- A sustainable power system consisting of batteries and solar panels, in order to cope with a power demanding payload.
- Down/up link with the ground to send commands and receive high-resolution payload data.
- Propulsion system to maintain the orbit throughout the mission.
- An active ADCS system, to provide the necessary attitude and stability for the Earth observation operation - which is the main topic of this dissertation.

1.3 Thesis Objectives

The first objective of this dissertation is to design a set of attitude control algorithms which are able to:

- Detumble the satellite upon orbit injection.
- Maintain the reference attitude, which is subject to change according to the mission phase.
- Desaturate the reaction wheels using the magnetorquers.
- Avoid close to null velocity points of the reaction wheels.

The second objective is to validate the decentralized architecture and the underlying dot product distribution as reliable methods for designing an ADCS for small satellites.

Both objectives imply the development of a Simulink model of *Infante* in orbit, in order to test, develop and validate the control algorithms.

1.4 Structure of the thesis

This document is segmented into several chapters, each regarding a certain aspect of the overall model of the *Infante* satellite in the space environment:

Section 2 is the State of the Art in Attitude Control, providing historic and technological context to this thesis.

Section 3 explains the mission requirements for this particular satellite in detail, necessary to understand the details in the development of *Infante* and its ADCS.

Section 4 includes a brief review on attitude parameterization methods and the equations for attitude kinematics and dynamics, which are necessary to elaborate a dynamic model of the attitude. The reference frames used throughout this document are also defined here.

Section 5 reviews Kepler's laws for orbital motion and presents the equations used to develop a simple orbit propagator, describing the spacecraft's motion around the Earth, as well as other necessary algorithms for time keeping.

In Section 6, environment models are introduced and the attitude disturbances of gravity gradient torque and aerodynamic torque are analyzed in detail.

Section 7 presents accurate models of the sensors, magnetorquers and reaction wheels.

Section 8 summarizes the development of a Multiplicative Extended Kalman Filter, which is used to get a better estimate of the current attitude.

Section 9 introduces the topic of decentralized architecture, where the torque distortion problem is analyzed for the common reaction wheel assemblies.

Section 10 presents attitude control laws for regulation, detumbling, desaturation of the reaction wheels and management of the reaction wheel's momentum. For reaction wheel regulation, the optimum control optimal gains are derived and the effect of the torque distortion introduced by the decentralized architecture is studied. A comparison with the traditional architecture is also done here.

Section 11 presents the science modes and ADCS programming designed for *Infante* so it can become fully autonomous.

Lastly, Section 12 provides simulation results for a mission lasting several orbits.

2 State of the Art

2.1 State of the Satellite Industry

The satellite industry is an engineering branch that is very recent and growing quite fast. According to the United Nations Office for Outer Space Affairs's annual report of 2019 [1], there are 2100 functional satellites currently in orbit, with 300 being launched in 2018. This is 27% compared to a total of around 8000 satellites that have been deployed since the first satellite, Sputnik, in 1957 [1]. A survey by NATO concludes that this number is expected to rise to 7,000 within several years [2]. The manufacturing sector of the satellite industry has seen a growth in revenues of 26%, while the launch sector has seen a growth of 34% in 2018, as seen in this year's report of the Space Industry Association (SIA) [3]. SIA argues there is currently a new "golden age" of space exploration, due to recent technological innovations, and ESA, in its 2016 report [4], refers to "fresh opportunities for space in the form of new technologies of which bio-inspired systems, self-healing, additive manufacturing, nano-tubes, in-space production and assembly, artificial intelligence and quantum technology". ESA also refers to a new era of "Space 4.0", analogous to the "Industry 4.0", the unfolding of the fourth industry revolution [5].

This recent growth and optimistic forecast are due to the fall of the costs and risks associated with this activity. Even tough space exploration has always been profitable, the costs associated with it were extremely prohibitive in its early stages, and could only have been supported by national governments such as the United States and the Soviet Union. There has been a huge increase in the private sector and in globalization. Private firms like Space X and Blue Origin are no longer operating as contractors, but becoming key protagonists in space, while changing the way companies perceive this industry. Space X has expressed its concern with maximum standardization and re-usability, illustrated by their rockets whose boosters can be re-used [2].

Table 1: Classification of Satellites with examples, taken from [6]

Name	**Mass**	**Cost**	**Classification**
Immar-Sat 4	5945 Kg	\$0.1 -2 B	Large
GIOVE-A	660 Kg	\$50-100 M	Medium
UK-DMC	166 Kg	\$10-50 M	Mini
PicoSat	67 Kg	\$2-10 M	Micro
SNAP	6.5 Kg	\$0.2 - 2 M	Nano
PalmSat	1 Kg	\$20-200 K	Pico
PCB Sat	250 g	\$3 -20 K	Femto

Despite this, manufacturing of satellites remains relatively centralized in a handful of companies that possess the skilled workforce, the scale, engineering capacity and financial resources needed to produce reliable complex systems. Satellites are still very expensive, but costs are declining with the advent of small satellites. The Airbus OneWeb satellite programme aims to launch 900 small satellites - that weigh between 10 and 20 kilograms - that will collectively provide affordable internet access to the entire world. The first of these micro satellites will be deployed in 2019. This is a highly ambitious trans-Atlantic project that has also demanded very rapid satellite production lines relying on robotic assembly lines [7] [2]. SpaceX has also launched a similar initiative called "Starlink", and 60 of its satellites are already in orbit [8].

2.2 State of the art in Attitude Control

The Attitude Determination and Control System (ADCS) of a satellite is always designed taking the mission primary objective, the desired accuracy, stabilization requirements, size constraints, the disturbance torques, system budget and costs into account [9]. There are a wide variety of sensors and actuators available, each with specific performance ranges. A company can develop their own attitude sensors and actuators, buy them individually from manufacturers or buy complete ADCS solutions [10].

There are three main types of attitude control: passive stabilization, spin stabilization and active stabilization. Passive stabilization relies on gravity gradient or magnetic torques, supplied by either a "boom" or permanent magnet rods. They are used on satellites with very coarse attitude requirements, since this only provides stabilization. An example of a mission using this is the QuakeSat, which measured low frequency oscillations in the magnetic field to predict earthquakes [11]. Spin stabilization relies on conservation of angular momentum, stabilizing a satellite by having it deployed while spinning at a considerable rate. The most infamous case is the USA's first satellite Explorer 1 which entered in flatspin shortly after being deployed in orbit [9]. Active stabilization involves the use of attitude sensors and actuators to provide control torques.

The most common attitude sensors are: sun sensors, horizon sensors, Earth sensors, magnetometers, star trackers and gyroscopes. Sun sensors acquire attitude by estimating the Sun's position and comparing it to an internal model of the Sun's position. Horizon sensors work using the same principle, detecting Earth's horizon using an infrared camera, as well as magnetometers, that measure Earth's magnetic field. A star tracker is essentially a camera that tracks stars in its field of view, is able to hold certain attitude measurements and compute the attitude by consulting a star catalog. It is the only sensor that can provide full knowledge of the attitude, without needing any extra instrumentation [9] [10] [12].

In attitude control, the exisiting technologies include: reaction wheels, magnetorquers and thrusters. Reaction wheels are inertia wheels attached to an electric motor, that create a reaction torque by accelerating them. These can be sold individually or in assemblies, depending on the mission's objectives, since three wheels are neeed to control the spacecraft in all three axis. Magnetorquers produce torque by interacting with Earth's magnetic field and they are a popular option for LEO missions in small satellites. Thrusters work by ejecting fluid, producing torque when their direction is misaligned with the center of mass. They have a limited mission lifetime and are usually more adequate for bigger satellites, because of their high power density and difficult miniaturization [10] [12].

Modern attitude estimation usually employs an on-board Kalman Filter to obtain a higher accuracy. A Kalman filter is an estimation tool that uses an observer model of the system to evaluate if the measures being taken are coherent with the system's states. Often, Kalman Filters are able to take multiple, redundant measures to tell which is the most probable system state, based on a priori information regarding sensor noise. The early uses of this tool are not openly documented since they were directed towards military uses, but its most famous application was its use on the Apollo missions' trajectory estimation. Its first attitude applications were attempted in the 60s, without initial success. These first Kalman filters assumed uncoupled rotation axes and used Euler Angles representations, and were later used by NASA in some missions with success. Since spacecraft attitude dynamics are non-linear, an Extended Kalman Filter is more commonly used, with different parameterizations such as the direction

cosine matrix, the quaternion and the Gibbs vector, to provide accurate results, which had a slow adoption due to the threat of non-convergence [13] [9]. State of the art filters in attitude estimation include the Multiplicative Extended Kalman Filter (MEKF), reviewed by this document, and two variants: the Isotropic Kalman Filter (IKF) and the Unscented Kalman Filter (UKF). The IKF assumes the same covariance in all directions, reducing computational effort, while the UKF propagates the state and the covariance using the Unscented Transformation, reducing linearization errors [14] [15] [16].

Closed-loop control of a spacecraft's attitude can also be achieved through multiple approaches, depending on the system's complexity, and desired performance, efficiency and robustness. The classic regulation approach is achieved by combining either a PD or a PID controller with a representation of the rotation error in quaternion form or in Euler Angles. This is the most common controller when reaction wheels are being used and large maneuvers are not solicited, providing simple implementation, analysis and system stability. Sliding mode control is commonly used when tracking a certain attitude trajectory, which is the case of the WMAP spacecraft presented in [9], providing robust control but usually at the cost of high control action. An alternative robust controller that is gaining popularity is H_∞, that is ideal for complex, flexible satellites structures with stringent attitude requirements, since it has proofed to be efficient at rejecting unmodeled disturbances [17] [18] [19]. An optimal control strategy known as Linear Quadratic Regulator (LQR) is also gaining popularity in satellite attitude control, as seen for example in [16] and in [20]. Other modern control strategies like *fuzzy logic* have shown success in satellites using thrusters and magnetorquers for attitude control [21] [22]. Finally, adaptive control has been applied to systems with modeling uncertainty regarding its inertia [23] and actuators [24].

3 Mission Requirements

In this chapter the mission requirements and objectives are presented in detail, in order to provide context to the algorithms developed later.

The *Infante* satellite is set to be launched in 2020 and injected into a midnight-noon sun-synchronous Low Earth Orbit of altitude 500 km, as an auxiliary payload of a larger mission. In this orbit regime the satellite passes near the North and South poles, while the orbit plane is forced to rotate with the Earth around the Sun. This phenomenon occurs for certain orbital inclinations due to the Earth's non-spherical gravity; an illustration is provided in Figure 1. This orbit provides good lighting conditions for Earth observation that change only seasonally, local observation time is always around midnight or noon.

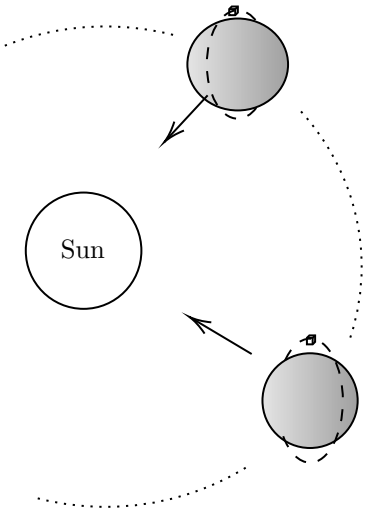


Figure 1: Sun Synchronous Orbit

Infante shall be deployed by a *P-Pod*, a rectangular aluminum container that pushes the satellite into space using the force of a loaded spring. *P-Pod*s usually follow standard dimensions of *CubeSat*s but the current development of *Infante* led to a non-standardized size and a *P-Pod* will be manufactured specially for this mission. In order to fit inside the rectangular *P-Pod*, *Infante* has foldable solar panels that only open after detumbling. Figure 3 illustrates the *Infante* foldable solar panels. Tumbling is what is called to the high-rate rotation that is expected upon orbit injection. The payload is to be located underneath the spacecraft with the solar panels facing up, casting a shadow over it when pointing at the Sun, keeping it cool.

Infante is still under development so final dimensions and moment of inertia matrices are unavailable. For the purpose of this study a temporary model is considered. *Infante* has a shape close to a rectangular cuboid of dimensions $890 \times 292 \times 250\ mm^3$ with thin foldable solar panels that span across an area of $892 \times (4 \times 219)\ mm^2$ each. The estimated weight is 71.58 Kg and its inertia is larger if the solar panels are deployed. Two inertia matrices are considered at the center of mass, in case the solar panels are either closed or open:

$$J_{closed} = \begin{bmatrix} 0.9154 & 0 & 0 \\ 0 & 5.0469 & 0 \\ 0 & 0 & 5.2522 \end{bmatrix} \quad J_{open} = \begin{bmatrix} 5.1658 & 0 & 0 \\ 0 & 6.1260 & 0 \\ 0 & 0 & 10.3434 \end{bmatrix} [kgm^2] \quad (1)$$

These dimensions can be seen on Figure 2 and in Table 3.

The satellite's ADCS (Attitude Determination and Control System) is made up of attitude sensors (a coarse Sun Sensor, two magnetometers, a Star Tracker, a coarse and a precise gyroscope) and attitude actuators (an assembly of four reaction wheels and three magnetorquers), which are explained in detail later. The ADCS is responsible for determining and holding the spacecraft's attitude within certain requirements, using the least amount of power as possible.

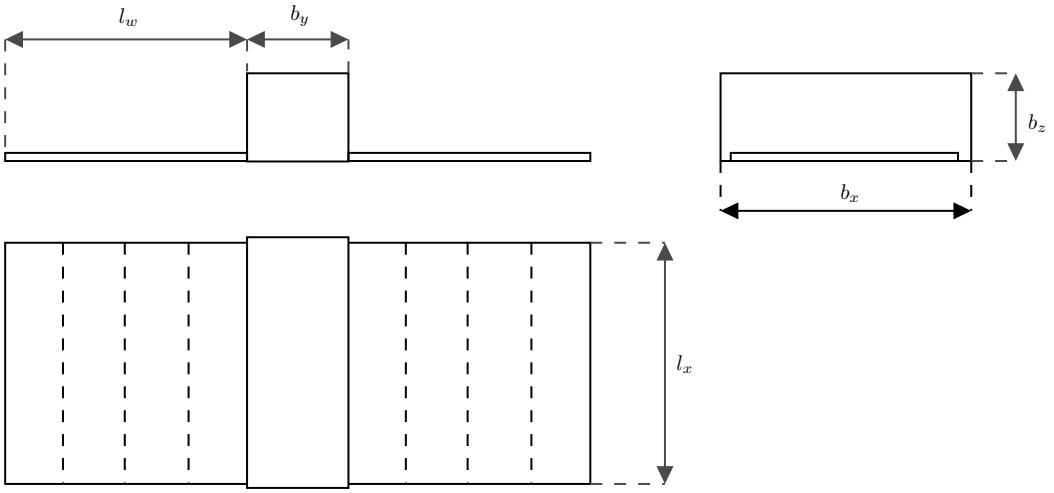


Figure 2: Dimensions of the *Infante* Satellite

Table 2: Dimensions of *Infante* in millimeters

b_x	b_y	b_z	l_x	l_w
890	292	250	892	4×292

The Synthetic Aperture Radar (SAR) is an imaging radar mounted on a moving platform where electromagnetic waves are transmitted sequentially, the echoes are collected, stored and sent to ground for signal processing. One of the great appeals of this radar is that it's capable of producing high resolution images from space in spite of daytime and weather conditions. The SAR to be used in *Infante* requires that its beam hits the Earth at different angles, usually in the region of 30^o. Since the radar is fixed to the spacecraft's body, *Infante* needs to perform a roll before a SAR scan, as depicted in Figure 4.

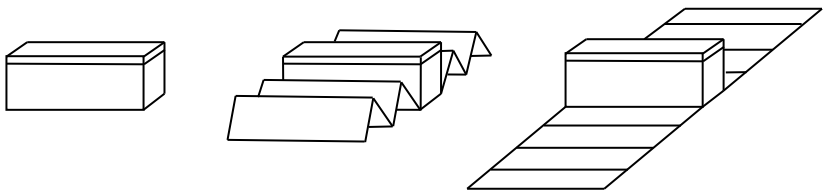


Figure 3: Deployment of the Solar Panels

During a SAR mission the attitude requirements are strict. The attitude knowledge is bounded by a $3\sigma = 0.25^o$ in all axes and the overall attitude error is bounded by $3\sigma = 0.75^o$ and axes. When the satellite is standing by, the overall attitude error can amount to a total of $3\sigma = 5^o$. During the actual SAR scan it is also desirable to reduce the attitude control dynamics of the system since a control dynamic that is too active may blur the obtained images.

A reaction wheel is a high inertia flywheel attached to a brushless electric motor, that provides torque by accelerating and braking. Several reaction wheels are assembled so they can provide three axis torque control, which is used for fine pointing of the satellite. Over time, as wheels

accelerate, their speed may increase until a certain limit is reached where they can no longer provide torque; this is called saturation. The opposite case can also be problematic since reaction wheels have poor torque control at very low speeds. In *Infante*, reaction wheels will operate in a four-wheel pyramid assembly to provide redundancy in case of wheel failure and enabling control over the distribution of their own momentum, ensuring that the wheels are operating in their optimal region.

Detumbling and desaturation of the reaction wheels is done by magnetorquers. A magnetorquer is a controllable electromagnet that creates a magnetic dipole, which interferes with Earth's magnetic field, providing useful torque. Magnetorquers exhibit residual magnetic dipoles or hysteresis, which can disturb the spacecraft. To avoid this, a decaying dither signal is applied to minimize this effect. In addition, when the magnetorquers are active the magnetometers must be turned off, otherwise they will measure the magnetic field generated by the satellite itself as interference.

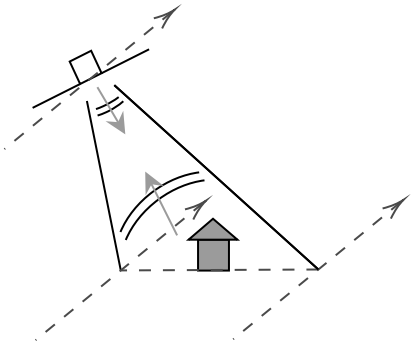


Figure 4: SAR Scanning Procedure

There are two magnetometers in *Infante*: a low precision one that is inside the satellite's structure and a more precise one that is used outside. The inner magnetometer measures the magnetic field upon detumbling, providing information on how to trigger the magnetorquers only. After detumbling, the more precise magnetometer is deployed outside the satellite's structure and is used to get better readings of the magnetic field, this time, for attitude acquisition.

Infante is equiped with two gyroscopes: a low power coarse gyro and another of great precision. The coarse gyro uses low power and should be turned on for the entire mission while the precision gyro uses more power and should only be turned on if necessary.

The coarse Sun Sensor is made up of six photo-resistors, one for each face of *Infante*. By measuring and comparing the voltage across each photo-resistor it is possible to estimate the Sun direction, compare it with a model and estimate the attitude.

4 Spacecraft Attitude Modeling

In this section the definitions and notations used throughout this document are presented, as well as a brief review of the kinematic and dynamic equations used to model the attitude of the spacecraft. This material is based on the works of [25] and [9]. Vectors $\mathbf{x}$ are written in bold and quaternions $\bar{q}$ with an overbar.

4.1 Attitude Parameterizations

Attitude is defined as the orientation of the satellite body frame regarding a certain reference frame. This orientation is usually expressed through a 3×3 rotation matrix. The rotation matrix A that relates the spacecraft's body frame B with an arbitrary reference frame I is denoted A_{BI}, so that a vector in frame I can be written in frame B by the equation:

$$\mathbf{x}_B = A_{BI}\mathbf{x}_I \tag{2}$$

There are, however, other attitude representations that are more compact and intuitive, such as Euler angles, unit quaternions and the Gibbs Vector.

The Euler angles ϕ, θ, ψ, represent a series of three consecutive rotations upon a given set of axis. The classical $3-2-1$ Euler sequence is a very common description of spacecraft's attitude, where the rotations are performed around the $x-$, $y-$ and $z-$axis of the satellite body. The angles are usually denoted by roll, pitch and yaw angles.

The individual rotation matrices are given by:

$$A_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \tag{3}$$

$$A_\theta = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{4}$$

$$A_\psi = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

The rotation matrix given by a $3-2-1$ Euler angle sequence can be obtained by $A_{321}(\phi,\theta,\psi) = A_\phi A_\theta A_\psi$. On the other hand, a matrix representing a $1-2-3$ Euler sequence would be obtained by inverting the order of the rotations, becoming roll, pitch and yaw. Another common Euler Angle sequence is the $3-1-3$ sequence; here the rotation is first around the yaw axis, then around the roll axis, and lastly around the yaw axis again.

This representation provides an intuitive description of attitude but is prone to be numerically unstable due to the existence of singularities. To avoid this problem the quaternion parameterization is often used.

Euler's eigenaxis theorem states that any matrix transformation can be expressed by a rotation angle ϕ over a fixed eigenaxis $\boldsymbol{e}$. The unit quaternion is a four-parameter representation based upon this theorem that is defined by:

$$\bar{q} = \begin{bmatrix} e_1 \sin(\phi/2) \\ e_2 \sin(\phi/2) \\ e_3 \sin(\phi/2) \\ \cos(\phi/2) \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{q}}_{1:3} \\ \bar{q}_4 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{6}$$

Where $\bar{\boldsymbol{q}}_{1:3}$ is the vector part of the quaternion and $\bar{q}_4$ is the scalar part. This convention where the scalar part comes fourth is used in this document. A convention where the scalar part comes first is also used very often, and is used for example in MATLAB's native quaternion functions. The quaternion product is defined as the following operation:

$$\bar{q} \otimes \bar{\bar{q}} = \begin{bmatrix} \bar{q}_4 \bar{\bar{\boldsymbol{q}}}_{1:3} + \bar{\bar{q}}_4 \bar{\boldsymbol{q}}_{1:3} - \bar{\boldsymbol{q}}_{1:3} \times \bar{\bar{\boldsymbol{q}}}_{1:3} \\ \bar{q}_4 \cdot \bar{\bar{q}}_4 - \bar{\boldsymbol{q}}_{1:3} \cdot \bar{\bar{\boldsymbol{q}}}_{1:3} \end{bmatrix} \tag{7}$$

The quaternion product provides a convenient way to perform vector transformations. Let $\bar{q}_{BI}$ be the quaternion representing a rotation analogous to the rotation matrix in equation (2). It can be shown that the quaternion product provides an equivalent matrix transformation through the expression:

$$\mathbf{x}_B = \bar{q}_{BI} \otimes \mathbf{x}_I \otimes \bar{q}{*}_{BI} \tag{8}$$

Where $\bar{q}{*}_{BI}$ is the conjugate quaternion given by the expression $\bar{q}{*}_{BI} = \begin{bmatrix} -\bar{\boldsymbol{q}}_{1:3} \\ \bar{q}_4 \end{bmatrix}$, which is also its inverse $\bar{q}^{-1}$ if it's a unit quaternion. Another option is to compute the rotation matrix of a quaternion directly:

$$A(\bar{q}) = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1 q_2 + q_3 q_4) & 2(q_1 q_{q3} - q_2 q_4) \\ 2(q_1 q_2 - q_3 q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2 q_3 + q_1 q_4) \\ 2(q_1 q_3 + q_2 q_4) & 2(q_2 q_3 - q_1 q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \tag{9}$$

Therefore, describing attitude in quaternions has computational advantage since singularities are avoided and vector transformations can be performed without the use of trigonometric functions.

The last parameterization to be introduced is the Gibbs Vector Representation:

$$\boldsymbol{g}(\bar{q}) = \frac{\bar{\boldsymbol{q}}_{1:3}}{\bar{q}_4} \tag{10}$$

With the equivalent expression in terms of Euler axis and angle:

$$\boldsymbol{g}(\boldsymbol{e}, \phi) = \boldsymbol{e} \tan(\phi/2) \tag{11}$$

This representation is very useful to represent small rotation errors with the disadvantage of becoming infinite for a 180° rotation.

4.2 Attitude Kinematics

When two frames are in relative motion their Euler axis and angle exhibit a time dependency $\boldsymbol{e}(t)$ and $\phi(t)$. The rotating body frame B is considered at time instants t and $t + \Delta t$, whose attitude is defined in terms of an inertial frame I. The B frame in the first time instant can be related to the next instant by a rotation axis $\boldsymbol{e}'$ and a small rotation angle $\Delta\phi'$. We define the angular rate at which the frames B and I are rotating relative to each other as the following limit:

$$\boldsymbol{\omega}_B^{BI} = \lim_{\Delta t \to 0} \frac{\Delta \phi'}{\Delta t} \boldsymbol{e}' \tag{12}$$

Where the superscript BI means that the considered velocity is the body's, with respect to the inertial frame I, and the subscript B means that the angular rate is represented on the body frame. For convenience, the subscript is omitted for body vectors, since this is usually the case.

The kinematic relation between these rotating frames can also be expressed by the derivative of the quaternion. This is given by a similiar limit:

$$\dot{\bar{q}}_{BI}(t) = \lim_{\Delta t \to 0} \frac{\bar{q}_{BI}(t + \Delta t) - \bar{q}_{BI}(t)}{\Delta t} \tag{13}$$

It can be shown that an equivalent, more convenient form for this equation is:

$$\dot{\bar{q}}_{BI}(t) = \boldsymbol{\omega}^{BI} \otimes \bar{q}_{BI} \tag{14}$$

Which is known as the kinematic equation of the quaternion.

4.3 Attitude Dynamics

Consider a spacecraft equipped with an assembly of n reaction wheels, each labeled by the index l. Each wheel has its spin axis direction represented by the body vector $\boldsymbol{w}_l$ and corresponding wheel angular velocity of magnitude ω_l^w. Since each wheel is axis-symmetric we use this fact to decompose the wheel's moment of inertia into two components: the component along the spin axis $J_l^{\parallel}$ and the component $J_l^{\perp}$ along any other perpendicular axis; as illustrated in Figure 5. This decomposition can be described in terms of the following equation:

$$J_l^w = J_l^{\perp}(I_3 - \boldsymbol{w}_l \boldsymbol{w}_l^T) + J_l^{\parallel} \boldsymbol{w}_l \boldsymbol{w}_l^T \tag{15}$$

Let $\tilde{J}$ represent the spacecraft's moment of inertia at the center of mass without reaction wheels. We can add the transverse component of the wheels for convenience since this component is ineffective for control purposes:

$$J = \tilde{J} + J^{\perp} = \tilde{J} + \sum_{l=1}^{n} J_l^{\perp}(I_3 - \boldsymbol{w}_l \boldsymbol{w}_l^T) \tag{16}$$

Since *Infante* has a moment of inertia that is much larger than the wheel's transverse component, and since at the current project stage, its inertia is not well determined, we can assume that $J \approx \tilde{J}$ for simplicity.

The total angular momentum of the spacecraft can then be calculated as:

$$\boldsymbol{H} = J\boldsymbol{\omega}^{BI} + \boldsymbol{H}^{w}, \tag{17}$$

$$\boldsymbol{H}^{w} = \sum_{l=1}^{n} J_{l}^{\parallel}(\boldsymbol{w}_{l} \cdot \boldsymbol{\omega}^{BI} + \omega_{l}^{w})\boldsymbol{w}_{l} \tag{18}$$

$$\approx \sum_{l=1}^{n} J_{l}^{\parallel}(\omega_{l}^{w}\boldsymbol{w}_{l}) \tag{19}$$

The axial momentum of the wheels due to spacecraft motion can be neglected, since they are usually spinning at rates much higher than the spacecraft. By deriving equation (17) and using the chain rule, the following differential equation is obtained:

$$\dot{\boldsymbol{H}} = J\dot{\boldsymbol{\omega}}^{BI} + \dot{\boldsymbol{H}}^{w} + \boldsymbol{\omega}^{BI} \times (J\boldsymbol{\omega}^{BI} + \boldsymbol{H}^{w}) \tag{20}$$

The derivatives of the angular momentum are just the same as the applied torques $\boldsymbol{L}$ and $\boldsymbol{L}^{w}$, so this equation can be rearranged and rewritten as:

$$\dot{\boldsymbol{\omega}}^{BI} = J^{-1}\left[\boldsymbol{L} - \boldsymbol{L}^{w} - \boldsymbol{\omega}^{BI} \times (J\boldsymbol{\omega}^{BI} + \boldsymbol{H}^{w})\right] \tag{21}$$

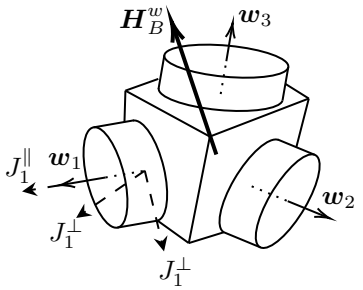


Figure 5: Reaction Wheel Schematics

This is the state space equation used to model the attitude dynamics of the spacecraft, which returns the state $\boldsymbol{\omega}^{BI}(t)$. The kinematic equation of the quaternion provides the remaining state of attitude $\bar{q}(t)$. The final block diagram is presented on Figure 6, showing how to perform a Simulink implementation of the model above.

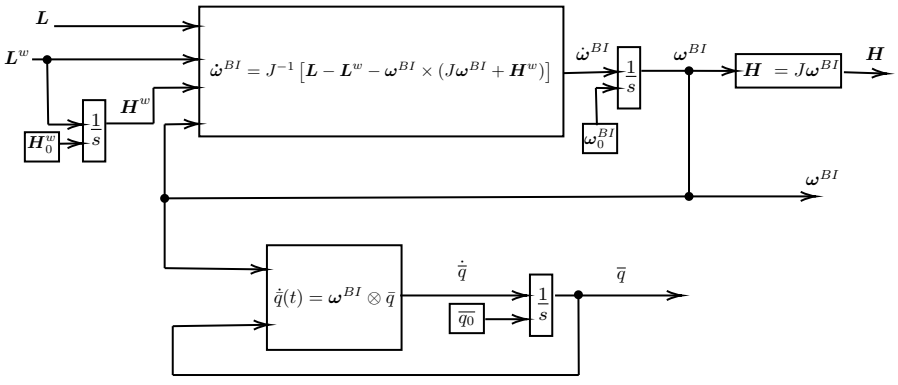


Figure 6: Block Diagram of Spacecraft Attitude Model

4.4 Reference Frames

This subsection addresses the definition of the different reference frames used throughout this document.

4.4.1 Spacecraft Body Frame

The body frame is defined at the center of mass for the purpose of this study for convenience. Generally, spacecraft body frames are located in a rigid *navigation base*, a subsystem including attitude sensors and payload. The $x-$axis points ahead, the $y-$axis to the right side and the $z-$axis points down, defined according to the local frame and space navigation conventions, as depicted in Figure 7.

4.4.2 GCI Frame

The GCI frame (*Geocentric Inertial* Frame) is an approximate inertial frame that has its origin at the Earth's center of mass. Its $z-$axis is aligned with Earth's North Pole and its $x-$axis is aligned with the *vernal equinox*, the intersection of the Earth's equatorial plane with the plane of the Earth's orbit around the Sun on the first day of spring. Since this frame orbits the sun at the rate of only one revolution per year, it's considered a practical and good enough approximation of an inertial frame for the analysis of spacecraft attitude. It's denoted in Figure 8 as $\{\boldsymbol{i}_1, \boldsymbol{i}_2, \boldsymbol{i}_3\}$.

4.4.3 ECEF Frame

The ECEF frame (*Earth-Centered/Earth-Fixed* Frame), denoted by $\{\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2, \boldsymbol{\epsilon}_3\}$ in Figure 8, is similar to the GCI frame since they both share the $z-$axis but with the difference that ECEF rotates with the Earth. The $x-$axis points toward Earth's prime meridian and the angle between GCI and ECEF frames is denoted by θ_{GMST} (*Greenwich Mean Sideral Time*). Computation of this angle is explained later in the *Epoch* calculations chapter.

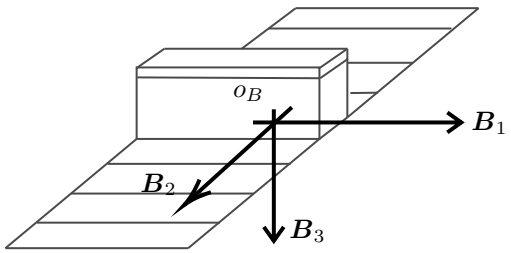


Figure 7: Body Frame of *Infante*

4.4.4 LVLH Frame

The LVLH frame (*Local-Vertical/Local-Horizontal* Frame) is the local frame defined upon the spacecraft's orbit, denoted by $\{\boldsymbol{o}_1, \boldsymbol{o}_2, \boldsymbol{o}_3\}$. The attitude control problem is usually defined in this frame since it coincides with the spacecraft's body frame for Earth pointing. The $z-$axis points toward the center of the Earth, the $y-$ is perpendicular to the orbit plane and the $x-$axis completes the orthogonal reference frame, pointing ahead. For eccentric orbits the speed direction does not coincide with the $x-$axis, which is not the case for *Infante's*.

4.4.5 NED Frame

The NED frame (*North-East-Up Frame*) is a commonly used local frame, based on the Earth's rotation axis. The matrix relating this frame with the *ECEF* Frame can be obtained if the spacecraft's latitude ϕ and longitude λ are known:

$$A_{ECEF-NED} = \begin{bmatrix} -\sin(\phi)\cos(\lambda) & -\sin(\lambda) & -\cos(\phi)\sin(\lambda) \\ -\sin(\phi)\sin(\lambda) & \cos(\lambda) & -\cos(\phi)\sin(\lambda) \\ \cos(\phi) & 0 & -\sin(\phi) \end{bmatrix} \tag{22}$$

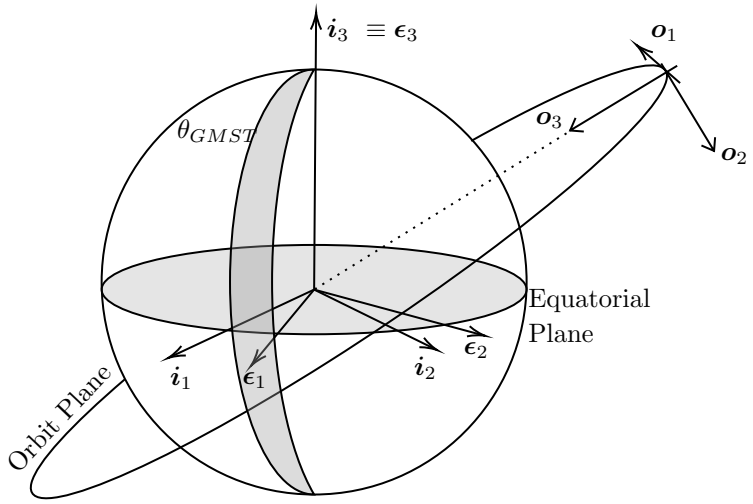


Figure 8: GCI, ECEF and LVLH Frames

5 Spacecraft Orbital Dynamics

In order to provide a realistic environment for the attitude model of *Infante*, its orbit needs to be described since the spacecraft's position and speed are crucial for simulating a real mission. The model needs to account events like solar eclipses, geographic passages and proper aerodynamic and magnetic effects, which are a consequence of the spacecraft's motion around the Earth. In this section, the mathematical tools are provided so a simple yet effective orbit model can be built.

5.1 Keplerian Motion and Orbital Propagation

Let $\boldsymbol{r}$ be the spacecraft's position in the GCI frame and $\dot{\boldsymbol{r}}$ be its time derivative, the spacecraft's speed. The problem of interest consists in finding a realistic algorithm that returns $\boldsymbol{r}(t)$ and $\dot{\boldsymbol{r}}(t)$ for a certain set of initial conditions. The simplest way to produce this algorithm is to consider the spacecraft's orbit as an ellipse with the Earth at its focus, and then to propagate the spacecraft's motion through some angle $E(t)$. This approximation is known as *Keplerian Motion*, since Kepler was the first to demonstrate that the planetary orbits are ellipses, by using Newton's laws of gravity and motion. This is not completely true however, disturbances like aerodynamic drag and non spherical gravity are not accounted for in this model. Satellites in LEO have decaying orbits and exhibit relatively small deviations from the elliptic orbit. For the purpose of this thesis Kepler's orbit model is sufficient, since these perturbations do not greatly affect attitude analysis.

As a consequence of the elliptic shape of a planetary orbit, Kepler proposed the classical orbital elements as a mean to describe an orbiting body's trajectory. The dimensional elements are:

- a: semimajor axis (size of orbit)
- e: eccentricity (shape of orbit)
- M_0: initial mean anomaly (related to the initial position in orbit)

These elements describe the elliptical motion of the spacecraft in the *Perifocal* frame, denoted by $\{\boldsymbol{i}_p, \boldsymbol{i}_e, \boldsymbol{i}_h\}$ in Figure 9. The $x-$axis $\boldsymbol{i}_p$ points along the *periapsis*, which is the point closest to the focus, the $z-$axis $\boldsymbol{i}_h$ points outside the orbit plane with the $y-$axis $\boldsymbol{i}_e$ completing the direct frame. Once the orbital ellipse is defined, it is necessary to define its orientation relative to the Earth, which is given by the orientation elements, also illustrated in Figure 9:

- i: inclination (angle between orbit plane and the equatorial plane)
- Ω: right ascension of the ascending node (angle between $\boldsymbol{i}_1$ and line of nodes $\boldsymbol{i}_n$, the intersection of the orbit plane with the equatorial plane)
- ω: argument of the *periapsis*

As stated before, the propagation problem consists of finding an algorithm which returns the spacecraft's position and speed at any point in time, given some arbitrary initial conditions. The most intuitive method would be calculating $\upsilon(t)$, the true anomaly of the ellipse, since the position in the perifocal frame can be calculated by:

$$x_p = r(\upsilon)\cos(\upsilon(t)) \quad y_p = r(\upsilon)\sin(\upsilon(t)) \tag{23}$$

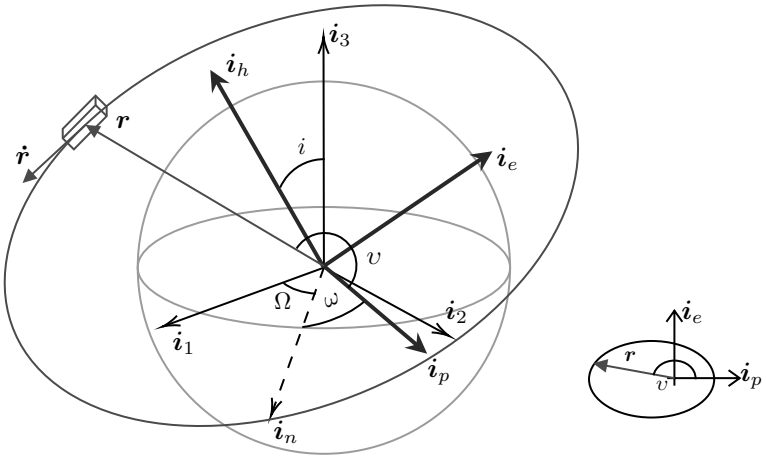


Figure 9: Perifocal Frame

Kepler's second law states that "the line joining the satellite to the Earth sweeps out equal areas in equal times", which means that the function $\upsilon(t)$ is not a linear time function. Furthermore, no closed-form solutions exist for this function and it is easier to compute the solution for orbit propagation in the perifocal frame using an angle called the *eccentric anomaly* E.

The *eccentric anomaly* E is the angle between the $x-$ axis direction and the line joining the center of the ellipse to a vertical projection of the point of interest in a concentric circle, as shown in Figure 10.

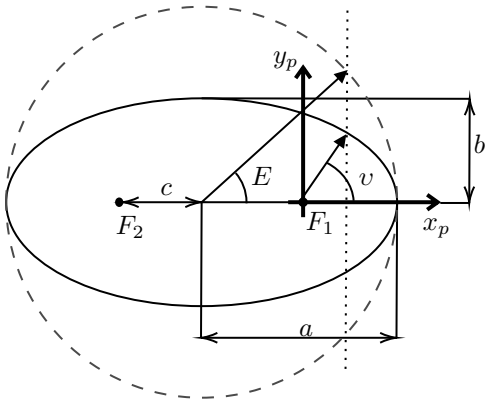


Figure 10: Ellipse Dimensions

This should not be confused with the *mean anomaly* M that is not represented. The *mean anomaly* M is an argument that is a linear function of time, as if the orbit was a circle, defined by the expression:

$$M(t) = M_0 + n(t - t_0), \tag{24}$$

$$n = \frac{2\pi}{T} \tag{25}$$

where n is the *mean motion* (corresponding to the mean angular velocity) and T is the orbital period. It can be shown that the mean motion is given by:

$$n = \sqrt{\frac{\mu}{a^3}}\,, \quad \mu \approx GM \tag{26}$$

where μ is the gravitational parameter, Newton's gravity constant times the mass of the Earth. The mean anomaly is easy to calculate but does not provide the true rate of elliptic motion. Kepler arrived at the following relation between the mean anomaly and the eccentric anomaly E in an orbit:

$$M(t) = E(t) - e\sin(E(t)) \tag{27}$$

which is known as Kepler's Equation. Even tough no direct form for $E(M)$ exists, this can be computed by iteration of the function:

$$f(E) = M - (E - e\sin(E)) \tag{28}$$

which is zero for the correct E. If $f(\hat{E})$ returns a residue that is too big, the following iteration can be given by the following update:

$$\hat{E}^{+} = \hat{E}^{-} + \frac{M - (\hat{E}^{-} - e\sin(\hat{E}^{-}))}{1 - e\cos(\hat{E}^{-})} \tag{29}$$

Having arrived at a value of the eccentric anomaly for a current time $E(t)$, the current position, velocity and distance to the center of the Earth can be calculated in the perifocal frame:

$$x = a(\cos(E) - e) \qquad y = a\sqrt{1 - e^2}\sin(E) \tag{30}$$

$$\dot{x} = -\left(\frac{na^2}{r}\right)\sin(E) \qquad \dot{y} = \left(\frac{na^2}{r}\right)\sqrt{1 - e^2}\cos(E) \tag{31}$$

$$r = a(1 - e\cos(E)) \tag{32}$$

Once the spacecraft's position and speed are calculated in the orbit plane, they can be transformed into *GCI* through a matrix transformation. The *GCI* frame and the perifocal frame can be related by a $3 - 1 - 3$ Euler Angle sequence of angles Ω, i, ω :

$$\boldsymbol{r} = A_{313}^{T}(\Omega, i, \omega)\begin{bmatrix} x \\ y \\ 0 \end{bmatrix} \qquad \dot{\boldsymbol{r}} = A_{313}^{T}(\Omega, i, \omega)\begin{bmatrix} \dot{x} \\ \dot{y} \\ 0 \end{bmatrix} \tag{33}$$

5.2 Sun-syncrhonous Orbit

The midnight-noon Sun-synchronous orbit was introduced previously in Section 3. The satellite's orbit plane is able to rotate due to the Earth's equatorial bulge: Earth's diameter is 43 km bigger in the equatorial plane than in the poles, which means that Earth's gravitational field is stronger in the equatorial plane, perturbing any satellite orbit that is not aligned with Earth's symmetry. This perturbation, also known as $J2$, causes the orbit plane to slightly rotate with time, at a rotation rate depending on the orbit size a, inclination i and its eccentricity e. Correctly tuning these parameters lets us achieve the desired midnight-noon orbit. It can be shown that for an eccentricity of zero and a mean height of 500 km, the inclination must be $i = 97.461^o$. [9]

The current model does not account for these perturbations and the classical orbit elements are constant. A right ascension rate $\dot{\Omega}$ is added to simulate the rotating effect, which corresponds to the rate of one revolution per year:

$$\Omega(t) = \Omega_0 + \dot{\Omega} \cdot t\,, \quad \dot{\Omega} = \frac{2\pi}{365.25 \times 24 \times 3600} = 1.991 \times 10^{-7} \quad [rad\,s^{-1}] \tag{34}$$

The offset Ω_0 must be calculated taking in account the simulation epoch in order to make sure that the orbit plane is oriented at noon. To calculate the Sun's position in the *GCI* frame one must first define the current time in terms of astronomical events, which is called the *epoch*.

5.3 *Epoch* Calculations

In this section are presented the algorithms used to calculate the simulation's *epoch*, which can be expressed simply as a date and a time, but also in more standardized formats like the Julian Date and Day of Year. The current calendar standard is the Gregorian Calendar and the current time standard is called the Universal Time, based on Earth's rotation.

5.3.1 Julian Date

The Julian Date is the standard format of declaring a certain *epoch* in navigation, a scalar corresponding to the time elapsed since a specific reference time. The default reference starts on midday of November 22 of the year 4714 BC, the start of the Julian Calendar. The following expression returns the number of Julian Days, for a given year Y, month M, day D, hour h, minute m and second s of the Gregorian Calendar:

$$\begin{aligned} JD = 1,\,721,\,013.5 + 367Y - INT\left\{\frac{7}{4}\left[Y + INT\left(\frac{M+9}{12}\right)\right]\right\} \\ +INT\left(\frac{275M}{9}\right) + D + \frac{60h + m + s/60^*}{1440} \quad [days] \end{aligned} \tag{35}$$

where INT denotes the integer part and 60* denotes using 61 seconds for days with leap seconds. Leap seconds are rather uncommon and can be ignored to a good approximation since only a total of 27 leap seconds have been added since the year 1972 to the present year of 2019. The next predicted leap second should be added on June 30, 2020. The Julian Date can be offset to a common reference that is J2000, which refers to midday of January 1 of the year 2000.

$$JD_{J2000} = JD - 2,\,451,\,545 \quad [days] \tag{36}$$

5.3.2 Day of Year

Models like the NRLMSISE, discussed in the atmospheric model chapter, use Day of Year as a measure of the current epoch. This can easily be calculated by subtracting the Julian Date of the current year at midnight of January 1 to the current Julian Date:

$$DOY = JD(Y, M, D, h, m, s) - JD(Y, 0, 0, 0, 0, 0) \tag{37}$$

5.4 Sun and Earth Position

Earth orbits the Sun at the rate of one revolution per year in an elliptic orbit. This motion causes an apparent motion of the Sun around the Earth that can be described through the ecliptic reference frame: an Earth centered frame that maps the apparent position of the Sun, illustrated on Figure 11. Describing the Sun's position follows steps similar to the orbit propagation problem, where it is first described on the ecliptic orbit plane and then transformed to the *GCI* frame. The following algorithm presents a good approximation to the position of the Sun, with a precision better than 1^o. [26]

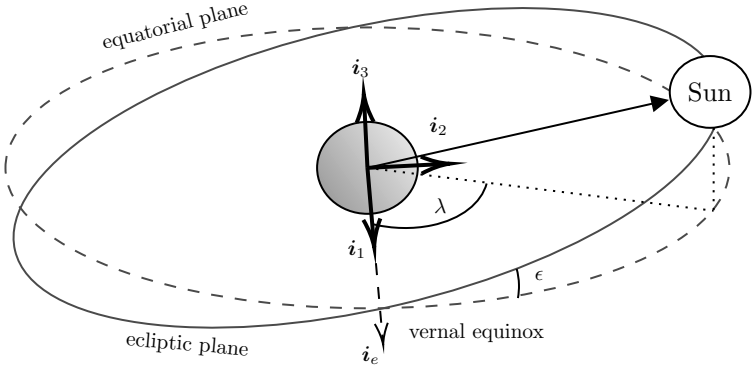


Figure 11: Ecliptic Plane

The mean longitude of the Sun, corrected for the aberration of light, is:

$$L = 280.460^o + 0.9856474^o \cdot JD_{J2000} \tag{38}$$

The mean anomaly of the Sun in the ecliptic frame is:

$$g = 357.528^o + 0.9856003^o \cdot J_{J2000} \tag{39}$$

These angles can be put in the range of 0^o to 360^o by:

$$L \equiv L \pmod{360^o}, \quad g \equiv g \pmod{360^o} \tag{40}$$

The ecliptic longitude of the Sun is:

$$\lambda = L + 1.915^o \sin(g) + 0.020^o \sin(2g) \tag{41}$$

Since the ecliptic latitude of the Sun is nearly zero, the distance from the Sun to the Earth can be given in astronomical units by:

$$R = 1.00014 - 0.016171 \cos(g) - 0.00014 \cos(2g) \tag{42}$$

Where an astronomical unit corresponds to $AU = 1,49597871 \times 10^{11}$ meters. The position is now completely defined in the ecliptic coordinate system and needs to be converted into *GCI*. First we compute the *obiquity of the ecliptic* ϵ, the angle between Earth's rotational axis and the Sun's orbital axis, also known as the axial tilt:

$$\epsilon = 23.439^o - 0.0000004 \times JD_{2000} \tag{43}$$

The right ascension of the orbital plane can be calculated:

$$\alpha = \arctan(\cos(\epsilon), \tan(\lambda)) \tag{44}$$

And its declination:

$$\delta = \arcsin(\sin(\epsilon), \sin(\lambda)) \tag{45}$$

Finally, it can be computed in *GCI* by:

$$x_i = R\cos(\epsilon)\cos(\lambda)\,, \quad y_i = R\cos(\epsilon)\sin(\lambda)\,, \quad z_i = \sin(\epsilon) \tag{46}$$

Referring back to the sun-synchronous orbit problem of finding a value for Ω_0, a midnight-noon orbit can be defined easily by letting $\Omega_0 = \lambda$, ensuring that the orbit plane is pointing in the direction of the Sun.

5.5 Latitude and Longitude

Once the spacecraft's position is fully defined in the *GCI* Frame it is necessary to describe it in the *ECEF* Frame in order to specify the spacecraft's position on Earth itself. To achieve this, θ_{GMTS} must be computed in terms of the *epoch*: first we calculate the number of Julian centuries elapsed from the J2000 reference to zero hours of the date in question:

$$T_0 = \frac{JD_{2000}(Y,\ M,\ D,\ 0,\ 0,\ 0)}{36,\ 525} \quad [centuries] \tag{47}$$

Once this is calculated, the current θ_{GMTS} can be calculated in seconds:

$$\begin{aligned} \theta_{GMTS} = 24,\ 110.54841 + 8,\ 640\,184.812866\,T_0 + 0.093104\,T_0^2 \\ -6.2 \times 10^6\,T_0^3 + 1.002737909350795(3600h + 60m + s) \quad [s] \end{aligned} \tag{48}$$

Then it is reduced to a range from 0 to 86400 s (number of seconds in a day) and converted to seconds by dividing by 240, since $86400/360^{\circ} = 240\,s/^o$:

$$\theta_{GMTS} = \frac{\theta_{GMTS}\ (\mathrm{mod}\ 86400^o)}{86400} \quad [^o] \tag{49}$$

Converting from *GCI* to *ECEF* is done simply by the matrix:

$$A = \begin{bmatrix} \cos(\theta_{GMTS}) & \sin(\theta_{GMTS}) & 0 \\ -\sin(\theta_{GMTS}) & \cos(\theta_{GMTS}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{50}$$

Most Earth models do not use the *ECEF* frame to describe a geographical location but rather geodetic latitude μ, longitude λ and height h. These values are calculated through an algorithm available in Matlab, based on the World Geodetic System (WGS84). The WGS84 is the most popular Earth model standard, used by GPS systems and others [27]. Simulation results for an ascending sun-synchronous orbit, starting at midday of the day 20/03/2019, can be seen below on Figure 12. It is interesting to see that in the first passage the satellite is near the 0^o and the 180^o longitudes, as expected.

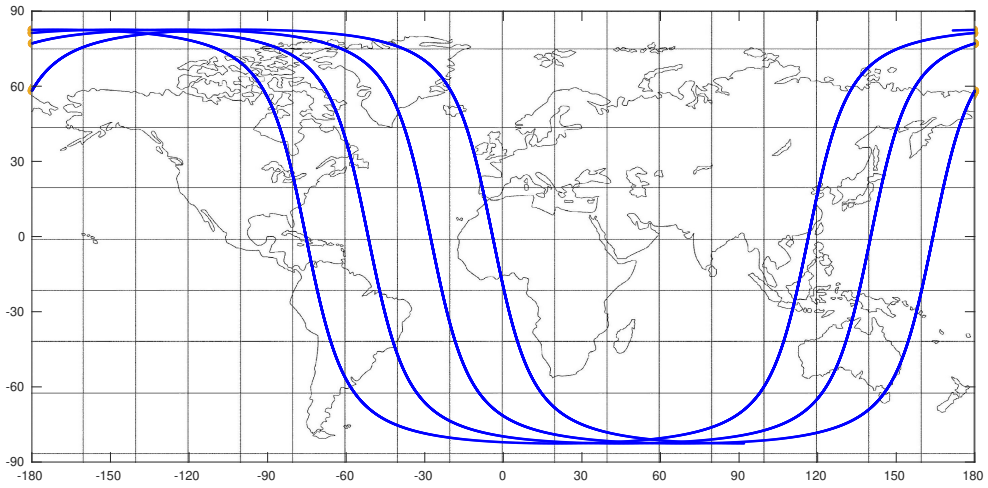


Figure 12: Simulation results for a few orbits of *Infante* plotted on a world map

5.6 Eclipse

Given the position of the Sun it is possible to determine if the Earth is eclipsing the spacecraft, an important information for orientation and energy purposes. The eclipse condition is determined by the following algorithm, inspired by [28].

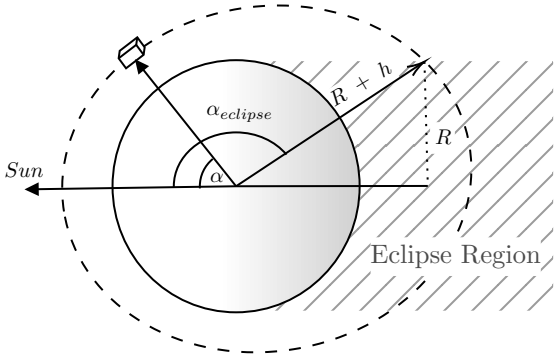


Figure 13: Eclipsed Region

In this model, the distance from the Earth to the Sun is considered great enough when compared to the Sun's radius to make the assumption that the Sun is a point of light at an infinite distance, generating a shadow region like the one in Figure 13. In reality, there is a transition region in this shadow called the *penumbra*, which takes a few seconds to cross in orbit.

Consider the plane that contains the Sun direction as well as the spacecraft's direction from the center of the Earth, represented in Figure 13. The angle α which describes the position of the satellite from the Sun in this plane can be calculated:

$$\alpha = \arccos\left(\frac{\boldsymbol{r}_{Sun} \cdot \boldsymbol{r}}{||\boldsymbol{r}_{Sun}|| \cdot ||\boldsymbol{r}||}\right) \tag{51}$$

The angle $\alpha_{eclipse}$, which describes the angle at which an eclipse starts, is given in radians by.

$$\alpha_{eclipse} = \pi - \arcsin\left(\frac{R}{R-h}\right) \tag{52}$$

where h is the spacecraft's height and R is the Earth's radius at the eclipse latitude. Since *Infante* has a high inclination, low altitude, midnight-noon orbit, R can be approximated to the Earth's radius in the poles, $R \approx 6357$ km. The condition for eclipse can then be expressed as $\alpha > \alpha_{eclipse}$.

The atmosphere plays an important role in the eclipse phenomenon because it distorts light as the Sun reaches the horizon. It makes sense to define another angle $\alpha'_{eclipse}$, calculated by the same expressions above but with $R' = R + h_{atm}$.

6 Environment Modeling and Disturbances

In this section the main attitude disturbances are discussed and the respective equations are presented, as well as the inherent magnetic and atmospheric models . The main attitude disturbances can be separated in the following torques:

- Solar Radiation Pressure Torque
- Gravity Gradient Torque
- Aerodynamic Torque
- Magnetic Torque

Solar Radiation Pressure torque is caused by the momentum exchange of photons hitting the satellite, creating torque depending on the spacecraft's attitude toward the Sun. This disturbance is considered to be dominated by the other three at the altitude of 500 km in which *Infante* operates, so it will not be covered by this thesis.

Gravity Gradient Torque occurs in any non symmetrical body subject to a gravity field. The underlying concept is: a satellite in space has parts of its body closer to the Earth than others, which are under the influence of a slightly stronger gravitational force. This difference in gravitational force can create a small torque if the attitude is not aligned with the field. In Figure 14, the satellite has the higher tip of the wing subjected to an acceleration $\boldsymbol{g}$, while the lower tip has a slightly stronger acceleration $\boldsymbol{g} + \delta\boldsymbol{g}$, creating a small roll torque. The Gravity Gradient $\delta\boldsymbol{g}$ is stronger the larger the body and the closer it is to the Earth.

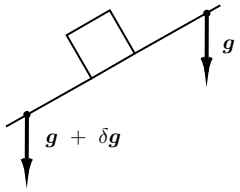


Figure 14: Satellite under Gravity Gradient Torque

In LEO, the predominant disturbance is usually the aerodynamic torque, due to a higher air density. In a similar way to the Gravity Gradient Torque, it can be made null if the proper attitude regarding the air speed is taken, which will be studied in this section.

Finally, Magnetic Disturbance Torque is due to the residual magnetic dipole of the magnetorquers after they have been operated. The residual magnetic dipole is not modeled since in modern magnetorquers a "dither" signal is applied, to minimize this effect. The time of actuation of this signal is taken in account.

6.1 Gravity Gradient Torque

The Local *LVLH* frame is a useful tool to study the Gravity Gradient Torque since the $z-$axis always points down in the direction of gravity field $\boldsymbol{g}$. Markley and Crassidis [9] present a solution for Gravity Gradient torque in the local frame for a $3-2-1$ Euler Angle representation:

$$\boldsymbol{L}_g = \frac{3\mu}{||\boldsymbol{r}^3||} \begin{bmatrix} (J_3 - J_2)\cos(\theta)^2 \cdot \cos(\psi) \cdot \sin(\psi) \\ (J_3 - J_2)\cos(\theta) \cdot \sin(\theta) \cdot \cos(\psi) \\ (J_1 - J_2)\cos(\theta) \cdot \sin(\theta) \cdot \sin(\psi) \end{bmatrix} \tag{53}$$

It is interesting to notice that it does not depend on the yaw angle ϕ and that it decreases with the cube of the distance to the center of the Earth.

6.2 Aerodynamic Torque and Drag

This section uses the methods developed in [29] to model the Aerodynamic Torque of the spacecraft, where parts of the text have been adapted:

The highly rarefied atmosphere of LEO requires a different approach to aerodynamics than that employed in the continuum regime that exists at aircraft flight altitudes. Due to the low density, the flow regime of low Earth orbit spaceflight is commonly described as free molecular. This means that the mean-free path (the mean distance between consecutive collisions) is many times greater than the characteristic dimension of a body immersed in the flow. Therefore, collisions between molecules are extremely rare in the flow field around the body, such that the flow can be assumed collisionless and it cannot be considered as a continuum medium anymore. Instead it is particulate in nature.

To quantify the validity of the collisionless assumption a non-dimensional parameter known as Knudsen number is used:

$$Kn = \frac{\lambda}{l_{ref}} \tag{54}$$

where l_{ref} is the characteristic dimension of the spacecraft and λ is the mean free path. A high Kn indicates that the flow is particulate in nature and a low Kn indicates that the flow is continuum. It is assumed that the free molecular flow assumption is valid for $Kn \geq 10$. The Knudsen number for an altitude of 500 km and a mean length of 1 m is usually between 10^4 and 10^6, indicating that the free flow assumption is reasonable not only for *Infante*, but for all kinds of satellites. The molecular mean free path can vary up to two orders of magnitude with magnetic and solar activity, which are responsible for large density and temperature variations in the thermosphere.

Another important parameter in free molecular flow is the molecular speed ratio:

$$s = \frac{V}{V_a} \tag{55}$$

where V_a is the thermal speed, which defines temperature as the most probable molecular speed of a gas, and V is the velocity of gas in the reference frame. It can be shown that the most probable speed of the gas is given by:

$$V_a = \sqrt{2RT} \tag{56}$$

where R is the specific gas constant of air and T is the temperature in K. The molecular speed ratio s provides an indication of the extent to which the flow behaves like a collimated beam of molecules (hyperthermal flow), where the bulk velocity of the gas is many times greater than the thermal velocity of the gas, or a chaotic drifting Maxwellian flow (hypothermal flow), where the high random thermal motion of the atmospheric gas constituents means that the free stream gas flow cannot be treated as a collimated beam of molecules anymore.

The specific constant of air can be calculated using the molar mass of air which is approximately 15 g/mol at 500 km: [30]

$$R = \frac{\overline{R}}{M} \approx \frac{8.314}{15 \times 10^{-3}} = 554.267 \tag{57}$$

At 500 km of altitude temperatures can easily reach values of 1500 K and simulation results indicate a mean air speed of $V = 8\,km/s$, considering that the atmosphere rotates with the Earth. Taking in account the equations above, the molecular speed ratio can be calculated:

$$s = \frac{8000}{\sqrt{2 \times 554.267 \times 1500}} = 6.20 \tag{58}$$

For practical implementations it is assumed that the hyperflow assumption is valid for $s > 5$, which is the case.

Aerodynamic torque is caused by the momentum exchange of particles with the spacecraft so it is necessary to characterize the interaction of the particles with the body. Upon collision with the spacecraft's surface, particles can be reflected with the same angle (specular reflection), or they can be reflected at a random angle (diffuse reflection). A proper gas-surface interaction model (GSIM) must be applied to the study of the satellite in order to describe the occurrence of these two cases.

The force $d\boldsymbol{F}$ acting on a spacecraft element dA is defined upon the moment exchange of a particle hitting the surface at an angle θ_i and speed V_i, reflected at angle θ_r and speed V_r, as illustrated in Figure 15. The momentum exchange can be calculated by the difference of momentum fluxes:

$$\frac{d\boldsymbol{F}}{dA} = -(p_i + p_r)\boldsymbol{n} + (\tau_i - \tau_r)\boldsymbol{t} \tag{59}$$

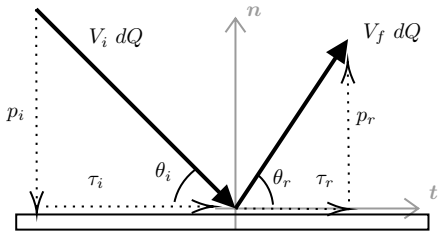


Figure 15: Reflection of a particle

Where p_i and p_r are the normal momentum fluxes and τ_i and τ_r are the tangent momentum fluxes. These incident momentum fluxes p_i and τ_i depend on incident velocity V_i and mass flux $dQ = \rho V_i \sin(\theta_i)$, and can be calculated:

$$p_i = V_i \sin(\theta_i) dQ \qquad \tau_i = V_i \cos(\theta_i) dQ \tag{60}$$

The reflected momentum flux does not follow a direct calculation, p_r and τ_r depend on gas properties such as chemical composition, density, temperature and speed, surface properties such as material, roughness, cleanliness and temperature and finally, the incidence angle θ_i. To estimate the reflected momentum a gas-surface interaction model (GSIM) must be studied.

The proposed GSIM is the Schamberg model, since it is well covered in [29], with a simple solution for this case. It considers that the reflected particles are emitted with a scattering pattern of a cone, as illustrated in Figure [16]. The reflected beam of particles is described by a mean reflection angle θ_r and a half angle ϕ_0. This model considers a uniform distribution of velocities in the scattering cone, so the mean of particles can be considered to be reflected at the mean reflection angle θ_r, affected by a scalar function $\Phi(\phi_0)$. In the diffuse case, ϕ_0 takes the value of $\pi/2$ and the respective scalar Φ equals 2/3.

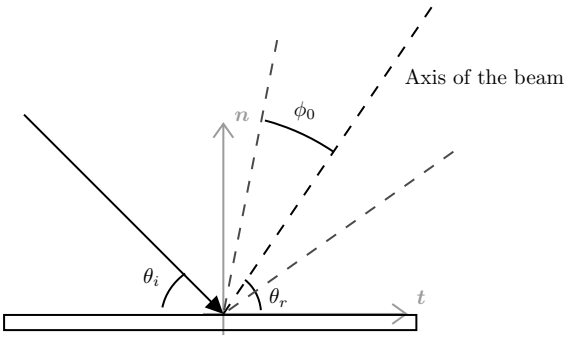


Figure 16: Schamberg's scattering model

The momentum of the reflected particle also depends on what is known as the accommodation coefficient. Particles arriving at the spacecraft's surface can interact and exchange energy with it, losing momentum accordingly. Complete thermal accommodation implies $\alpha = 1$, the particle is re-emitted with energy corresponding to temperature of the surface, sometimes neglected due to high spacecraft speed. No energy exchange implies $\alpha = 0$ and the particle is reflected with the same velocity. This coefficient is calculated:

$$\alpha = \frac{E_i - E_r}{E_i - E_w} \approx \frac{E_i - E_r}{E_i} \tag{61}$$

where E_i, E_r and E_w are incident, reflected and surface energies respectively.

Angular distribution of reemitted particles and the accommodation coefficient are affected mainly by the type of molecules of the incident gas and surface contamination. Clean surfaces have low accommodation coefficients while contaminated surfaces have high coefficients, since they exhibit a process known as adsorption: atmospheric molecules become trapped in the surface and incident molecules are more likely to be re-emitted diffusely. This effect is very pronounced at altitudes between 200 and 700 km due to the presence of atomic oxygen contaminating the surface. For clean surfaces the accommodation coefficient is very dependent of the material and its roughness, while for contaminated surfaces it has been shown that it is almost independent [29] [31]. This has been measured in several spacecraft: it is very close to 1 for altitudes up to 170km but it starts decreasing with altitude as adsorption gets weaker, primarily due to increasingly lower air density. The article [31] also argues that experiments on Earth have been found to be very little representative of the true values in space because of difficulties in replicating the space environment.

Due to high adsorption, it is assumed that the angular distribution is almost purely diffuse: particles are re-emitted at random because they become trapped in the surface due to contamination. In orbit experiments measuring α make this assumption achieving good results. The article [30] provides a table of recommended accommodation coefficients based on satellite experimental data where a completely diffuse re-emition is assumed, but only to an altitude of 325 km ($\alpha = [0.89 - 0.93]$). The coefficient increases in times of high solar and magnetic activity since this has an influence in atmosphere composition.

Taking in account this information and that the atmospheric density decays exponentially, the following assumptions are made in the *Infante* aerodynamic model:

- The reflected molecules are re-emitted diffusely
- The accommodation coefficient is $\alpha = 0.85$

Solving the Schamberg model for diffusely reflected particles in a flat plate dA, as described in Figure 15, results in the reflected moment fluxes:

$$p_r = \Phi V_r \sin(\theta_r) dQ \qquad \tau_r = \Phi V_r \cos(\theta_r) dQ \tag{62}$$

Substituting $dQ = \rho V_i \sin(\theta_i)$, $\Phi = 2/3$ for the diffuse case, and including the accommodation effect :

$$p_r = \frac{2}{3}\rho V_i^2 \sin(\theta_i)^2 r \qquad \tau_r = \frac{2}{3}\rho V_i^2 \cos(\theta_i)\sin(\theta_i) r \tag{63}$$

where r is the speed ratio, that can be related to the accommodation coefficient by:

$$r = \frac{V_r}{V_i} = \sqrt{1-\alpha} \tag{64}$$

Assuming that the thermal speed is negligible in comparison with spacecraft speed. Substituting Schamberg's solution in equation (59) and integrating it, the expression for the aerodynamic force in a flat plate of area A becomes:

$$\boldsymbol{F} = A\rho V_i^2 \sin(\theta_i) \left[\left(\frac{2}{3} r - 1 \right) \sin(\theta_i) \boldsymbol{n} + \left(1 - \frac{2}{3} r \right) \cos(\theta_i) \boldsymbol{t} \right] \tag{65}$$

For the purpose of this aerodynamic model, *Infante* is considered as collection of eight flat panels. Knowing the air speed, it is possible compute the aerodynamic torque by summing the torque contribution of each panel: a cross product between the aerodynamic force and the distance from the center of pressure to the center of mass. A step further is taken by considering the shadow cast by the body and wings on each panel, in a method known as Ray Tracing Panel Method (RTP). [29]

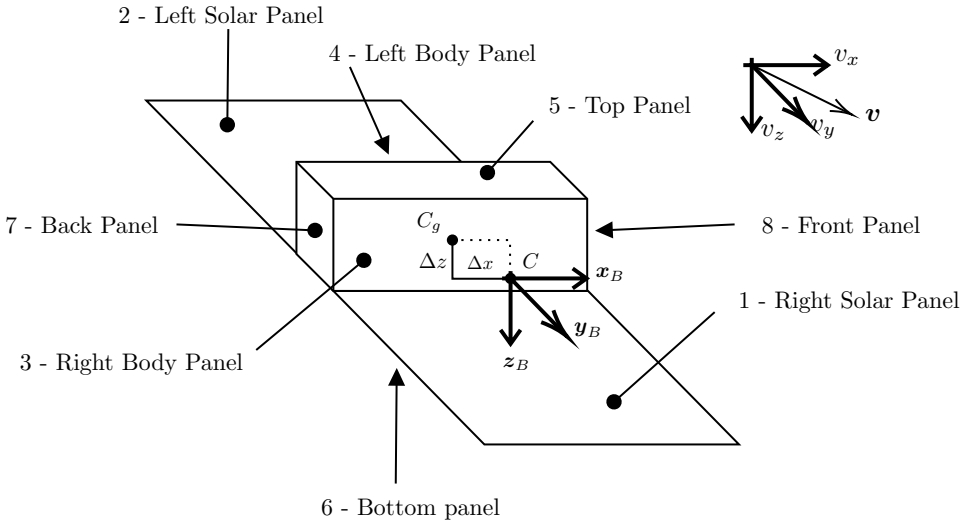


Figure 17: Panel Model of Infante

A panel is described by the normal unit vector $\boldsymbol{n}$ pointing outside of the panel. When the body is subject to air speed, characterized by the vector $\boldsymbol{V}$, Ray Tracing is used to figure out if the panel is being shielded or partially shielded by the flow. Take the complex case of the Panel number 1 as an example, represented in Figure 19. The other panels follow an identical method.

The normal vector characterizing this panel is:

$$\boldsymbol{n} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \tag{66}$$

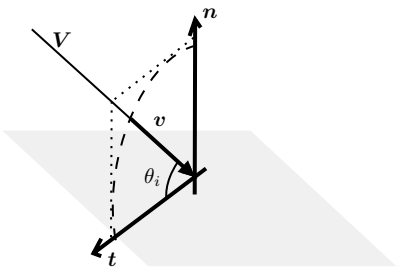


Figure 18: Tangent Direction of Air Speed

The tangent direction $\boldsymbol{t}$ depends on the air speed. Consider the unit vector characterizing air speed direction $\boldsymbol{v} = [v_x, v_y, v_z]^T$. The trigonometric functions of the incidence angle can be calculated by:

$$\sin(\theta_i) = -\boldsymbol{v} \cdot \boldsymbol{n} \qquad \cos(\theta_i) = \sqrt{1 - \sin(\theta_i)^2} \tag{67}$$

The vector $\boldsymbol{t}^*$ can be obtained by subtracting the normal component of the air direction and normalizing:

$$\boldsymbol{t}^* = \boldsymbol{v} - \sin(\theta_i)\boldsymbol{n}\,, \quad \boldsymbol{t} = \frac{\boldsymbol{t}^*}{||\boldsymbol{t}^*||} \tag{68}$$

Once the normal and tangent vectors are defined it is necessary to compute the panel area. Looking at Figure 19 it is possible to see that the panel will be shielded if $\boldsymbol{V_z}$ is negative (flow hits bottom panel instead) or if the shadow angle γ_i is smaller than 90^o.

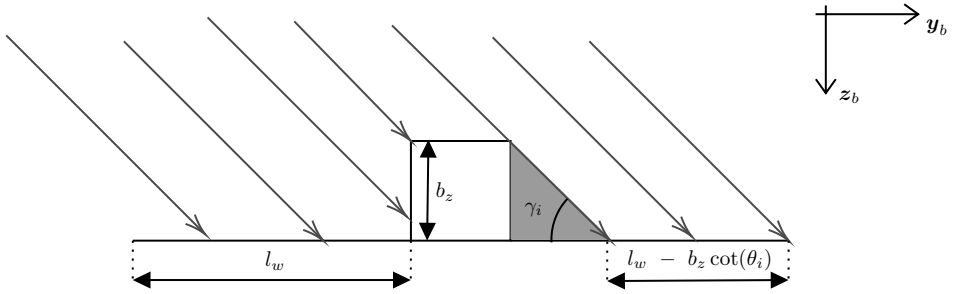


Figure 19: Ray Tracing of the Wings

The area for the right panel can be computed if there is no shadow by the simply

$$A = l_w \times l_x \tag{69}$$

If there is shadowing, the area can be calculated in terms of the angle γ_i:

$$A = (l_w - l_{shadow}) \times l_x \tag{70}$$

where the shadow length is:

$$l_{shadow} = b_z \cot(\gamma_i) = b_z \frac{v_y}{v_z} \tag{71}$$

and the shadow condition becomes $l_{shadow} < l_w$ and v_z, $v_y > 0$. Now that there is knowledge of the effective surface area and of the air speed, the aerodynamic force can be calculated using equation (65). Computing the right wing torque can be done as long as the distance from the center of mass to the center of pressure is known. If there is no shielding:

$$\boldsymbol{d}_{C-C_p} = \begin{bmatrix} -\Delta x \\ (b_y + l_w)/2 \\ -(b_z - \Delta z) \end{bmatrix} \tag{72}$$

and if there is body shielding:

$$\boldsymbol{d}_{C-C_p} = \begin{bmatrix} -\Delta x \\ l_w - (l_w - l_{shadow})/2 + b_y/2 \\ -(b_z - \Delta z) \end{bmatrix} \tag{73}$$

Finally, the aerodynamic torque of this panel can be computed by the cross product:

$$\boldsymbol{L}_1 = \boldsymbol{d}_{C-C_p} \times \boldsymbol{F}_1 \tag{74}$$

The remaining panels follow the same procedure: the left solar panel is symmetric to the right solar panel and equations can be easily adapted by changing signs. Right and left body panels suffer shielding from the solar panels when the air speed is coming from the side at some angle. Because the solar panels are much bigger in length than the height of the body and because the side surface area is considerably smaller than the area of the solar panels, it is considered that the side panels suffer no shielding when $v_z > 0$ and are not exposed when $v_z < 0$. Since the remaining top, bottom, front and back panels suffer no partial shielding, the existence of flow on those surfaces be declared easily from the signs of v_x, v_y and v_z.

The total aerodynamic force and torque can be calculated simply by adding the contributions of each panel.

$$\boldsymbol{F}_{aero} = \sum_{i=1}^{8} \boldsymbol{F}_i \,, \qquad \boldsymbol{L}_{aero} = \sum_{i=1}^{8} \boldsymbol{L}_i \tag{75}$$

For confirmation purposes, the drag coefficient can be calculated when the satellite is moving forward, using the frontal area as the reference area:

$$A_{ref} = b_z \times b_y \tag{76}$$

$$F_{drag} = \boldsymbol{F}_{aero} \cdot \boldsymbol{v} \tag{77}$$

$$C_d = \frac{2F_{drag}}{\rho A_{ref} V^2} \tag{78}$$

Simulations of this aerodynamic model and stated assumptions indicate a drag coefficient $C_d \approx 2.1$ which is within the values of $C_d = 2$ to $C_d = 2.5$ expected for satellites in LEO, as indicated in literature [9].

6.3 Atmosphere Model

The Atmosphere Model used in this thesis was the *2001 United States Naval Research Laboratory Mass Spectrometer and Incoherent Scatter Radar Exosphere* (NRLMSISE-00) [32], an extensive empirical model of the upper atmosphere. It is available as a Simulink block in MATLAB's Aerospace Toolbox; it returns total air density and composition if height, latitude, longitude and epoch are provided. An optional feature is the addition of solar and magnetic parameters, which become extremely important at LEO altitudes as mentioned previously. These parameters, like the A_p index for geomagnetic activity, can be obtained accurately for the present but cannot be determined for future times, since they are part of unpredictable phenomenons known as space weather.

6.4 Optimal Attitude and Stability

Given the extensive modeling in the previous aerodynamic torque chapter, the goal of this chapter is to study which attitude minimizes the mean disturbance torque.

If the attitude was to be defined with the body axes coincident with the local frame (case (1) in Figure 20), the gravity gradient torque would be null, but the aerodynamic torque would have a positive component in pitch, assuming that the incoming air has the opposite direction of speed. This is due to the fact that the front panel has a pressure center higher than the center of mass, and when the air hits it, it creates a torque that tends to pitch the satellite backwards.

As the satellite tilts backwards (case (2) in Figure 20), the aerodynamic force in the bottom panel tends to create an opposite torque, since the satellite's center of mass is slightly ahead of the pressure center. If the pitch angle is high enough, the total aerodynamic torque becomes null but the total disturbance torque is not null since the gravity gradient torque is acting, tilting the satellite backwards. Even tough the bottom area is much greater, its torque arm, distance from the center of pressure to the center of mass, is smaller than the torque arm of the front plate. Therefore, a great inclination is needed in order to compensate for the front panel torque and the gravity gradient torque.

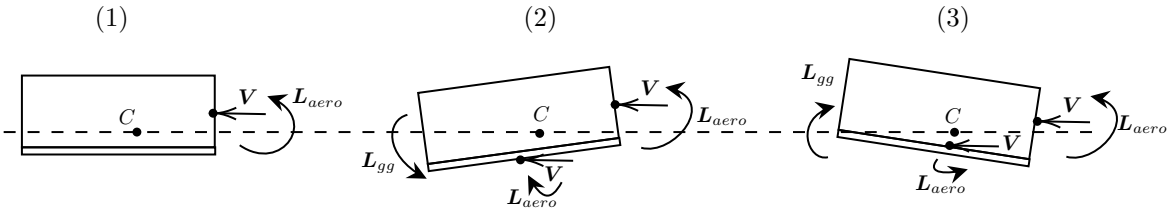


Figure 20: Different Attitude Pitch and Disturbance Torques

In the case of negative pitch (case (3) in Figure 20), the top and solar panel plates act to compensate the front panel torque but the gravity gradient torque acts favorably, reducing the amount of pitch that is needed in order to achieve zero mean disturbance torque.

In Figure 21 the simulated disturbance torques are plotted in terms of the pitch angle, assuming the wind speed matches the spacecraft speed. For the specified case of a height of 500 km and a torque arm of $\Delta x = 11$ mm, it is visible the gravity gradient torque plays a major role in the attitude disturbance, increasing almost linearly with pitch. Due to the small torque arm the aerodynamic torque remains similar for most pitch angles. A minimum point in the total disturbance curve is found for a small negative pitch of -0.424^o, which is the optimum flying pitch.

Figure 21 also shows that the satellite is pitch unstable: for a pitch under the optimum pitch the disturbance torque is negative, while for a pitch larger than the optimum point, the disturbance is positive.

Next, the same simulation was run but with the center of mass pushed significantly forward, so that $\Delta x = 100$ mm. The objective is to evaluate if pushing the center of mass forward enough makes the optimum pitch point attractive. In Figure 22, it is visible that the spacecraft now becomes aerodynamically stable, but the overall disturbance torque is still unstable due to the gravity gradient torque. Also, the optimal pitch was found at -0.44^o, very similar to the original case.

While *Infante* will spend most of its lifetime at the height of 500km, it will necessarily make its way through the 400 km region when de-orbiting, where the aerodynamic torque is much more pronunciated. In Figures 23 and 24 is plotted the pitch influence in the disturbance torques, for $\Delta x = 11$ mm and $\Delta x = 100$ mm, at the height of 400 km.

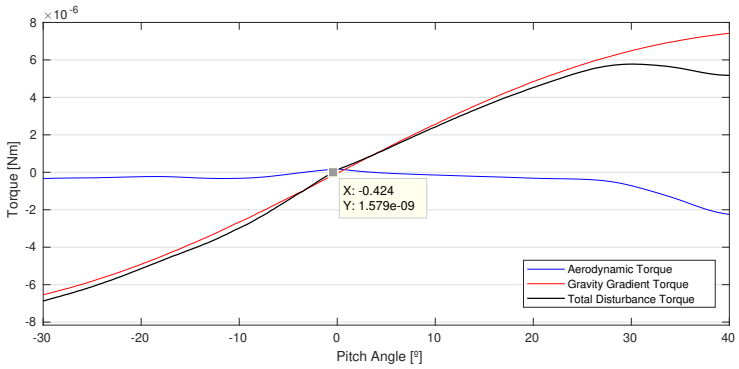


Figure 21: Disturbance Torques with Pitch Angle, $\Delta x = 11$mm, h = 500 km

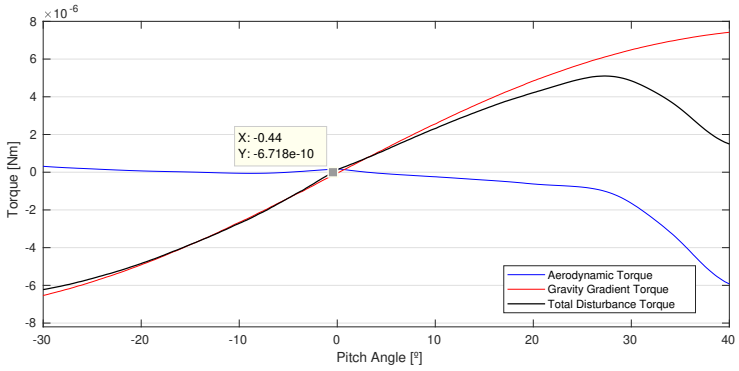


Figure 22: Disturbance Torques with Pitch Angle, $\Delta x = 100$mm, h = 500 km

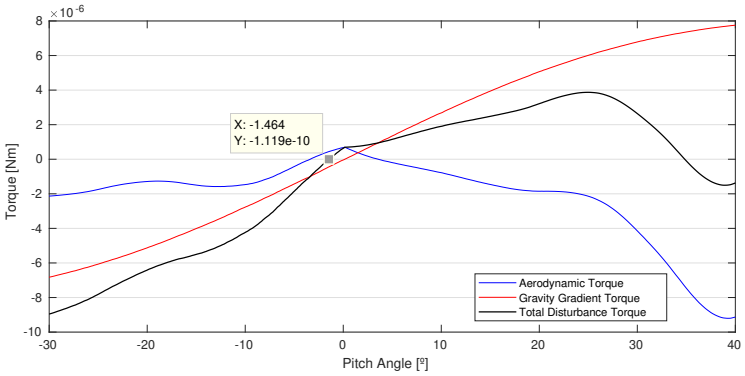


Figure 23: Disturbance Torques with Pitch Angle, $\Delta x = 11$mm, h = 400 km

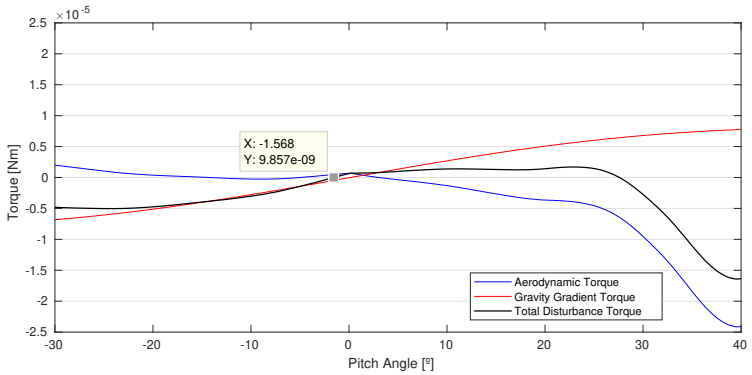


Figure 24: Disturbance Torques with Pitch Angle, $\Delta x = 100$mm, h = 400 km

Figure 24 shows that for a lower altitude and a center of mass that is placed significantly forward, the spacecraft tends to stability but the optimal point is still unattractive. For orbits in the range of 300-400 km, the engineering effort of placing the spacecraft's center of mass as forward as possible is rewarded with less disturbance and stability. However, in the case of *Infante* this effort is not significant since it will be operated at a height in the region of 500 km for most of its mission lifetime.

6.5 Earth's Magnetic Field

The Magnetic Field Model used in this study is the *World Magnetic Model 2015* (WMM2015), a complex model produced by the United States' National Geospatial-Intelligence Agency (NGA) and the United Kingdom's Defence Geographic Centre (DGC), that is available to the general public in [33]. The current model is valid up to December 2019; unfortunately the next model update, the WMM2020, is yet to be released. This model is also available in MATLAB's Simulink software as a block that returns the magnetic field direction and intensity from height, latitude, longitude, and *epoch*.

6.6 Earth's Albedo

The main source of error in a Coarse Sun is Earth's Albedo, the reflection of sunlight from Earth's surface. It is assumed that the reflected sunlight comes from Earth's center and that its intensity depends of Earth's mean reflectance and Sun angle. This is a coarse approximation since Earth's albedo near the ice covered poles is 0.84 and in dark forest regions only 0.14. The mean albedo is assumed to be around 0.3, as indicated by [34].

The sunlight shining upon the spacecraft is divided in two components, direct sunlight and reflected light:

$$\boldsymbol{l} = \boldsymbol{l}_{sun} + \boldsymbol{l}_{reflected} \tag{79}$$

Direct sun light is equal to the Sun's solar constant, $I = 1361\,W/m^2$, the amount of incoming electromagnetic power, times the Sun direction vector.

$$\boldsymbol{l}_{sun} = I\boldsymbol{s} \tag{80}$$

The reflected sunlight is assumed to be coming from the center of Earth and that it is proportional to $\cos(\beta)$, as represented in Figure 25:

$$\boldsymbol{l}_{reflected} = aI\cos(\beta)\boldsymbol{n} \tag{81}$$

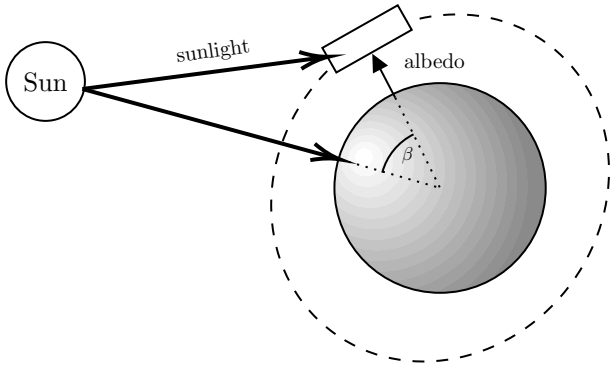


Figure 25: Albedo Model

where a is Earth's mean reflectance and $\boldsymbol{n}$ is the nadir vector, the vector pointing down from the spacecraft's to Earth's center. Sunlight shining upon the satellite can be further reduced to the following expression:

$$\boldsymbol{l} = I[\boldsymbol{s} + a(\boldsymbol{s}\cdot\boldsymbol{n})\boldsymbol{n}] \tag{82}$$

7 Sensors and Actuators

This section aims to explains the characteristics of the sensors and actuators and how they were modeled to provide a realistic control behavior.

7.1 Gyroscopes

Three-axis gyroscopes are instruments that measure the angular velocity of the body they are attached to. By integrating the angular velocity, it is possible to get an attitude estimation in the present from attitude measurements in the past, making gyroscopes particularly useful in attitude estimation. The quality of these estimations degrades over time because the angular rate errors are also integrated and accumulated. A gyroscope is usually subject to the following error sources:

- Rate White Noise
- Angular White Noise
- Systematic Bias
- Bias Instability
- Misalignment Errors
- Scale Factor Errors

Rate White Noise is simply the zero mean noise that happens naturally upon measuring the angular rate, mostly due to environment factors. This is considered a white noise source, in which the error variance does not depend on frequency. Integration of this noise source results in a phenomenon called Angular Random Walk (ARW).

Angular White Noise (AW), similar to rate white noise, is the measurement noise that is added upon integration of the angular velocity.

Systematic Bias is the systematic error that is present in the gyroscopes' axis. Integrating this bias results in a linear function with time, proportional to the bias. The bias is usually estimated on-board by the Kalman Filter, to be evaluated in the next section, and compensated for.

The bias in a gyroscope is not constant over time, therefore, the gyros's Bias Instability must be evaluated. It is assumed that the bias is a first order Gauss-Markov noise process. This is also called Random Rate Walk (RRW), but with a fundamental difference between this process and the Angular Rate Walk, that must be well understood. While the Angular Walk is a true unbounded Random Walk process, the Random Rate walk is bounded due to the nature of the Gauss-Markov process, therefore is not a true random walk in the strict sense.

When a gyro is mounted on a body, its axis might not be aligned with the body axis pretended to be measured, giving rise to misalignment errors. The output of the gyro might also be slightly different from the true rate by a scale factor. These two error sources can be compensated by gyro calibration. Due to this fact, these errors are assumed to be compensated and are out of the scope of this thesis.

Taking in account the information above, the following gyro model is implemented in Simulink:

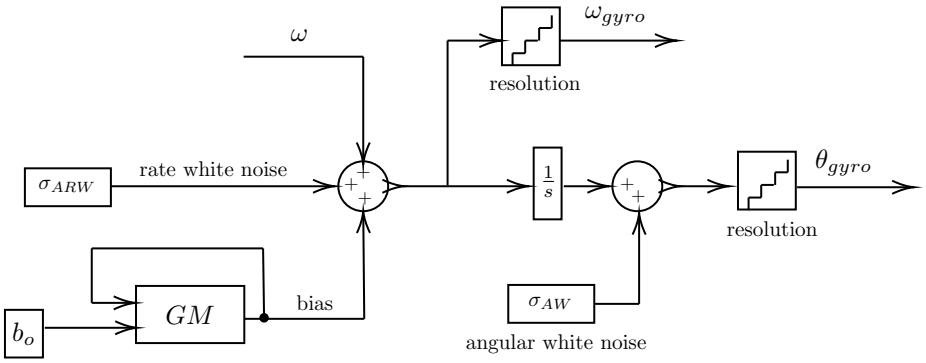


Figure 26: Block Diagram of the Gyroscope model

The square blocks with variances σ^2 are random number generators which produce Gaussian white noise at simulation step time, and the block GM represents a first order Gauss-Markov noise process. This model was inspired by articles [35] and [36].

A first order Gauss-Markov process is a self-correlated process that can be written in the following discrete form:

$$x_k = (1 - \beta\Delta t)x_{k-1} + \mu_k \tag{83}$$

where μ_k corresponds to Gaussian white noise $\mu_k \sim \mathcal{N}(0, \sigma)$ and the scalar β is the inverse of the correlation time T_c:

$$\beta = \frac{1}{T_c} \tag{84}$$

The higher the correlation time, the longer the process will stay self correlated. The noise deviation σ is related to the correlation time by the expression:

$$\sigma = \sigma_{GM}\sqrt{1 - e^{(-2\beta\Delta t)}} \tag{85}$$

where σ_{GM} is the standard deviation of the noise process, which in this case, is the Bias Stability of the gyroscope. If the gyro is standing still, it can be shown that for an infinite amount of samples, the expected value of the square of the noise process is:

$$\lim_{k\to\infty} \mathbb{E}(x_k^2) = \frac{\sigma^2}{1 - \alpha^2} \tag{86}$$

where $\alpha = 1 - \beta\Delta t$, therefore proving the convergence of the process. By contrast, a random walk process y_k, produced by integrating Gaussian white noise of variance σ^2 is divergent. The expected value of the square of the noise process after N samples is:

$$\mathbb{E}(y_N^2) = \sigma^2 N \tag{87}$$

The challenge of gyro modeling is not the error model itself, but quantifying the variances of the errors sources. Even tough information regarding these parameters is available in the datasheets of the used gyroscopes, it is worth mentioning the method used to extract this information. The most common method of measuring Angular White Noise, Angular Random Walk and Random Rate Walk is the Allan Variance Method, which will be used on the data produced by the simulation to assess if the gyro model is realistic.

Allan variance is a measure of noise stability used in clocks, oscillators, amplifiers and gyros. Suppose there is a certain amount of discrete gyro data with N samples over a total time t. The Allan Variance $\sigma^2(\tau)$ is the variance of the means of n consecutive smaller groups of data, with cluster time τ, where $\tau n = t$. It is estimated as:

$$\sigma^2(\tau) = \frac{1}{2\tau^2(N-2n)} \sum_{k=1}^{N-2n} (\omega_{k+2n} - 2\omega_{k+n} + \omega_k)^2 \tag{88}$$

It is represented as a curve: the variance as function of cluster time, as seen on Figure 27. For cluster time equal to the sampling time, the Allan Variance becomes simply the sample variance σ^2. The influence of Angular Random Walk and Rate Random Walk can be seen in this curve: presence of Angular Random Walk noise causes a decrease of the Alan Variance at smaller cluster times with a slope of -0.5 and presence of Rate Random Walk noise causes an increase with 0.5 at higher cluster times. The Allan Variance reaches a minimum when the cluster time reaches the Gauss-Markov correlation time.

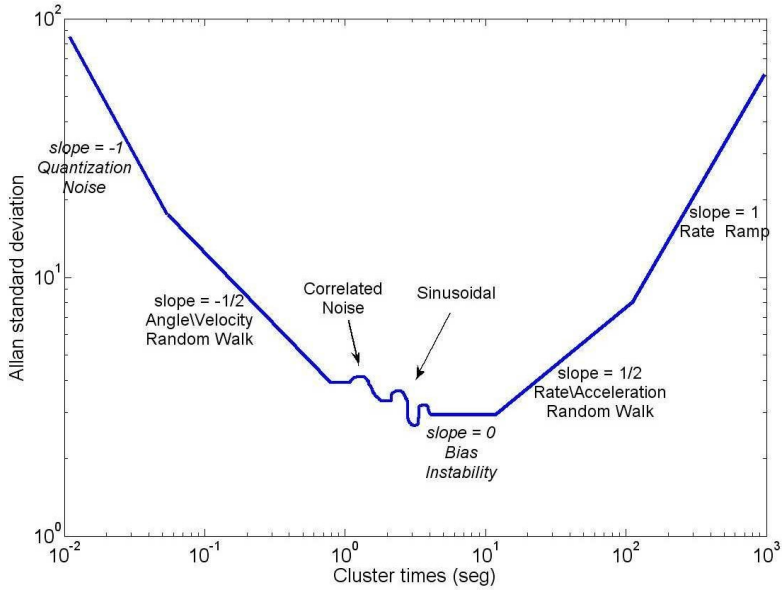


Figure 27: Allan Standard Deviation of an Hypothetical Gyro Measurement [36]

7.1.1 Coarse Gyro

It is expected that the Coarse Gyroscope to be mounted onboard the *Infante* Mission is the *Bosch BMG*250 low power gyroscope. Relevant aspects of the datasheet are gathered below [37]:

Table 3: Noise characteristics of the *Bosch BMG*250 gyro

Random Walk	Bias Instability	Correlation Time	Maximum Drift
$0.007^o/\sqrt{s}$	$10^o/h$	$6.350s$	$3^o/s$

This gyro comes with no Allan Variance plot and so it must be modeled with the information above only. The correlation time does not come in the data-sheet but was calculated as:

$$CT = \left(\frac{ARW}{BI}\right)^2 \tag{89}$$

Where ARW and BI are Angular Random Walk and Bias Instability respectively. As mentioned previously, the challenge for proper gyro modeling is choosing the correct variance for the noise generators in the model displayed in Figure 26. These variances depend on simulation step time, since the same variance produces a higher noise density for a smaller step. This is one of the reasons the simulation step is conveniently set to 1 second, since it avoids this problem. Different time steps would need a time correction in the noise variances. The Bias Instability needs to be converted to degrees per second: $\sigma_{GM} = \sigma_{BI} = 0.0028^o/s$. The standard deviation corresponding to the Angular Random Walk is set to $\sigma_{ARW} = 0.007^o/\sqrt{s}$. The Root Allan variance plot was calculated and is represented in Figure 28.

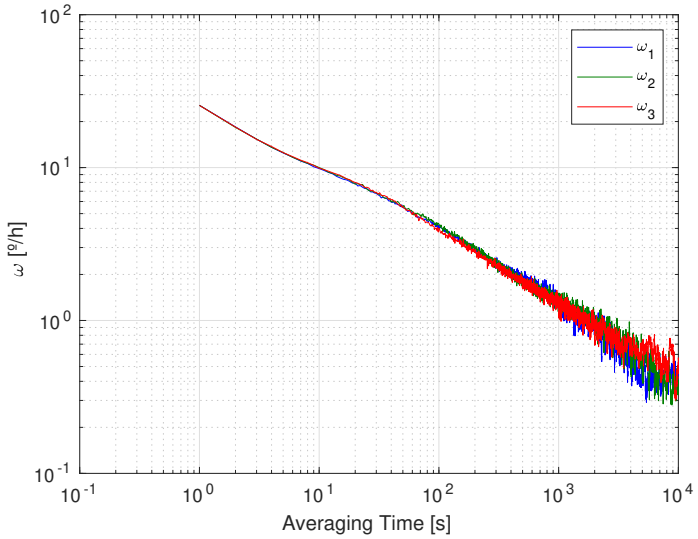


Figure 28: Root Allan Variance of $BMG250$ from Simulink Data

7.1.2 Precise Gyro

The precision gyro that is expected to be onboard the *Infante* is the $STIMM210$ gyroscope. The relevant information of the precise gyro was taken from the datasheet [38]:

Table 4: Noise characteristics of the $STIMM210$ gyro

Random Walk	Bias Instability	Correlation Time	Maximum Drift
$0.0025^o/\sqrt{s}$	$0.3^o/h$	$900s$	$0.07^o/s$

which was used as input on the *Simulink* model, like in the coarse gyroscope. This gyro comes with an Allan Deviation plot in the datasheet, so data from the simulation can be extracted to generate an Allan Deviation plot for comparison. In Figure 29 is the Allan Deviation from the datasheet and in Figure 30 is the one generated by the gyro model. It can be seen that they are a close match, validating the model. The datsheet's plot is smoother only because it contains more data, as a smaller step size was considered.

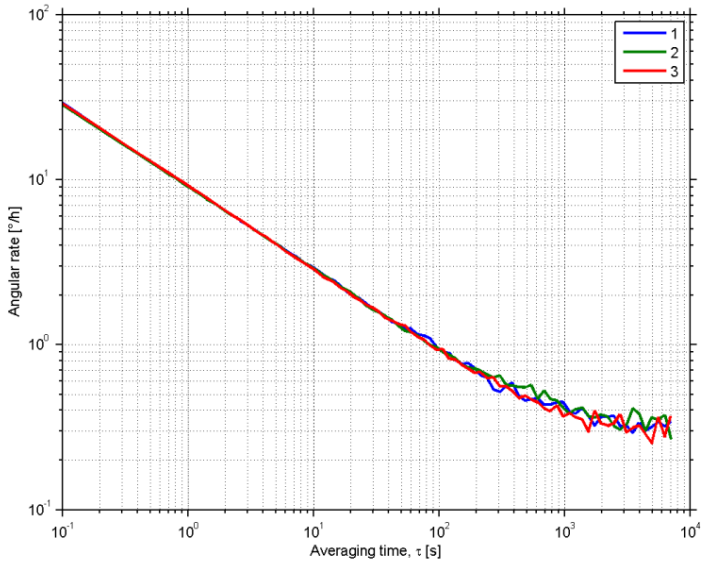


Figure 29: Root Allan Variance taken from Datasheet [38]

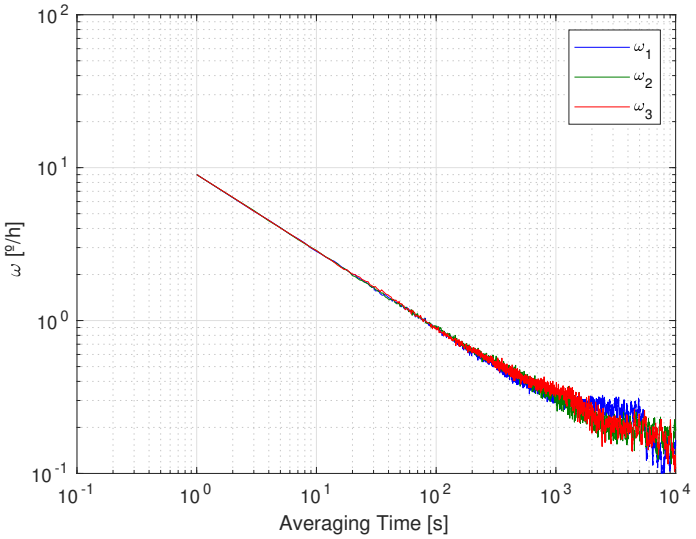


Figure 30: Root Allan Variance generated from Simulink Data

7.2 Coarse Sun Sensor

The Coarse Sun sensor is a collection of six photo-resistors. each mounted on a face of the spacecraft. By comparing the electrical current across each resistor, it is possible to estimate the solar radiation on each face.

Consider two opposite faces along the spacecraft's $x-$axis, whose normal vectors are $\boldsymbol{n}_1 = [1, 0, 0]$ and $\boldsymbol{n}_2 = [-1, 0, 0]$. Knowing the photo-resistor gain K_s, it is possible to estimate the solar radiation in the $x - direction$ $\hat{\boldsymbol{s}}_x$ by measuring the current difference $I_1 - I_2$.

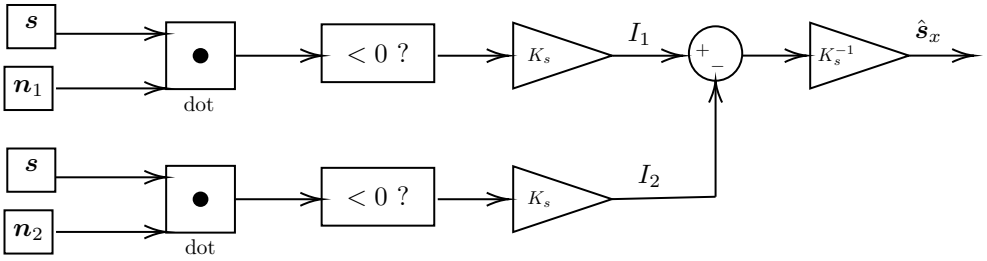


Figure 31: Block Diagram of the Sun Sensor

Figure 31 illustrates the block diagram for the $x-axis$ of the sun sensor. Similar blocks apply for the $y-$ and $z-$ axis. Each axis component is added along with white noise and a normalization procedure. The inverse algorithm of the albedo model discussed previously is used to compensate for Earth's reflection. Here is introduced the greatest source of error, where the algorithm is forced to overestimate the albedo.

7.3 Magnetometer

As mentioned previously, there are two magnetometers in *Infante*: an inside magnetometer which gives magnetic field information for detumbling, and an outside precise magnetometer that is used for navigation. Both are modeled the same way but with corresponding values taken from the data-sheet. The measurement is considered to be the real vector affected by white noise and a bounded first order Gauss-Markov process bias, in a similar way to the gyroscopes mentioned previously. The respective block diagram is illustrated in Figure 32.

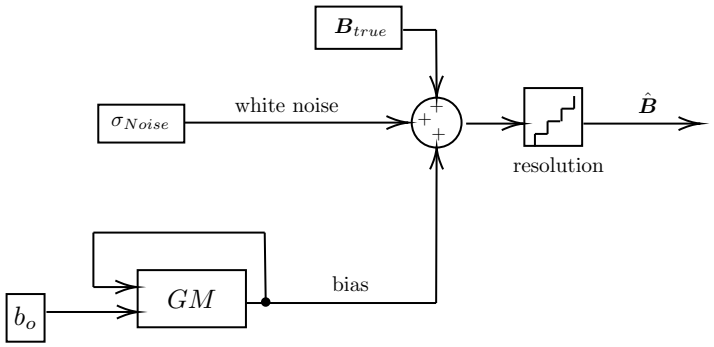


Figure 32: Block Diagram of a Magnetometer

The magnetometers that are expected to be onboard the *Infante* are the ST IIS2MDC magnetometer (low precision) and the Bosch BMM150. The relevant aspects of the corresponding datasheets were gathered below in Table[5]:

Table 5: Information regarding the used magnetometers [39] [40]

	ST IIS2MDC	Bosch BMM150
Sensitivity/ Resolution	0.15 μT	0.3 μT
RMS Noise	0.3 μT (3σ)	0.6 μT (1σ)
Sensor offset / Accuracy (3σ)	6 μT	2.5^o $(B = 30\mu T)$

The accuracy of the Bosch BMM150 can be converted to micro-Tesla by assuming the error equal to $30\,\mu T \cdot \sin(2.5^o) \approx 1.3\,\mu T$. The assumed correlation time for the bias is ten minutes.

7.4 Star Tracker

Star Trackers are digital cameras that follow multiple stars in the sky, match them to an internal catalog and compute the tracker's attitude with respect to a celestial frame. A typical tracker has an accuracy of a few arcseconds in the boresight (axis of the camera), losing precision as stars move to the limit of the field of view. Unlike the Sun Sensor and the Magnetometer, this sensor has the ability of computing the attitude without other information, returning the correspondent attitude quaternion. It has the inconvenient of consuming a considerable amount of power and not working if the Sun is in its field of view. It is typical to find a baffle in the camera, protecting it from stray lights while narrowing the field of view angle α_{FOV}, as seen on Figure 33. The Star Tracker will be equipped on top of the *Infante* so it is expected that the Sun enters its field of view during daytime, near the equator.

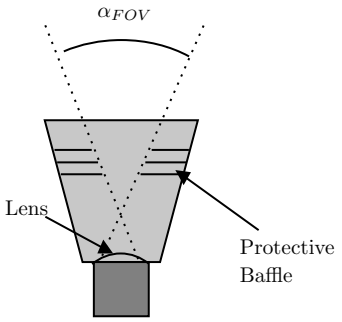


Figure 33: Angle of Exclusion in a Star Tracker Camera

The proposed equipment is the Leuven Star Tracker, a compact tracker for CubeSats. The relevant information is presented below in Table [6]:

Table 6: Information regarding the KU Leuven Star Tracker [41]

Cross Boresight Error (1σ)	Around Boresight Error (1σ)	Sun Rejection Angle
2 arcsec	10 arcsec	40^o

Evaluating if the Sun is in the field of view can be done by analyzing the dot product between the sun direction and the camera's boresight direction. The camera is pointing upwards in the direction of $-\boldsymbol{o}_3$ and the field of view angle is $\alpha_{FOV} = 40^o$, so the Star Tracker can't be used if:

$$-\boldsymbol{s} \cdot \boldsymbol{o}_3 \geq \cos(20^o) \tag{90}$$

Modeling a Star Catalog and multiple star tracking is a task too complicated for the benefit of this study, so it is assumed that the true error lies between the two indicated values. Also, Star Trackers suffer from not only random noise, but systematic errors, which are due to: image pixelization, inhomogeneity of components, different sensivity to each star's emission spectre, light aberration and other optical effects [42].

The chosen model is very simple, but sufficient: a walking error under the form of rotation is added to the true quaternion, followed by white noise. The addition of noise violates the quaternion norm so it needs to be normalized. Finally, a resolution effect is added.

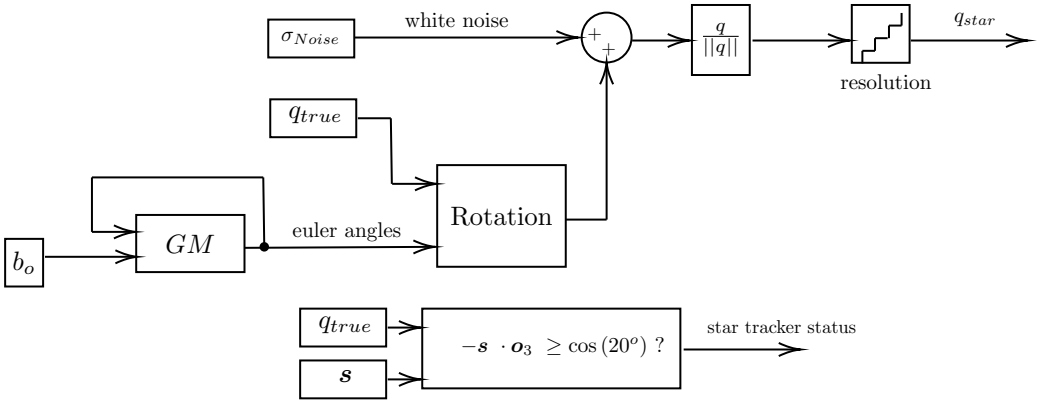


Figure 34: Simulink Model of the Star Tracker

The systematic error is modeled as a first order Gauss-Markov process like the bias in the gyroscopes mentioned previously. This error is essentially a bounded random walk of error Euler Angles, with a deviation of $\sigma = 8$ arcseconds, a mean error near the worst case scenario. An arbitrary correlation time of 1000 s was chosen, since it is considered that the systematic error moves very slowly. The resolution of the Star Tracker is not disclosed but is assumed to be near 1 arcseconds in one axis. Translating this in quaternion resolution:

$$\sin(\phi/2) = \sin\left(\frac{1}{60 \cdot 2}\right) = 1.45 \times 10^{-4} \tag{91}$$

Therefore, the resolution considered is 10^{-4} in each of the quaternion's components. The white noise added has a standard deviation of this value.

7.5 Magnetorquers

Magnetorquers are controllable electromagnets which create a magnetic dipole moment $\boldsymbol{m}$. This magnetic dipole interferes with Earth's magnetic field $\boldsymbol{B}$ creating a magnetic torque:

$$\boldsymbol{L}_{mag} = \boldsymbol{m} \times \boldsymbol{B} \tag{92}$$

This magnetic dipole is created by producing a current over a coil wrapped in a ferromagnetic core, where $m = kNIA$, where k is an dimensionless constant, N is the number of loops, I is current and A is the loop area. It is worth mentioning that it's not possible to produce torque in the direction of the magnetic field with magnetorquers, however, full control is usually available over the course of an orbit. Care is needed when evaluating magnetic control and simulations should be longer than 24h to ensure there is no unfavorable magnetic field geometry at some point [9].

The magnetorquers expected to be used in *Infante* shall be manufactured by *Tekever*. The only information that is available at the moment is that they will be developed to ensure a dipole moment of 0.52 Am^2 in the $x-$ and $y-$ body axis and a dipole moment of 1 Am^2 on the $z-$ axis. These magnetorquers are assumed to be completely controllable and linear, so they are simply modeled using equation (92).

7.6 Reaction Wheels

The Reaction Wheel Assembly is the fundamental pointing instrument of *Infante*, so this is a topic requiring some attention, since the control error will directly depend on the error model of the wheels. A major problem with the choice of a simulation step time of only one second is the difficulty in representing the reaction wheels' error dynamics, which spin at rates of thousands of rpm. The model must accomplish acceptable performance while ignoring fast phenomenon like torque ripple.

The circuit of a brushless electric DC motor is adequately modeled by the following equation:

$$U_{in} = RI + L\frac{dI}{dt} - K_e\omega \tag{93}$$

where K_e is the back electromotive force constant and U_{in} is the voltage applied to the motor. The inductance effects are neglected since the simulation step of one second overcomes this effect, usually in the range of milliseconds. The output torque is a function of the motor's current and speed:

$$L = L_{DC} - L_d = K_m I - C_c sign(\omega) - C_d\omega \tag{94}$$

where K_m is the motor's mechanical gain and C_c and C_d are the Coulomb and Viscous drag coefficients respectively. The wheel's dynamic equation is simply:

$$L = \dot{H} = J\dot{\omega} \tag{95}$$

The motor's internal Hall sensor measures the circuit's current, feeding this information to an internal PID controller, where the current torque is estimated. This fact is simplified, as it is assumed that the controller obtains perfect information about torque. The block diagram of this reaction wheel model can be seen on Figure 35.

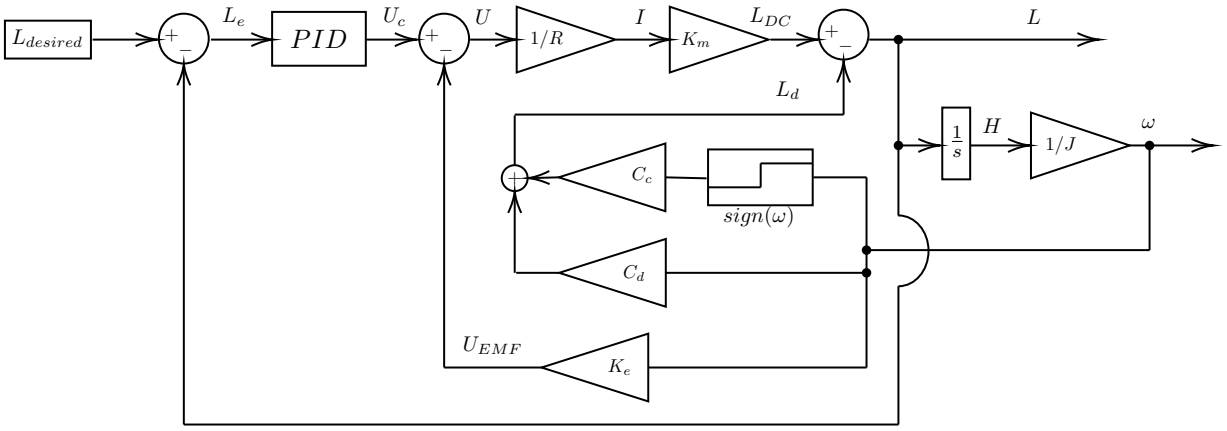


Figure 35: Block Diagram of the Reaction Wheels

The reaction wheel used in this study is a wheel developed by *Tekever*, based on a *MAXON EC*20 351005 *Flat motor* with Hall Sensor [43], attached to a high inertia wheel. Relevant information about this reaction wheel is gathered below on Table [7.6].

Table 7: Reaction Wheel Specifications [43]

Nominal Voltage	6 V
No load speed	9350 rpm
Max Continuous Torque	7.59 mNm
Max Continuous Current	1.31 A
Resistance	2.05 Ω
Inductance	0.189 mH
Mechanical Time Constant	30.3 ms
Torque Constant	5.88 mNm/A
Electric Constant	5.89 mV/rads
Wheel Inertia	861.37 gcm^2
Max Momentum (9000 rpm)	83.9 mNms

The mechanical time constant and inductance are small enough to be neglected, the reason why these were ignored in this model. The datasheet lacks information regarding the drag coefficients C_c and C_d, and controller gains. The overall drag can be estimated taking in account the speed achieved with no load, summing the torques acting on the motor's shaft:

$$C_d\,\omega_{max} + C_c = K_m\left(\frac{6\,V - K_e\,\omega_{max}}{R}\right) \approx 0.50\,mNm \tag{96}$$

Assuming a $20\% - 80\%$ distribution of the Coulomb and Viscous friction at this speed, the drag coefficients are estimated:

$$C_c = 0.1\,mNm \qquad C_d = 4.085 \times 10^{-4}\,mNms\,rad^{-1} \tag{97}$$

The PID gains are hand tuned until a satisfactory performance is obtained. It is important to take in account the poor wheel performance at low and high speeds, specially when null velocity points are crossed. It is also important to mention that the integral part of the PID needs anti-windup to ensure fast response times when the wheel is saturated and the torque changes direction.

To test the model a random desired torque trajectory was generated, in a way that the wheel exhibits null velocity points, direction changes and saturation. Results are displayed on Figures 36 and 37.

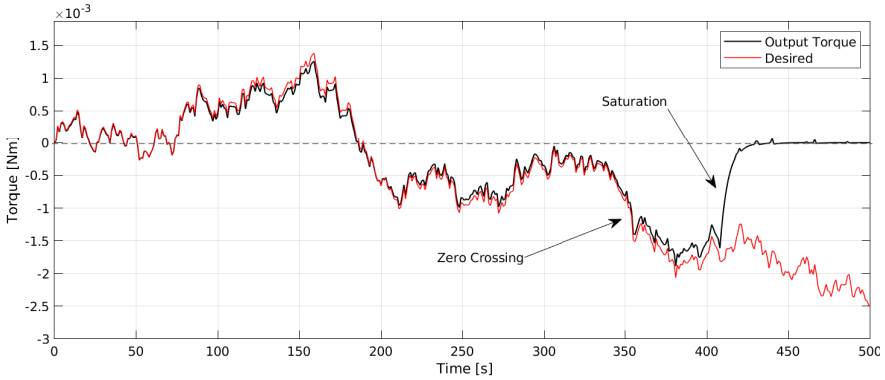


Figure 36: Torque Trajectory of Modeled Reaction Wheel

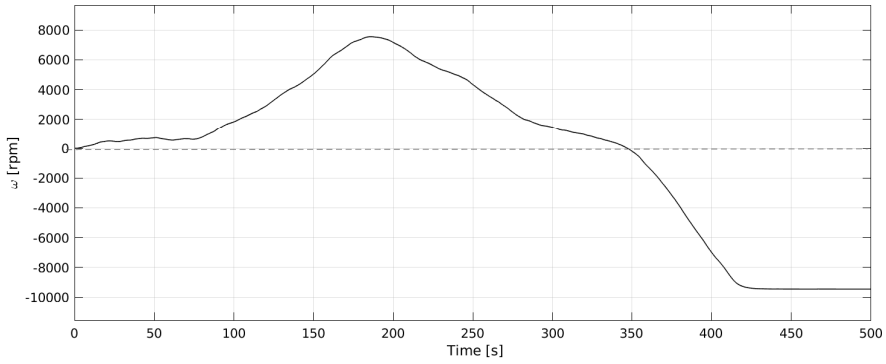


Figure 37: Speed Trajectory of Modeled Reaction Wheel

7.6.1 Reaction Wheel Assemblies

The most common reaction wheel assembly is a pyramid configuration of four wheels, each wheel making a 30^o angle with the baseline, like the one represented in Figure 38. The body vectors of each wheel are described by:

$$\boldsymbol{w}_1 = \begin{bmatrix} \frac{\sqrt{3}}{2} \\ 0 \\ 0.5 \end{bmatrix}, \quad \boldsymbol{w}_2 = \begin{bmatrix} -\frac{\sqrt{3}}{2} \\ 0 \\ 0.5 \end{bmatrix}, \quad \boldsymbol{w}_3 = \begin{bmatrix} 0 \\ \frac{\sqrt{3}}{2} \\ 0.5 \end{bmatrix}, \quad \boldsymbol{w}_4 = \begin{bmatrix} 0 \\ -\frac{\sqrt{3}}{2} \\ 0.5 \end{bmatrix} \tag{98}$$

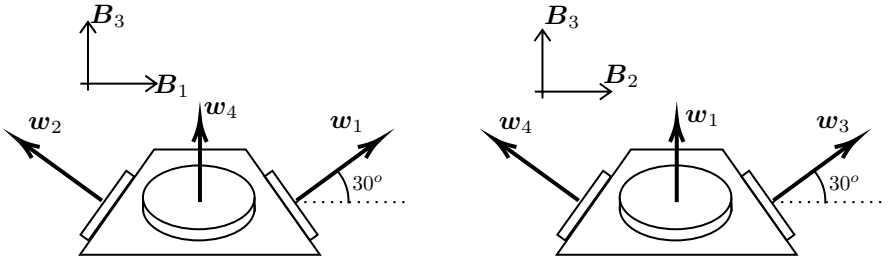


Figure 38: Reaction Wheel Pyramid Configuration

Let $D = [\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3, \boldsymbol{w}_4]$ be the wheel frame matrix. The overall torque in the body frame is calculated:

$$\boldsymbol{L}_B^w = D\boldsymbol{L}_w^w\,, \qquad \boldsymbol{L}_w^w = \left[L_1, L_2, L_3, L_4\right]^T \tag{99}$$

The inverse problem of calculating how torque is distributed by the wheels, for a given desired body torque, has multiple solutions. Since the D matrix is not square but rather 3×4, it is said that the reaction wheel assembly has a one dimensional null-space: any torque configuration in the wheel's null-space equals zero torque in the body frame. The null space of a pyramid configuration such as the one expected to be used in *Infante* is:

$$\boldsymbol{n}_4 = k \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \tag{100}$$

Where k is a free scalar. The solution of interest is the one that minimizes any torque in the null space, since this has no effect overall. It is shown in [9] that this distribution is given by the *pseudoinverse* of D, denoted by D^p. The *pseudoinverse* is given by:

$$D^p = D^T(DD^T)^{-1} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0.5 \\ -\frac{1}{\sqrt{3}} & 0 & 0.5 \\ 0 & \frac{1}{\sqrt{3}} & 0.5 \\ 0 & -\frac{1}{\sqrt{3}} & 0.5 \end{bmatrix} \tag{101}$$

And the ideal torque distribution is done by:

$$\boldsymbol{L}_w^w = D^p\boldsymbol{L}_B^w \tag{102}$$

8 Attitude Estimation and Filtering Algorithms

Multiplicative Extended Kalman Filter (MEKF)

The use of a Kalman Filter allows us to get a better estimate of the current attitude state by combining multiple sensor measurements with a dynamic model of the system. It is a technique that has been widely employed with success on spacecraft navigation for many decades. This chapter focuses on the development of a Multiplicative Extended Kalman Filter (MEKF), a special formulation designed to work with the unit quaternion parameterization and information from the spacecraft's gyros. It is expected that the reader is familiar with the basic concepts of a Kalman Filter. This filter model was developed based on the works of [13], [44] and [9].

As discussed previously, the unit quaternion has many advantages in representing attitude but the norm constraint makes it impossible to implement a linear additive update and its components are not independent from each other. Another disadvantage is that since the quaternion is a four parameter representation, the 4×4 covariance matrix becomes singular because the information contained in a quaternion is redundant; a quaternion can be reduced to a three parameter representation. To overcome these problems this approach regards the filter as a three component error vector, while still representing the global attitude by a quaternion.

8.0.1 Model

The following truth model is defined:

$$\bar{q}_{true} = \delta\bar{q}(\boldsymbol{a}) \otimes \hat{\bar{q}} \tag{103}$$

Where $\bar{q}_{true}$ is the true attitude quaternion, $\hat{\bar{q}}$ is the current best estimate and $\delta\bar{q}(\boldsymbol{a})$ is the the quaternion representing the estimation error rotation, represented in the spacecraft body frame. The estimation error quaternion is also represented by a three parameter representation $\boldsymbol{a}$, which is what is used in the filter equations. The task of the filter is to estimate the current error $\hat{\boldsymbol{a}}$ and perform the following multiplicative update:

$$\hat{\bar{q}}^{+} = \delta\bar{q}(\hat{\boldsymbol{a}}) \otimes \hat{\bar{q}}^{-} \tag{104}$$

Where $\hat{q}^{+}$ is the *a posteriori* estimate after a measurement and $\hat{q}^{-}$ is the *a priori* estimate. Many choices for the parameterization of the error '$\boldsymbol{a}$' are possible but the quantity of two times the Gibbs Vector was chosen, since it provides numerical stability and it has been well covered by literature. The used Gibbs Vector representation is given by:

$$\boldsymbol{a}_g = 2\boldsymbol{g} \tag{105}$$

$$\delta\bar{q}(\boldsymbol{a}_g) = \frac{1}{\sqrt{4 + \|\boldsymbol{a}_g\|^2}} \begin{bmatrix} \boldsymbol{a}_g \\ 2 \end{bmatrix} \tag{106}$$

A further advantage of the this Gibbs vector parameterization is that the update can be performed in few steps by the following equations:

$$\boldsymbol{\rho} = \begin{bmatrix} \boldsymbol{a}_g \\ 2 \end{bmatrix} \otimes \hat{\bar{q}}^{-} \tag{107}$$

$$\hat{\bar{q}}^{+} = \boldsymbol{\rho}/|\boldsymbol{\rho}| \tag{108}$$

The angular speed is measured through the use of gyros, which provide noisy, biased information. In order for the bias to be estimated and compensated for, the gyro and angular speed are modeled in following way:

$$\boldsymbol{\omega}(t) = \hat{\boldsymbol{\omega}}(t) - \boldsymbol{b}(t) - \boldsymbol{\eta}_1(t) \tag{109}$$

$$\dot{\boldsymbol{b}}(t) = \boldsymbol{\eta}_2(t) \tag{110}$$

where $\boldsymbol{\eta}_1(t)$ and $\boldsymbol{\eta}_2(t)$ are zero mean white noise processes of variances σ_u^2 and σ_v^2 , $\hat{\boldsymbol{\omega}}(t)$ is the gyro output and $\boldsymbol{b}(t)$ is the gyro bias.

The six component state vector to be estimated is comprised of the error parameterization and the gyro bias, with the state space being described by the following:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{a}_g} \\ \dot{\boldsymbol{b}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}(\boldsymbol{a}_g, \boldsymbol{\omega}) \\ \boldsymbol{\eta}_2(t) \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} \boldsymbol{a}_g \\ \boldsymbol{b} \end{bmatrix} \tag{111}$$

The filtering algorithm can be divided in the following major steps:

- Propagation
- Update
- Reset

8.0.2 Propagation

Since the expected value of the gyro noise process $\boldsymbol{\eta}_1(t)$ is zero, the current angular velocity is computed by the equation:

$$\hat{\boldsymbol{\omega}}_k = \hat{\boldsymbol{\omega}}_{gyro\,k} - \boldsymbol{b}_k \tag{112}$$

Each propagation is made at the filter's sampling time and the spacecraft's attitude dynamics can be considered low enough to make the assumption that the direction of $\boldsymbol{\omega}$ is constant over this time interval. A rotation angle vector can then be computed by:

$$\Delta\boldsymbol{\theta} = \int_t^{t+\Delta t} \boldsymbol{\omega}(\tau) d\tau \approx \hat{\boldsymbol{\omega}}_k \Delta t \tag{113}$$

The solution for the attitude propagation of $\hat{\bar{q}}$ is given by the equation:

$$\hat{\bar{q}}_k^- = M(\Delta\boldsymbol{\theta})\hat{\bar{q}}_{k-1}^+ \tag{114}$$

where the matrix $M(\Delta\boldsymbol{\theta})$ is given by:

$$M(\Delta\boldsymbol{\theta}) = \cos(||\Delta\boldsymbol{\theta}||/2) I_{4\times4} + \frac{\sin(||\Delta\boldsymbol{\theta}||/2)}{||\Delta\boldsymbol{\theta}||}\Omega(\Delta\boldsymbol{\theta}) \tag{115}$$

$$\Omega(\Delta\boldsymbol{\theta}) = \begin{bmatrix} 0 & \Delta\theta_3 & -\Delta\theta_2 & \Delta\theta_1 \\ -\Delta\theta_3 & 0 & \Delta\theta_1 & \Delta\theta_2 \\ \Delta\theta_2 & -\Delta\theta_1 & 0 & \Delta\theta_3 \\ -\Delta\theta_1 & -\Delta\theta_2 & -\Delta\theta_3 & 0 \end{bmatrix} \tag{116}$$

During propagation the obtained estimate for the attitude is assumed to be currently the best, so the error vector $\boldsymbol{a}_g$ is kept at zero. Similarly, the bias $\boldsymbol{b}$ has a derivative that is a zero mean white noise process so it is held at its current value. However, the certainty level to which we know these values decreases with time, so the covariance matrix must be propagated:

$$P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + Q_k \tag{117}$$

A closed-form expression for the state propagation matrix Φ_k exists and can be computed by the following:

$$\Phi = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix} \tag{118}$$

$$\Phi_{11} = I_3 - [\hat{\boldsymbol{\omega}}\times]\frac{\sin(||\hat{\boldsymbol{\omega}}||\Delta t)}{||\hat{\boldsymbol{\omega}}||} + [\hat{\boldsymbol{\omega}}\times]^2\frac{1-\cos(||\hat{\boldsymbol{\omega}}||\Delta t)}{||\hat{\boldsymbol{\omega}}||^2} \tag{119}$$

$$\Phi_{12} = [\hat{\boldsymbol{\omega}}\times]\frac{1-\cos(||\hat{\boldsymbol{\omega}}||\Delta t)}{||\hat{\boldsymbol{\omega}}||^2} - I_3\Delta t - [\hat{\boldsymbol{\omega}}\times]^2\frac{||\hat{\boldsymbol{\omega}}||\Delta t - \sin(||\hat{\boldsymbol{\omega}}||\Delta t)}{||\hat{\boldsymbol{\omega}}||^3} \tag{120}$$

$$\Phi_{21} = 0_3 \qquad \Phi_{22} = I_3 \tag{121}$$

The noise covariance matrix Q can be approximated by the matrix below if the sampling rate is faster than Nyquist's limit, which is the case:

$$Q_k \approx \begin{bmatrix} \left(\sigma_v^2\Delta t + \frac{1}{3}\sigma_u^2\Delta t^3\right) I_3 & -\left(\frac{1}{2}\sigma_u^2\Delta t^2\right) I_3 \\ -\left(\frac{1}{2}\sigma_u^2\Delta t^2\right) I_3 & \left(\sigma_u^2\Delta t\right) I_3 \end{bmatrix} \tag{122}$$

8.0.3 Update

A filter update occurs when a measurement is performed, which can be presented to the filter in two ways: a quaternion measurement from the Star Tracker or a vector measurement from the Magnetometer or the Coarse Sun Sensor. In case of a quaternion measurement it is possible (and more convenient) to transform the measurement directly into the error parameterization $\boldsymbol{a}_g$ by comparing it to the current attitude estimate $\hat{\bar{q}}_k$. In this case, the sensitivity matrix is simply equal to the identity matrix in the attitude error part, and null in the gyro bias part since this part is not observable:

$$\boldsymbol{a}_{star} = 2\,\delta\bar{\boldsymbol{q}}_{1:3}/\delta\bar{q}_4\,, \qquad \delta\bar{q} = \bar{q}_{star}\otimes\hat{\bar{q}}^{-1} \tag{123}$$

$$H_k = \begin{bmatrix} I_3 & 0_3 \end{bmatrix}, \qquad \boldsymbol{z} = \boldsymbol{a}_{star} \tag{124}$$

This method of observation comes with the small inconvenient that the Star Tracker's error covariance matrix R needs to be defined as a 3×3 matrix representative of attitude errors in the body frame.

In case of a vector measurement the input of the filter becomes the vector difference between the measured and predicted state $\Delta\boldsymbol{z}$:

$$\Delta\boldsymbol{z} = \boldsymbol{z} - \hat{\boldsymbol{z}} \tag{125}$$

The sensitivity matrix in this case is given by the cross matrix $[\hat{\boldsymbol{z}}\times]$ for the attitude error part. This transforms the vector difference $\Delta\boldsymbol{z}$ into a vector $\hat{\boldsymbol{z}}\times\Delta\boldsymbol{z}$ representing the rotation error $\boldsymbol{e}\sin\phi$, shown in the right by Figure 39. It is shown in [9] that this vector is equal to the error parameterization $\boldsymbol{a}_g$ for a first order approximation.

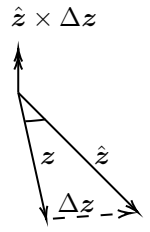


Figure 39: Cross product illustration

$$H_k = \begin{bmatrix} [\hat{\boldsymbol{z}}\times] & 0_3 \end{bmatrix} \tag{126}$$

It is worth noticing that a unit vector is an insufficient measure of the attitude and because of this, in the case of only one vector observation, the attitude estimation will spin away in the vector axis, at a rate depending on the gyro's bias instability. In the common case of multiple observations the sensitivity matrix becomes a $3m \times 6$ matrix, where m is the number of observations, and the input vector becomes a $3m$ concatenation of the several results. Presented below is the case where the Star Tracker, the Sun Sensor and the Magnetometer are in use:

$$\boldsymbol{z}_k = \begin{bmatrix} \boldsymbol{a}_{star} \\ \Delta \boldsymbol{z}_{sun} \\ \Delta \boldsymbol{z}_{magnt} \end{bmatrix} \qquad H_k = \begin{bmatrix} I_3 & 0_3 \\ [\hat{\boldsymbol{z}}_{sun}\times] & 0_3 \\ [\hat{\boldsymbol{z}}_{magnt}\times] & 0_3 \end{bmatrix} \tag{127}$$

Obtaining the filtering gain K follows the same procedure as the standard Kalman Filter, which produces the optimal gain:

$$K_k = P_k^- H_k^T \left[H_k P_k^- H_k^T + R \right]^{-1} \tag{128}$$

where R is the measurement covariance matrix, a $3m \times 3m$ diagonal matrix of the variances of the sensors being used. The covariance matrix P is updated through Joseph's formula to ensure numerical stability:

$$P_k^+ = \left[I_6 - K_k H_k \right] P_k^- \left[I_6 - K_k H_k \right]^T + K_k R K_k^T \tag{129}$$

The state update equation is given simply by:

$$\boldsymbol{x}_k^+ = \boldsymbol{x}_k^- + K_k \boldsymbol{z}_k \tag{130}$$

Finally, the global quaternion parameterization is updated through equations (107) and (108)

8.0.4 Reset

After the global attitude update, the error part of the state needs to be reset to zero, since the filter has arrived at the best estimate. This is simply done by:

$$\boldsymbol{x}_k^+(1,2,3) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{131}$$

8.0.5 Filter Convergence

When the filter is turned on and when gyros are commuted, it may take some time for the filter to converge into accurate results. Large estimation errors, even if temporary, can lead to undesirable and potentially dangerous control actions if some precautions are not taken. Upon initialization, gyro biases are unknown and can be significantly different than zero. It is necessary to turn on a sufficient number of instruments, which completely describe attitude, and no control action must be taken until results are stable. The algorithm can also commute between the coarse and the precise gyros, so the respective biases need to be stored and interchangeable. Not changing biases upon changing gyros will most likely cause a spin in the estimated attitude that may only be recovered after some time. Storing and switching the last covariance matrix before commuting gyros also helps achieve better results.

8.0.6 Application

An example of filter performance using multiple sensors can be seen on Figures 40, 41 and 42. In the first figure is plotted the true quaternion $\bar{q}_{GCI-B}$, translating the attitude of the body regarding the *GCI* frame, and the filter estimated $\hat{\bar{q}}_{GCI-B}$, obtained from the sensors modeled in the previous section. The next figure represents the rotation error in Euler Angles, to better understand the magnitude of the error. The last figure is the data logging taken from the simulation, showing which sensors are active.

It is visible that filter convergence is not instantaneous, it takes a couple seconds to find a close solution. At around 250 seconds of simulation time the sun sensor is unavailable due to eclipse, attitude knowledge comes only from the magnetometer, which is insufficient. To hold the attitude knowledge the precision gyro is turned on. Even tough the attitude information isn't correct, due to the magnetometer and sun sensor's biases, the precision gyro doesn't let the accuracy degrade until the sun sensor is available again around the 2500 second mark. At 3000s the magnetometer is deliberately turned off , showing that the accuracy degrades slowly with the coarse gyro until the star tracker is turned on multiple times to compensate for this drift. At 6000 seconds the star tracker and precision gyro are turned on to give a very precise attitude estimate, whose maximum error is around 0.06^o, or 3.6 arcseconds. Using the filter with the star tracker and the precision gyro provides better accuracy than using the star tracker alone. Turning on the biased sun sensor when the star tracker is used has no negative influence, since the filter rejects the sun sensor error automatically.

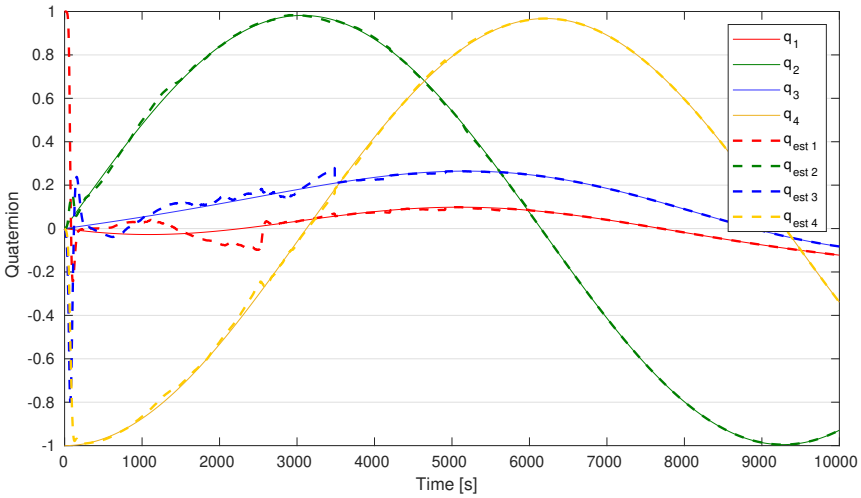


Figure 40: Kalman Filter Quaternion

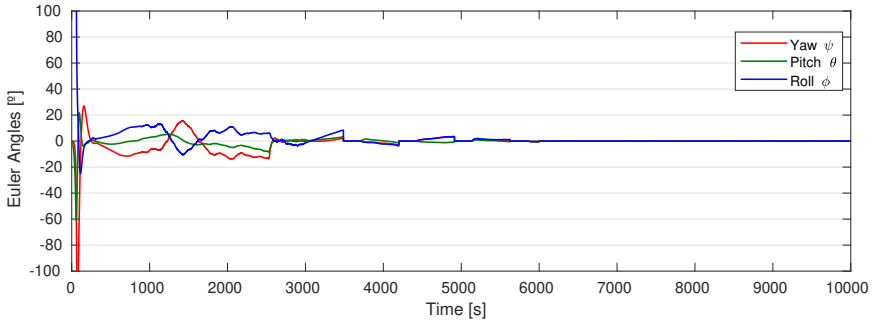


Figure 41: Kalman Filter Error in Euler Angles

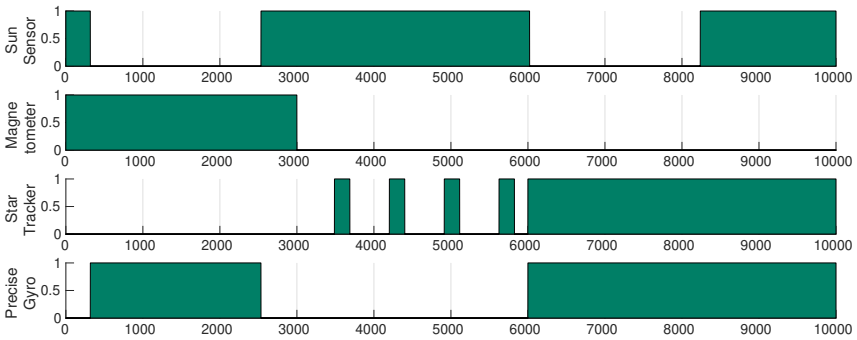


Figure 42: Sensors Used in the Kalman Filter

9 Decentralized Architecture

Traditionally, the internal architecture of a modern satellite with attitude control would look something like the scheme represented in Figure 43. The attitude sensors give information to the ADS (Attitude Determination System), where the Kalman Filter estimates the current attitude. The ADS also gives commands to the sensors, regarding the use of different modes, and is able to turn them on or off to save energy. The ADS communicates with the ACS (Attitude Control System), by giving instructions about the current attitude, which are needed to compute the control actions, while the ACS can return information about actuators' status. Both ADS and ACS form a larger system, the ADCS (Attitude Determination and Control System), which is often only one construction comprising the two subsystems. The ADCS communicates with the OBC (On-Board Computer), the central authority of the satellite, which gives the ADCS instructions about the desired attitude and its precision requirements. It also gives valuable information regarding the satellite's current position, taken from orbit propagation and GPS measurements, which are needed to compute the local frame, position of the Sun and expected magnetic field.

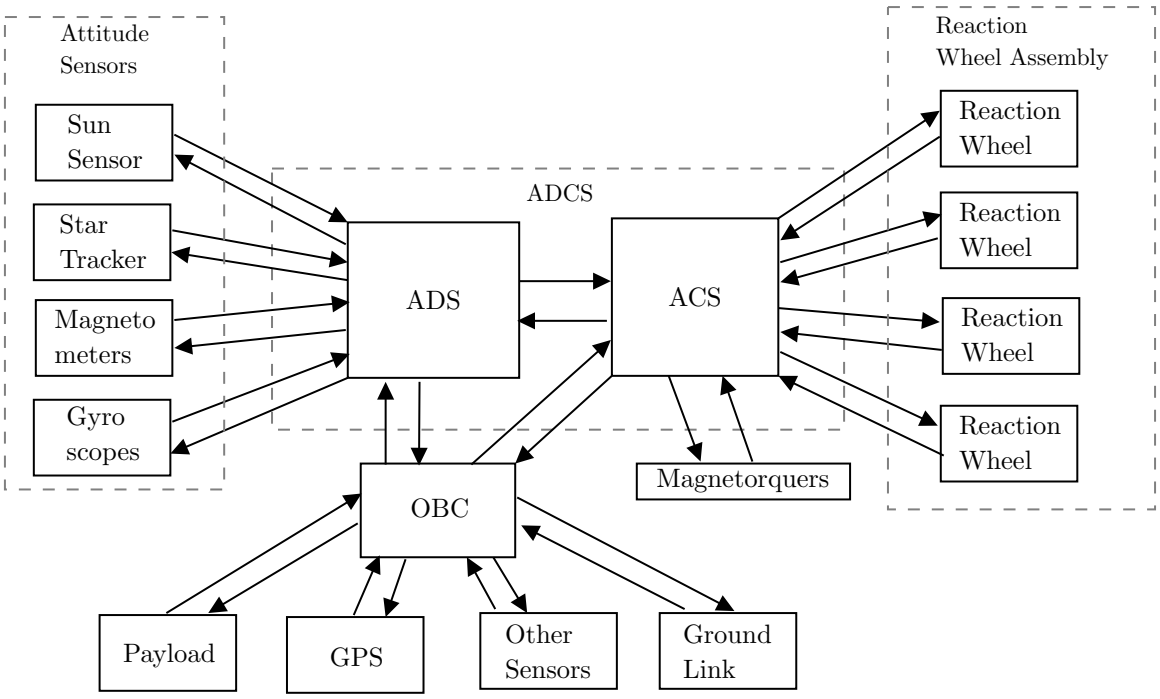


Figure 43: Example of a common satellite architecture

The ACS handles torque or speed commands to the reaction wheels, and voltage commands to the magnetorquers. It is assumed that each wheel has an internal PID controller, as modeled previously in the Actuators section, although torque control can be made within the ACS. Reaction wheels usually feedback status, wheel velocity/momentum, and sometimes the current output torque to the ACS, while the magnetorquers can feedback its status and electrical current, although the latter can be made with no feedback at all. Quite often, manufacturers sell reaction wheels in an assembly that is locally controlled by a different subsystem, which simply receives torque or speed commands from the ACS and then controls all four wheels simultaneously. This is another example of a typical control architecture, displayed in Figure 44.

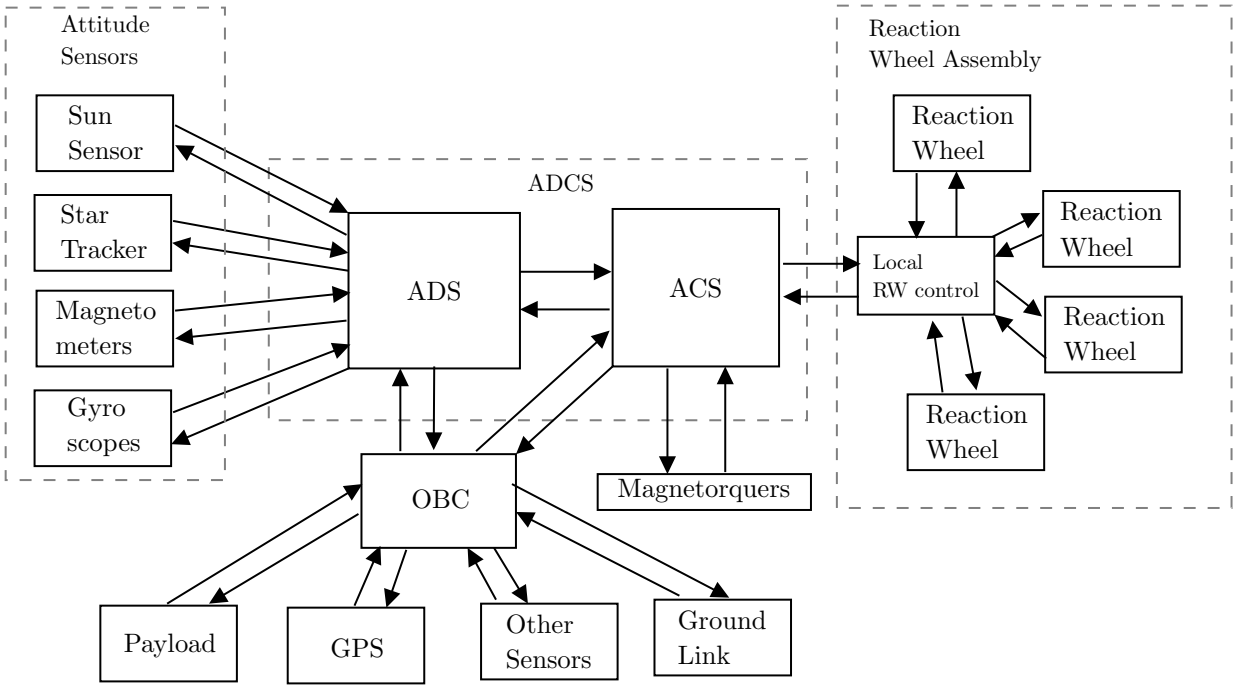


Figure 44: Another Example of a Common Satellite Architecture

This thesis proposes a decentralized architecture, where the ACS is partitioned and migrated to the reaction wheels and magnetorquers, as illustrated in Figure 45. In this case, the ADS gives attitude requirements to the several ACSs and only expects feedback from the wheel's speed and magnetorquer's status, since this is the only information needed by the ADS, to allow the desaturation of the reaction wheels and the use of the magnetometer. The ACSs of the reaction wheels work independently, without any knowledge of the other wheels's directions, actions and status, under the assumption that each component is doing its job of minimizing the control error, and that the overall system is stable as long as this is true. The two advantages of this architecture are increased modularity and ACS redundancy.

The output of the ADS is built as an "attitude bus", that is divided and routed to the several ACSs of the reaction wheels, all using the same communication protocol. The reaction wheels become fully independent and the only information that is known to them is: their own direction $\boldsymbol{w}_l$, an estimate of the inertia matrix of the spacecraft J, two control parameters (natural frequency ω_n and damping coefficient ζ), and the current and desired attitudes. The local ACS does not have information regarding the direction of the other wheels because this is not needed to achieve a stable system. With this independence, modifications can be made to the reaction wheels without having to fully re-program the ADCS.

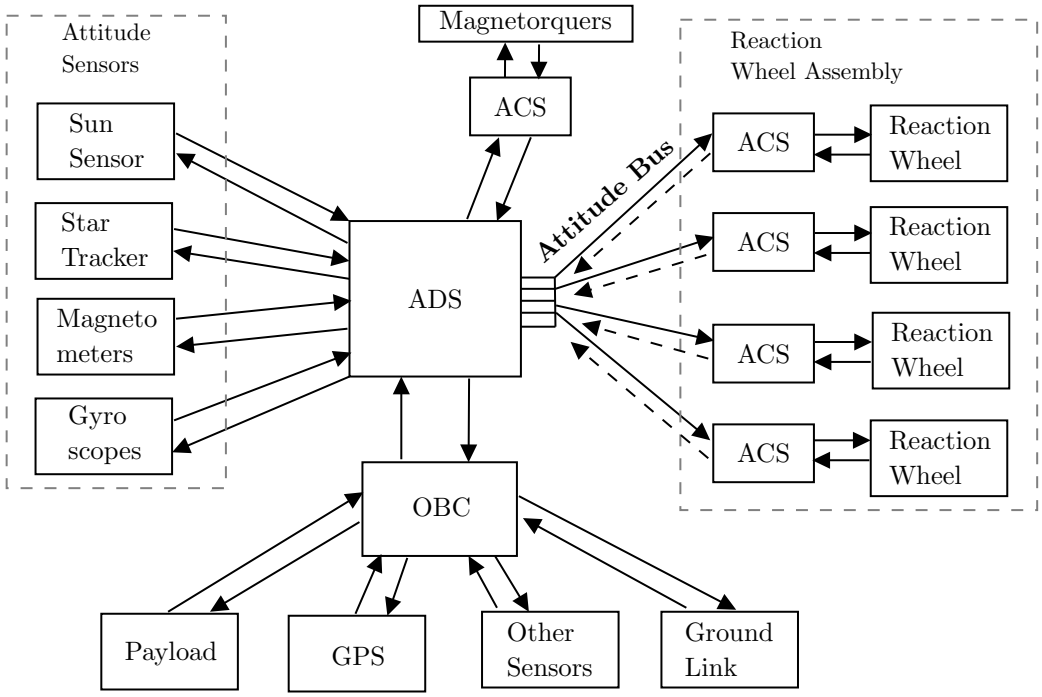


Figure 45: Proposed Decentralized architecture

Suppose that during the development stage, where an engineering model is being tested, the attitude performance of the satellite reveals itself as insufficient. With a decentralized architecture engineers can add a reaction wheel, or change its model or its direction, in a practical way, which does not involve reprogramming the other local ACSs. The same philosophy applies to the opposite case: during the test phase, if the engineers reach the conclusion that the spacecraft is too heavy and that a reaction wheel must be changed or removed, they can do so without worrying about modifications to the other wheels. Ultimately, reaction wheels can be manufactured with an incorporated ACS using a standardized protocol, in which a satellite developer can simply attach a wheel to a spacecraft, connect the "attitude bus" and declare the wheel's direction vector. After this, the spacecraft is ready for testing and eventual tuning.

The trade-off of this architecture is that the control torque becomes slightly distorted, and the obtained control action differs from the programmed one. It will be shown later that this distortion acts like a small deviation of the calculated control gains, and that it does not bring instability to the system. To compute how much torque a reaction wheel should apply, each individual ACS computes the control torque that should be applied to the spacecraft, and then computes how much of this torque can be performed by the wheel using a dot product distribution:

$$\boldsymbol{L}_w^{wi} = \boldsymbol{L}_B^w \cdot \boldsymbol{w}_l^i \tag{132}$$

This torque distribution does not guarantee an optimum torque distribution like the pseudo-inverse method, and for most of the cases it returns a torque that is bigger than the computed torque. The pseudo-inverse method gives different results depending on the type of assembly, so reaction wheel assemblies must be analyzed one by one. For a three wheel orthogonal assembly, the pseudo-inverse method simply becomes the inverse method, which gives the same result

for the dot product distribution, which is equation (132). In this case, as long as the wheel's axes are orthogonal, the dot distribution provides the optimal solution. For the other cases, the difference between the pseudo-inverse and the dot distributions causes the control action to be distorted, and this distortion can be characterized by the *distortion matrix* $\mathbb{D}$:

$$\boldsymbol{L}_{RW} = \mathbb{D}\boldsymbol{L}_{Control}\,, \quad \mathbb{D} = \begin{bmatrix} D_x^T/D_x^p & 0 & 0 \\ 0 & D_y^T/D_y^p & 0 \\ 0 & 0 & D_z^T/D_z^p \end{bmatrix} \tag{133}$$

A practical example is given to explain these concepts: consider a classical four-wheel pyramid assembly, represented in Figure 46, which is characterized by the following distribution matrix and its pseudo-inverse:

$$D = \begin{bmatrix} a & -a & 0 & 0 \\ 0 & 0 & c & -c \\ b & b & d & d \end{bmatrix} \qquad D^p = \frac{1}{2}\begin{bmatrix} a^{-1} & 0 & b/(b^2+d^2) \\ -a^{-1} & 0 & b/(b^2+d^2) \\ 0 & c^{-1} & d/(b^2+d^2) \\ 0 & -c^{-1} & d/(b^2+d^2) \end{bmatrix} \tag{134}$$

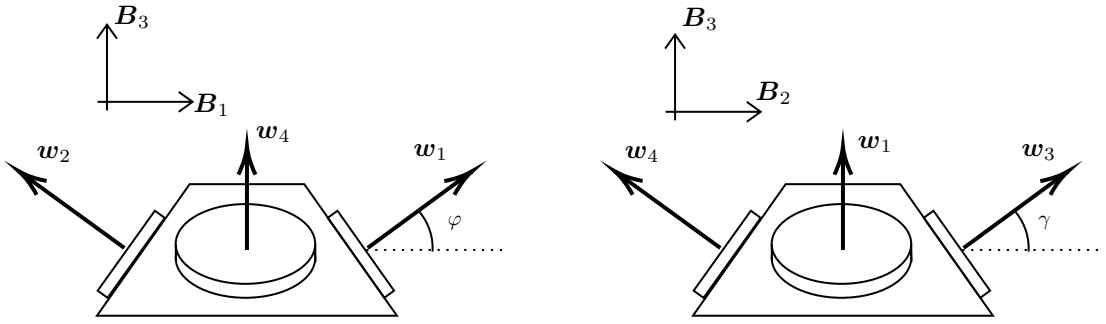


Figure 46: Four Reaction Wheel Assembly in a Pyramid

Where $a = \cos(\varphi)$, $b = \sin(\varphi)$ and $c = \cos(\gamma)$, $d = \sin(\gamma)$. The dot distribution gives the following distribution matrix:

$$D^{dot} = D^T = \begin{bmatrix} a & 0 & b \\ -a & 0 & b \\ 0 & c & d \\ 0 & -c & d \end{bmatrix} \tag{135}$$

Considering a symmetric assembly with with $\varphi = \gamma$, the distortion matrix is:

$$\mathbb{D} = \begin{bmatrix} 2\cos(\gamma)^2 & 0 & 0 \\ 0 & 2\cos(\gamma)^2 & 0 \\ 0 & 0 & 4\sin(\gamma)^2 \end{bmatrix} \tag{136}$$

Most of times the pyramid is symmetric, with $\varphi = \gamma$, and this inclination angle is in the region of 20^o to 60^o. For angles outside of this region, the assembly starts losing the ability to act effectively in all directions: for $\varphi = 0^o$ there is no component in $\boldsymbol{B}_3$, while for $\varphi = 90^o$ there is torque in $\boldsymbol{B}_3$ only. For these intermediate angles, the numeric value of the dot distribution matrix approaches the value of the pseudo-inverse matrix, so the distortion matrix approaches

the identity matrix I_3. In Figure 47 is a plot of the numerical values for both pseudoinverse (blue) and dot (red) distributions , for the xy directions and the z direction, as a function of the wheel's inclination angle γ. Below, the dot distribution's relative deviation to the optimal solution is represented, showing the percentage of torque distortion in the xy and z directions.

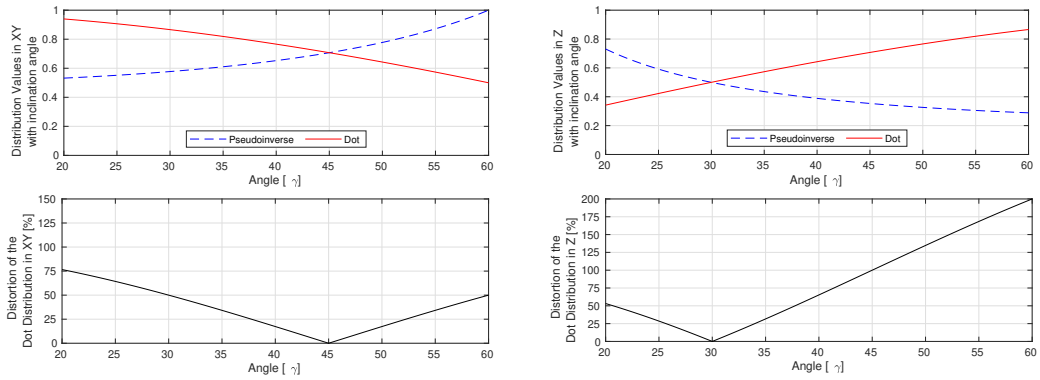


Figure 47: Dot and Pseudoinverse Distributions for a pyramid assembly with four wheels

For angles between 25^o and 45^o, the distortion is less 100% in any direction, with no distortion in xy for $\gamma = 45^o$ and in z for $\gamma = 30^o$. For a classical, symmetrical pyramid assembly with $\varphi = \gamma = 30^o$, the distortion matrix becomes:

$$\mathbb{D} = \begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{137}$$

This will be used as a reference for the development of the control laws for *Infante*. What is interesting is that the consequence of this distortion is not a loss in control performance, but rather a small increase of the control gains in that axis, where the satellite responds slightly faster than expected, by using the dot distribution. Furthermore, this distortion can be interpreted as a the result of a pseudo-inverse distribution law for a different set of higher control gains in the xy direction.

Another typical assembly is the six reaction wheel pyramid, represented in Figure 48, in which the angle β defines the pyramid's inclination.

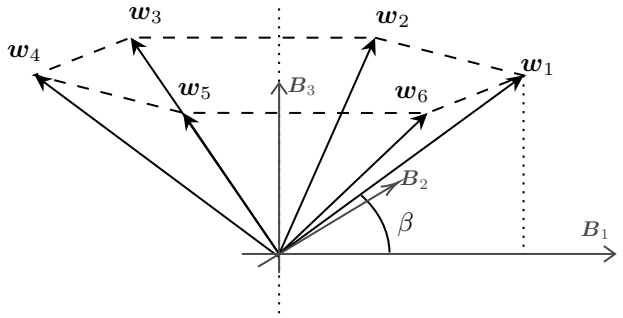


Figure 48: Six Reaction Wheel Assembly in a Pyramid

The pyramid is around the $\boldsymbol{B}_3$ axis for analytical simplicity, they are commonly built around the $\boldsymbol{B}_2$ axis to achieve higher capacity in pitch. The distribution matrix for this assembly is given by :

$$D = \begin{bmatrix} a & a/2 & -a/2 & -a & -a/2 & a/2 \\ 0 & a\sqrt{3}/2 & a\sqrt{3}/2 & 0 & -a\sqrt{3}/2 & -a\sqrt{3}/2 \\ b & b & b & b & b & b \end{bmatrix} \tag{138}$$

Where $a = cos(\beta)$ and $b = sin(\beta)$. The pseudo-inverse is given by the matrix:

$$D^p = \frac{1}{6}\begin{bmatrix} 2a^{-1} & a^{-1} & -a^{-1} & 2a^{-1} & -a^{-1} & a^{-1} \\ 0 & \sqrt{3}a^{-1} & \sqrt{3}a^{-1} & 0 & -\sqrt{3}a^{-1} & -\sqrt{3}a^{-1} \\ b^{-1} & b^{-1} & b^{-1} & b^{-1} & b^{-1} & b^{-1} \end{bmatrix}^T \tag{139}$$

The dot distribution matrix is again $D^{dot} = D^T$. Similarly, the numerical values of the pseudo-inverse matrix are similar to the dot distribution matrix for a certain reasonable window of the angle β. The distortion matrix is calculated as:

$$\mathbb{D} = \begin{bmatrix} 3\cos(\beta)^2 & 0 & 0 \\ 0 & 3\cos(\beta)^2 & 0 \\ 0 & 0 & 6\sin(\beta)^2 \end{bmatrix} \tag{140}$$

In Figure 49 is the analysis for this assembly: of the values of dot product algorithm (red) and the optimum method of the pseudo-inverse (blue), and the produced distortion in percentage in both directions. For the reasonable interval of inclination angles from 30^o to 50^o, the dot distribution algorithm is within 250% of the optimum pseudo-inverse. A small distortion is achieved between 35^o and 40^o. These are also the angles that are typically used, where the assembly provides a more uniform performance.

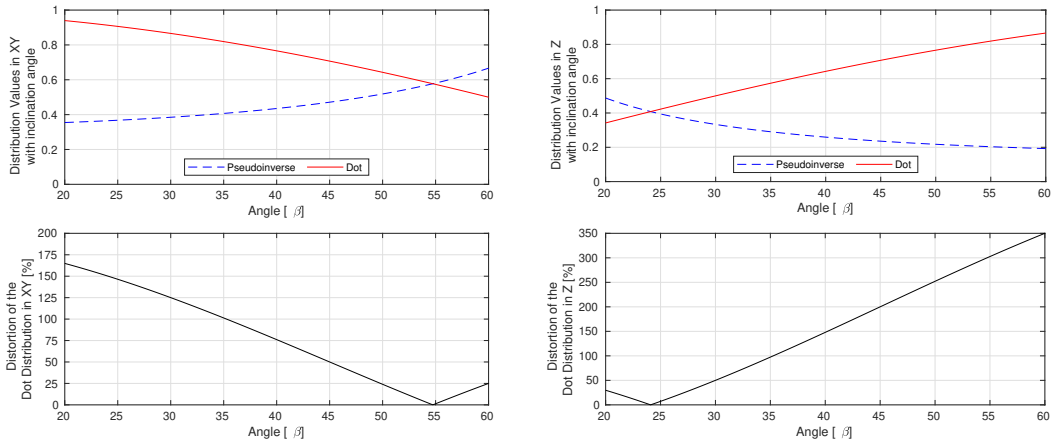


Figure 49: Dot and pseudoinverse distributions for a pyramid assembly with six wheels

These results can be generalized for pyramid assemblies of n wheels: the graphics exhibit similar shapes and the dot product distortion increases with n, and if the inclination angles become too low or too high. For $n = 8$ reaction wheels, the distortion is twice the one in Figure 47 and the same can be said for $n = 12$, which produces a distortion twice as large as the one seen in Figure 49.

In the next section, attitude control laws are developed using a dot distribution with the originally planned four-wheel pyramid assembly for *Infante*, with $\varphi = 30^o$. Then a comparison is made with the pseudo-inverse method, and the effects of adding and removing reaction wheels is studied.

10 Attitude Control Laws

In this chapter, the developed control laws are explained in detail, along with simulation results showing their effectiveness. First, a general algorithm for regulation is developed, and its simulation results are obtained using the dot distribution, not the pseudo-inverse. The control parameters are tuned using these results, ignoring the torque distortion caused by this distribution. Then, the relation is derived between the calculated control gains, as calculated by the pseudo-inverse, and the actual gains, as distorted by the dot distribution. This will show that the distortion introduced by the decentralized architecture keeps the system stable, and does not introduce significant changes to its behavior.

In the second part of this section, algorithms for detumbling the spacecraft, desaturating the reaction wheels and managing its momentum are also presented, along with their simulation results.

10.1 Regulation

Regulation control consists of keeping a certain commanded attitude with the reference angular velocity as zero. It is the control law that keeps the satellite pointed to the Earth, to the Sun, or to other convenient point. The first part consists of defining the desired quaternion:

The target attitude is defined in the local *LVLH* frame but represented in the *GCI* frame which is static. The desired attitude is described as a roll rotation followed by a pitch and a yaw rotation in the local frame, or more simply a $1-2-3$ Euler Angle representation. This strategy lets us to define a roll component first, that can be used for either SAR or for rolling the spacecraft with the Sun, and then to define a pitch component to minimize the disturbance torque. The yaw component can ultimately be ignored since this has no practical utility, it exists solely to enable any attitude command that might be necessary in any special case.

The rotation quaternion that defines the attitude is:

$$\bar{q}_{attitude} = \bar{q}_{yaw} \otimes \bar{q}_{pitch} \otimes \bar{q}_{roll} \tag{141}$$

Where:

$$\bar{q}_{roll} = \begin{bmatrix} \sin(\phi/2) \\ 0 \\ 0 \\ \cos(\phi/2) \end{bmatrix} \quad \bar{q}_{pitch} = \begin{bmatrix} 0 \\ \sin(\theta/2) \\ 0 \\ \cos(\theta/2) \end{bmatrix} \quad \bar{q}_{yaw} = \begin{bmatrix} 0 \\ 0 \\ \sin(\psi/2) \\ \cos(\psi/2) \end{bmatrix} \tag{142}$$

Provided that there is knowledge of the attitude of the local frame, the desired attitude in the geo-centric inertial frame is obtained by the product:

$$\bar{q}_{desired} = \bar{q}_{LVLH-GCI} \otimes \bar{q}_{attitude} \tag{143}$$

The control error is given by the rotation between the estimated and desired attitudes. The rotation is itself represented by the error quaternion $\delta\bar{q}$:

$$\hat{\bar{q}}_{GCI-B} = \delta\bar{q} \otimes \bar{q}_{desired} \tag{144}$$

$$\delta\bar{q} = \hat{\bar{q}}_{GCI-B} \otimes \bar{q}^{-1}_{desired} \tag{145}$$

The chosen regulation control law is a non-linear PD controller in the quaternion form, adapted from [9], where it is shown that is asymptotically stable:

$$\boldsymbol{L}_{control} = K_p sign(\delta\bar{q}_4)\delta\bar{\boldsymbol{q}}_{1:3} + K_d(1 + \delta\bar{\boldsymbol{q}}_{1:3}^T\delta\bar{\boldsymbol{q}}_{1:3})\boldsymbol{\omega}_{local} \tag{146}$$

This law was chosen since it is a simple yet effective PD controller, with a non-linear gain in the derivative part, increasing stability in large maneuvers. The disturbances are relatively small in magnitude so no integral action is used, assuring stability in exchange for a negligible steady state error. It is important to notice that this feedback law has no minus sign, due to the nature of the reaction wheels: if you apply a positive torque in a reaction wheel, it will respond with a negative torque in the spacecraft. The torque distortion effect of the dot distribution is neglected, as it will be shown to have little effect.

The term $sign(\delta\bar{q}_4)$ is used to prevent a phenomenon called unwinding, since $\bar{q}$ and $-\bar{q}$ represent the same attitude. The controller could complete a 360^o maneuver to "unwind" the quaternion if this was not considered.

The subscript in $\boldsymbol{\omega}_{local}$ means that it is the angular velocity regarding the local frame. The local frame has a non-negligible orbital component that needs to be subtracted, so that when the spacecraft is static in this frame, it must have zero angular velocity.

The desired feedback gains are derived making some approximations: the axis are considered uncoupled, the disturbances negligible and the error angles small. Later it is shown that these approximations are reasonable and provide an excellent starting point for the control design. Consider that there is just one rotation axis in the satellite, the $x-$axis, and that the control objective is to keep the reference angle $\theta_{ref} = 0$. The sum of torques in that axis, using Newton's second law is:

$$L_{disturbances} + L_{control} = J_x\ddot{\theta} \tag{147}$$

Now consider the control law of equation (146) in the $x-$axis only. The vector part of the error quaternion is the Euler axis and angle product $\delta\bar{\boldsymbol{q}}_{1:3} = \boldsymbol{e}\sin(\theta/2)$. Assuming again the error of the $x-$axis only, and using the first order approximations:

$$\sin(\theta/2) \approx \theta/2\,, \qquad (1 + \delta\bar{\boldsymbol{q}}_{1:3}^T\delta\bar{\boldsymbol{q}}_{1:3}) = [1 + \sin(\theta/2)^2] \approx 1 \tag{148}$$

The control action is written in the approximate form:

$$L_{control} = -\frac{K_p}{2}\theta - K_d\omega^x\,, \quad \omega^x = \dot{\theta} \tag{149}$$

Assuming that disturbances are negligible and substituting the control action in equation (147):

$$-\frac{K_p}{2}\theta - K_d\dot{\theta} = J_x\ddot{\theta} \tag{150}$$

This can be written as a second order differential equation in its canonical form:

$$\ddot{\theta} + \frac{K_d}{J_x}\dot{\theta} + \frac{K_p}{2J_x}\theta = 0 \tag{151}$$

This differential equation's coefficients are closely related to a natural response frequency ω_n and a damping coefficient ζ:

$$\frac{K_p}{2J_x} = \omega_n^2\,, \qquad \frac{K_d}{J_x} = 2\zeta\omega_n \tag{152}$$

Rearranging these expressions, it is possible to represent the gains as a function of the desired behavior:

$$K_p = 2J_x\omega^2\,, \qquad K_d = 2J_x\zeta\omega_n \tag{153}$$

Generalizing for all three axis, the control law in equation (146) is rewritten as:

$$\boldsymbol{L}_{control} = 2J\left[\omega_n^2 sign(\delta\bar{q}_4)\delta\bar{\boldsymbol{q}}_{1:3} + \zeta\omega_n(1+\delta\bar{\boldsymbol{q}}_{1:3}^T\delta\bar{\boldsymbol{q}}_{1:3})\boldsymbol{\omega}_{local}\right] \tag{154}$$

Where the natural response frequency ω_n and damping coefficient ζ are the adjustable parameters. These are only representative of the true behavior for a certain interval of natural frequencies and damping coefficients. Setting a value too low for the natural frequency will cause the control action to be too small, and the effect of the disturbances too evident. Setting a frequency too high will cause the satellite to chatter and vibrate, and to not converge in a static attitude state, even with a high damping coefficient.

A natural frequency of $\omega_n = 0.1$ rad/s was chosen for a tight control of the attitude, where a good balance of stability, maneuver time and accuracy was found. The chosen damping coefficient is $\zeta = 1$, which in theory prevents overshoots and oscillations, which translate in wasted energy and higher convergence time. In Figure 50 is plotted the local quaternion for a step response of the controller with these parameters, and in Figure 51 the respective error represented in Euler Angles. The spacecraft is kept aligned with the local frame until a step input of 30^o occurs at the 2500 second mark. It is shown that the controller behaves well, with a minimal overshoot of 0.7^o. This shows that the first order approximations performed when deriving the gains were reasonable, but not exact, as no overshoot was expected, and also that the dot distribution has little effect in the behavior of the spacecraft.

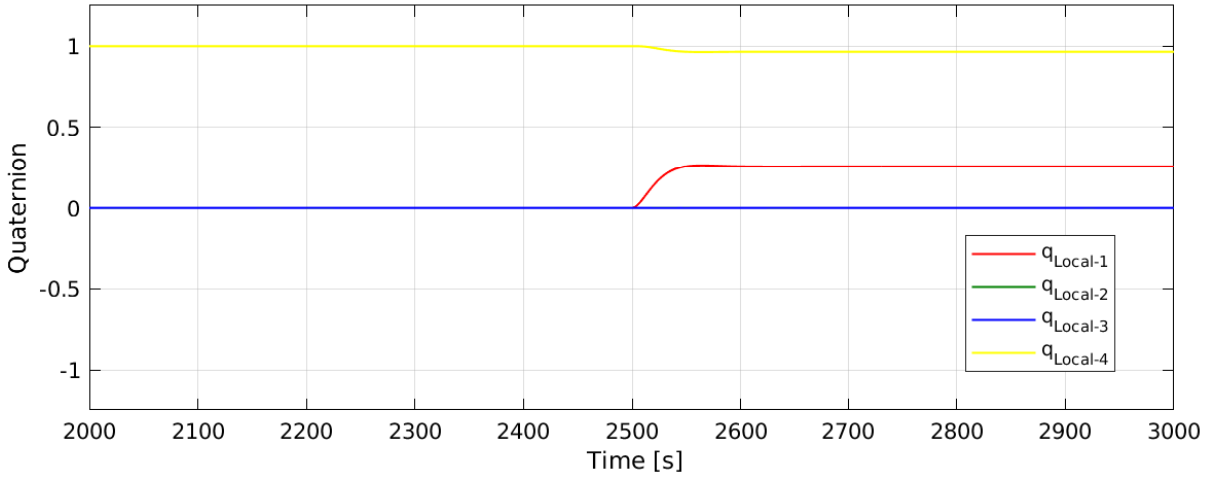


Figure 50: Controller Step Response for 30^o in roll, $\omega_n = 0.1$ rad/s and $\zeta = 1$

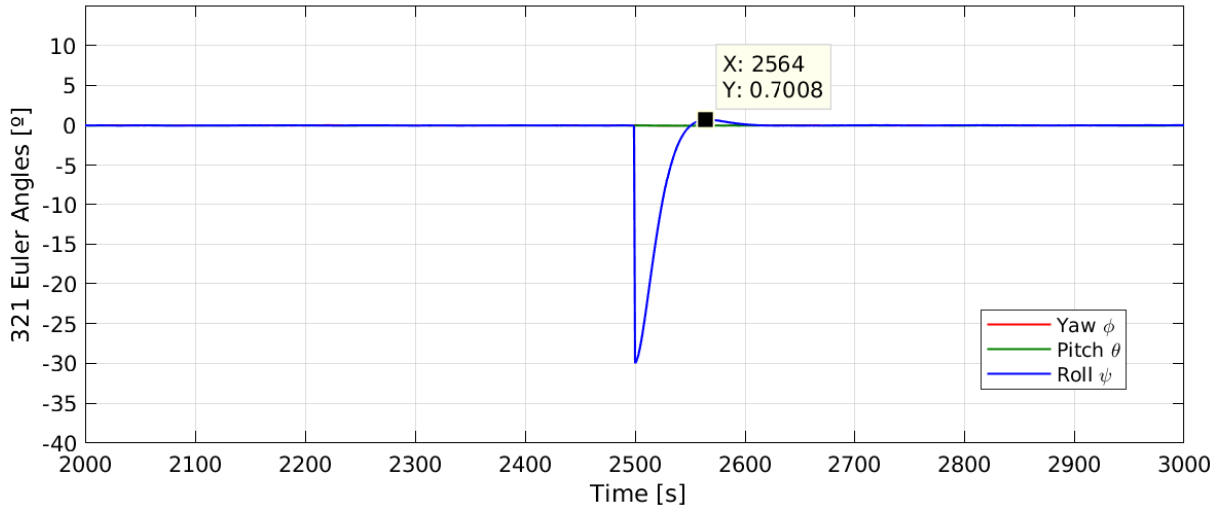


Figure 51: Control Error in Euler Angles for 30^o in roll, $\omega_n = 0.1$ rad/s and $\zeta = 1$

Performance starts to degrade when the controller is asked to act on multiple axis. The more complicated is the maneuver, the bigger the overshoot, mainly due to the assumption that the rotation axes act as uncoupled. In Figure 52 is represented the attitude quaternion in the local frame for a step response of 60^o in roll, pitch and yaw. The spacecraft takes approximately 2 and a half minutes to complete the maneuver with an overshoot of 7.748^o in the $y-$axis.

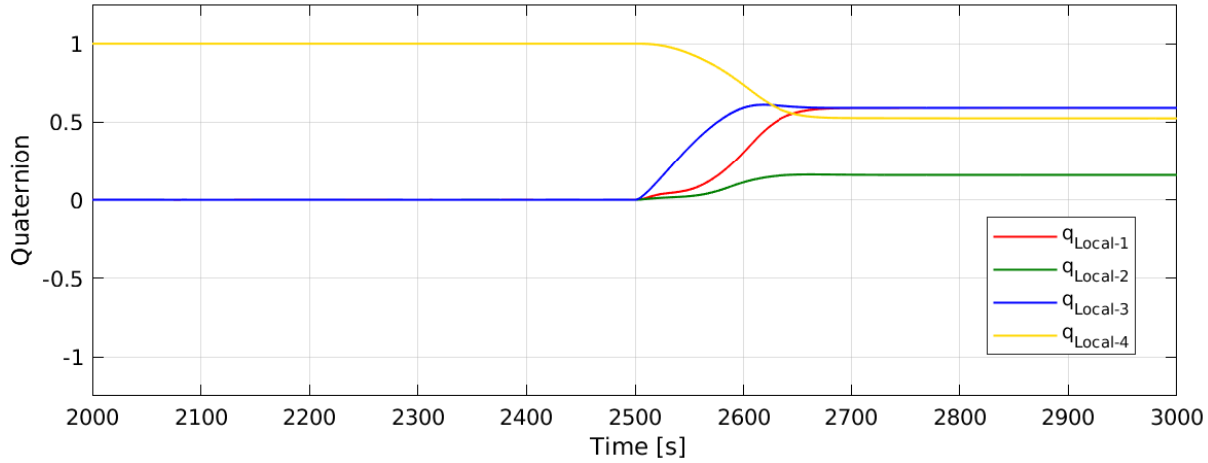


Figure 52: Controller Step Response for 60^o in all axis, $\omega_n = 0.1$ rad/s and $\zeta = 1$

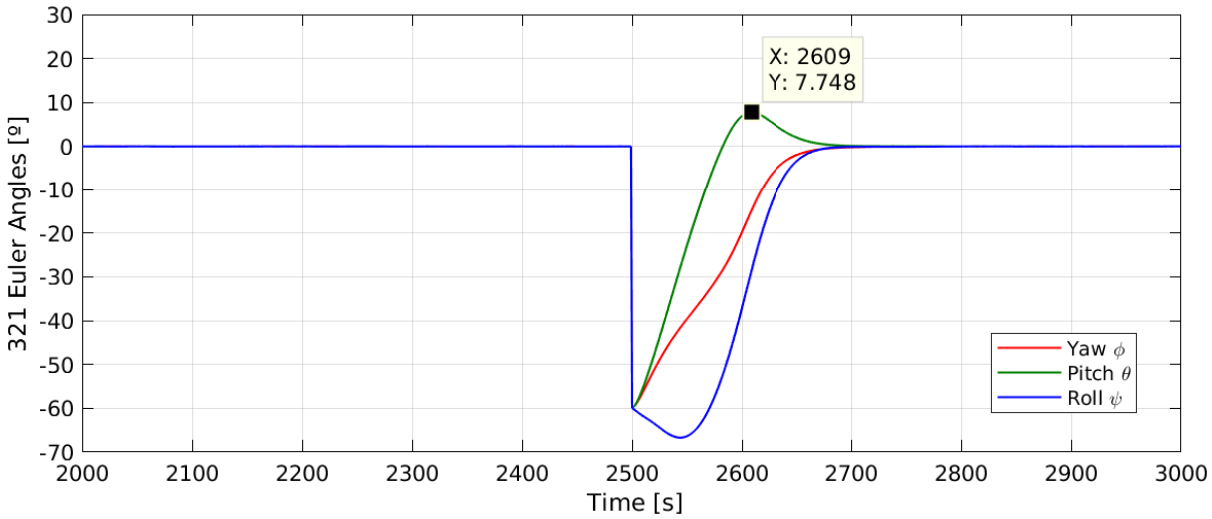


Figure 53: Control Error in Euler Angles for 60^o in all axis, $\omega_n = 0.1$ rad/s and $\zeta = 1$

This problem is reduced by increasing the damping coefficient. Simulation results show that a damping coefficient of $\zeta = 1.5$ is sufficient so no overshoot happens in the vast majority of maneuvers. However, asking for a 90^o shift in all angles is a special case that exhibits angle overshoot for damping coefficients as high as $\zeta = 3$. For practical cases, this rotation is unlikely, and, with such a damping coefficient, maneuvers would take much longer to be completed. Sometimes it is better to have a couple degrees of angle overshoot in exchange for a large economy in maneuver time.

In Figures 54 and 55 is a plot of the local quaternion and the respective control error expressed in $3-2-1$ Euler Angles, for a 60^o step input in all axes, but this time, with a damping coefficient of $\zeta = 1.5$. The maneuver takes 261 seconds to complete with negligible overshoot.

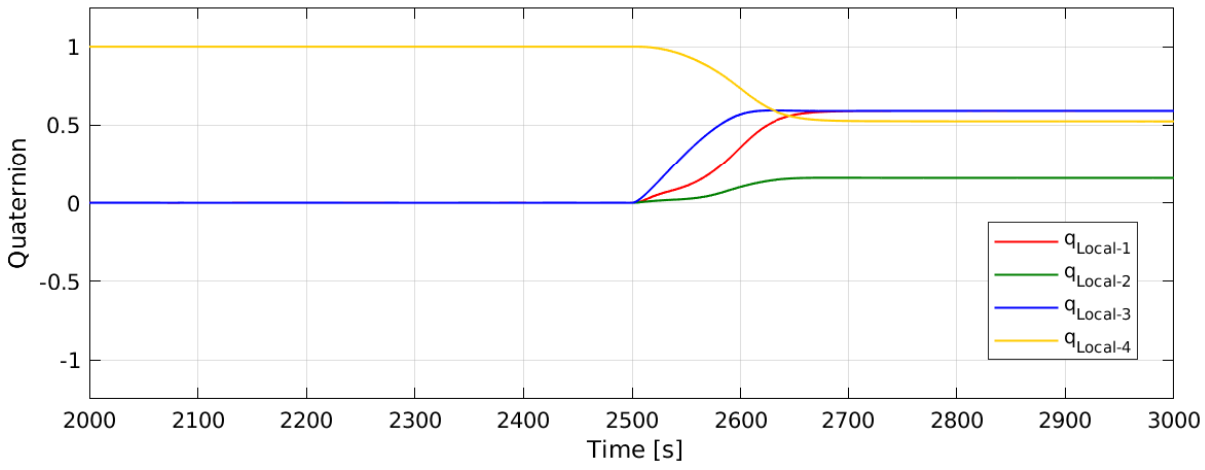


Figure 54: Controller Step Response for 60^o in all axis, $\omega_n = 0.1$ rad/s and $\zeta = 1.5$

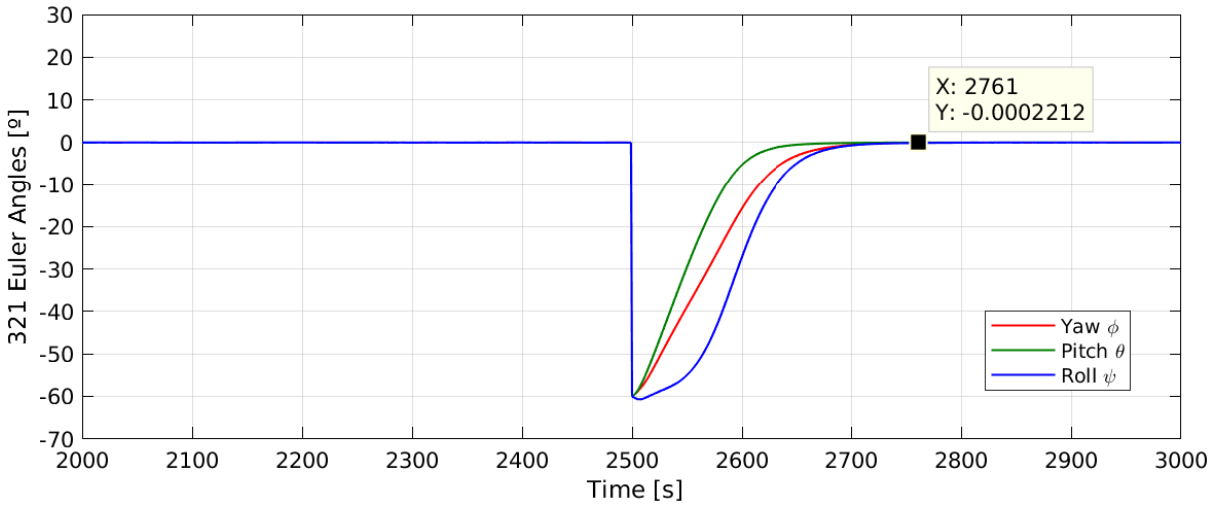


Figure 55: Control Error in Euler Angles for 60^o in all axis, $\omega_n = 0.1$ rad/s and $\zeta = 1.5$

In satellite attitude control, it is usually best practice to have a slightly over-damped system than a faster, under-damped system. The satellite does not need to have fast dynamics but rather slow, efficient movements, so the chosen parameters for a dynamic control were defined as $\omega_n = 0.1$ rad/s and $\zeta = 1.5$. In Figure 56 is a plot of the steady state error in attitude Euler Angles, when the spacecraft is using the Star Tracker, the Precise Gyroscope and the developed attitude law. The expected error, estimation error plus control error, is bounded to 0.05^o degrees. The pitch error is the largest one due to the effect of the aerodynamic torque. No integral action is needed since no steady state error is noticeable and since the error is within the specification by a large margin.

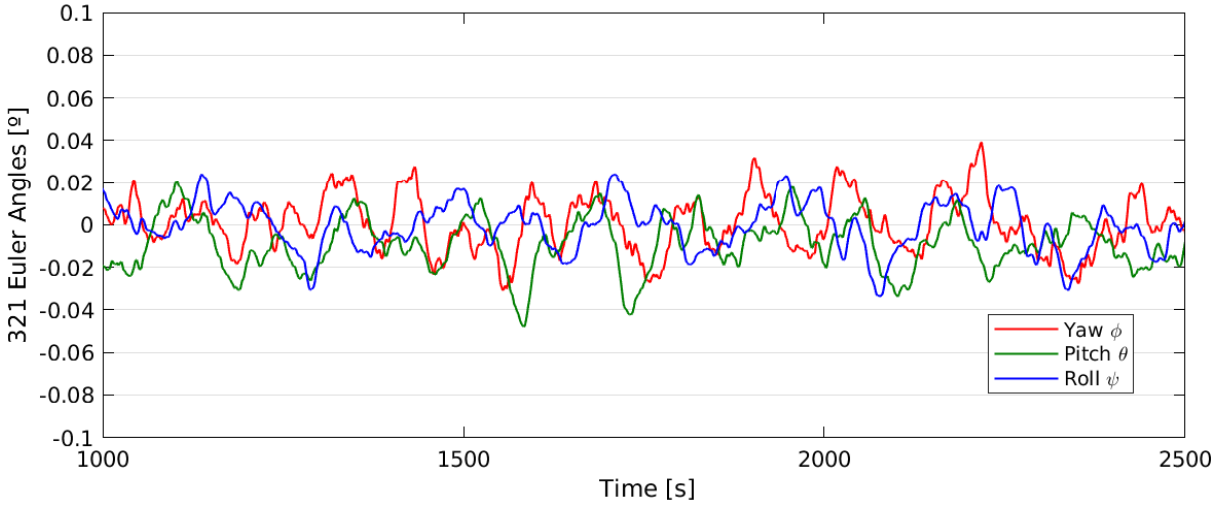


Figure 56: Attitude Error when Aligned with local frame, $\omega_n = 0.1$ rad/s and $\zeta = 1.5$

In Figure 56 the total attitude error is plotted again but this time with the spacecraft rolled 30^o, simulating a SAR maneuver. The total attitude error is similar to the previous case, being bounded on a $\pm 0.05^o$ interval again. It is worth mentioning that these simulation results are very optimistic and they do not represent the actual expected error for *Infante*, since a lot of error sources were neglected throughout this study.

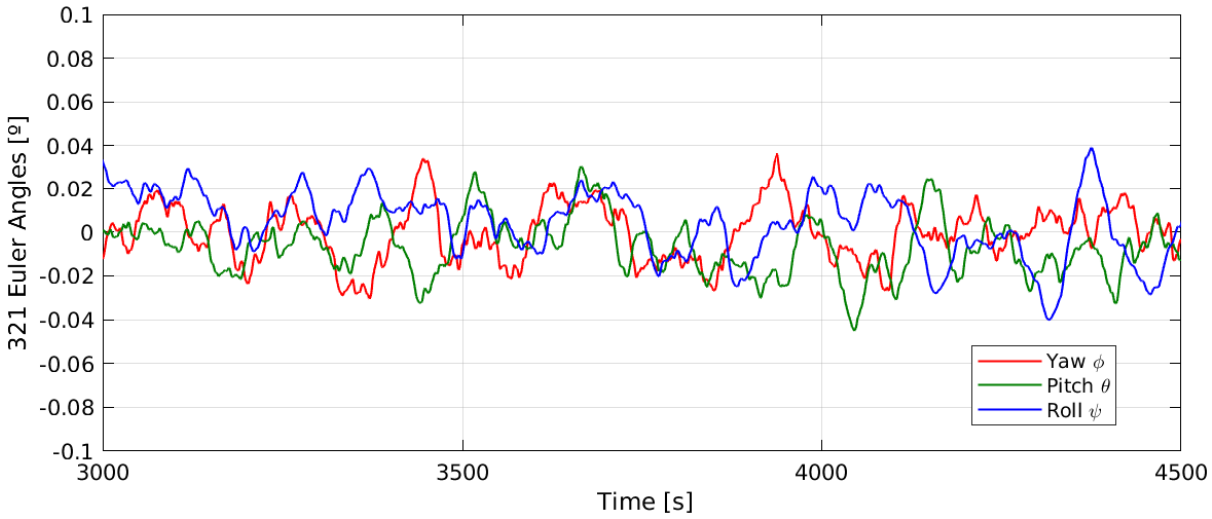


Figure 57: Attitude Error of the Spacecraft when rolled 30^o, $\omega_n = 0.1$ rad/s and $\zeta = 1.5$

When *Infante* is standing by, the attitude sensors are kept at a minimum to save energy. Consider a case where the sun sensor, magnetometer and coarse gyro are used during orbit daytime, and that the precise gyro is turned on just before the eclipse, when the sun sensor is powered off. This is a situation subject to large estimation errors, due to the nature of these sensors, and a situation where the control action should be minimized. A combination of parameters such as a natural frequency of $\omega_n = 0.01$ rad/s and a damping coefficient of $\zeta = 3$ was chosen, since it produces slow control dynamics with two great advantages: first, energy is saved by performing less control effort and secondly, it is less sensitive to error spikes in attitude estimation.

In Figures 58 and 59 is represented the estimation error and the control error for several orbits using this control law. In Figure 60 the total attitude error is represented. The largest estimation errors occur during daytime and can amount up to a 10^o error, due to coarse gyro drift, magnetometer bias and albedo effect in the Sun Sensor. For such significant estimation errors, nothing can be done but reducing the controller dynamics.

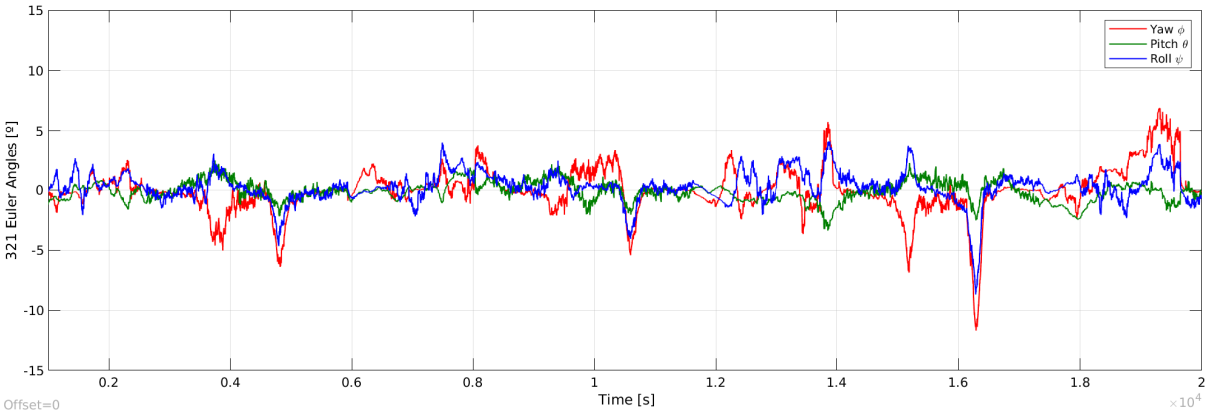


Figure 58: Estimation Error of the Spacecraft in cruise mode $\omega_n = 0.01$ rad/s and $\zeta = 3$

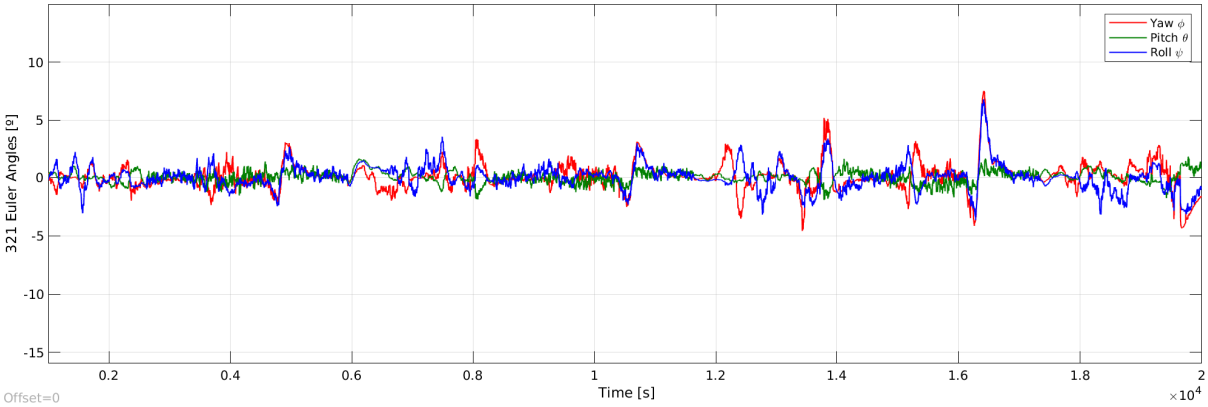


Figure 59: Control Error of the Spacecraft in cruise mode $\omega_n = 0.01$ rad/s and $\zeta = 3$

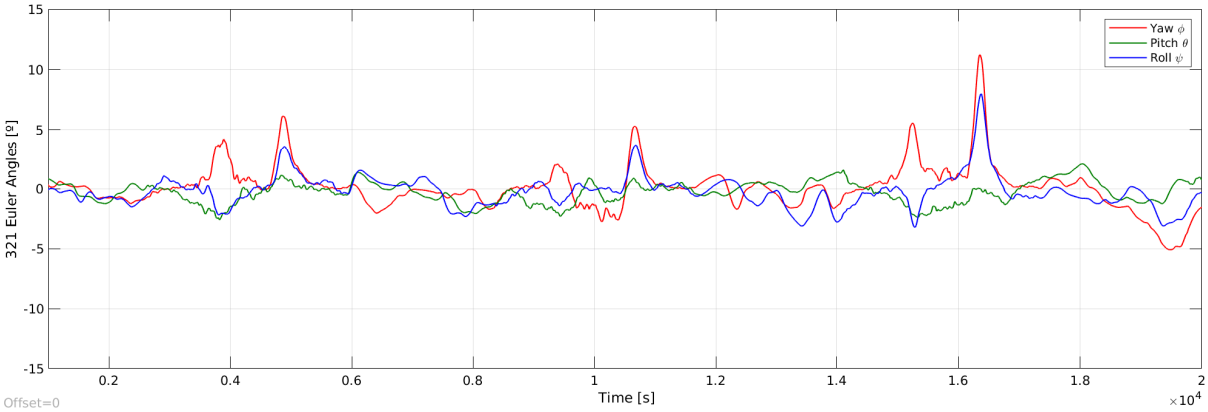


Figure 60: Total Attitude Error of the Spacecraft in cruise mode $\omega_n = 0.01$ rad/s and $\zeta = 3$

In Figure 61 it is possible to appreciate the difference in control effort for each set of gains, during the regulation of attitude.

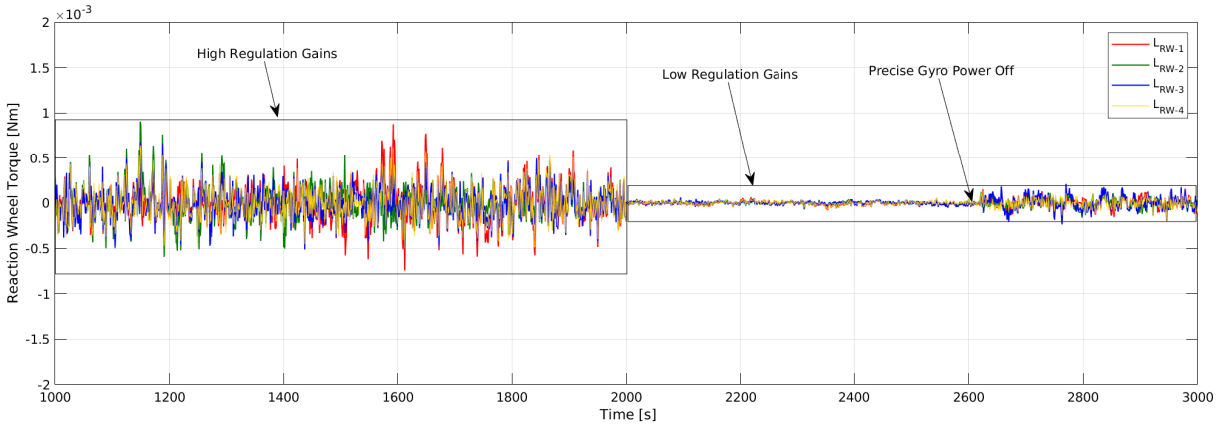


Figure 61: Reaction Wheel Control Effort with High and Low Gains

10.2 Influence of torque distortion in control

The use of a decentralized architecture and a dot product distribution law means that the output torque is distorted, interfering with optimum control. This distortion can be characterized by a distortion matrix $\mathbb{D}$, as discussed previously in the section regarding the Decentralized Architecture:

$$\boldsymbol{L}_{RW} = \mathbb{D}\boldsymbol{L}_{control} \tag{155}$$

Substituting the control law of equation (154), the true, distorted reaction wheel torque becomes:

$$\boldsymbol{L}_{RW} = 2J\left[sign(\delta\bar{q}_4)\omega_n^2\mathbb{D}\delta\bar{\boldsymbol{q}}_{1:3} + \zeta\omega_n\mathbb{D}(1+\delta\bar{\boldsymbol{q}}_{1:3}^T\delta\bar{\boldsymbol{q}}_{1:3})\boldsymbol{\omega}_{local}\right] \tag{156}$$

The true natural frequency $\check{\omega_n}$ and the true damping coefficient $\check{\zeta}$ become two diagonal matrices, since the system is not isotropic anymore:

$$[\check{\omega_n^2}] = \omega_n^2\mathbb{D}\,, \qquad [\check{\zeta}\check{\omega_n}] = \zeta\omega_n\mathbb{D} \tag{157}$$

And the following relation can be extrapolated:

$$[\check{\omega_n}] = \begin{bmatrix} \check{\omega_n^x} & 0 & 0 \\ 0 & \check{\omega_n^y} & 0 \\ 0 & 0 & \check{\omega_n^z} \end{bmatrix} = \omega_n \begin{bmatrix} \sqrt{\mathbb{D}^x} & 0 & 0 \\ 0 & \sqrt{\mathbb{D}^y} & 0 \\ 0 & 0 & \sqrt{\mathbb{D}^z} \end{bmatrix} \tag{158}$$

$$[\check{\zeta}] = \begin{bmatrix} \check{\zeta^x} & 0 & 0 \\ 0 & \check{\zeta^y} & 0 \\ 0 & 0 & \check{\zeta^z} \end{bmatrix} = \zeta \begin{bmatrix} \sqrt{\mathbb{D}^x} & 0 & 0 \\ 0 & \sqrt{\mathbb{D}^y} & 0 \\ 0 & 0 & \sqrt{\mathbb{D}^z} \end{bmatrix} \tag{159}$$

The true system frequency and dampness, taking in account the dot distribution law, are represented by these matrices, which increase with the square root of the torque distortion. The actual natural frequency and damping coefficients for the developed control law, affected by

the dot distribution in a four wheel pyramid assembly, can be calculated by substituting the distortion matrix calculated previously in (137):

$$[\breve{\omega_n}] = \omega \begin{bmatrix} 1.2247 & 0 & 0 \\ 0 & 1.2247 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad [\breve{\zeta}] = \zeta \begin{bmatrix} 1.2247 & 0 & 0 \\ 0 & 1.2247 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{160}$$

The distortion created by using the dot distribution method, in this case, only produces a 22.5% increase in the calculated natural frequency and damping coefficient in the xy directions. This variation is quite small, showing that the dot distribution does not change the behavior of the system significantly. Having an xy natural frequency of $\breve{\omega} = 0.123$ instead of $\omega = 0.1$, and having a damping coefficient of $\breve{\zeta} = 1.84$ instead of $\zeta = 1.5$, is not critical: the control is still in an optimum region. Furthermore, an increase in system dampness compensates the increase in dynamics by the natural frequency.

More interestingly, torque distortion tends to be greater than unity for most assembly angles, and tends to increase with wheel number. This means that as reaction wheels are added to the spacecraft, its natural frequency and damping coefficient tend to increase slightly, making the satellite automatically faster and more damped. Care is needed when removing reaction wheels, because it may cause a decrease of the damping coefficient to values below one, making the spacecraft under-damped.

Consider the addition of two reaction wheels, so that the assembly becomes a six wheel pyramid with $\beta = 30^o$. Substituting equation (140), the distortion matrix for this assembly becomes:

$$\mathbb{D} = \begin{bmatrix} 2.25 & 0 & 0 \\ 0 & 2.25 & 0 \\ 0 & 0 & 1.5 \end{bmatrix} \tag{161}$$

And the new natural frequency and damping coefficients become:

$$[\breve{\omega_n}] = \omega \begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 1.2247 \end{bmatrix}, \quad [\breve{\zeta}] = \zeta \begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 1.2247 \end{bmatrix} \tag{162}$$

Which is also an example of the low effect torque distortion has on the overall system behavior.

To better evaluate the difference in system behavior caused by using a dot product distribution, the cases that were studied were simulated using the developed method. A 60^o step input in this is given in all axis at 1000 second simulation time. In Figure 62 is the comparison of the control error in Euler Angles for four different cases: the classical pyramid assembly of four reaction wheels using both pseudo-inverse and dot product distribution, the six reaction wheel pyramid using dot distribution and two equal sets of the four wheel pyramid assembly, also using a dot product distribution.

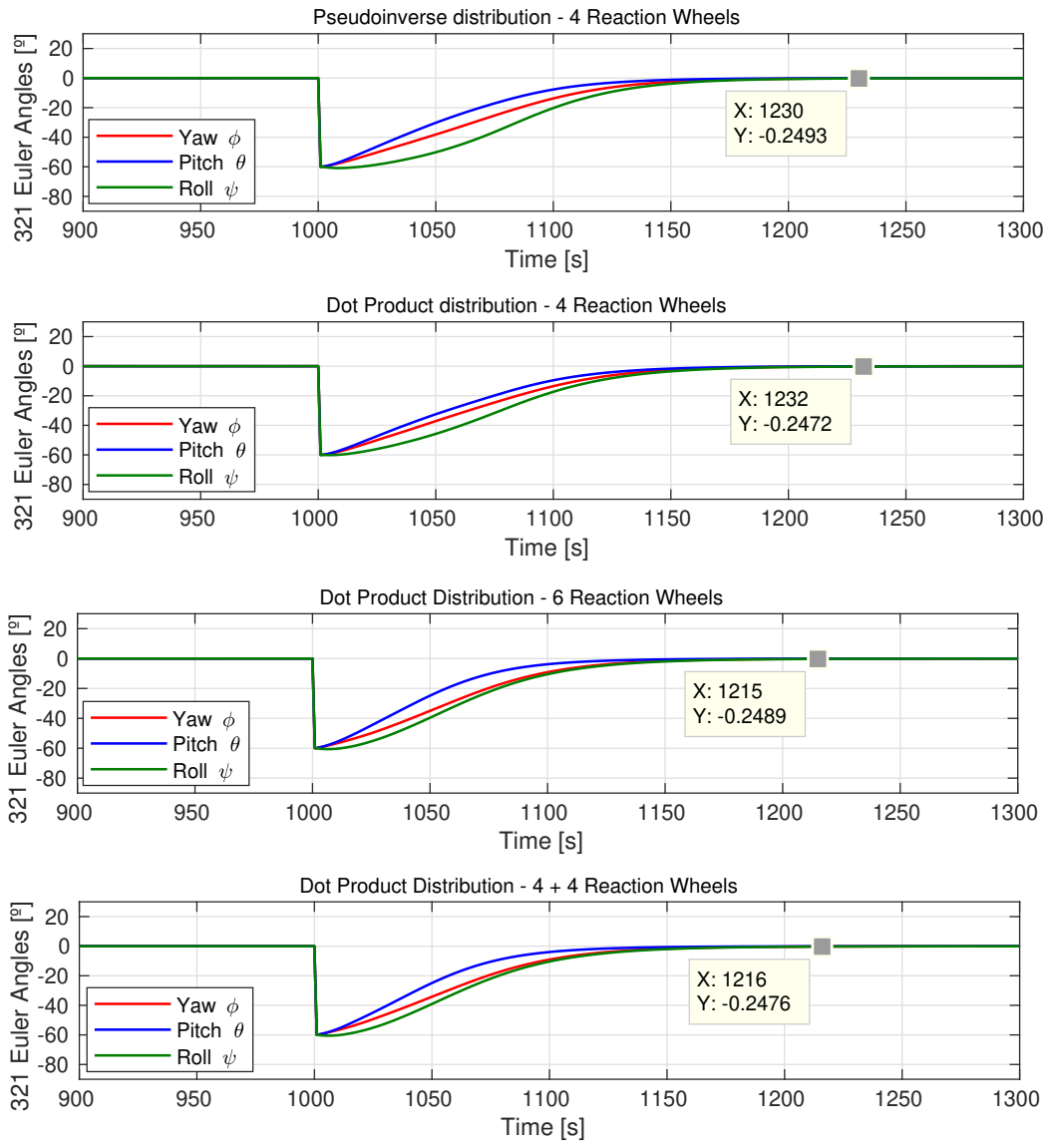


Figure 62: Performance comparison between assemblies and distributions

The maneuver is considered complete when the control error is less than 0.25^o for all axis. The pseudoinverse method is only two seconds faster than the dot distribution for the four wheel assembly, showing that in practice, performance is nearly identical. One of the reasons for this is that the actuators are using its maximum torque during the maneuver, making control differences even smaller. For a six wheel pyramid assembly, the maneuver was completed 15 seconds earlier than the previous cases, due to the increased actuation power. Using two pyramid assemblies of four wheels, however, provides a performance similar to the six wheels case because the system becomes too over-damped. These simulations show that the dot product distribution is a valid, functional method, which is sufficiently close to the optimum solution to be used in practice.

10.3 Detumbling

When a satellite is deployed from its launcher, it spins in an uncontrolled way, which must be stopped in order to proceed with the mission. Although this can happen naturally, it can take months or years, so a detumbling action is done by the magnetorquers by producing a torque opposite to the angular velocity. This can be challenging in some cases because attitude sensors like the Star Tracker, the Sun Sensor and some types of gyroscopes, do not typically work at high spinning rates. There are several types of control laws for the magnetorquers which will be studied and compared, depending on their energy efficiency, detumbling time and resources used.

Assuming that angular velocity is available from a gyro measurement, [9] proposes the control law:

$$\boldsymbol{m} = \frac{k}{||\boldsymbol{B}||^2}\boldsymbol{\omega} \times \boldsymbol{B} \tag{163}$$

which produces the control torque:

$$\boldsymbol{L}_{mag} = -k(I_3 - \boldsymbol{b}\boldsymbol{b}^T)\boldsymbol{\omega} \tag{164}$$

where $\boldsymbol{b} = \boldsymbol{B}/||\boldsymbol{B}||^2$. This control torque acts in the opposite direction of $\boldsymbol{\omega}$ and can be shown to be Lyapunov stable. Sometimes gyro measurements are unavailable because they become saturated. This is frequent in small CubeSats, so they use a variation of this control law called the *B-dot* control, where the angular velocity is replaced by the derivative of the measured magnetic field $\dot{\boldsymbol{B}}$, which is estimated on board. The time derivative is:

$$\dot{\boldsymbol{B}} = A_{BI}\dot{\boldsymbol{R}} - \boldsymbol{\omega} \times \boldsymbol{B} \approx -\boldsymbol{\omega} \times \boldsymbol{B} \tag{165}$$

where $\dot{\boldsymbol{R}}$ is the magnetic field derivative due to the spacecraft's motion in orbit, which can be neglected for a considerable tumbling rate. The control law then becomes:

$$\boldsymbol{m} = -\frac{k}{||\boldsymbol{B}||}\dot{\boldsymbol{B}} \tag{166}$$

Another variation of this control law is also popular, called the *bang-bang* control, based on an on-off controller. Here each magnetorquer is controlled individually:

$$m_i = -m_i^{max} sign(\boldsymbol{u_i} \cdot \dot{\boldsymbol{B}}) \tag{167}$$

The advantages of this controller is that it has a bigger actuation capacity and it is less sensitive to noise. The field derivative $\dot{\boldsymbol{B}}$ is calculated by a backward finite difference:

$$\dot{\boldsymbol{B}}(t) \approx \frac{\boldsymbol{B}(t) - \boldsymbol{B}(t - \Delta t)}{\Delta t} \tag{168}$$

Worse performance can be expected in the simulation than in real conditions, due to a simulation time constraint of one second. These control laws can be used without gyros and attitude acquisition but they cannot completely detumble the spacecraft, as a residual angular velocity is left, due to the orbital variation of the magnetic field. Nevertheless, this is enough for starting reaction wheel actuation and these strategies have become very popular.

The optimal gain k has been calculated and suggested by several authors in literature [45] [9] [46]. It is worth mentioning that a higher gain doesn't always mean a better time performance. Setting a gain too high will not only make the detumbling take longer, but also less energy efficient. Optimal performance sits below the calculated optimal detumbling time, balancing energy and detumbling time [45]. The authors of [47] indicates that the optimal gain is given by:

$$k^* = 2\omega_0(1 + \sin(i))J_{min} \tag{169}$$

where ω_0 is the orbit rate, i is the orbit inclination and J_{min} in the minimum inertia. The articles [46] and [47] propose the use of a changing gain, which becomes stronger if the tumbling rate is higher, and decreases as detumbling reaches the end phase. This allows for a significant decrease in consumed energy with small changes in overall detumbling time. The gain is calculated:

$$k^{**} = \frac{k^*}{\varphi p + \varepsilon} \tag{170}$$

where φ is the rate factor, ε is the tuning parameter and p is the tumbling parameter:

$$p \approx \frac{||\hat{\boldsymbol{\omega}}||}{\sqrt{3}|\omega_{max}|} \tag{171}$$

where $|\omega_{max}|$ is the maximum sensible angular velocity by the attitude sensors.

Two important parameters are the frequency and the duty cycle of the magnetometer and the magnetorquers. When the magnetorquers are functioning the magnetometer does not provide realistic measurements as the magnetic field created by the spacecraft interferes with the readings of Earth's magnetic field. Increasing magnetometer time provides better readings but decreases actuation time, and vice-versa. The article [46] argues that the best performance happens in the case where the magnetorquers and the magnetometer change as fast as possible, and assumes a 50% duty cycle. Taking this into account and that the simulation's minimum step time is only of one second, the chosen cycle is two seconds for the magnetometer, in order to provide a finite difference, two seconds for the magnetorquer actuation and one second for the desaturation signal.

The five types of controllers were simulated with several gains and rate factors, registering the detumbling time as well as the energy used. In the five cases the spacecraft was given an initial velocity of $\omega = 0.05$ rad/s in each axis, which is representative of initial tumbling conditions. Detumbling ends when the module of the angular velocity reaches 0.02 rad/s and the reaction wheels are activated with the regulation control law.

Results are plotted for each controller in Figures 63 and 64, showing how the absolute angular velocity decreases with time and how much power is used. The results are then summarized in Table 8, where units $[kJ] \times c$ mean that the actual energy used is unknown, but the algorithms are compared using an unknown power constant c.

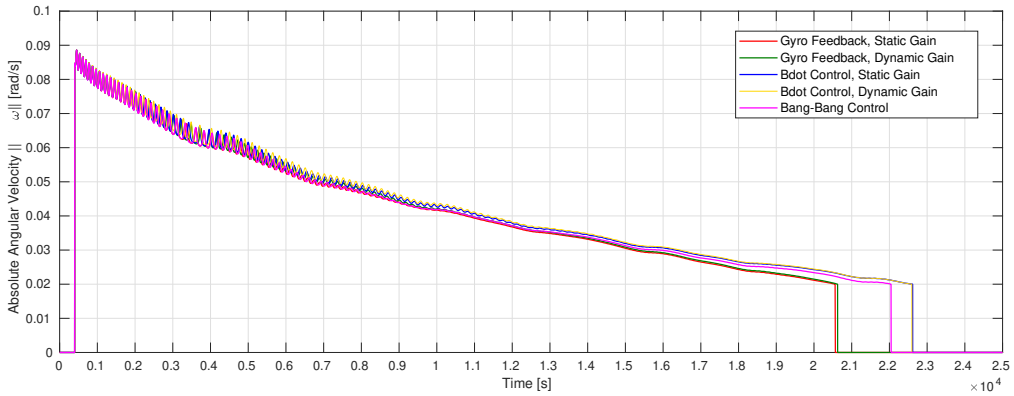


Figure 63: Absolute Angular Velocity $||\omega||$ during Detumbling for various controllers

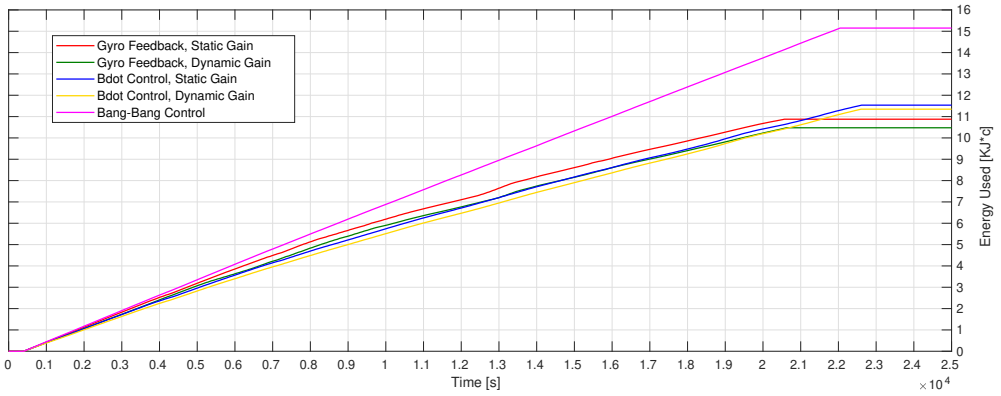


Figure 64: Energy used during Detumbling for various controllers

Table 8: Chosen gains and simulation results for five different detumbling control laws

Control Law	**Gain**	φ	**Detumbling Time** $[s]$	**Energy Used** $[kJ] \times c$
Gyro Feedback, Static Gain	$0.75\,k^*$		20174	10.88
Gyro Feedback, Dynamic Gain	k^*	12	20228	10.48
Bdot, Static Gain	k^*		22233	11.54
Bdot, Dynamic Gain	$1.5\,k^*$	12	22206	11.35
Bang-Bang	∞		21618	15.15

It is shown that the Gyro Feedback laws achieve the best performance. The chosen gain for the simple gyro feedback controller of equation (163) is $0.75\,k^*$, since it gives a better balance between used power and detumbling time than the theoretical gain k^* . The dynamic gain law, where gain is calculated by equation (170), was dimensioned using the gain k^* and $\varepsilon = 1$. The tumbling parameter $\varphi = 12$ was found to give a similar detumbling time while being able to save a bit of energy. This is the best controller to be implemented due to this small improvement.

The case where there is no feedback from the gyroscopes must be considered, since gyroscope saturation can occur for high tumbling rates, and because these are simpler controllers

to implement. The *B-dot* controllers perform slightly worse than the previous when speed is slowed down, due to the approximation made in equation (165). Similarly, there is also a small improvement on energy efficiency by using a dynamic gain, which should be considered.

Lastly, the *Bang-Bang* control law provides a smaller detumbling time than the other magnetic feed-back controllers but at the cost of losing energy efficiency. A *B-dot* controller is prefered for this reason.

10.4 Desaturation of the Reaction Wheels

10.4.1 Desaturation of the Reaction Wheels using Magnetorquers

Over the course of the mission, the disturbances on the spacecraft force the reaction wheels to accelerate, up to a point where they can reach saturation. To prevent this, the magnetorquers act to reduce reaction wheel momentum on the spacecraft, by creating a torque opposite to the wheels's accumulated momentum.

In a similar way to the detumbling algorithm of equation (163), [9] proposes a desaturation control law in the form of:

$$\boldsymbol{m} = \frac{k}{||\boldsymbol{B}||^2}\, \boldsymbol{h}^w \times \boldsymbol{B}\,, \qquad \boldsymbol{h}^w = \frac{\boldsymbol{H}^w}{||\boldsymbol{H}^w||} \tag{172}$$

where $\boldsymbol{H}^w$ is the reaction wheel momentum in the body frame, that should be minimized. Notice that minimizing the wheel momentum in the body space is very different than keeping the individual wheels' momentum near zero. In fact, the objective is to keep the velocity of each wheel at a desired speed, while the overall momentum is zero, by taking advantage of the wheel's redundant configuration and null space. This shall be discussed further ahead, and in this chapter the only concern is minimizing the body component of the wheel's momentum. It can be shown that the resulting torque of this control law is:

$$\boldsymbol{L}_{mag} = -k(I_3 - \boldsymbol{b}\boldsymbol{b}^T)\boldsymbol{h}^w \tag{173}$$

The effect of this control law is illustrated in Figures 65 and 66, for various control gains from $0.25k'$ to $5k'$, where k' is the optimal gain adapted from the detumbling gain equation (169):

$$k' = \omega_0(1 + \sin(i)) = \frac{k^*}{2J_{min}} \tag{174}$$

In this demonstration of the desaturation control law, the spacecraft is tumbling with a momentum of $||\boldsymbol{H}|| = 0.5$ Nms, that is transfered to the reaction wheels when they activate and stabilize the satellite. Soon after, the magnetorquers start desaturating the reaction wheels until its momentum in the body space reaches $||\boldsymbol{H}^w_B|| = 0.005$ Nms. Both Figures 65 and 66 show that the gain k' provides a good balance between desaturation time and energy efficiency. It is worth mentioning that attitude errors do not increase significantly if the magnetorquers are acting do desaturate the wheels, but they do increase significantly if there is a lot of accumulated momentum. Because of this, it is common to have the desaturation algorithms always turned on during the mission [9].

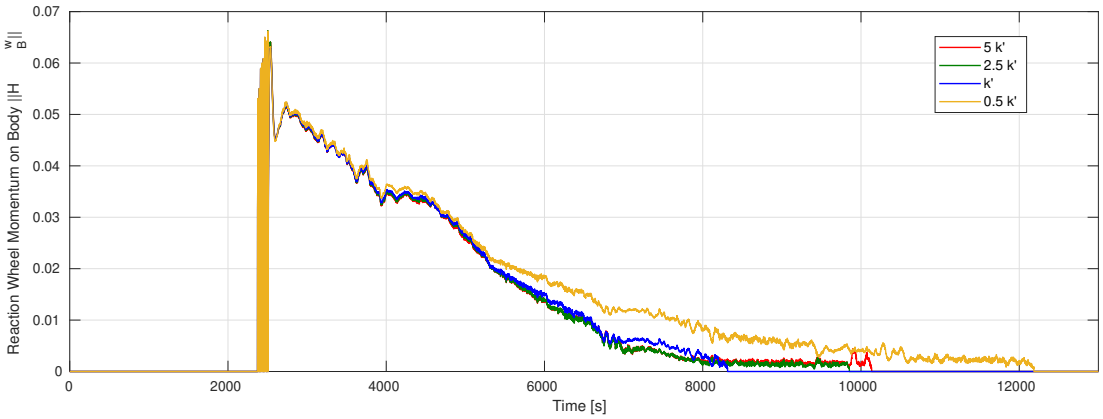


Figure 65: Total momentum of the Reaction Wheels during desaturation

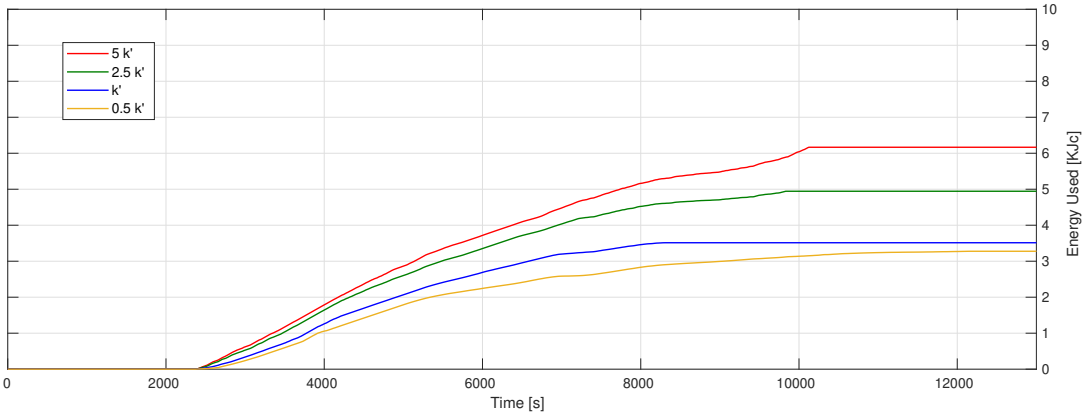


Figure 66: Energy used during desaturation for various control gains

10.4.2 Desaturation of the Reaction Wheels using Attitude

Considering the analysis of the aerodynamic and gravity gradient torque disturbances performed previously in the Disturbances section, it can be noticed that for the height of 500 km the relation between pitch and torque is close to linear, as seen on Figure 21. Even tough this relation is a coarse approximation, since it assumes that the atmosphere is static, some authors have suggested using attitude disturbances to the benefit of the satellite, for example, do desaturate the reaction wheels. The main advantage of this strategy is that there is no energy spent by the magnetorquers. Such control law, in pitch, would be:

$$\theta_{desat} = -kH_{RW}^{y} \tag{175}$$

10.5 Momentum Management of the Reaction Wheels

Due to the redundancy of the reaction wheels when mounted in an assembly, it is possible to have an overall momentum of zero, while the wheel's individual momentum is different than zero. This allows the wheels to be kept at optimal speed without interfering with the attitude control by applying a torque in the wheel's null space. The article [48] suggests an approach where the torque done by the wheels is divided into two: a component for spacecraft attitude control and a component driving the wheel to their optimal speed:

$$\boldsymbol{L}_B^w = D(\boldsymbol{L}_w^w + L_{management}\boldsymbol{n}) = D\boldsymbol{L}_w^w \tag{176}$$

where $L_{management}$ is the torque driving the wheel's speed to the desired state and $\boldsymbol{n}$ is the assembly's null space. By the definition of null space, the management torque has no effect in the body, and therefore it does not interfere with the control action. The management torque is defined by the article as:

$$L_{management} = PID\left(\omega_{desired} - \boldsymbol{n}^T\boldsymbol{\omega}_{RW}\right) \tag{177}$$

The problem with this solution is that it is incompatible with decentralized architecture, since the local ACSs do not have access to the other wheels' speed. This thesis proposes a blind method of driving the reaction wheels to the desired state. In the same manner as the article, each reaction wheel performs a control torque with two components:

$$L_{RW} = L_{control} + L_{management} \tag{178}$$

where the management torque is driven by a proportional controller:

$$L_{management} = k_{management}\left[sign(H_{RW})H_{desired} - H_{RW}\right] \tag{179}$$

where $H_{desired}$ is the desired momentum in the null space and H_{RW} is the current momentum. Due to the lack of information regarding the other wheels, the $sign(H_{RW})$ term drives the wheel's speed the closest of the desired states. This management works on the assumption that all the wheels will be managing the their momentum as well, so the overall system will converge into a desired null space. Unlike the first algorithm, it is not guaranteed that this algorithm won't interfere with the attitude control, so the management torque must be limited in a way that for a worst case scenario, a wheel failure, this won't degrade control performance. This is done by setting a management gain that is low enough, which for this case it was found that $k_{management} = 10^{-6}$ Nm/Nms produced good results. A low management gain means that reaching the desired state might take a considerable amount of time, but this isn't critical since once it is reached it will unlikely diverge unless the reaction wheels absorb a lot of momentum. The desired momentum was set as 0.01 Nms, around 10% of the maximum, where the wheels are spinning at low speed and far from the saturation limits. The maximum management torque produced by this algorithm in a worst case scenario is 10^{-7} Nm, 10% of the expected disturbance torque.

Simulation results of this algorithm can be seen in Figures 67 and 68, which show the evolution of the reaction wheel's momentum in their wheel space and body space respectively. In this simulation the spacecraft is kept pointing to the Earth while performing several orbits. Convergence to the desired null space takes a few hours due to the low gain, and is only possible if there is desaturation of the reaction wheels. A sinusoidal pattern at orbit rate can be observed because the reaction wheels aren't completely desaturated, and this causes an interchange of the residual momentum between them, keeping the angular momentum of the wheels pointing at the same direction in the inertial frame.

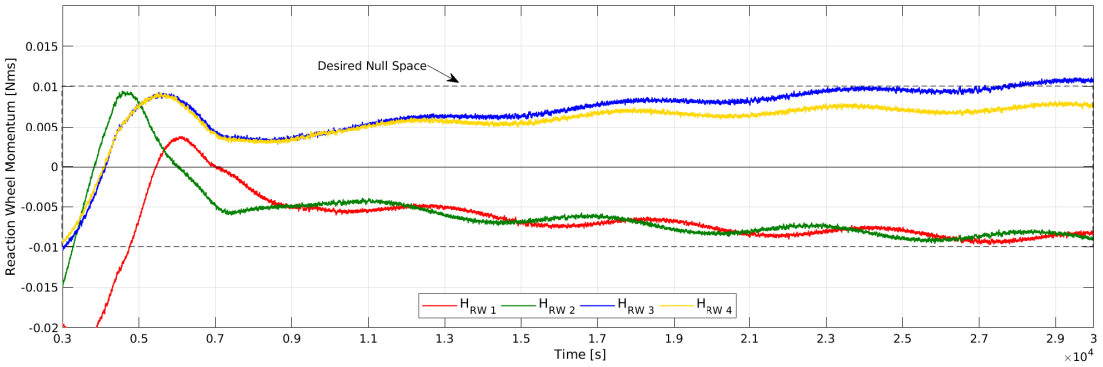


Figure 67: Total momentum of the Reaction Wheels during desaturation

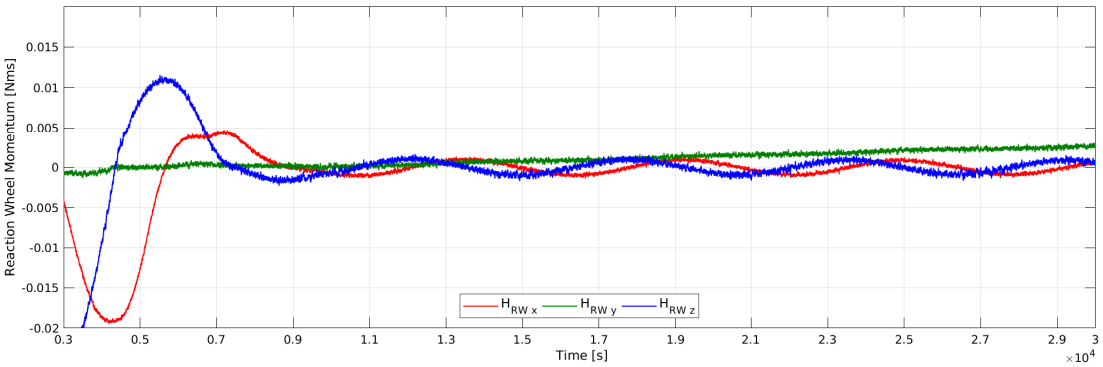


Figure 68: Total momentum of the Reaction Wheels during desaturation

11 ADCS Modes and Programming

In this section is proposed a global ADCS logic structure, whose goal is to make *Infante* fully autonomous. It is assumed that the ADCS communicates with the On-Board Computer (OBC) and the attitude sensors: the OBC gives a desired SAR epoch, and ADCS replies if it is ready to perform each mission phase.

This logic was designed as a state machine and implemented in the *Simulink* model using MATLAB's state machine compiler *Stateflow*. The goal was to design a logic structure as simple and robust as possible, while still being able to fulfill the requirements for this mission. The control program handles a set of logical variables that can be separated in system inputs, system outputs and internal variables. A list of these variables can be found in Tables [9], [10] and [11]:

Table 9: Output Variables of the ADCS

Variable	**Type**	**Description**
detumbling_flag	Bool	Requests detumbling control
regulation_flag	Bool	Requests regulation control
desaturation_flag	Bool	Requests desaturation of the RW
solar_panels_flag	Bool	Requests solar panel deployment
outer_mag_deploy_flag	Bool	Requests deployment of the outer magnetometer
dynamic_control_flag	Bool	Requests higher dynamics in regulation control
sun_sensor_flag	Bool	Requests use of the Coarse Sun Sensor
star_tracker_flag	Bool	Requests use of the Star Tracker
coarse_gyro_flag	Bool	Requests use of the coarse gyroscope
precise_gyro_flag	Bool	Requests use of the precise gyroscope
inner_magnetorquer_flag	Bool	Requests use of the inner magnetorquer
outer_magnetometer_flag	Bool	Requests use of the outer magnetometer
dither_flag	Bool	Requests dithering of the magnetorquer
sar_ready_flag	Bool	Is positive if spacecraft is ready for SAR
ready_flag	Bool	Is positive if spacecraft is mission ready
roll_command	Double	Sets desired roll
pitch_command	Double	Sets desired pitch
yaw_command	Double	Sets desired yaw

Table 10: Internal Variables of the ADCS

Variable	**Type**	**Description**
ads_start	Bool	Returns positive if ADS is functioning properly
mission_mode	Bool	Requests accurate attitude determination
magnetometer_flag	Bool	Requests the activation of any magnetometer
eps	Double	Measure of attitude determination accuracy
JD	Double	Current Julian Date
delta_t	Double	Time left until SAR
sensor_time	Double	Time before SAR where strict attitude determination starts
maneuver_time	Double	Time before SAR where maneuver starts
t_min	Double	Minimum preparation time for mission

Table 11: Input Variables of the ADCS

Variable	**Type**	**Description**
eclipse_status	Bool	Returns positive if the spacecraft is in eclipse, small margin
pre_eclipse_status	Bool	Returns positive if the spacecraft is in eclipse, large margin
sun_sensor_status	Bool	Returns positive if sun sensor is available
star_tracker_status	Bool	Returns positive if star tracker is available
inner_magnetometer_status	Bool	Returns positive if inner magnetometer is available
outer_magnetometer_status	Bool	Returns positive if outer magnetometer is available
outer_mag_deploy_status	Bool	Returns positive if outer magnetometer is deployed
coarse_gyro_status	Bool	Returns positive if coarse gyroscope is available
precise_gyro_status	Bool	Returns positive if precision gyroscope is available
solar_panels_status	Bool	Returns positive if solar panels are deployed
magnetorquer_status	Bool	Returns positive if magnetorquers are available
sar_status	Bool	Returns positive if OCB is doing SAR
H	Double	Total Momentum of Spacecraft in Local Frame
H_wheels	Double	Total Momentum on RW's body space
battery_level	Double	Energy available on the batteries
JD_mission	Double	Julian Date at mission Time

The system is divided into several state machines that are run in parallel, each with variables that are bounded to the states of the other state machines. The first state machine, labeled "a", handles the main states of the satellite, and is represented in Figures 69 and 70. All machines start in a "stand-by" state, which jumps to the next one when the starting bit "adcs_start" is triggered by the ADCS after start-up. It is assumed that the ADCS will be turned on after being ejected into orbit, and that the first effective state, "a10", should be detumbling. The program transitions into state "a20" when the total estimated momentum is below a certain threshold, where regulation control is possible. The reaction wheels absorb the rest of the momentum, so this threshold should be somewhat below the maximum momentum that can be supported by the reaction wheels. When the momentum is very small, the program transitions into the next state "a30", where the solar panels are deployed. It is important to mention that this momentum is measured regarding the local frame. In this state, the flag "outer_mag_deploy_flag" is also set, so that the outer magnetometer can be deployed, so it can substitute the inner magnetometer. Upon confirmation that the solar panels have been deployed, the program enters in what is intended to be its definitive state, "a40", and sets an output as "ready". If, for any given reason, the spacecraft's momentum becomes too large, the program goes back to detumbling mode, returning to "a40" after this is completed. The opposite threshold for detumbling must be larger than any momentum that the satellite might gain during a maneuver, or this state might be reached by accident.

Regulation control is defined by groups of two child states, defined inside the parent states "a20", "a30" and "a40", as illustrated in Figure 70. Regulation is only possible if there is enough energy stored in the batteries and the variable "eps" is low enough. This is an internal variable ε that correlates with the uncertainty of the attitude determination, that has been defined as one thousand times the trace of the covariance matrix of the Kalman Filter:

$$\varepsilon = 1000 \times trace(P) \tag{180}$$

In Figures 71 and 72 can be seen how this variable correlates with the estimation error during the satellite's regulation, while using only the sun sensor, magnetometer and gyroscopes. This strategy is to turn off the regulation control of the reaction wheels whenever the value of ε becomes too large, which is a consequence of when the filter regards its estimation as uncertain. This should be used with care, however, since this is a correlation, and the value of ε can be high for a correct estimation, and vice-versa.

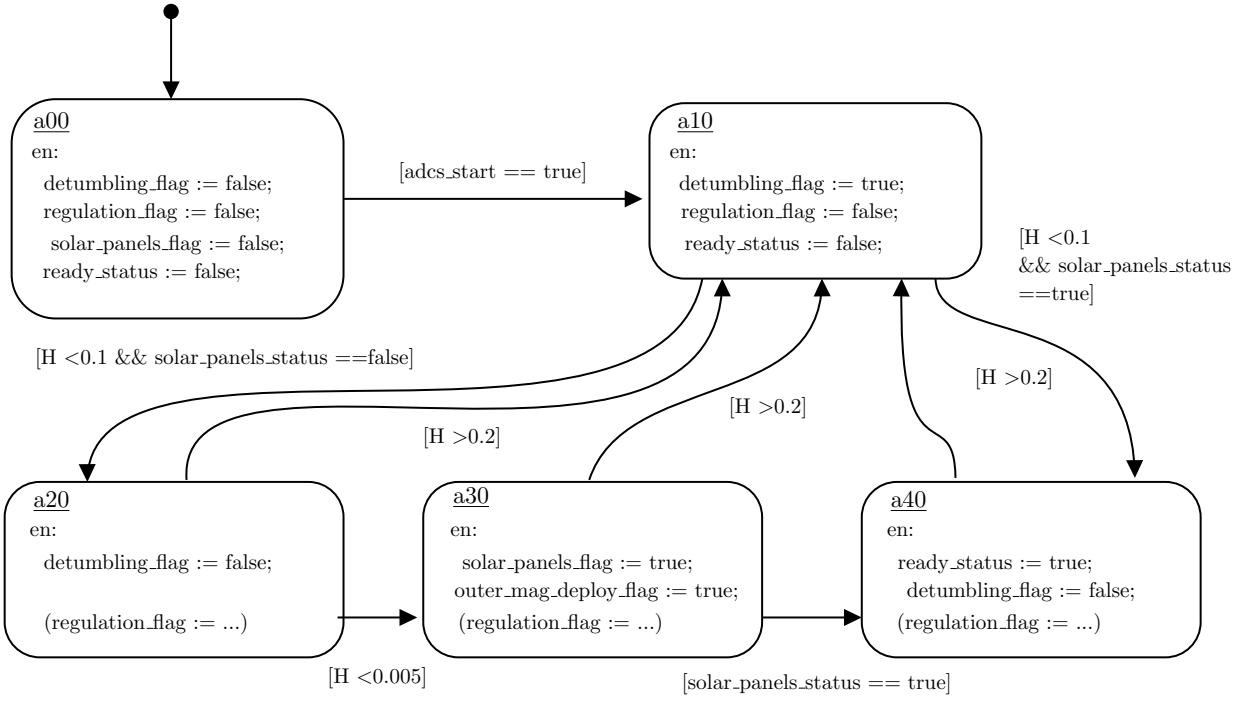


Figure 69: State Machine for Main Attitude Modes (Parent)

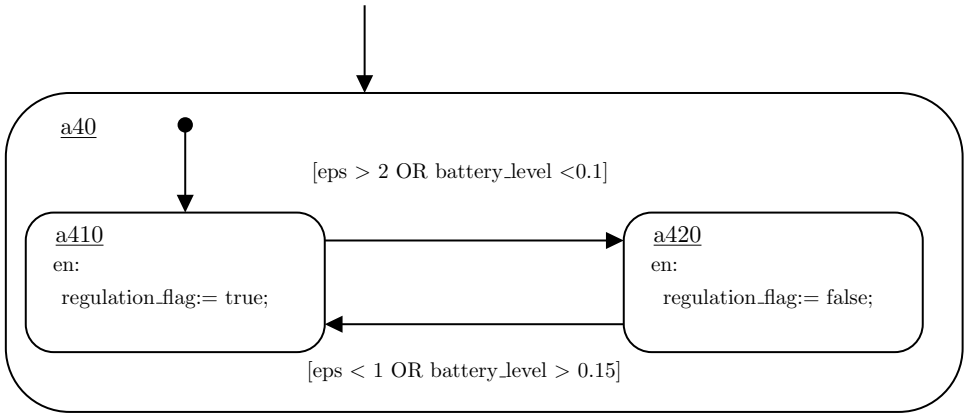


Figure 70: State Machine for Main Attitude Modes (Child)

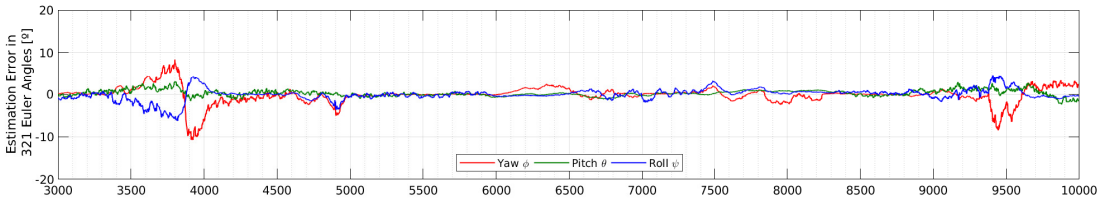


Figure 71: Estimation error of *Infante* during regulation control

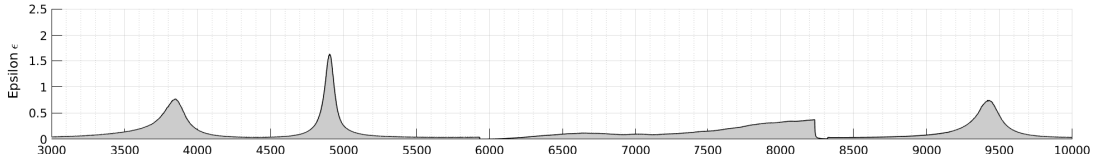


Figure 72: Variable "ε" and its correlation with the estimation error

The second state machine "b" handles the execution of SAR missions, and is represented on Figure 73. The first state is a stand-by state in which no mission is performed, and *Infante* shall remain here for the great majority its lifetime. The On-Board Computer gives the ADCS a certain mission epoch in Julian days, using a variable called "*JD_mission*", which is the calculated epoch when *Infante* will be flying over a specific target. What this state machine does is calculate the time interval Δt of time left until the passage over the target, in order to anticipate actions like activating the Star Tracker, the precision gyroscope and maneuvering the spacecraft.

There is a transition into the mission states if the following conditions are gathered: the variable "*delta_t*" is within a certain interval of time, the ADCS has completed its initial procedures and there is enough energy in the batteries. The variable "*delta_t*" is computed:

$$\Delta t = (JD\_mission - JD) \times 24 \times 60 \quad [min] \tag{181}$$

The number of minutes before the scan in which the sensors activate, here described as "*sensor_time*", can be either fixed, or declared as a function of the variable "*eps*", allowing the precision sensors to be turned on later if *Infante* has already good attitude accuracy.

When entering the first mission state, the flags for precision attitude and control are triggered, the program jumps into its next state as the mission comes closer, and the spacecraft starts maneuvering. As seen earlier, it takes approximately four minutes to perform any maneuver on *Infante*, so something like seven minute time window is reasonable. An example maneuver is given in this state, although in practice the OBC would communicate the desired attitude required to hit a target. Once the maneuver is complete, the program transitions into the next state, where a ready flag is activated and the ADCS stands by. If the OBC replies with a SAR status, the dynamic control is deactivated to avoid the possibility of any corrections blurring the image, by having the reaction wheels responding too fast. The final state is triggered after SAR is over, the spacecraft rolls back into its normal position and then goes back to stand-by.

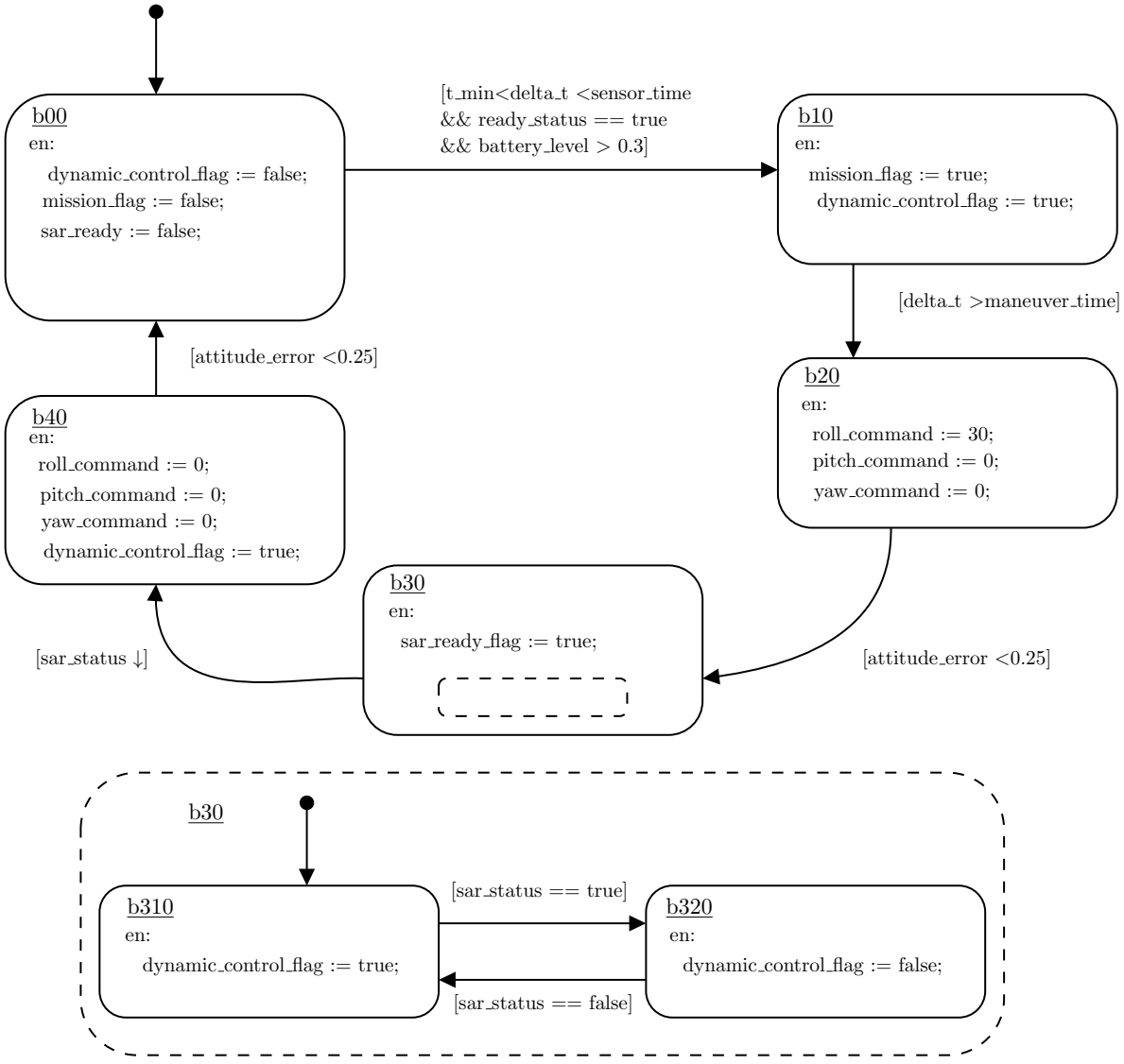


Figure 73: State Machine for Mission Handling

Each attitude sensor has a state machine assigned to its behavior. They are all quite similar, with a starting state, a turn-off state and an active state. It is important to clarify that the term "status" is used to indicate that a sensor is able to give reasonable data if turned on at any given moment. For example: in a Star Tracker, its status is positive if the sensor has not failed and if the Sun is not in its field of view. This model is used on the state machine since most of the sensors can be operated with power-saving modes, in which they are standing by and not performing attitude measurements, but still transmitting status data and using low power. This type of stand-by mode is available in the magnetometers [40] [39] and the coarse gyroscope [37], but no information is given for the Star Tracker [41] and the Sun Sensor. Still, it is assumed that it is possible to know if the sun is in the Star Tracker's field of view or if there is an eclipse. The precision gyro doesn't have a stand-by mode [38], but this gyro is always available unless there is a failure, which would then be recorded by the ADCS.

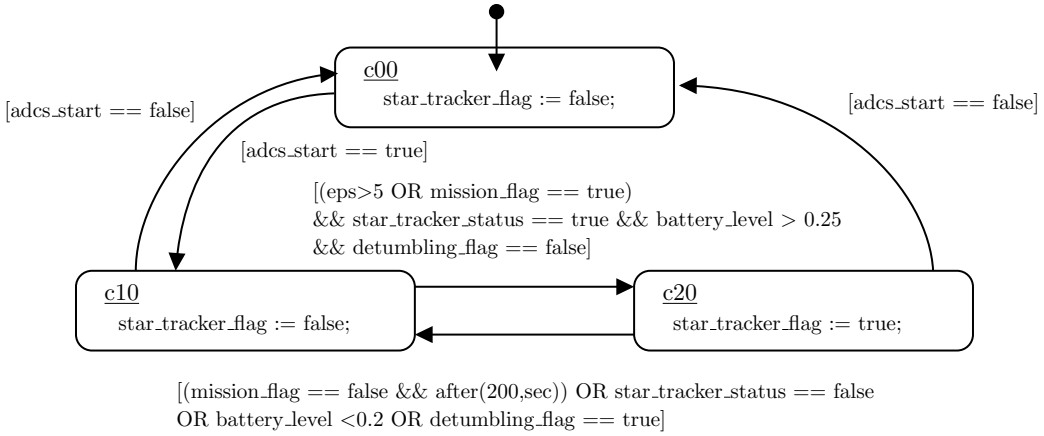


Figure 74: State Machine for the Star Tracker

In Figure 74 is represented the state machine for the Star Tracker. As mentioned, it goes into stand-by mode "c10" once the ADCS has started, and goes into an active state if the following conditions are gathered: accurate attitude determination is needed, either because the covariance of the Kalman Filter is too high or because the spacecraft is on a mission, there is enough energy in the batteries, the star tracker is available and the spacecraft is not detumbling. The star tracker is turned off if there is no mission and sufficient time has passed to ensure good attitude knowledge, or if there is low battery level or the sensor becomes unavailable, or the satellite starts detumbling.

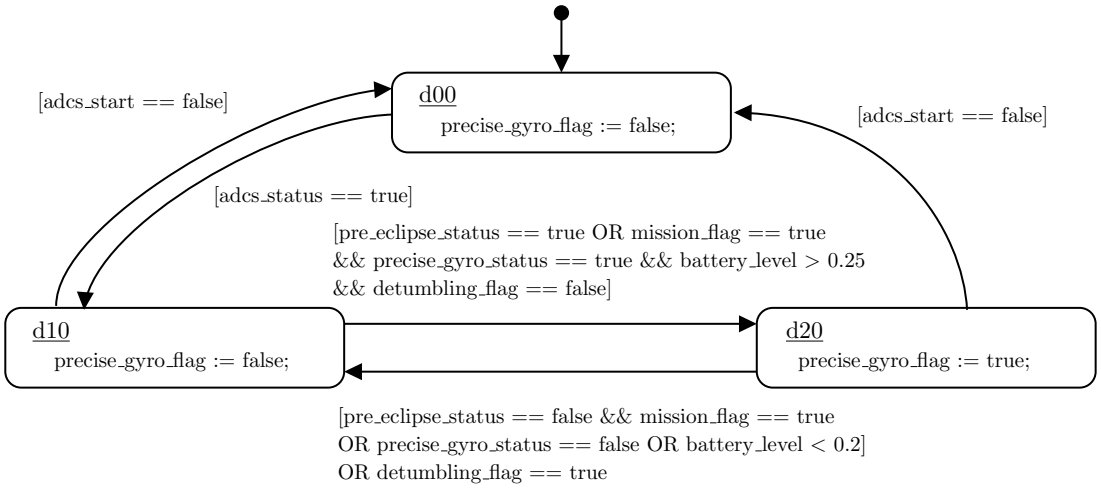


Figure 75: State Machine for the Precision Gyroscope

In Figure 75 is the state machine for the precision gyroscope. It is quite similar to the state machine of the Star Tracker, but instead of activating the gyro with an increase of the variable "*eps*", the gyro is activated by the variable "*pre_eclipse_status*", which is one of the two computed eclipse conditions. The eclipse conditions are set using GPS and orbit propagation data by the OBC, rather than being set by looking at the actual Sun, intentionally triggering the eclipse status sooner than the physical eclipse, and returning it to false after the eclipse period has

ended. The variable "*pre_eclipse_status*" is more conservative than the variable "*eclipse_status*", allowing the precision gyroscope to be turned on before the sun sensor is turned off. This holds the attitude estimation performed by the magnetometer and the sun sensor, which is better than the insufficient one performed by the magnetometer only. The use of the variable "*eclipse_status*" can be seen on the state machine for the sun sensor in Figure 76, where it is used to power the sensor off during the night. The logic of this sensor is also similar to the previous ones:

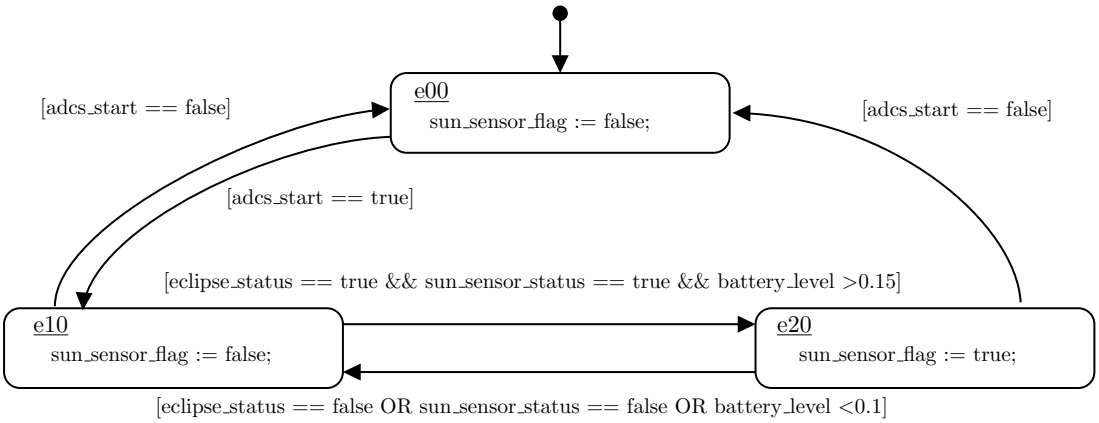


Figure 76: State Machine For Main Attitude Modes

The coarse gyro follows a simple logic, so its state machine is not represented: after the ADCS has started, the coarse gyroscope is always turned on until an eventual failure occurs, or energy levels become critical. Having the coarse gyroscope continuously functioning is a decision made by *Tekever* based on the fact that it consumes low power, it's not mission critical and because it can eventually be replaced by the precision gyro.

In Figure 77 is represented the state machine handling the behavior of the magnetometers and the magnetorquers. Before explaining this diagram, it is important to clarify the difference between two behavior options that can be chosen for a spacecraft with this technology: either ignore magnetometer readings while actuating the magnetorquers, or place the magnetometers in stand by while the magnetorquers are active. The first option should be more robust while the second option can save energy. This is a choice that depends on factors like the cycle times attributed to this state machine and the electronic properties of the magnetometer, and falls beyond the scope of this thesis. For this study it is assumed that the magnetometers are placed in stand by during magnetorquer actuation, which does not mean that it will be the case for *Infante*. If the continuous operation case were to be chosen, it would mean erasing the variable "*magnetometer_flag*" from the state machines. Then, the magnetorquer status would be used to tell the Kalman Filter that magnetometer measurements should be ignored.

The first active state of Figure 77 is bounded to the activation of any magnetometer, whose choice depends of the state machine "h". If the spacecraft is detumbling or desaturating the reaction wheels, if there is enough energy in the batteries, if the magnetorquers are available and if there has been enough time for the magnetometer to perform readings, a magnetorquer loop is executed, through states "f20" and "f30". In these states the magnetorquers act according to the active control algorithm, and then dither after some time, in order to avoid the effect of residual momentum.

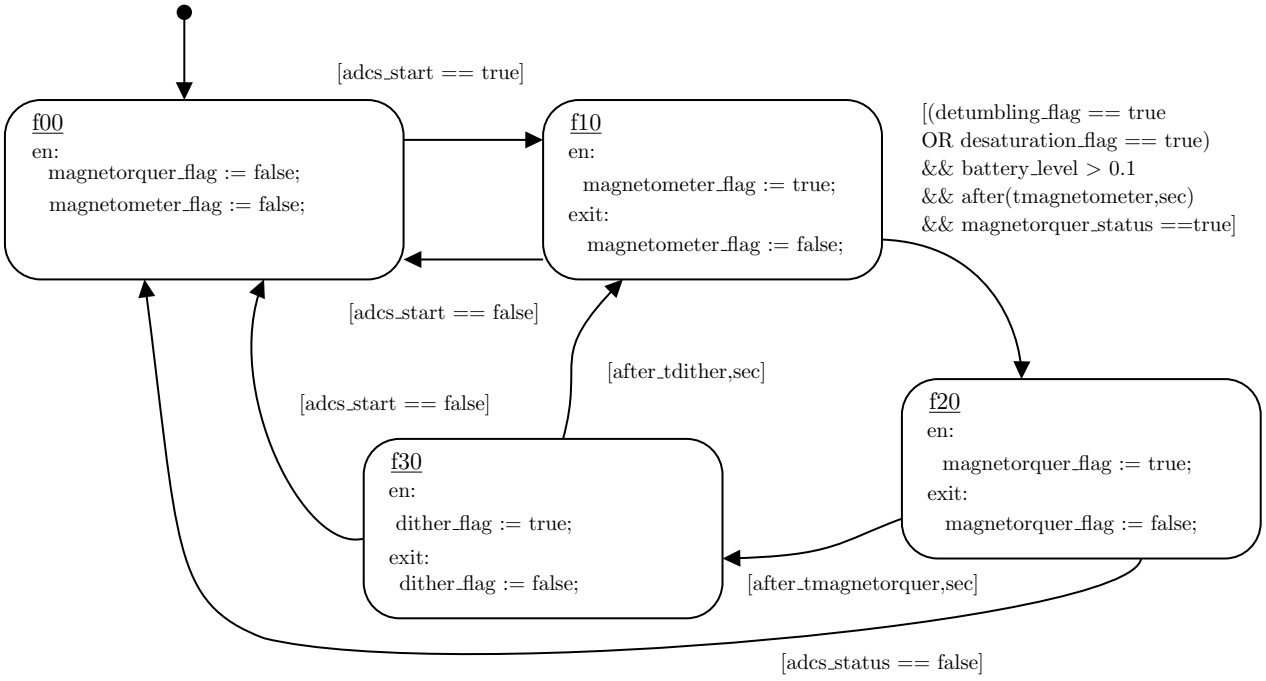
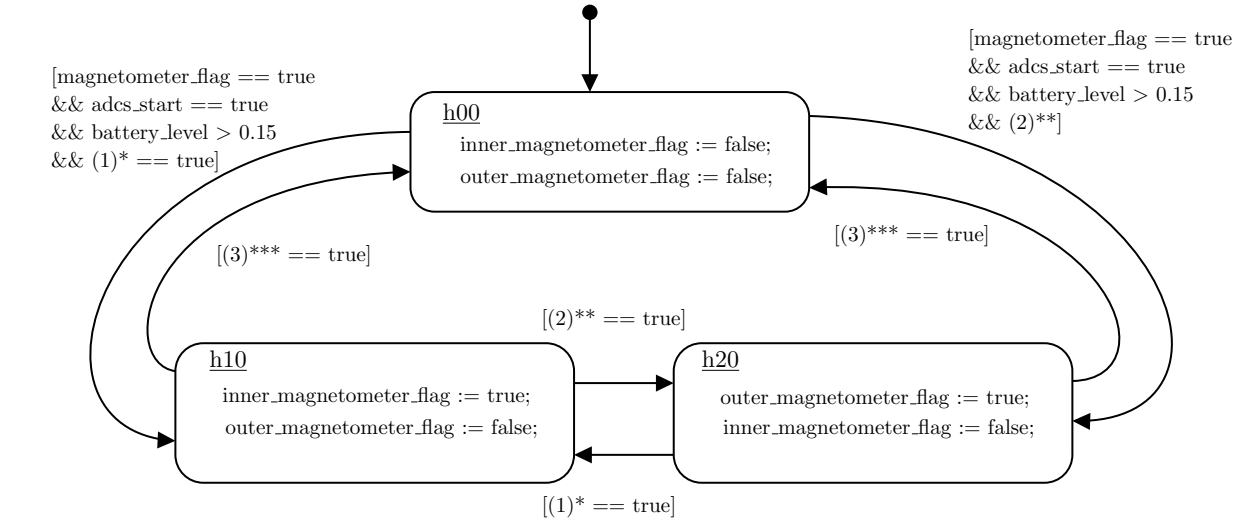


Figure 77: State Machine Handling Magnetometers and Magnetorquers

In Figure 78 is represented the state machine for the choice of magnetometers. One of the two magnetometers becomes active if the magnetometer flag is triggered and if there is enough battery. The inner and outer magnetometer are chosen depending if they fulfill conditions (1) and (2) respectively, which are mutually exclusive as long as both magnetometers don't fail, which is a part of condition (3). The inner magnetometer is chosen if the outer magnetometer has failed or hasn't been deployed successfully to the outside of the spacecraft. The outer magnetometer replaces the inner magnetometer upon failure or as soon as it's deployed.

Figure 79 represents the state machine for the desaturation of the reaction wheels. Desaturation is performed as long as there is a minimum amount of momentum in the reaction wheels' body space, if there is enough battery, if there is no current mission (since the magnetorquers can perturb precise attitude control) and if there is enough battery. The inverse, analogous case prohibits desaturation of the reaction wheels.

These eight, parallel state machines have shown to be sufficient to make the spacecraft fully autonomous, which is demonstrated in the next section. The reader is reminded that this is a global proposal for the ADCS logic and that a lot of details were ommited in the design of these state machines.



*(1): (outer_mag_deploy_status == false OR outer_magnetometer_status == false)
&& inner_magnetometer_status == true
**(2): outer_magnetometer_status == true && (outer_mag_deploy_status == true
OR inner_magnetometer_status == false)
***(3): (outer_magnetometer_status == false && inner_magnetometer_status == false)
OR adcs_start == false OR magnetometer_flag == false OR battery_level <0.1

Figure 78: State Machine for Inner and Outer Magnetometers

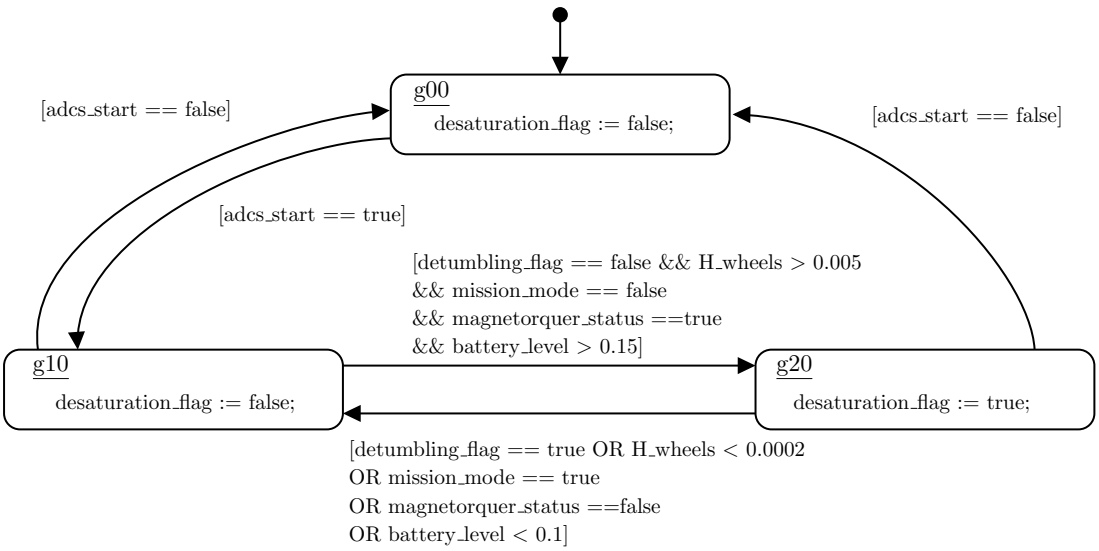


Figure 79: State Machine for Desaturation of the Reaction Wheels

12 Simulation Results

To demonstrate the concepts discussed on this thesis, a final simulation was designed to include all mission phases, including two SAR scans, scenarios such as loss of battery power and non-critical failures of the precision gyroscope and a reaction wheel.

This simulation starts arbitrarily at 22:15:00 of the day 10th of July 2019, with the satellite being injected into orbit with the angular velocity of $\omega_0 = [1.72; 2.29; 2.29]$ degrees per second. Its starting point is at -50^o of latitude, performing a circular sun synchronous orbit with the characteristics mentioned previously. It has two SAR missions scheduled for 12:49:50 (Universal Time) on the 11th of July, when it will be flying over Portugal, and for 06:15:00 (Universal Time) on the 12th of July when it will be flying over Asia. After the first SAR, the spacecraft will experience periods of low power at 20 and 22 hours of simulation time. At 25 hours of simulation time the precision gyroscope will fail, followed by a failure of reaction wheel number four, at 27 hours of simulation time, just before the second SAR scan is performed.

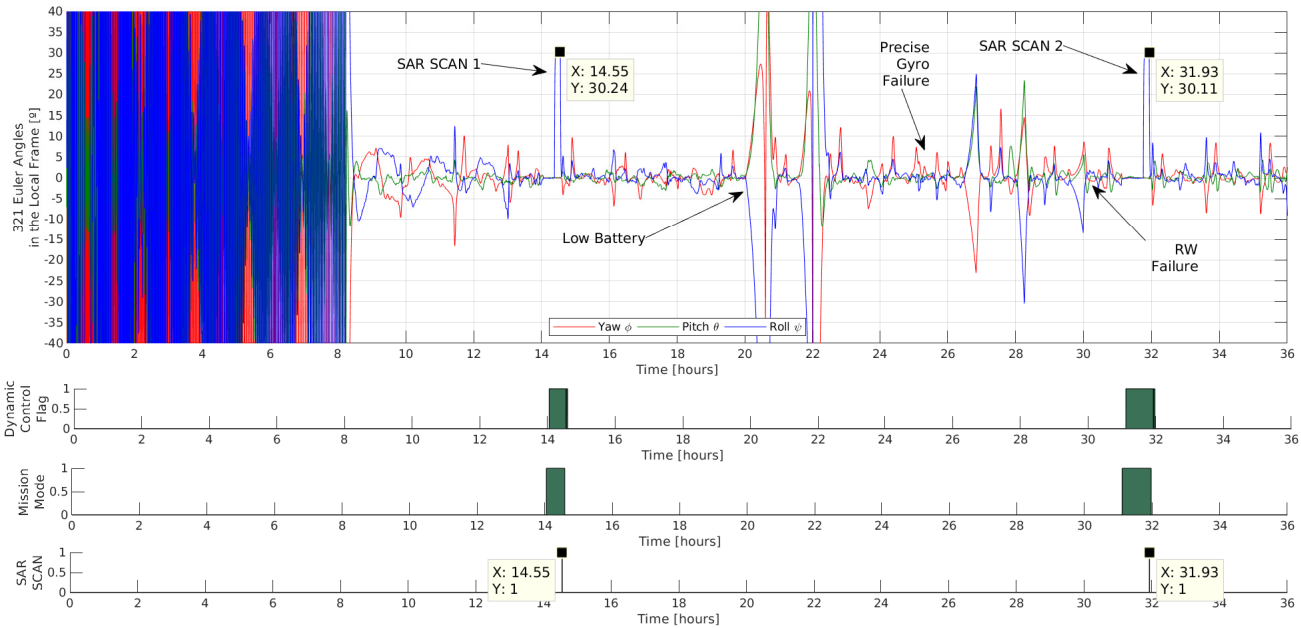


Figure 80: Attitude of the Spacecraft in the Local Frame, states of internal variables

The first Figure 80 shows the 321 Euler Angles of the spacecraft attitude regarding the local frame, which is the most intuitive way to perceive how the spacecraft behaves during the simulation. In Figure 80 are also the states of the dynamic control, mission mode and SAR scan variables, showing the two moments when missions are performed. In the first eight hours of simulation it is visible that the satellite is experiencing detumbling, and stabilizes around this time. The moments where the satellite rolls 30^o for the SAR scans are evident in the Euler Angles plot, when the other variables are also active. It is also evident the effect of the power losses, as the satellite slowly tumbles when actuators and attitude sensors are turned off. The gyroscope failure also causes a visible loss of attitude precision during nightime, as well as the reaction wheel failure which causes a temporary error around 5^o. Despite these drawbacks, the spacecraft is still able to keep the attitude within the 0.25^o requirement during SAR, as illustrated by the data cursors.

On Figure 81 is represented the control and estimation errors during the simulation. Figure 82 is a time diagram of the different control laws and the evolution of the variable ε, representing the filter's uncertainty, and in Figures 83 and 84 is a time diagram of the attitude sensors and eclipse conditions.

Turning our attention to Figure 82, it can be seen that the detumbling control law is used during the first 8 hours, being replaced by the regulation and desaturation control algorithms as soon as the satellite slows down enough. The variable ε is extremely high in the beginning of the simulation because the only sensors turned on are the Sun Sensor (during day), the inner magnetometer and the coarse gyroscope, which can be seen in Figure 83. The inner and outer magnetometers' states are illustrated in a lighter color in the moments where the magnetorquers are active (detumbling and desaturation control) because they are being turned on and off every few seconds, due to the duty cycle mentioned earlier.

When the satellite is stabilized after 8 hours of simulation, it enters its stand-by state: the outer magnetometer replaces the inner magnetometer, the Sun Sensor is turned on during daytime and the precise gyro is turned on during night time, as seen on Figure 83. During this phase, the reaction wheels are desaturated, reducing the momentum left from the initial detumbling or that was absorbed by the disturbances throughout time. It can be seen in Figure 82 that this is active for most of the time, being turned off during SAR and low power mode. Similarly, regulation is always active except for the low energy mode, and when the variable ε becomes too high. Comparison of Figure 81 with Figure 80 shows that this strategy of turning off regulation with ε is successful at minimizing attitude error during high uncertainty, since the actual error is considerably less than the estimation and control error at certain moments. Moments like this occur during the night time after the failure of the precise gyroscope, because the attitude information received becomes insufficient, and the coarse gyro is unable to hold attitude for long. The Star Tracker is also turned on to compensate this lack of knowledge, as seen on Figures 82 and 83, but not for long to save energy. It can be seen on Figure 81 that attitude precision is mostly adequate, except for the power outages and the failure of the precise gyroscope.

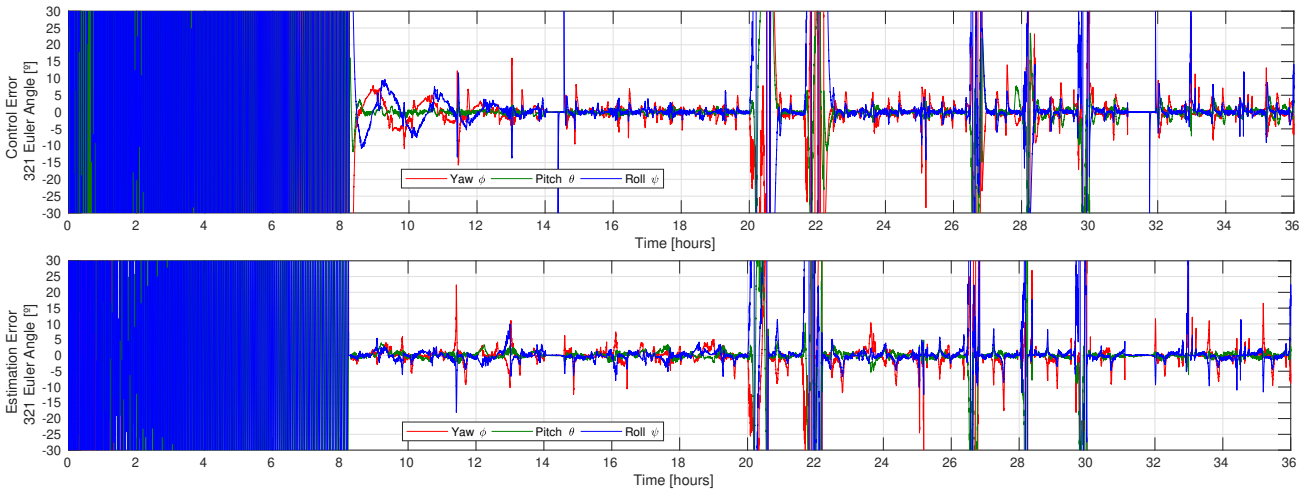


Figure 81: Control and Estimation Errors represented in 321 Euler Angles

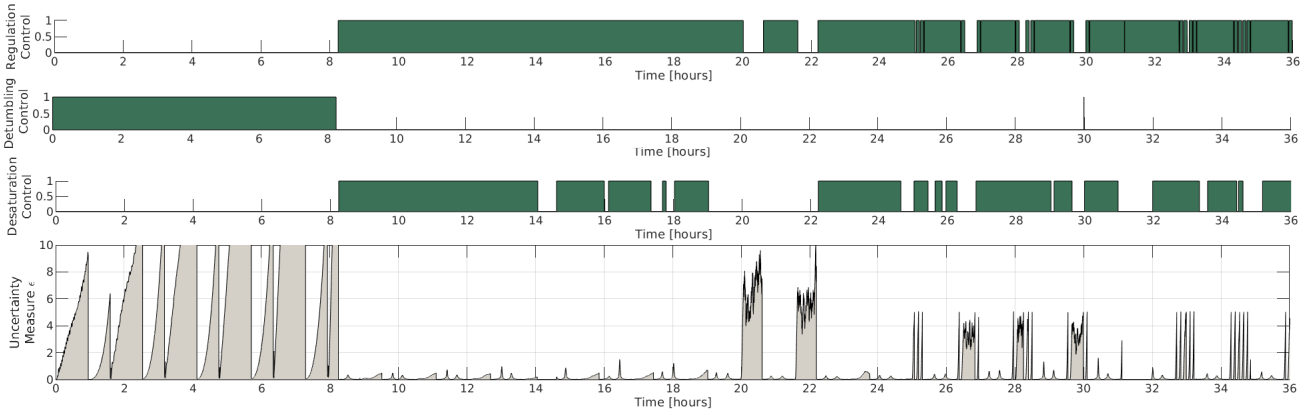


Figure 82: Time Diagram of Control laws and uncertainty measure "ε"

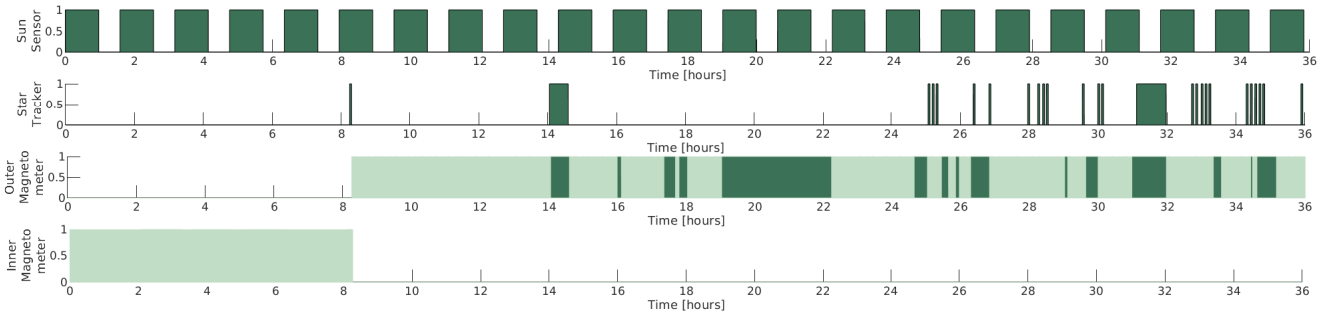


Figure 83: Time Diagram of Attitude Sensors

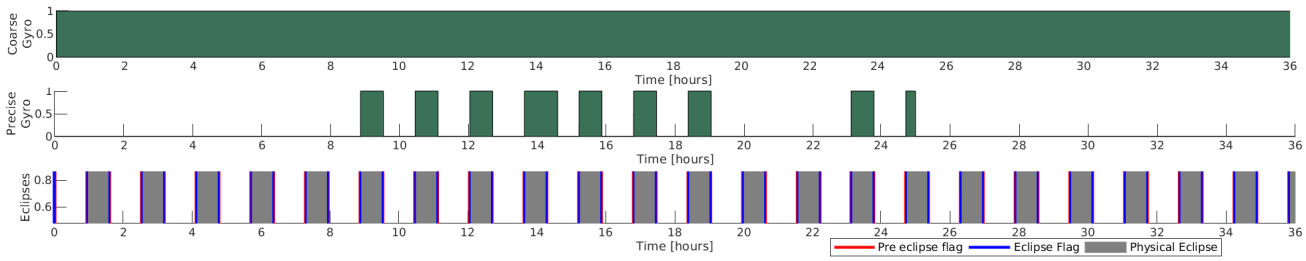


Figure 84: Time Diagram of Gyroscopes and the Occurrence of Eclipse

In Figure 84 can be seen how the coarse gyroscope is kept on for the entirety of the mission, and how the precise gyroscope and the sun sensor activate with the different eclipse variables.

On Figure 85 is plotted the momentum of the reaction wheels in the body and the wheel space during the simulation. The reaction wheels respond with the activation of the regulation control, and absorb the rest of the momentum that was not absorbed by the magnetorquers. This momentum, represented on the body frame, changes direction during several orbits, due to the conservation of momentum in the inertial frame, until the desaturation algorithm drives it to zero. On the wheel space, as the global momentum approaches zero, each of the four wheels

approaches its optimal speed at $H_{RW} = 0.01$ [Nms]. The SAR roll causes temporary spikes on the wheel's momentum, since they supply torque to rotate the spacecraft. It is visible how upon the failure of reaction wheel number 4 at hour 27, the other three wheels lose their null space, so they lose the ability of having a non-zero speed for a global momentum of zero. It can be seen on Figure 80 that the error introduced by a reaction wheel failure is only around 15^o.

On Figure 86 is visible a world map with the orbit trajectories and the targets scanned during this simulation.

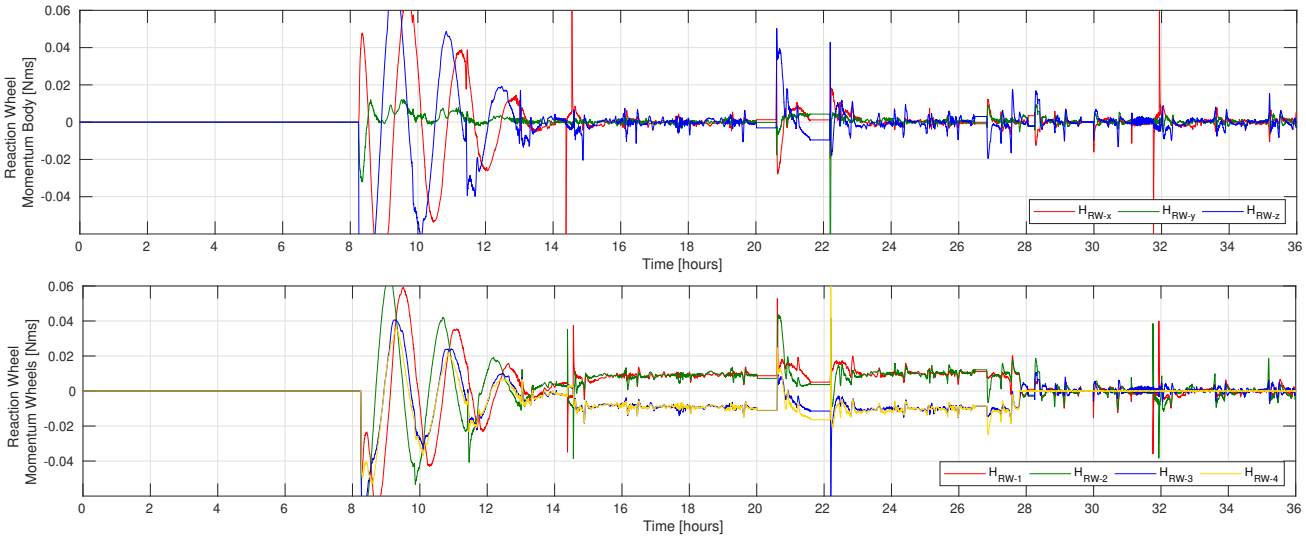


Figure 85: Reaction Wheel Momentum on Body and Wheel Space

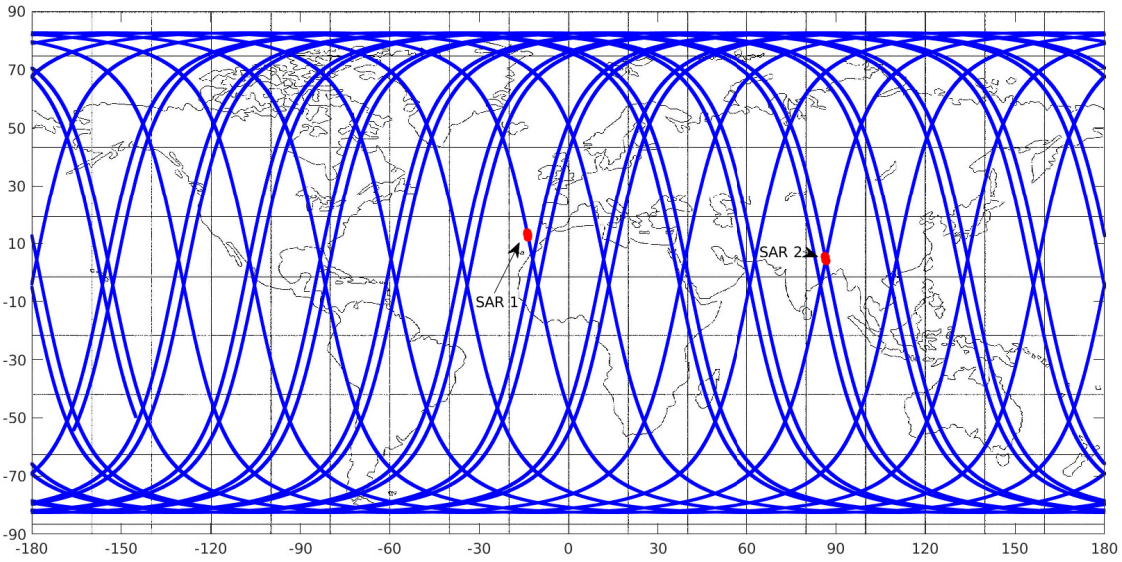


Figure 86: World Map with SAR targets and Orbits

13 Conclusions

In this dissertation were proposed two main objectives: a global solution for the ADCS of the satellite *Infante*, and the analysis of a new type of control architecture, called decentralized architecture, for which the *Infante* satellite served as a case study. To validate the proposed control algorithms and the architecture, the development of a simulation model is also necessary.

13.1 Contributions

As seen by the results of the previous section, the proposed solution for the ADCS of *Infante* has fulfilled the requirements given on the start of this dissertation: the control algorithms developed are able to detumble the spacecraft, maintain a reference attitude, desaturate the reaction wheels and keep them away from null velocity states. The first problem of detumbling was solved with time and energy efficiency in mind, optimizing currently used algorithms for this case. The regulation control problem was solved using the dot distribution inherent to the decentralized architecture, for which it was shown to be stable, efficient and robust. The proposed solution for the desaturation of the reaction wheels was also designed keeping in mind time and energy efficiency using a traditional algorithm. A new way to drive the reaction wheels to their desired state was also proposed, since the traditional method of directly applying torque in the null space cannot be used with decentralized architecture. This new method is functional and of simple implementation, but riskier and slower than the current method. It was also proposed a simple programming logic using state machines, which ensured mission success, energy economy and robustness to non-critical failures. A law for desaturating the reaction wheels using attitude, based on an rigorous analysis of the disturbances, was proposed but unfortunately was not implemented.

A *Simulink* model was developed in the scope of this thesis, in an effort to validate the proposed solutions. This model includes, among other things, rigorous attitude dynamics, classical orbital propagation, complete environment models from third parties, precise astronomical mechanics, self-modeled attitude sensors and actuators, a Multiplicative Extended Kalman Filter and an ADCS state machine capable of simulating real missions. This model can be used to not only aid the development of *Infante* but also to simulate any other kind of small to medium satellites.

The topic of decentralized architecture was proposed, explained and analyzed in depth. It has been shown that using this control architecture provides an increase in development flexibility at the cost of a small change in effective control behavior. This change was formally described by a "distortion matrix" and its consequences were analyzed for the most common reaction wheel assemblies. For the classic 30^o four wheel pyramid assembly, the damping coefficient and the natural frequency increased by 22.5% in the xy directions; for a classical six wheel assembly it increased by 22.5% and 50% in xy and z directions. This is interpreted as a change in the control gains, that has low impact on system behavior, since the increase in dampness compensates the increase in the natural frequency, and since the control action becomes saturated easily. This was confirmed by using the developed *Simulink* model to compare the dot distribution with the pseudoinverse method.

As stated previously, the goal of this architecture is to allow a more flexible methodology in the development of an ADCS. Technicians can add, modify or remove reaction wheels without compromising the global design of the control system, which is highly suitable for using with a design-test-iterate philosophy.

13.2 Future Works and Recommendations

The next step regarding the topic of decentralized architecture is to continue the mathematical description of the other assemblies that were not described here, like asymmetrical configurations and non-pyramid configurations. It would also be productive to evaluate if there are other "blind" distributions, other than the dot product distribution, that might be more efficient and flexible. Finally, it would be important to see what control algorithms, other than the presented PD, are stable using this architecture.

The next step for *Infante* is continuing the development of its ADCS, whether this is from a theoretical point of view or if it is actual implementation. A topic that I would have liked to have contributed to, is the study of the solar panel flexibility. This is a critical analysis that cannot be ignored: if the solar panels' natural frequency is too close to the controller's natural frequency, they will resonate and most likely cause the solar panels to break. Preliminary analysis from *Tekever* has shown that this is unlikely but a deeper study must be made.

The developed model uses some crude approximations, which can be improved in order to provide more realistic results. Such approximations include: the use of a classical propagator, neglecting solar light pressure, small simulation step size, perfect on-board magnetic model, etc.

References

[1] "United nations office for outer space affairs annual report 2018," United Nations, Tech. Rep., 06 2019.

[2] J.-M. Bockel, "The future of the space industry," North Atlantic Treaty Organization, Tech. Rep., 11 2018.

[3] "22nd state of the satellite industry report," Satellite Industry Associations, Tech. Rep., 05 2019.

[4] "European space agency annual report 2016," European Space Agency, Tech. Rep., 2017.

[5] "ESA", ""what is space 4.0?"," "2016". [Online]. Available: "https://www.esa.int/About_Us/Ministerial_Council_2016/What_is_space_4.0"

[6] H. Khalil, N. Abas, and S. Rauf, "Energy efficient wsn-sat system for surveillance of satellites," *Wireless Sensor Network*, vol. 7, pp. 129–136, 10 2015.

[7] "Oneweb", ""one web satellite - revolutionizing the economics of space"," "2019". [Online]. Available: "https://onewebsatellites.com/"

[8] "SpaceX", ""starlink mission"," "2019". [Online]. Available: "https://www.spacex.com/news/2019/05/24/starlink-mission"

[9] F. L. Markley and J. L. Crassidis, *Fundamentals of Spacecraft Attitude Determination and Control*. Springer, 2014.

[10] NASA, "State of the art - small spacecraft technology," 12 2018. [Online]. Available: https://sst-soa.arc.nasa.gov/

[11] C. M. Pong, "High-precision pointing and attitude estimation and control algorithms for hardware-constrained spacecraft," Ph.D. dissertation, Massachussets Institute of Technology, Department of Aeronautics and Astronautics, 2014.

[12] R. Burton, S. Weston, E. Agasid, and R. Shimmin, "State of the art in guidance, navigation and control: A survey of small satellite gnc components," 02 2016.

[13] E. Lefferts, L. Markley, and M. Shuster, "Kalman filtering for spacecraft attitude estimation," 01 1982.

[14] F. Reichel, P. Bangert, S. Busch, K. Ravandoor and K. Schilling, "The attitude determination and control system of the picosatellite uwe-3," *IFAC Symposium on Automatic Control in Aerospace*, pp. 271–276, 9 2013.

[15] V. Grigore, "Unscented kalman filters for attitude and orbit estimation of a low earth orbit cubesat," Ph.D. dissertation, KTH Royal Institute of Technology, 2015.

[16] J. Zhou, "Attitude determination and control of the cubesat mist," Ph.D. dissertation, KTH Royal Institute of Technology, 2016.

[17] D. Navarro Tapia, "Structured h-infinity and linear parameter varying control design for the vega launch vehicle," 07 2017.

[18] A. Capua, A. Shapiro, and D. Choukroun, "Spacecraft attitude control using nonlinear h-infinity output-feedback," 08 2013.

[19] N. Guy, "A new attitude control design approach for flexible satellites," 09 2013, pp. 500–505.

[20] Y. Yang, "Analytic lqr design for spacecraft control system based on quaternion model," *Journal of Aerospace Engineering*, vol. 25, pp. 448–453, 07 2012.

[21] R. Chiang and J.-S. Jang, "Fuzzy logic attitude control for cassini spacecraft," 07 1994, pp. 1532 – 1537 vol.3.

[22] A. Walker, P. Putman, and K. Cohen, "Fuzzy logic attitude control of a magnetically actuated cubesat," in *AIAA Infotech at Aerospace (I at A) Conference*, 08 2013.

[23] H. Leduc, C. Pittet, and D. Peaucelle, "Adaptive attitude control of a microsatellite during payload deployment," 07 2017, pp. 56–61.

[24] W. Mackunis, K. Dupree, N. Fitz-Coy, and W. E. Dixon, "Adaptive satellite attitude control in the presence of inertia and cmg gimbal friction uncertainties," *The Journal of the Astronautical Sciences*, vol. 56, 08 2007.

[25] L. Lindblad, "Modelling and Simulation of GNC/AOCS Systems for Conceptual Studies," Master's thesis, Royal Institute of Technology, Department of Mechanics SE-100 44 Stockholm, Sweden, 2013.

[26] United States Naval Observatory, *The Astronomical Almanac*. Nautical Almanac Office, 2019.

[27] National Geospatial-Intelligence Agency, "World geodetic system," 2019. [Online]. Available: "https://gisgeography.com/wgs84-world-geodetic-system/"

[28] C. D. Brown, *Elements of Spacecraft Design*. American Institute of Aeronautics and Astronautics inc., 2002.

[29] D. M. Prieto, "Spacecraft drag modeling," Ph.D. dissertation, School of Mechanical, Aerospace and Civil Engineering, University of Manchester, 2017.

[30] M. M. Moe, S. D. Wallace, and K. Moe, "Recommended drag coefficients for aeronomic satellites," *The Upper Mesosphere and Lower Thermosphere: A Review of Experiment and Theory*, pp. 349–356, 1995.

[31] K. Moe and M. M. Moe, "Gas-surface interactions in low-earth orbit," *AIP Conference Proceedings*, vol. 1333, pp. 1313–1318, 05 2011.

[32] U.S. Naval Research Laboratory, "Empirical modeling of the upper atmosphere: Nrlmsise-00, hwm07, and g2s," 2019. [Online]. Available: "https://www.nrl.navy.mil/ssd/branches/7630/modeling-upper-atmosphere"

[33] National Geospatial-Intelligence, "The world magnetic model," 2019. [Online]. Available: "https://www.ngdc.noaa.gov/geomag/WMM/DoDWMM.shtml"

[34] NASA Earth Obsertavory, "Measuring earth's albedo," 2019. [Online]. Available: "https://earthobservatory.nasa.gov/images/84499/measuring-earths-albedo"

[35] A. G Quinchia, G. Falco, E. Falletti, F. Dovis, and C. Ferrer, "A comparison between different error modeling of mems applied to gps/ins integrated systems," *Sensors (Basel, Switzerland)*, vol. 13, pp. 9549–88, 08 2013.

[36] Q. Lam, N. Stamatakos, C. Woodruff, and S. Ashton, "Gyro modeling and estimation of its random noise sources," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, vol. 5562, 08 2003.

[37] Bosch, *BMG250 Low noise, low power triaxial gyroscope*, 7 2016.

[38] Sensonor, *STIM210 Multi-Axis Gyro Module*, 8 2018, rev. .18.

[39] ST, *High-accuracy, ultra-low-power, 3-axis digital output magnetometer*, 9 2017, rev 1.

[40] Bosch, *BMM150 Geomagnetic Sensor*, 4 2019, revision 1.2 042019.

[41] KU Leuven, *KU Leuven Star Tracker*.

[42] M. Tuchin, A. Biryukov, M. Nickiforov, M. Prokhorov, and A. Zakharov Sternberg, "On random and systematic errors of a star tracker," in *27th Annual AIAA/USU Conference on Small Satellites*, 01 2013.

[43] Maxon, *EC 20 Flat Motor ϕ20mm, brushless, 5 Watt*, 5 2018.

[44] L. Markley, "Attitude error representations for kalman filtering," *Journal of Guidance Control and Dynamics*, vol. 26, pp. 311–317, 03 2003.

[45] T. B. G. Juchnikowski1 and J. Lisowski, "Optimal control gain for satellite detumbling using b-dot algorithm," in *2nd CEAS Specialist Conference on Guidance, Navigation and Control*, 4 2013.

[46] R. Fonod and E. Gill, "Magnetic detumbling of fast-tumbling picosatellites," in *69th International Astronautical Congress*, 10 2018.

[47] M. Leomanni, "Comparison of control laws for magnetic detumbling," 10 2012.

[48] A. Kron, A. St-Amour, and J. de Lafontaine, "Four reaction wheels management: Algorithms trade-off and tuning drivers for the proba-3 mission," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 19, pp. 9685–9690, 01 2014.