TÉCNICO LISBOA

Control Algorithm for ISTsat-1

Diogo Pessanha Neves

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisors: Prof. Paulo Jorge Coelho Ramalho Oliveira
Prof. Rui Manuel Rodrigues Rocha

Examination Committee

Chairperson: Prof. Fernando José Parracho Lau
Supervisor: Prof. Paulo Jorge Coelho Ramalho Oliveira
Member of the Committee: Prof. Rita Maria Mendes de Almeida Correia da Cunha

July 2019


IST SAT ONE
INSTITUTO SUPERIOR TÉCNICO • UNIVERSIDADE DE LISBOA

To my family, for their unwavering support and infinite patience through all these years.

Acknowledgments

I would like to thank my supervisor professor Paulo Oliveira for his guidance, time and knowledge passed onto me during this work.
I would like to thank my supervisor professor Rui Rocha for his unbelievable commitment to the ISTsat project, its students and for the iron will to guide them.
Finally to my ISTsat-1 colleagues and friends for all their support, aid and stimulating arguments but specially for the hours spent together working hard as a real team, turning my time in the project into a very rewarding experience.

Resumo

O ISTsat-1 é um satélite pertencente à classe de cubesats que foi concebido por estudante e alunos do Instituto Superior Técnico (IST) ao abrigo do programa educacional *Fly Your Satellite* da Agência Espacial Europeia (ESA). O objectivo desta missão é fazer seguimento dos sinais ADS-B de aviões e a respectiva caracterização. Uma das prioridades da plataforma será garantir que o satélite aponta para a direcção desejada de forma a cumprir a missão, assim como garantir que o satélite não adquire rotação excessiva. Este trabalho foca-se no desenvolvimento do Sistema de Determinação e Controlo de Atitude que garanta estas condições. O sistema foi desenvolvido para ser integrado na plataforma física previamente desenvolvida para o computador de bordo. O obstáculo que se pretende vencer é garantir que o satélite cumpre os requisitos de orientação impostos pela missão usando uma plataforma de baixo custo e de baixo consumo energético. Para avaliar possíveis soluções usou-se a ferramenta *Simulink* que permitiu modelar o satélite e o ambiente físico a que estará sujeito. Em seguida estudou-se o problema da atitude testando algoritmos de estimação de atitude com baixos requisitos computacionais. Finalmente a solução para a determinação de atitude permitiu estudar soluções para o problema de controlo de atitude também estas focadas na eficiência computacional. Os resultados mostram que é possível usar soluções rápidas, eficientes e de baixo custo para desenvolver satélites com baixos requisitos de orientação.

Palavras-chave: Satélite, CubeSat, ISTsat, Atitude, Estimação, Controlo

Abstract

The ISTsat-1 is a cubesat satellite developed by students from Instituto Superior Técnico (IST) under the ESA educational program Fly Your Satellite. The objective of this mission is to monitor and characterize the ADS-B signals from aircraft. One of the priorities of the satellite platform is to point it into the desired direction with a certain level of accuracy to accomplish the mission as well as to ensure the satellite does not get excessive rotation. This work focuses the development of the Attitude Determination and Control System which will ensure these conditions. The system was developed to be integrated into the physical platform previously designed for the onboard computer. The challenge to surpass is to ensure the satellite reaches the pointing requirements posed by the payload while using a low-cost and low-power platform. To evaluate the possible solutions the Simulink tool was used, allowing to model the satellite and the environment to which the satellite will be subjected. Afterwards, the problem was studied using computational light attitude estimation algorithms. Finally, the attitude determination solutions allowed to study control solutions also focused on computational efficiency. The results demonstrate that it is possible to use quick, efficient and low-cost solutions to develop satellites with low orientation requirements.

Keywords: Satellite, CubeSat, ISTsat, Attitude, Estimation, Control

Contents

List of Tables

List of Figures

Nomenclature

Reference Frames Symbols

$\hat{\langle}$	Normal vecotr of Orbit plane.
$\mathcal{B}$	Body Frame.
$\mathcal{E}$	Earth Centered, Earth Fixed Frame (ECEF).
$\mathcal{I}$	Earth Inertial Frame (ECI).
$\mathcal{O}$	Orbit Frame.
ϕ_{lla}	Spherical Coordinate Longitude of ECEF.
θ_{lla}	Spherical Coordinate Latitude of ECEF.
b_1 , b_2 , b_3	Orthonormal basis of Body Frame.
e_1 , e_2 , e_3	Orthonormal basis of Earth Centered, Earth Fixed Frame (ECEF).
i_1 , i_2 , i_3	Orthonormal basis of Earth Inertial Frame (ECI).
o_1 , o_2 , o_3	Orthonormal basis of Orbit Frame.
r_{lla}	Spherical Coordinate Radius of ECEF.

Matrices Symbols

$[R_i^j]$	Rotation matrix from frame i to frame j.
$[]_\times$	Matrix representation of the cross product left vector
$I_{3\times3}$	3 by 3 identity matrix
A	Attitude Matrix.

Vectors Symbols

$\bar{e}_r$	Position vector normalized
$\mathbf{n}$	Rotation vector axis
$\mathbf{q}$	Quaternion vector

$\mathbf{V}$	Satellite velocity in ECI frame.
$\otimes$	Quaternion product
θ	Angular displacement of rotation vector
v	Rotation vector
q_ϵ	Quaternion imaginary vector
q_η	Quaternion real coordinate

Glossary

ADCS	Attitude Determination and Control System
ADS-B	Automatic dependent surveillance-broadcast
COSPAR	Committee on Space Research
CSS	Coarse Sun Sensor
ECF	Explicit Complementary Filter
EGM2008	Earth Gravitational Model 2008
EGM96	Earth Gravitational Model 1996
EPS	Electric Power System
FOV	Field of vision
GPS	Global Positioning System is satellite based navigation system.
IAGA	International Association of Geomagnetism and Aeronomy
IGRF	International Geomagnetic Reference Field
IMU	Inertial Measurement Unit
ISS	International Space Station
LASER	Light Amplification by Stimulated Emission of Radiation
LEO	Low Earth Orbit
MCU	Micro-Controller Unit
MEKF	Multiplicative Extended Kalman Filter
OBC	On-Board Computer
PSD	Power Spectral Density
PWM	Pulse-Width Modulation
RMSE	Root Mean Square Error
RMS	Root Mean Square
S/C	Spacecraft
w.r.t	With Respect To

Chapter 1

Introduction

1.1 Motivation

The satellites are unmanned vehicles that flight through space without the possibility of human direct intervention. As such, the Attitude Determination and Control System plays a crucial role in tasks related to the satellite orientation, e.g. pointing the communication antennas to the Earth or pointing the payload devices to the desired direction. These are some of the most common critical actions the ADCS must ensure autonomously throughout the satellite life and the main reasons why most of the satellites include one. Although the satellites fly in space they are not exempt from external torques and forces and even despite them the satellite must point in the right direction. To provide an estimation of towards where the satellite is oriented a set of sensors is used to gather data which will then be used in estimation algorithms to guess the satellite attitude. The work around the attitude estimation, got a major boost due the space race between U.S.A. and the U.S.S.R. during the Cold War. After the war ended space agencies continued to design space exploration missions, driving the growth of the satellite attitude estimation technology along with it.

Most launchers were built to put in orbit very large satellites and most of the times they were used the payload was smaller than what they were design for. To monetize the available payload space companies started to sell the remaining cargo space not used by the main payload providing a piggyback launch. To this smaller payload a proportional smaller price was associated allowing for entities to launch smaller budget satellites promoting the development of small satellite technology. The reduced cost of the smaller satellite open the doors of this exclusive industry, year by year more space companies are created.

The small satellites categories according to their size are established as: the biggest minisatellite which comprises of satellites between $100\,kg$ to $500\,kg$, then the microsatellite for satellites between $100\,kg$ to $500\,kg$, the nanosatellite designating the satellites in the $10\,kg$ to $1\,kg$ range and the smallest the picosatellite for satellites below the $1\,kg$ threshold.

In 1999, the California Polytechnic State University along with its partners created a nanosatellite standard to facilitate the access for universities to acquire space components. By creating a standard,

companies could now provide off-the-shelf compatible components to integrate satellites decreasing the design time and overall cost. CalPoly established the geometric unit as a $10\,cm$ cube with up to $1.33\,kg$, due to its geometry this format was designed as CubeSat. The development of this standard promote the industrial and academic growth of space technology, as seen in the following graph the yearly number of launched satellites has been growing.

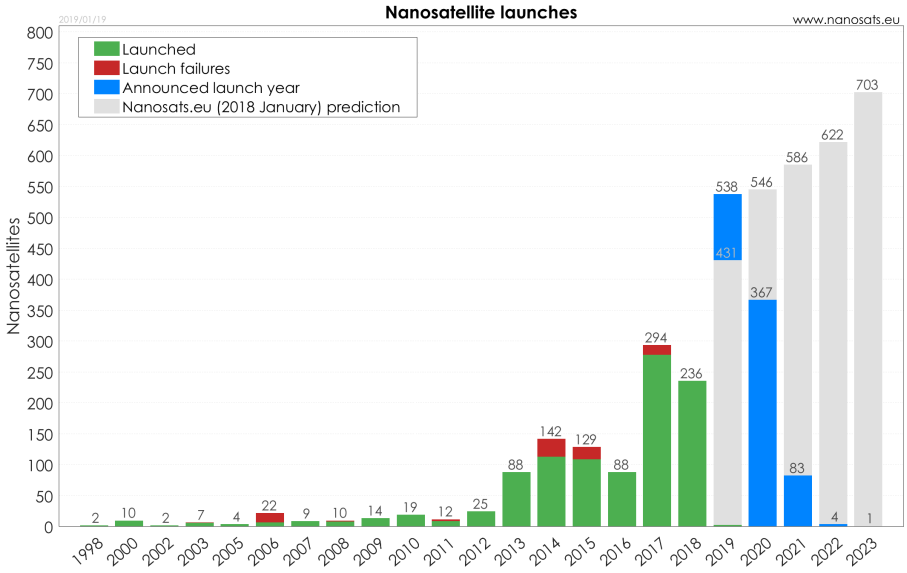

Figure 1.1: Number of launched nanosatellites per year (taken from Nanosats Database at www.nanosats.eu)

1.2 Problem Statement

The ISTsat-1 satellite is the first satellite mission of a group of students from the Instituto Superior Técnico called NanoSat. The purpose of this mission is to study the Automatic Dependent Surveillance-Broadcast (ADS-B) signals from low earth orbit (LEO). The ADS-B is an automatic signal that broadcasts the aircraft position posing as a monitoring signal for ground stations and other aircraft. With the increasing air traffic worldwide the transmission of ADS-B is becoming mandatory for commercial aircraft.

Being a 1U cubesat, the ISTsat-1 has serious geometry limitations i.e. needs to use solar panels to replenish its energy but the small area available reduces the available energy per orbit. As a consequence the power budget of each system is limited and the design of each system is centred in maximising energy efficiency. The average power consumption per orbit must be lower than $40\,mW$. The communication subsystem also has some demands for the ADCS system, namely to establish a good communication link the satellite must not surpass the limit angular rate of $5\,^o/s$. ISTsat-1 payload antenna was design to cover a $600\,km$ diameter area and it needs some insurance the satellite is pointing to NADIR to be able to characterize the ADS-B signals thus it was established the pointing error should be less than $20\,^o$. At $400\,km$ this error limit corresponds to a deviation of $145.6\,km$ from the NADIR point. Lastly,

there are some concerns regarding tumbling after being launched and while not pointing to NADIR. Consequently, the ADCS must have a mode where it only reduces the excess of angular rotation. The actual tumbling rate after launch is unknown. However it is known the satellite will be launched from ISS through JAXA robotic arm. This will limit the tumbling to a very low value but was specified the satellite should be able to stop tumbling from at least $30^o/s$ angular rate. To save space, during the mission design it was established the ADCS was to be embedded into the Onboard Computer (OBC) module. Both system will share the same computational resources using a real time operating system to manage each subsystem task. This fusion emphasizes the efficiency of ADCS algorithms options.

1.3 Thesis Outline

The proposed approach for this thesis started with a briefly technical background review presenting the coordinate frames that will be used in chapter 2.
Then, in chapter 3, a review of the basic mechanics involved in 3D rigid body dynamics was presented highlighting relevant specific orbital flight environment forces which were coded into the orbital simulator. Next, in chapter 4, followed an explanation about the physical satellite model focusing the most important components of the ADCS and how their characteristics were mathematically modelled into the simulator. Having finished presenting the mechanics background needed to simulate the satellite movement in chapter 5 some solutions for the attitude problem were presented, starting with the simpler solutions, then including dynamic models and finally gyro bias estimation.
After solving the attitude problem the focus was steered into solving the control problem. In chapter 6 this problem was divided into two parts based on the mission requirements: detumbling and NADIR pointing.
After all the theoretical presentation of the solutions the attitude determination and control methods were finally tested in a simulator whose configuration and results were discussed in chapter 7.
Lastly the conclusions from the results and remaining work were discussed in chapter 8 with a small remark for future work and possible future approaches.

Chapter 2

Theoretical Background

2.1 Reference Frames

To understand the context of the three dimensional space dynamics quantities involved in this thesis and also to describe them accurately four different reference coordinate frames will be introduced based on the literature [1], [2], [3].

2.1.1 Earth Centered Inertial

The Earth-Centered-Inertial (ECI) frame is centred in Earth and it is represented through its basis $\mathcal{I} = \{i_1\,,\, i_2\,,\, i_3\}$. The i_1 vector belongs to the ecliptic plane and it is pointing to the vernal equinox direction, i_3 vector is aligned with the Earth spin axis and it points toward the North pole, finally the i_2 completes the right-handed orthogonal reference frame.

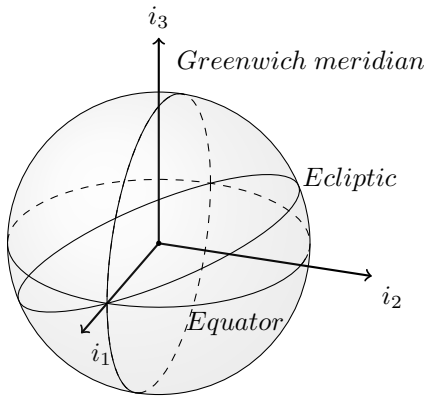

Figure 2.1: Earth Centered Inertial Reference Frame.

Although the name suggest that the ECI is not a true inertial frame, due to a precession in the ecliptic pole which usually is neglected as the period of that precession, around 26000 years, it is much greater than the lifetime discussed in most space flights.

2.1.2 Earth Centered Earth Fixed

The Earth-Centered-Earth-Fixed reference frame is a frame also centred in the Earth and will be represented with the basis $\mathcal{E} = \{e_1\,,\, e_2\,,\, e_3\}$. The e_3 vector is aligned with Earth spin axis, pointing to the North pole, the e_1 vector points towards the intersection of prime meridian (Greenwich) with the equatorial plane, the e_2 vector points in the direction that completes the right-handed orthogonal reference frame. As it can be seen in figure 2.2, the ECEF frame relates to the ECI through a rotation in the

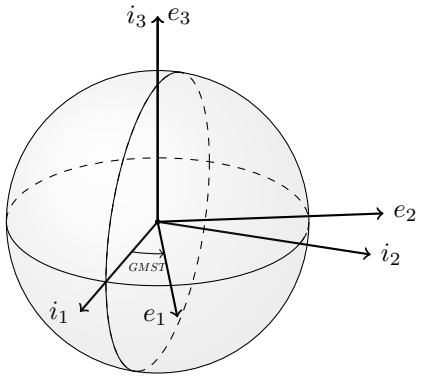

Figure 2.2: Earth Centered Earth Fixed Reference Frame.

spin axis of the Earth i_3 which is coincident with e_3, this displacement is known as the Greenwich Mean Sidereal Time (GMST).

Sometimes it is useful to represent vector and points in this frame with a set of spherical coordinates known as Global Position System (GPS) coordinates or LLA, whose three basis are θ_{lla}, ϕ_{lla} and r_{lla} [2]. The first is known as latitude and represents the angular displacement to equatorial plane, the second known as longitude represents the angular distance to the prime meridian or axis e_1 and finally r_{lla} which represents the distance to the center of the frame.

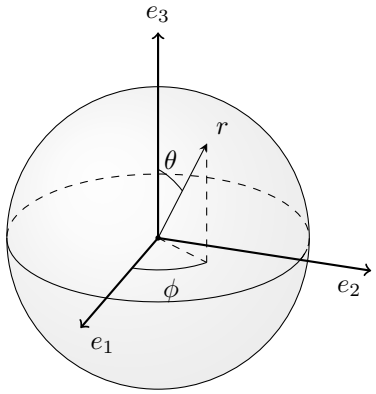

Figure 2.3: GPS spherical coordinates on ECEF frame.

2.1.3 Orbit

The Orbit frame follows the movement of the satellite in regard to the Earth, being quite helpful for decoupling position and motion regarding the Earth. The orbit frame is defined as $\mathcal{O} = \{o_1\,,\, o_2\,,\, o_3\}$, the

second base vector o_2 has the same direction as the satellite orbit plane normal, o_3 is collinear to vector from Earth to Satellite and lastly the o_1 completes the right-handed orthogonal system. The frame was defined with o_2 and o_3 purposely different from the literature for convenience of the control reference orientation.

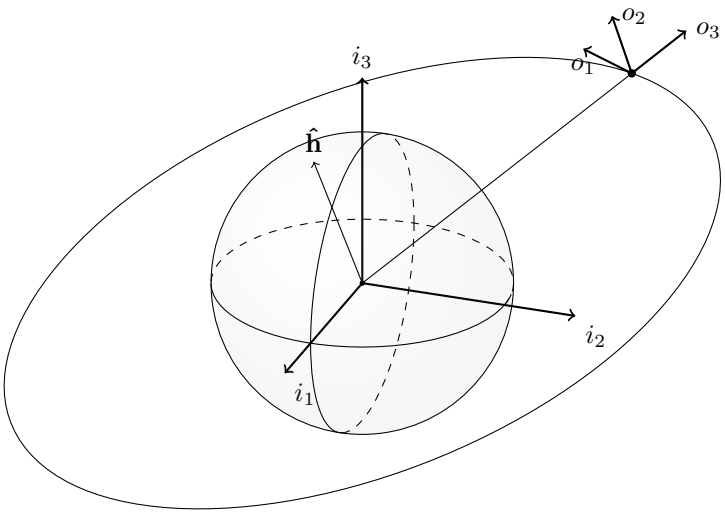

Figure 2.4: Orbit Reference Frame.

2.1.4 Body

The body reference frame is the spacecraft/satellite own coordinate frame, with the origin in its center of mass and designated with the basis $\mathcal{B} = \{b_1\,,\, b_2\,,\, b_3\}$. As the CUBESAT standard has established b_3 vector is perpendicular to the launcher rails, b_1 is perpendicular to the access port face and b_2 completes the right-hand coordinate system.

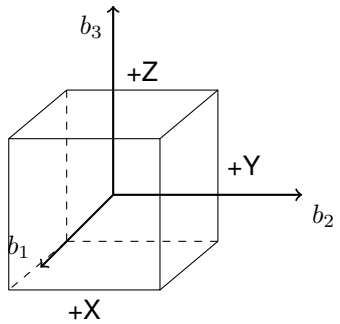

Figure 2.5: Diagram of the satellite $\mathcal{B}$ reference frame.

This reference frame is useful to represent how the satellite perceives and interacts with the environment, how it evaluates the system state. Therefore is usually the preferred main reference frame, where the control is designed to operate. From the body frame another reference frame will be derived used

mainly for the attitude representation which is the estimator reference frame and depicts the satellite assessment of the body frame.

2.2 Rotation Matrix

The rotation matrices has a special structure denoted as orthogonal three by three matrices, i.e. belongs to the $\mathbb{R}_3$ that represent tridimensional rotations allowing to rotate frames, vectors and quantities over different reference frames. The matrix $\left[R_i^j\right]$ represents a transformation from the space a to b and when multiplied by a generic vector v_a expressed in a it generates the representation of the vector v_a in frame b.

$$v_b \quad = \quad \left[R_a^b\right] v_a \tag{2.1}$$

The rotation matrices are orthogonal matrices thus their determinant should be unitary and multiplying them with vectors never changes the norm of the vector. Also as a consequence, the matrix used to represent the inverse rotation, from frame b to a is the transpose matrix used to rotate from a to b.

$$[R_b^a] \quad = \quad \left[R_a^b\right]^{-1} \quad = \quad \left[R_a^b\right]^T \tag{2.2}$$

Sequential rotations are achieved simply by multiplying the rotations matrices, such that a rotation from frame a to frame b could be perceived as a rotation from frame a to frame b and then from frame b to frame c. This is useful as enables the separation of a rotation operation into two or more known rotations.

$$[R_a^c] \quad = \quad [R_b^c] \cdot \left[R_a^b\right] \tag{2.3}$$

2.3 Attitude Representation

To understand rigid body dynamics in space it is necessary to first establish a method to represent the orientation of the satellite. The orientation of the satellite could be represented with the coordinates of the satellite body frame in the reference frame of choice or as a rotation from the reference frame to the satellite body frame. There are many different methods which could be used each with its own set of advantages but in throughout this work the approaches used to represent the attitude were the rotation matrix, the rotation vector and the quaternion [4].

2.3.1 Attitude Matrix

The attitude of a satellite can be represented as a rotation from the reference frame to the body frame and throughout this work the standard reference frame shall be the ECI. Being a rotation by definition allows it to be written as one of the rotation matrices referred in the previous chapter. To differentiate from other rotation matrices this attitude shall be identified as A.

$$A \quad = \quad \left[R_{\mathcal{I}}^{\mathcal{B}}\right] \tag{2.4}$$

2.3.2 Rotation Vector

Although matrices are very useful to rotate vectors they aren't very helpful to understand the rotation nor the attitude except in a few very specific cases. Another much simpler alternative is to see the attitude as a rotation along an axis described in the global frame. Therefore an axis and an angle displacement allow to reduce the need from 9 variables to 4, 3 for the axis and 1 for the rotation. With this representation is much easier to visualize the attitude. With this form there is also a constrain in the axis of rotation where its magnitude has to be unitary.

$$v \quad = \quad \begin{bmatrix} \theta \\ \mathbf{n} \end{bmatrix} \quad , \quad ||\mathbf{n}|| = 1 \tag{2.5}$$

The rotation vector v can be used to express the attitude matrix A with the following form:

$$A(v) \quad = \quad \mathbf{I}_{3\times 3} \, - \, \sin\theta \, [\mathbf{n}]_{\times} \, + \, (1 \, - \, \cos\theta) \, [\mathbf{n}]_{\times}^{2} \tag{2.6}$$

Using this representation the attitude visualization and analysis is easier but this method is not very convenient for computations. For further explanation see [5].

2.3.3 Quaternion

The quaternion was a concept introduced by Hamilton in 1844 as a hyper-complex extension of the complex space $\mathbb{Z}$ space with four components $q = q_0 + \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3$ with the following rules:

$$i^2 + j^2 + k^2 = -1 \tag{2.7a}$$

$$ij = -ji = k \tag{2.7b}$$

$$jk = -kj = i \tag{2.7c}$$

$$ki = -ik = j \tag{2.7d}$$

$$\mathbf{q}^T \, \mathbf{q} \quad = \quad 1 \tag{2.7e}$$

There are other possible forms to establish the quaternion frame but the presented shall be the one used throughout this document. The complex part of the quaternion, $\boldsymbol{\epsilon}$, represents the three coordinates of

the rotation axis while the real part, η is represents the angular displacement of the rotation.

$$\mathbf{q} \quad = \quad \begin{bmatrix} \eta \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} \tag{2.8}$$

The transformation from the attitude quaternion to the attitude matrix is given by the following equation.

$$A(q) \quad = \quad \left(\eta^2 - ||\boldsymbol{\epsilon}||^2\right) I_{3\times 3} \; - \; 2\eta \left[\boldsymbol{\epsilon}_\times\right] \; + \; 2\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T \tag{2.9a}$$

$$= \quad \begin{bmatrix} \eta^2 + \epsilon_1^2 - \epsilon_2^2 - \epsilon_3^2 & 2(\epsilon_1\epsilon_2 + \eta\epsilon_3) & 2(\epsilon_1\epsilon_3 - \eta\epsilon_2) \\ 2(\epsilon_1\epsilon_2 - \eta\epsilon_3) & \eta^2 - \epsilon_1^2 + \epsilon_2^2 - \epsilon_3^2 & 2(\epsilon_2\epsilon_3 + \eta\epsilon_1) \\ 2(\epsilon_1\epsilon_3 + \eta\epsilon_2) & 2(\epsilon_2\epsilon_3 - \eta\epsilon_1) & \eta^2 - \epsilon_1^2 - \epsilon_2^2 + \epsilon_3^2 \end{bmatrix} \tag{2.9b}$$

The inverse operation of generating a quaternion $\mathbf{q}$ equivalent to a specific rotation $[R]$ is also possible and shall be identified with the notation $q\,(R)$ but due the more complex operation the transformation shall be inserted into annex.

$$\left[R_a^b\right] \rightarrow \quad q\left(\left[R_a^b\right]\right) \quad = \quad \mathbf{q}_a^b \tag{2.10}$$

Similarly to the rotation matrix, the inverse rotation of a particular quaternion is described as it's inverse more so in quaternion notation that is equal to the conjugate.

$$\mathbf{q}_b^a \quad = \quad \left(\mathbf{q}_a^b\right)^{-1} \quad = \quad \left(\mathbf{q}_a^b\right)^* \tag{2.11}$$

The most important quaternion operation to understand is the product which shall be presented with the operator $\otimes$ and shall follow the rules of the bibliography [4]. The product of a quaternion $\mathbf{q}$ by another quaternion $\mathbf{p}$ is described as the following operation:

$$\mathbf{q} \otimes \mathbf{p} \quad = \quad \begin{bmatrix} q_\eta \cdot p_\eta - \mathbf{q}_\epsilon^T \cdot \mathbf{p}_\epsilon \\ q_\eta \cdot \mathbf{p}_\epsilon \; + \; p_\eta \cdot \mathbf{q}_\epsilon \; - \; \mathbf{q}_\epsilon \times \mathbf{p}_\epsilon \end{bmatrix} \tag{2.12}$$

The sequential rotation property of the rotation matrices also applies to the quaternions with it's product.

$$\mathbf{q}_a^c = \mathbf{q}_b^c \otimes \mathbf{q}_a^b \tag{2.13}$$

For basic understanding of the work presented here the most useful rules of the quaternions have been described here and detailed a little more in [Annex A] however a more broad approach of the quaternion rules are described in the bibliography [6],[4] and [7].

Chapter 3

Equations of Motion

Before starting to tackle the determination and control problem it is required to understand the physics of the space vehicle motion and rotation. This will complete the satellite physics model and later allow the accurately predict the system behaviour.

3.1 Environment Model

In order to accurately depict the satellite behaviour is necessary to describe the environment where he is going to flight [8] [9].

3.1.1 Gravity

A satellite per definition orbits a planet, such a phenomenon occurs due to the existence and major influence of gravity. According to Newton's law of Universal Gravitation the force a body exerts on another will be proportional to the product of their masses and inversely proportional to square distance between them. So the earth gravity would be written as:

$$\mathbf{F} \quad = \quad \frac{\mu_{\oplus} \cdot m_s}{|\mathbf{r}|^2} \cdot \mathbf{e}_r \, , \quad \mathbf{e}_r \; = \; \frac{\mathbf{r}}{|\mathbf{r}|} \tag{3.1}$$

where F is the gravity force vector, $\mu_{\oplus}$ is the Earth universal gravitation constant,m_s is the satellite mass and $\mathbf{r}$ is the position vector of the body written in ECI. This law interprets the bodies as point of mass and although this law is accurate enough for most of the problems, in reality the Earth is not a perfect sphere neither has a homogeneous composition. The planet has a deformation on the poles, which causes a variation in the gravity force. In order to accurately depict the Earth deformation other models were developed. The major perturbation, caused by the Earth oblateness is called J2 perturbation and can be easily corrected by adding to Newton's law a correction factor based on each axis coordinate. Though it may seem enough for a LEO, a more accurate model is preferred, the most well know model is the updated version of the Earth Gravitational Model written in 1996 (EGM96) which is the Earth Gravita-

tional Model 2008 (EGM2008). This model uses Legendre Polynomials to describe the gravity potential field as a function of the spherical coordinates. The accuracy of the model would depend on the order of coefficients used. The Earth gravity force $F_{\oplus}$ is described as a function of the satellite's ECEF position coordinate $\mathbf{r}_{\mathcal{E}}$.

$$\mathbf{F}_{\oplus} \quad = \quad f\left(\mathbf{r}_{\mathcal{E}}\right)_{egm} \tag{3.2}$$

Earth is the planet the satellite will orbit. As every body with mass has a gravitational field and given that are other celestial bodies in our solar system whose gravitational field have influence, the biggest contributors being the Moon and the Sun whose gravity forces from the perspective of our satellite orbit are classified as perturbations. The heterogeneous constitution verified with Earth does not pose a problem with these bodies as they are too distant to be noticed. So both of these can be modelled with Newton's Universal Gravitation Law.

$$\mathbf{F}_{\odot} \quad = \quad \frac{\mu_{\odot} \cdot m_s}{\mathbf{r}_s^2} \cdot \mathbf{e}_s \tag{3.3a}$$

$$\mathbf{F}_{\ominus} \quad = \quad \frac{\mu_{\ominus} \cdot m_m}{\mathbf{r}_m^2} \cdot \mathbf{e}_m \tag{3.3b}$$

Where $\mathbf{F}_{\odot}$, is the Sun gravity force, $\mu_{\odot}$ is the Sun gravitational constant, $\mathbf{r}_s$ is the distance vector between the satellite and the Sun written in the ECI frame, $\mathbf{e}_s$ is the direction vector from the satellite to the Sun written in ECI, $\mathbf{F}_{\ominus}$, represents the moon gravity force, $\mu_{\ominus}$ is the moon gravitational constant, $\mathbf{r}_m$ is the distance vector between the satellite and the Sun written in the ECI frame and $\mathbf{e}_m$ is the direction vector from the satellite to the moon written in ECI.

As seen before the heterogeneous density can causes some disturbances and not only the Earth but also the satellite own asymmetry can cause the disturbances. Let's first introduce the concept of center of gravity, which is the point where resultant gravity force would be applied. When this point is out of the satellite center of mass it results in a torque around the of mass center.

Being m the mass of the satellite the produced torque by gravity gradient τ_g in the $\mathcal{B}$ frame is given by:

$$\boldsymbol{\tau}_g \quad = \quad -\mu_{\oplus} \int \frac{\mathbf{r} \times \mathbf{r}_{\mathcal{B}}}{\mathbf{r}_{\mathcal{B}}^3} \tag{3.4}$$

Where $\mathbf{r}$ is the distance of the mass particle to center of mass and $\mathbf{r}_{\mathcal{B}}$ is the distance of the mass particle to the Earth center. Using J to represent the concept of inertia tensor, the torque can be expressed as:

$$\boldsymbol{\tau}_g \quad = \quad -3\left(\frac{\mu_{\oplus}}{\mathbf{r}_{\mathcal{B}}^3}\right)\hat{\mathbf{o}} \times J \cdot \hat{\mathbf{o}} \qquad , \qquad \hat{\mathbf{o}} = \frac{\mathbf{r}_{\mathcal{B}}}{|\mathbf{r}_{\mathcal{B}}|} \tag{3.5}$$

3.1.2 Magnetic Field

There is no precise date for it's discovery but Earth magnetic field is one of the most used features for navigation. Although there are some different theories about how exactly the planet core generates this field, even so the geomagnetic field models proposed throughout the years have proven to be very accurate. The first order approach, treats the magnetic field a magnetic dipole with the North-South axis 11^o rotated from the spinning axis. Near the poles the field magnitude reaches $60000nT$ and lowers until reaches it's minimum near the equator for a magnitude around $30000nT$. This rough approximation allows to generate an immediate and simple model for the geomagnetic field.

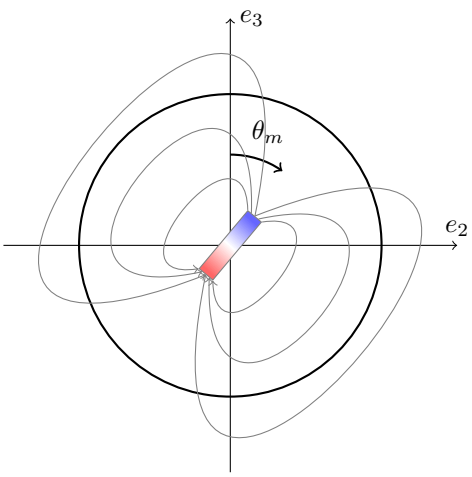

Figure 3.1: 2D diagram of Earth magnetic dipole field model with reference to the axis e_2 and e_3

However the moving inner core introduces irregularities and a slow change, t. This phenomena motivated the development of a more exact model. Nowadays, the model that offers the most accurate results is the International Geomagnetic Reference Field (IGRF) model, defining the field potential through a series of Legendre functions [10].

$$\mathbf{B} = -\nabla V \tag{3.6a}$$

$$V(r,\theta,\phi) = a\sum_{n=1}^{k}\left(\frac{a}{r}\right)^{n+1}\sum_{m=0}^{n}(g_n^m\cos m\phi + h_n^m\sin m\phi)P_n^m(\theta) \tag{3.6b}$$

Where r,θ,ϕ are the spherical coordinates radius, latitude and longitude, a is the Earth radius, g_n^m, h_n^m are the Gauss coefficients defined by IAGA (International Association of Geomagnetism and Aeronomy) and $P_n^m(\theta)$ is the Schmidt quasi-normalized Legendre polynomial. Using 3.6a and 3.6b the equation for

the geomagnetic field in each spherical coordinate direction is:

$$B_r = \sum_{n=1}^{k} \left(\frac{a}{r}\right)^{n+2} (n+1) \sum_{m=0}^{n} (g_n^m \cos m\phi + h_n^m \sin m\phi) P_n^m(\theta) \tag{3.7a}$$

$$B_\theta = -\sum_{n=1}^{k} \left(\frac{a}{r}\right)^{n+2} \sum_{m=0}^{n} (g_n^m \cos m\phi + h_n^m \sin m\phi) \frac{\partial P_n^m(\theta)}{\partial \theta} \tag{3.7b}$$

$$B_\phi = -\frac{1}{\sin\theta} \sum_{n=1}^{k} \left(\frac{a}{r}\right)^{n+2} \sum_{m=0}^{n} m \left(-g_n^m \sin m\phi + h_n^m \cos m\phi\right) P_n^m(\theta) \tag{3.7c}$$

According to Lorenz Law every time a electric current is induced in a conductor it creates a magnetic dipole. The satellite is an electric system and as such will have a resultant magnetic dipole which shall react to earth dipole and try to align create a torque. This is know as residual magnetic torque τ_B and is modelled through the following equation in the body frame:

$$\boldsymbol{\tau}_B = \mathbf{m}_r \times \mathbf{B}_\mathcal{B} \tag{3.8}$$

Where $\mathbf{m}_r$ is the satellite residual magnetic dipole vector and the $\mathbf{B}_\mathcal{B}$ is the geomagnetic field vector written in $\mathcal{B}$. The residual magnetic dipole is a satellite characteristic that varies in time and is difficult to measure therefore it will be modelled as a noise source, whose mean value will be based upon tested satellites.

3.1.3 Atmosphere

The atmosphere is a thin layer of air surrounding our planet and generate the largest non-gravitational perturbations for the LEO. The drag force the satellite sustains $\mathbf{F}_d$ in the ECI frame is a function of the velocity the satellite moves in relative to the air and is modelled from the following fluid mechanics equation:

$$\mathbf{F}_d = -\frac{1}{2} \cdot \rho \cdot C_d \cdot A_{sat} \cdot |\mathbf{v}|^2 \cdot \left[R_\mathcal{E}^\mathcal{I}\right] \frac{\mathbf{v}}{|\mathbf{v}|} \tag{3.9a}$$

$$\mathbf{v} = \mathbf{v}_\mathcal{E} + \mathbf{v}_w \approx \mathbf{v}_\mathcal{E} \tag{3.9b}$$

Where ρ is the air density, C_d is the drag coefficient, A_{sat} is the cross section area of the satellite, $\mathbf{v}_\mathcal{E}$ is the satellite velocity in the ECEF frame and the $\mathbf{v}_\omega$ is the velocity of the air in ECEF which can be neglected when compared to the orbital velocity in the ECEF frame. The satellite drag coefficient is a factor mostly determined by it's geometry and the direction of the velocity vector in the body frame. The

cross sectional area A_{sat} is also related to the satellite geometry and the attitude in regard to the velocity vector. From [11] the C_d of 1U cubesat is between 2.0 and 3.5 with a nominal value of 2.2 being the most used for non-varying C_d analysis and a mean cross-section area of $0.01\ m^2$. The air density depends on the temperature, altitude, solar activity, position, among other characteristics. The accuracy of most atmosphere models is low for high altitudes $(>\ 100\,km)$ but according to [1] the models approved by COSPAR are the most adequate. In bibliography the air density ρ is written as a function of the height h:

$$\rho(h) = \rho_0 e^{-h\left(\frac{kT}{mg}\right)} = \rho_0 e^{-\left(\frac{h}{H}\right)} \tag{3.10}$$

Where ρ_0 is the density at reference height, T is the air temperature, k is Boltzman's constant, m is the air molecular weight, g the gravity acceleration and H is the scalar height. The scalar height H is a empiric coefficient establish in model CIRA 72 according to the height.

h (km)	100	110	120	130	140	150	160	180	200	250	300	350	400
H (km)	6.15	8.06	11.6	16.1	20.6	24.6	26.3	33.2	38.5	46.9	52.5	56.4	59.4

Table 3.1: Scalar height reference values for different heights

Similarly to the gravity gradient origin, the resulting point where the aerodynamic force is virtually applied is called center of pressure and as it's constantly changing most of the time it doesn't coincide with the satellite center of mass. This distance will cause a torque which will rotate the satellite. From mechanics the resulting drag torque $\boldsymbol{\tau}_d$ written in the body frame is described by the following equation:

$$\boldsymbol{\tau}_d \quad = \quad \left(\left[R_{\mathcal{E}}^{\mathcal{I}}\right]\mathbf{F}_d\right) \times \mathbf{r}_{cp} \tag{3.11}$$

where $\mathbf{r}_{cp}$ is the position vector of the center of pressure in the $\mathcal{B}$ frame. From previous works [[11]] and [[12]] it is possible to see the distance between the center of pressure and the center of mass is around $1\,cm$. However the exact coordinates of the $\mathbf{r}_{cp}$ in time are not a simple problem to be solved and require an extensive aerodynamic study to create a realistic model. Instead, a rough approximation was used to simulate the movement of the center of pressure in time under the assumption of a free body having a natural equilibrium point where the drag is minimum and to where it will tend to.

$$\boldsymbol{\tau}_d \quad = \quad \left(\left[R_{\mathcal{E}}^{\mathcal{I}}\right]\mathbf{F}_d\right) \times r_{d_0} \begin{bmatrix} cos\left(\omega_x t + \lambda_1\right) \\ cos\left(\omega_y t + \lambda_2\right) \\ cos\left(\omega_z t + \lambda_3\right) \end{bmatrix} \tag{3.12a}$$

where r_{d_0} is the maximum distance of the center of pressure, ω_x, ω_y, ω_z are the axial components of the angular velocity $\boldsymbol{\omega}$ written in the $\mathcal{B}$. The coefficient λ is a random generated vector between 0^o and

360^o with a 120^o phase difference between coordinates representing the unknown vector position of the center of pressure.

3.2 Orbital Motion

Having described the most important external forces, in the following chapter the satellite movement around the Earth will be mathematically formulated following the information in [13]. According to Newton's Second Law the sum of all the forces actuating a body are described by a resulting acceleration $\mathbf{a}$ inversely proportional to the mass:

$$\sum \mathbf{F} \quad = \quad m\,\mathbf{a} \tag{3.13}$$

As described before, the forces actuating over the satellite are of gravity and aerodynamic nature, replacing equations (3.2), (3.3a), (3.3b) and (3.9a) into the previous equation (3.13) holds:

$$\begin{aligned} m\,\mathbf{a} \quad &= \quad \mathbf{F}_{\oplus} \,+\, \mathbf{F}_d \,+\, \mathbf{F}_{\odot} \,+\, \mathbf{F}_{\ominus} \\ &= \quad f\left(\mathbf{r}_{\mathcal{E}}\right)_{egm} \,+\, \frac{1}{2}\,\rho\, C_d\, A_{sat}\,|\mathbf{v}|\cdot\left(\left[R_{\mathcal{E}}^{\mathcal{I}}\right]\mathbf{V}\right) \\ &\qquad + \,\frac{\mu_{\odot}\cdot m_s}{\mathbf{r}_s^2}\cdot\mathbf{e}_s \,+\, \frac{\mu_{\ominus}\cdot m_m}{\mathbf{r}_m^2}\cdot\mathbf{e}_m \end{aligned} \tag{3.14}$$

Where the acceleration is written in the ECI frame and when integrated provides the variation of the satellite velocity in ECI frame.

$$\mathbf{v}(t) = \mathbf{v}_0 \,+\, \int_0^t \mathbf{a}\,d\,t \tag{3.15}$$

The coordinates of the satellite are generated integrating once again, using the equation (3.15) to achieve the position:

$$\mathbf{r}(t) = \mathbf{r}_0 \,+\, \int_0^t \mathbf{v}(t)\,d\,t \tag{3.16}$$

The initial values $\mathbf{v}_0$ and $\mathbf{r}_0$ are settled according to the Kepler parameters of the satellite's orbit. The expected motion of the satellite will be of an orbit, describing perturbed eclipses around the planet with the orientation stetted by its orbital parameters.

3.3 Angular Motion

This chapter will focus on the angular motion of the satellite, used to depict the satellite attitude kinematics. Starting with the Euler's equation and defining the relation between the resulting torques and the variation of angular momentum $\mathbf{H}$ in [1], as:

$$\dot{\mathbf{H}} = \sum \boldsymbol{\tau} \tag{3.17}$$

By definition the angular momentum around the center of mass is given by the product of the inertia tensor J by the angular velocity written in $\mathcal{B}$:

$$\mathbf{H} = J\boldsymbol{\omega} \tag{3.18}$$

For ease it's preferred to use the body frame to express the satellite rotations, as such the satellite angular momentum derivative expressed using the angular rate w.r.t. the inertial frame (ECI) gives the Euler equation used in satellite dynamics:

$$J\dot{\boldsymbol{\omega}} + (\boldsymbol{\omega} \times J\boldsymbol{\omega}) = \sum \boldsymbol{\tau} \tag{3.19}$$

The sum of all torques is given by the environment models presented in the previous section 3.1 and the S/C actuators.

$$\sum \boldsymbol{\tau} = \boldsymbol{\tau}_c + \boldsymbol{\tau}_g + \boldsymbol{\tau}_B + \boldsymbol{\tau}_d \tag{3.20a}$$

$$= \boldsymbol{\tau}_c - 3\left(\frac{\mu}{\mathbf{r}_{\mathcal{B}}^3}\right)\hat{\mathbf{o}} \times J \cdot \hat{\mathbf{o}} + \mathbf{m}_r \times \mathbf{B}_{\mathcal{B}} + \left(\left[R_{\mathcal{E}}^{\mathcal{I}}\right]\mathbf{F}_d\right) \times r_{d_0}\, cos\left(\boldsymbol{\omega} t + \boldsymbol{\lambda}\right) \tag{3.20b}$$

Now it is possible to write the differential equation of the angular rate joining the two previous equations (3.19) and (3.20a):

$$\dot{\boldsymbol{\omega}} = J^{-1}\left(\boldsymbol{\tau}_c + \boldsymbol{\tau}_g + \boldsymbol{\tau}_B + \boldsymbol{\tau}_d - \boldsymbol{\omega} \times J\boldsymbol{\omega}\right) \tag{3.21}$$

To solve this equation is necessary to provide an initial condition identified as $\boldsymbol{omega}_0$.

$$\boldsymbol{\omega} = \boldsymbol{\omega}_0 + J^{-1}\int \left[\boldsymbol{\tau}_c + \boldsymbol{\tau}_g + \boldsymbol{\tau}_B + \boldsymbol{\tau}_d - \boldsymbol{\omega} \times J\boldsymbol{\omega}\right] dt \tag{3.22}$$

Next we need to integrate the motion into a attitude representative unit. This is where the Euler angles reveal one more difficulty, the angle obtained from direct integration of the angular velocity through time would be difficult to interpret, unless the rotation is restricted to one axis at a time. As establish in the previous chapter 2.3 the attitude representation method chosen will be the quaternions. The quaternions derivative can be related to the angular velocity pure quaternion using the equation:

$$\dot{\mathbf{q}} = \frac{1}{2}\,\mathbf{q}_W \otimes \mathbf{q} \tag{3.23}$$

$$= \frac{1}{2}\,\Omega\left(\boldsymbol{\omega}\right)\cdot\mathbf{q} \tag{3.24}$$

Where Ω is a transformation from the vector to quaternion multiplication matrix based on the pure quaternion form $\mathbf{q}_W$ computed fixing the first coordinate to zero.

$$\mathbf{q}_W = \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} \tag{3.25}$$

With every quaternion operation is necessary to check if the unitary quaternion norm is being uphold and through this method is possible to guarantee it. Having defined the quaternion derivative the direct integration describes the quaternion evolution in time and consequently the attitude motion.

$$\mathbf{q} = \mathbf{q}_0 + \int \dot{\mathbf{q}}\,dt \tag{3.26}$$

3.3.1 Equation Summary

$\mathbf{a}$	$=$	$m^{-1}\left[f\left(\mathbf{r}_{\mathcal{E}}\right)_{egm} + \frac{1}{2}\,\rho\,C_d\,A_{sat}\,\lvert\mathbf{v}\rvert\cdot\left(\left[R_{\mathcal{E}}^{\mathcal{I}}\right]\mathbf{V}\right) + \frac{\mu_{\odot}\cdot m_s}{\mathbf{r}_s^2}\cdot\mathbf{e}_s + \frac{\mu_{\ominus}\cdot m_m}{\mathbf{r}_m^2}\cdot\mathbf{e}_m\right]$
$\mathbf{v}$	$=$	$\mathbf{v}_0 + \int \mathbf{a}\,dt$
$\mathbf{r}$	$=$	$\mathbf{r}_0 + \int \mathbf{v}\,dt$
$\boldsymbol{\omega}$	$=$	$\boldsymbol{\omega}_0 + J^{-1}\int\left[\boldsymbol{\tau}_t - \boldsymbol{\omega}\times J\boldsymbol{\omega}\right]\,dt$
$\boldsymbol{\tau}_t$	$=$	$\boldsymbol{\tau}_c - 3\left(\frac{\mu}{\mathbf{r}_{\mathcal{B}}^3}\right)\hat{\mathbf{o}}\times J\cdot\hat{\mathbf{o}} + \mathbf{m}_r\times\mathbf{B}_{\mathcal{B}} + \left(\left[R_{\mathcal{E}}^{\mathcal{I}}\right]\mathbf{F}_d\right)\times r_{d_0}\,cos\left(\boldsymbol{\omega}t+\boldsymbol{\lambda}\right)$
$\dot{\mathbf{q}}$	$=$	$\frac{1}{2}\,\Omega\left(\boldsymbol{\omega}\right)\,\mathbf{q}$
$\mathbf{q}$	$=$	$\mathbf{q}_0 + \int \dot{\mathbf{q}}\,dt$

Table 3.2: Summary of the equations of motion

Chapter 4

Satellite Model

The ISTsat-1 is a 1U[1] cubesat satellite comprising 5 subsystems: an Onboard Computer (OBC), Communication Data Handler (COM), a telecommunication system (TTC), an Electric Power System (EPS) and a payload system.

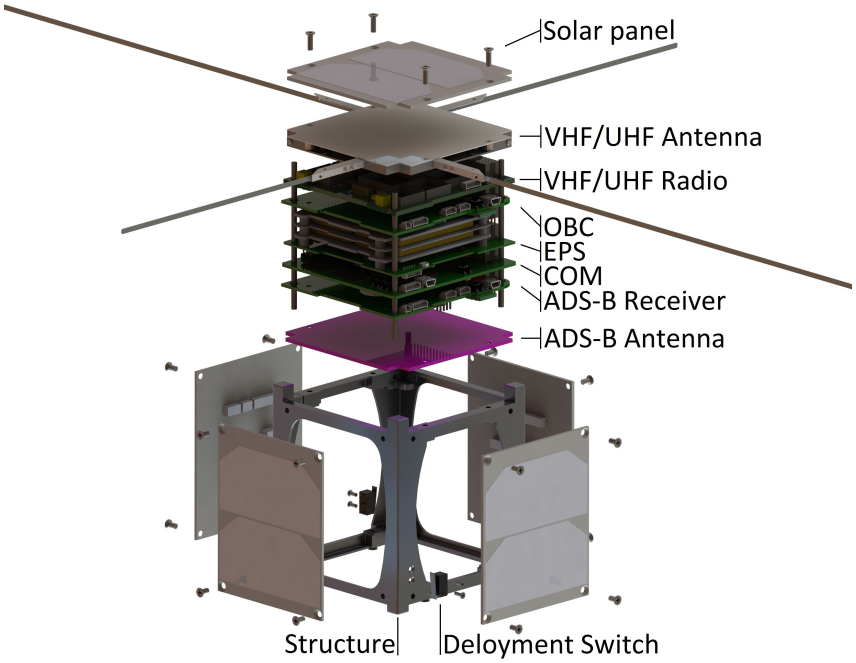

Figure 4.1: ISTsat-1 CAD assembly exploded view with the subsytems mockups. [Taken from ISTsat-1 engineering documentation]

[1] The industry standard cubesat unit of measurement was establish in [14] as a $10\,cm$ cube with $1.33\,kg$.

4.1 Inertia Matrix

To be able to understand the satellite dynamic response, one important feature is how the satellite mass is geometrically distributed. Dynamically this is known as Momentum of Inertia and fundamentally states the more sparse a body mass is the more difficult is to rotate it. This information is represented in a matrix J mentioned in previous chapters. J is a posive-definite matrix and if defined around the center of mass it is computed through the following equation:

$$[J] \quad = \quad \iiint r^2 \, dm \tag{4.1}$$

The satellite geometry is very close to a cube which allows to compare it to a $1\,kg$ cube of uniform mass whose diagonal inertia matrix $[J]_{cube}$ computation can be easily made from [15].

$$[J]_{cube} \quad = \quad \frac{s^2}{6} \, I_{3\times 3} \quad = \quad 1.66 \times 10^{-3} I_{3\times 3} \;\; (kg\, m^2) \tag{4.2}$$

From the CAD software used to model the satellite systems, it is possible to obtain an estimation of the satellite inertia matrix around the center of mass and check how much it differs from the homogeneous situation. The CAD model includes the mass properties of each component resulting in a heterogeneous mass distribution and non-zero cross-axis components in inertia matrix estimation.

$$\begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{yx} & J_{yy} & J_{yz} \\ J_{zx} & J_{zy} & J_{zz} \end{bmatrix} \quad = \quad \begin{bmatrix} 1.6194 & -0.0174 & 0.0113 \\ -0.0174 & 1.7603 & 0.0036 \\ 0.0113 & 0.0036 & 1.8415 \end{bmatrix} \times 10^{-3} \; kg\, m^2 \tag{4.3}$$

After de-tumbling the satellite will deploy the antennas which are $0.5\,m$ length steel rods that extend from the top face to each of the sides significantly changing the inertia matrix.

$$[J]_{deployed} \quad = \quad \begin{bmatrix} 1.6715 & -0.0406 & 0.0118 \\ -0.0406 & 2.1326 & 0.0036 \\ 0.0118 & 0.0036 & 2.2369 \end{bmatrix} \times 10^{-3} \; kg\, m^2 \tag{4.4}$$

4.2 Hardware

The ADCS is design to be integrated into the OBC, sharing the same hardware to accomplish its distinct functions. As depicted in [16], at the core of the OBC module is a MSP430F5438A, a low power 16-bit MicroController Unit (MCU) by Texas Instruments, where the ADCS algorithms and data will be computed. This specific MCU has 256 kB of flash memory and 16 kB of RAM and the clock frequency goes

up to 25 MHz. This is not the most powerful MCU although it is well known for it's low power consumption which was the main reason for it's choice. Besides the MCU, two more sets of devices are needed for the ADCS to be able to perform its duties. The first are the sensors, responsible for the digital translation of specific phenomenons. The second set are the actuators which are the satellite physical manifestation of control. The OBC system was design to have access to measurements from a gyroscope, Sun sensor and magnetometer. The most relevant operational requirement for this mission is the control over the satellite rotation which focused the choice on torque producing actuators like the chosen magnetorquers.

4.3 Inertial Sensors

4.3.1 Gyroscope

A gyroscope is an inertial sensor that measures an angular displacement in an inertial reference frame. Among the gyro sensors there is a specific group called the rate-gyros, which estimate the angular velocity by dividing the measured angular displacement over a period of time. This information can help understand the S/C dynamic state better and can even be used together with other attitude sensors to complement their information. There are many different technologies used for the gyroscopes like LASER ring, fiber-optic or nuclear-based although offering great performance these options require much more volume than a MEMS solution [17][18]. The size and mass being highly restricted are of most importance in satellite systems design because it impacts severely the launch cost. For this reason the most common option in cubesats is to use MEMS. During the design of the OBC board the gyroscope chosen was integrated into an Inertial Measurement Unit (IMU) manufactured by TDK Invensense designated by MPU9050 [19]. This 9-axes unit was design for low power applications with heading requirements such as smart-phones, motion-based controllers and and 3D remote controls among others. Another 5 gyroscopes devices are available, one in each solar panel, although they are not expected to be used to enforce redundancy it is wise to list him and check if it can be used as substitute to the main gyroscope. Unfortunately the manufacturer does not list a complete performance report in the specification sheet to enable a fair comparison with the main gyro. As stated in [20], a high accuracy maneuvering requires a higher fidelity gyroscope model and for ISTsat-1 this would be useful. As such the gyroscope reading $\hat{\boldsymbol{\omega}}$ as the real value $\boldsymbol{\omega}$ multiplied by a misalignment and a scaling matrix $[G_g]$, added by a bias $\boldsymbol{\beta}_g$ and noise factor $\mathbf{n}_{arw}$.

$$\hat{\boldsymbol{\omega}} = [G_g]\,\boldsymbol{\omega} + \boldsymbol{\beta}_g + \mathbf{n}_{arw} \tag{4.5}$$

$$\dot{\boldsymbol{\beta}}_g = \mathbf{n}_{rrw} \tag{4.6}$$

The noise factor $\mathbf{n}_{arw}$ is a white noise representing the sensor perturbation in reading the values known as Angle Random Walk. The bias is the sensor output when the input is zero and in rate-gyros this

Characteristic	Sensor	Sensor
Model	MPU-9250	Solar Panel
Type	Rate gyro	Rate gyro
ADC	$16\,bits$	-
Sampling	$4 - 8000\,Hz$	-
Scale[2]	$250\,^o/s$	$80\,^o/s$
Sensitivity	$131\,\frac{LSB}{^o/s}$	$54.6\,\frac{LSB}{^o/s}$
ΔSensitivity (-40^oC to $+85^oC$)	$4\,\%$	-
Linearity	$0.1\,\%$	-
Noise RMS	$0.1\,^o/s$	-
Rate Noise PSD	$0.01\,\frac{^o/s}{\sqrt{Hz}}$	-
LowPass Filter	$5\,-\,250\,Hz$	-

Table 4.1: MPU9250 and solar panel specifications [19]

property isn't constant each time the sensor is turned on nor does it remain constant through time. The latter is a phenomenon known as gyro drift or rate random walk and it's change can be interpreted also as white noise.

The processes $\mathbf{n}_{rrw}$ and $\mathbf{n}_{arw}$ are generally defined by their spectral densities which subsequently are computed through their Allen Variation σ_{rrw} and σ_{arw} respectively.

$$E\left[\mathbf{n}_{arw}\,\mathbf{n}_{arw}^T\right] = \sigma_{arw}^2\,I_{3\times3} \tag{4.7a}$$

$$E\left[\mathbf{n}_{rrw}\,\mathbf{n}_{rrw}^T\right] = \sigma_{rrw}^2\,I_{3\times3} \tag{4.7b}$$

While higher accuracy is desired there is a trade-off to be made in order to save some computational power to OBC, instead of computing the alignments and scaling factors on flight they will be calculated on ground through tests.

4.4 Attitude Sensors

For attitude determination, reference measurements are needed to compare the sensor readings to, so in space there a few options to use as reference. The most usual references relate to Earth and Sun features and to capture such features the most used sensors are Sun sensors, horizon sensors, magnetometers and star trackers [1]. The value these sensors truly measure can be computed through high accuracy empirical models asserting the devices as good attitude sensors. Normally the star trackers are high cost devices whose size is an exclusion cause for low budget small satellites. The most ade-

quate sensors for cubesats are the magnetometers, the Sun sensors and horizon sensors. With that in mind it was chosen to use a magnetometer and a coarse Sun sensor for the attitude sensors.

4.4.1 Sun Sensors

There many types of solar sensors divided according to how they detect the sunlight direction. Due to its optic needs the Sun sensors usually aren't small enough for the cubesats to accommodate. In order to decrease the volume and financial costs the choice was to use a coarse Sun sensor system. Even among COTS the volume of most coarse Sun sensors aren't viable in a 1U cubesat as it would imply the removal of one solar panel per face. Searching for another solution led to the challenge of using an array of photo-diodes as solar sensors. This was a solution already included into the products of some solar panels manufacturers. A photo-diode is a semiconductor device that in the presence of light allows to change their junction polarity and conduct current [21]. The photo-diode design was then added as a requirement for choosing the solar panel and after weighting all the parameters it was chosen to use EnduroSat solution. Unfortunately the sensor specification the manufacturer provided is not much detailed, just giving information about the spectrum bandwidth and half-angle sensitivity.

Name	Solar Panel
Type	Photo diode
Half-Sensitivity	60^o
Spectral bandwidth	$430 - 610\,nm$

Table 4.2: Endurosat solar sensor specifications [22]

According to the model in [23] the current i measured in a photodiode is a cosine function of the incident angle.

$$i \quad = \quad i_0 \cos\alpha + n_{ss} \tag{4.8}$$

Where α is the angle between the photodiode normal and the light direction, i_0 is the maximum current the diode allows for that wavelength and n_{ss} is measurement noise. The incident angle α is a product of the photodiode i normal surface $\bar{n}_i$ in $\mathcal{B}$ with the Sun position vector S read in $\mathcal{B}$ which can be approximated to the Sun position in frame $\mathcal{E}$ rotated to $\mathcal{B}$ and shall be identified as $\mathbf{S}_\mathcal{B}$.

$$\alpha \quad = \quad \cos^{-1}\left(\frac{\mathbf{S}_\mathcal{B} \cdot \bar{\mathbf{n}}_i}{|\mathbf{S}_\mathcal{B}|}\right) \qquad , \qquad |\bar{\mathbf{n}}_i| = 1 \tag{4.9}$$

This equation represents the unobstructed model however the line of sight to the sun is obstructed by

the Earth while in eclipse. The identification of an eclipse could be made using the angle ψ between the Sun and earth written in $\mathcal{B}$ and the limit angle of visibility ψ_{ecl}.

$$\alpha = \begin{cases} \cos^{-1}\left(\frac{\mathbf{S}_{\mathcal{B}} \cdot \bar{\mathbf{n}}_i}{|\mathbf{S}_{\mathcal{B}}|}\right) & \rightarrow \quad \psi < \psi_{ecl} \\ 90 & \rightarrow \quad \psi > \psi_{ecl} \end{cases} \tag{4.10}$$

The angular displacement between the vectors $\mathbf{S}_{\mathcal{B}}$ and $\mathbf{r}_{\mathcal{B}}$ is given by:

$$\psi = \cos^{-1}\left(\frac{\mathbf{r} \cdot \mathbf{S}_{\mathcal{B}}}{|\mathbf{r}| \cdot |\mathbf{S}_{\mathcal{B}}|}\right) \tag{4.11}$$

The limit region where the satellite is in eclipse is determined by the angle ψ_{ecl} which is a sinusoidal function of the ratio between orbit radius and Earth radius.

$$\psi_{ecl} = \sin^{-1}\left(\frac{R_{\oplus}}{R_{\oplus} + h}\right) \tag{4.12}$$

Where h represents the altitude of the satellite from the surface Earth. And the angle ψ is given by the angle between the Earth and Sun vector written in $\mathcal{B}$.

$$\psi = \cos^{-1}\left(\frac{\mathbf{S}_{\mathcal{B}} \cdot \mathbf{r}_{\mathcal{B}}}{|\mathbf{S}_{\mathcal{B}}| \cdot |\mathbf{r}_{\mathcal{B}}|}\right) \tag{4.13}$$

The transformation from current to Sun position vector written in the $\mathcal{B}$ is given by writing the sum of the opposites sides (same axis) measurements into the corresponding $\mathcal{B}$ axis.

$$\hat{\mathbf{S}}_{\mathcal{B}} = \frac{1}{i_0}\begin{bmatrix} i_{+x} - i_{-x} \\ i_{+y} - i_{-y} \\ i_{+z} - i_{-z} \end{bmatrix} \tag{4.14}$$

There is one important note to make about sunlight reflecting on the Earth surface introducing an error into the sensor. The satellite could include an albedo model as in [24], instead a simpler approach was chosen by setting the minimum current detected by the sensor to around 34% of the maximum current a little more than the Earth albedo 30% and imposed a limit in each photodiode FOV to 70^o.

4.4.2 Magnetometers

The magnetometers are sensors that measure the intensity of the magnetic field in each of their own coordinate frame orthogonal axis. The three measurements reveal the direction and magnitude of the magnetic field in its body reference frame. These readings will allow the spacecraft to establish attitude relations with the reference values generates through the IGRF model according to the S/C position in $\mathcal{E}$

frame. ISTsat has two of these sensors available, one in the IMU already mentioned and second in a different specialized IC manufactured by Honeywell, the HMC5983. The latter has a superior performance, much accurate readings and for that reason will be the main sensor from each most measurements will be read until a problem is diagnosed.

Characteristic	Sensor	Sensor
Name	HMC5983	MPU-9250
Type	3 axis	3 axis
ADC	$12bits$	$14bits$
Sample Rate	$0.75 - 220\,Hz$	-
Range	$\pm 1\,G$	$\pm 48\,G$
Sensitivity	$1370\,\frac{LSb}{G}$	$171\,\frac{LSb}{G}$
Linearity	0.1%	-
RMS Noise	$2mG$	-

Table 4.3: HMC5983 and MPU9250 sensor specifications [25] [19]

The vector read by the magnetometer is modeled as the true magnetic field vector multiplied by a misalignment and scaling matrices, summed to a bias $\boldsymbol{\beta}_B$ and in the end a noise component $\mathbf{n}_B$ is added.

$$\hat{\mathbf{B}} \quad = \quad [G_B]\,\mathbf{B} \,+\, \boldsymbol{\beta}_B \,+\, \mathbf{n}_B \tag{4.15}$$

The matrices of misalignment and scaling are multiplied sequentially and therefore can be joined together into the matrix $[G_B]$. The noise component is modeled as Gaussian white noise with a standard deviation σ_B furthermore in table [4.3] the value of σ_B for HMC5983 is specified as $200\,nT$.

$$E\left[\mathbf{n}_B\,\mathbf{n}_B^T\right] \quad = \sigma_B^2\,I_{3\times 3} \tag{4.16}$$

4.5 Magnetorquers

As important as the sensors are, the actuators are also of major importance to provide the necessary control actions. The actuators can be divided into two types, the attitude actuators and the orbital actuators. The result produced from the first type is predominantly a torque vector eventually changing

the attitude and the second type generates a force vector mostly used to perform corrections or change the orbit. The orbit corrections are necessary to extend the period of a satellite in a specific orbit or to change into a specific orbit. Neither of these types of corrections are necessary as the ISTsat-1 estimated lifetime is enough to accomplish the objectives established. Therefore it only needs to control the attitude. There are two other categories in which the attitude actuators can be divided the passive and active upon the possibility to change the actuator output. The passive solutions are a great option to save some power as most don't require energy supply to work although as a trade-off the attitude control won't be as stable, precise or flexible as the solutions allowed by active actuators. While in normal mode the satellite will have to be pointing NADIR which reduces the options normally used in cubesats to the Gravity Boom and two active: the Magnetorquers and Reactions Wheel.

Actuator	Power	Accuracy	Volume	Mass
Gravity Boom	Zero	Low	High	High
Magnetorequer	Low	High	Low	Low
Reaction Wheel	High	Very High	High	High

Table 4.4: ISTsat-1 possible actuators comparison

Analysing table 4.4 a few considerations need to be made, the first is that the satellite is small and the accommodation of more than 1 reaction wheel is difficult. The same can be said about the Gravity Boom considering that when it is not deployed it has to be accommodated inside the 1U structure. Furthermore, from previous cubesat missions team reports the gravity boom needs to be very carefully deployed otherwise when release the satellite may end up pointing in the opposite direction. A further disadvantage of the gravity boom is also the interference with the ADS-B path antenna located in the -Z surface of the satellite. For these reasons the most appealing solution was to use magnetorquers, the possiblity of integrate them into the solar panels assembly represents a big volume and mass advantage. The disadvantage of this choice is the power consumption leading the design of the controller to focus in reducing power consumption when possible, even so it will not be as high as the reaction wheels. The solar panels sold by Endurosat have an optional feature that may include a printed coil in the back of the panel to work as magnetorquers. To control 3 axes, 3 of the 5 solar panels needed this coil option and they are to be placed on $+X$, $+Y$ and $-Z$ faces.

Manufacturer	*EnduroSat*
Voltage	$3.3\,V$
DC Resitance	$42\,\Omega$
Current	$78\,mA$
Magnetic Dipole	$131\,mAm^2$

Table 4.5: Endurosat solar panel magnetorquer specification

The magnetoquers are a set of electromagnetic coils that generate a magnetic dipole when are current flows through them and this magnetic dipole will then react with the Earth Magnetic Field and produce a torque. The interaction between two magnetic dipoles is already described in equation [3.8], the same equation will describe the torque τ_c created by the magnetorquers magnetic dipole $\mathbf{m}_c$.

$$\boldsymbol{\tau}_c \quad = \quad \mathbf{m}_c \times \mathbf{B}_{\mathcal{B}} \tag{4.17}$$

The magnetic dipole can be approximated to a linear function of the current passing into the coils but is usually easier to control a circuit through voltage. Assuming a stationary case, the voltage will be directly proportional to the current and then it will be possible to write the magnetic dipole as a function of the voltage output V_c.

$$\mathbf{m}_c \quad = \quad \mathbf{m}_m \left(\frac{V_c}{V_{max}} \right) \tag{4.18}$$

Where V_{max} is the voltage necessary to provide the specified maximum magnetic dipole $\mathbf{m}_m$. For further simplification the instead of a variable voltage output the OBC shall provide a PWM signal whose duty-cycle is calculated as ratio between the desired output voltage and the maximum voltage.

$$D \quad = \quad \frac{V_c}{V_{max}} \tag{4.19}$$

The resulting voltage averaged over the actuation time is the same as the desired voltage however the averaged torque generated could be different as the torque depends on the magnetic field which is time-varying.

$$\frac{1}{T} \int_{t_0}^{t_0+T} (\mathbf{m}_m \sqcap) \times \mathbf{B}(t)\, dt \quad \neq \quad \frac{1}{T} \int_{t_0}^{t_0+T} \left(\frac{\mathbf{m}_m\, V_c}{V_{max}} \right) \times \mathbf{B}(t) dt \tag{4.20}$$

Where $\sqcap$ is a square function of dutycycle D. With this implementation the precision of the resulting magnetic dipole is defined based upon the dutycycle precision. Although for very small duty-cycle values the pulse is too quick and the driver may not be able to maintain a perfectly timed square wave. Also to avoid coils frequency problems a lower duty cycle limit as imposed which corresponds to pulse widths smaller than $0.1\,ms$.

On other hand, there is also a requirement to ensure the ADCS doesn't take magnetometer readings when the magnetoquers are being induced and sufficient time is provided so that the magnetometers can take unbiased readings. This problem was solved by limiting the duty-cycle to smaller value than 100% allowing at least a few milliseconds for the readings.

These limiting values will depend on the control sampling frequency which will be chosen in a later chapter but for safety reasons the duty-cycle was limited from 0.1% to 80% until the frequency is chosen.

$$0.0001 \quad \leq \quad D \quad \leq \quad 0.8 \tag{4.21}$$

Chapter 5

Attitude Determination Algorithms

In order to design the attitude and control system the previous chapters were essential to establish the base for developing the algorithms required. So in this chapter the attitude determination problem will be formally presented and recalling the satellite has a low power budget simple algorithms will be presented as possible solutions to satisfy the low computational power requirements. The solutions for the attitude problem corresponds to finding the rotation matrix with a positive determinant which for any set of measurements transform the sensor data into the reference vectors.

$$A\,\mathbf{r}_i \quad = \quad \mathbf{b}_i\,, \quad \texttt{for} \quad i = 1, 2 \tag{5.1}$$

In ISTsat-1 case, the number of measurements will be two so i can only be either 1 or 2, for the magnetometer can give a measurements in any normal condition but the sun sensor may not be able to due to eclipse conditions. The first two approaches presented in this chapter are the TRIAD and Wahba, both leading to deterministic solutions with simple and fast formulation. Then some more complex algorithms integrating the dynamic information will be presented in order to increase the system robustness and performance while still trying to maintain the low power requirement. The major advantage of most dynamic algorithms is the capability of predicting a vector evolution using only gyroscope measurements, which is quite useful when is not possible to register one measurement vector as it happens with the sun sensor in eclipse. The first dynamic algorithm presented here will be the EQUEST which is a simple and quick weighted quaternion combination solution. The second will be the celebrated Kalman Filter with a small variation in implementation for increased speed. Lastly a sensor fusion algorithm known as Complementary Filter which even though being simple reports showed good performance.

5.1 TRIAD

One of the first algorithms to solve the attitude problem is known for TRIAD or ”TRIAxial Determination algorithm”, developed by Harold Black [26]. This algorithm uses only 2 non-collinear measurements vectors and their respective references in the reference frame to provide a solution. The algorithm logic

consist in creating a triaxial orthogonal frame with the measurements and use its properties to find the attitude matrix. The measurements will be used to form a orthonormal triaxial space with the base vectors $\mathbf{w}_k$ and the same will be done with the reference vectors creating the base vectors v_k. As they are created using the same linear operations the attitude equation is applicable.

$$\hat{A}_{TRIAD}\,[\mathbf{v}_1\,\mathbf{v}_2\,\mathbf{v}_3] \quad = \quad [\mathbf{w}_1\,\mathbf{w}_2\,\mathbf{w}_3] \tag{5.2}$$

To find A the solution could be simply given by the multiplication of matrix w by the inverse of v. Now as both matrices are orthonormal instead of using long methods to compute the inverse of a 3 by 3 matrix the inverse could be easily computed by transposing it.

$$\hat{A}_{TRIAD} \quad = \quad [\mathbf{w}_1\,\mathbf{w}_2\,\mathbf{w}_3] \cdot [\mathbf{v}_1\,\mathbf{v}_2\,\mathbf{v}_3]^T \tag{5.3}$$

So the important step will be to create the vectors w_k and v_k in order to form an orthonormal group using the measurements vectors $\mathbf{b}_i$ and $\mathbf{r}_i$. The first base vector will be equal to the first measurement vector. In order to achieve better results the vector $\mathbf{b}_1$ should provide more accurate measurements than $\mathbf{b}_2$, as the algorithm is more sensible to the error associated to measurement vector 1.

$$\mathbf{w}_1 \quad = \quad \mathbf{b}_1 \tag{5.4}$$

The second vector $\mathbf{w}_2$ is the result of the cross product of $\mathbf{b}_1$ and $\mathbf{b}_2$.

$$\mathbf{w}_2 \quad = \quad \mathbf{b}_1 \times \mathbf{b}_2 \tag{5.5}$$

Lastly the third vector $\mathbf{w}_3$ will be orthogonal to the other two, a cross product will also be used to achieve this.

$$\mathbf{w}_3 \quad = \quad \mathbf{w}_1 \times \mathbf{w}_2 \tag{5.6}$$

The matrix of vectors $[V]$ is also built this way but using the reference vectors instead.

$$\mathbf{v}_1 \quad = \quad \mathbf{r}_1 \tag{5.7a}$$

$$\mathbf{v}_2 \quad = \quad \mathbf{v}_1 \times \mathbf{r}_2 \tag{5.7b}$$

$$\mathbf{v}_3 \quad = \quad \mathbf{v}_1 \times \mathbf{v}_2 \tag{5.7c}$$

Now that the orthonormal properties are guaranteed the attitude equation can be rewritten into:

$$\hat{A}_{TRIAD} \quad = \quad \mathbf{w}_1\,\mathbf{v}_1^T + \mathbf{w}_2\,\mathbf{v}_2^T + \mathbf{w}_3\,\mathbf{v}_3^T \tag{5.8}$$

Before trying out the algorithm in the simulator it is possible to analytically demonstrate its expected performance. The first assumption to be considered will be an expected value equal to zero. Assuming the error are random errors with a zero mean value their description will be made through their respective covariance. The zero mean of the observation and reference vectors does also lead to a zero mean error in the attitude matrix.

$$\langle \triangle A \rangle \quad = \quad \langle A \cdot \hat{A} \rangle \quad = \quad 0 \tag{5.9}$$

As there is no correlation between the reference and measurement vectors the total covariance matrix is the sum of the contributions from each set of vectors.

$$0 \quad = \quad \langle \triangle \mathbf{b}_1 \triangle \mathbf{b}_2^T \rangle \quad = \quad \langle \triangle \mathbf{r}_1 \triangle \mathbf{r}_2^T \rangle \tag{5.10a}$$

$$P \quad = \quad P_b \,+\, A P_r A^T \tag{5.10b}$$

Then using the results from [27] the covariance matrix is written in the following form:

$$\begin{aligned} P \quad = \quad \sigma_1^2 (I \,-\, \mathbf{v}_3\mathbf{v}_3^T \,+\, \mathbf{v}_1\mathbf{v}_1^T \,-\, \mathbf{v}_2\mathbf{v}_2^T) \,+\, \frac{\left(\sigma_1^2 \,+\, \sigma_2^2\right)}{|r_1 \times r_2|^2} \left(I \,-\, \mathbf{v}_1\mathbf{v}_1^T\right) \\ +\, \sigma_1^2 \frac{(\mathbf{r}_1 \cdot \mathbf{r}_2)}{|\mathbf{r}_1 \times \mathbf{r}_2|} \left(\mathbf{v}_1\mathbf{v}_3^T \,+\, \mathbf{v}_3 v_1^T\right) \end{aligned} \tag{5.11}$$

Where the standard deviations σ_1 and σ_2 are the measurement and reference vectors combined.

$$\sigma_1^2 \quad = \quad \sigma_{r_1}^2 \,+\, \sigma_{b_1}^2 \tag{5.12a}$$

$$\sigma_2^2 \quad = \quad \sigma_{r_2}^2 \,+\, \sigma_{b_2}^2 \tag{5.12b}$$

The usual attitude error parametrization through small angles approximation allow for an even more simplified equation as described in [5] but as that is only valid for small angles the equation will not be used until the errors are proven to be small enough.

Finally the error associated with the acquisition of the reference vectors will be ignored for now until the

error of the position algorithm is known. The error will regard only the noise in the measurement vectors which were characterized in previous report allowing to achieve numeric results of the covariance matrix for this method.

5.2 Wahba's Problem

In attitude determination one common formulation of the attitude problem was posed by Grace Wahba lead to many solutions [28]. This forms logic is to find the matrix which minimizes the cost function which is similar to least square minimum problem:

$$L(A) \quad = \quad \frac{1}{2}\sum_{k=1}^{n} a_k \, ||\mathbf{b}_k \, - \, A\mathbf{r}_k||^2 \tag{5.13}$$

Where a_k are the relative weights associated with the sensor and n the number of measurements in these particular case will be equal to 2. While using direction vectors the magnitude of vectors $\mathbf{b}_k$ and $\mathbf{r}_k$ will be unitary. This allows to transform the problem into a simpler equation:

$$||\mathbf{b}_k - A\mathbf{r}_k||^2 \quad = \quad ||\mathbf{b}_k||^2 \, - \, 2\,(\mathbf{b}_k \cdot A\mathbf{r}_k) \, + \, ||\mathbf{r}_k||^2 \tag{5.14a}$$

$$= \quad 2 \, - \, 2\,(\mathbf{b}_k \cdot A\mathbf{r}_k) \tag{5.14b}$$

To simplify even more the products of the vectors and weight were written in matrix B.

$$B \quad = \quad \sum_{k=1}^{n} a_k \, \mathbf{r}_k \cdot \mathbf{b}_k^T \tag{5.15}$$

These changes allow to write the cost function in the following form:

$$L(A) \quad = \quad \lambda_0 \, - \, tr\left(AB^T\right) \tag{5.16a}$$

$$\lambda_0 \quad = \quad \sum_{k=1}^{n} a_k \tag{5.16b}$$

Now the problem is inverted and finding the minimum cost is to find the maximum of $\left(AB^T\right)$ trace. Throughout the years different algorithms were developed to solve Wahba's problem. Some of the more known are: q-Method, QUEST, SVD and FOAM.

5.2.1 Davenport's q-Method

Davenport developed one of the first solutions to Wahba's problem using quaternions [28]. He proved the matrix AB^T trace is related to the attitude quaternion through a 3×3 gain matrix K.

$$tr\left(AB^T\right) \quad = \quad \mathbf{q}^T K \, \mathbf{q} \tag{5.17}$$

Where the matrices K and z are given by:

$$K \quad = \quad \begin{bmatrix} B + B^T - tr(B) \cdot I & z \\ z^T & tr(B) \end{bmatrix} \tag{5.18a}$$

$$z \quad = \quad \sum_{k=1}^{n} a_k \, (\mathbf{b}_k \times \mathbf{r}_k) \tag{5.18b}$$

Manipulating q in the equation is possible to come up to the eigenvector-eigenvalue equation which has n solutions, in this case being $n = 4$ there will be 4 possible solutions vectors. Lastly Davenport proved the corrected solution of $\mathbf{q}$ to the attitude problem was associated with the highest eigenvalue of the matrix K.

$$K\hat{\mathbf{q}} \quad = \quad \lambda_{max}\hat{\mathbf{q}} \tag{5.19a}$$

$$|\hat{\mathbf{q}}| \quad = \quad 1 \tag{5.19b}$$

Using the highest eigenvalue of K for λ_{max} the original cost equation turns to:

$$J(A) \quad = \quad \lambda_0 \, - \, \lambda_{max} \tag{5.20}$$

Davenport transformed Wahba's problem into a eigenvector problem, so in order to solve the attitude problem it's only necessary to compute the eigenvectors of K and use the one associated to the highest eigenvalue. One problem of this method is revealed when the two highest eigenvalue are not distinct resulting in a non-unique solution even so, this formulation is one of the best to solve Wahba's problem.

5.2.2 QUEST

Using the results of Davenport's q-method, Malcolm Shuster proposed the QUaternion ESTimator algorithm [29], abbreviated to QUEST, in order to avoid the complex computations to obtain the eigenvectors. To avoid analytically compute λ_{max} Shuster first wrote the λ polynomial equation and then used the well known numerical method, Newton-Rapshon, to rapidly achieve an optimal λ.

$$\psi(\lambda) \quad = \quad det(\lambda I_{4\times 4} \, - \, K) \tag{5.21}$$

To ease the use of Newton-Raphson the equation was rewritten to:

$$\psi(\lambda) \quad = \quad \lambda^4 \, - \, (a+b)\lambda^2 \, - \, c\lambda \, + \, (ab+cs-d) \tag{5.22}$$

where

$$a \quad = \quad tr(B)^2 \, - \, tr(adjB + B^T) \tag{5.23}$$

$$b \quad = \quad tr(B)^2 \, - \, z^t z \tag{5.24}$$

$$c \quad = \quad tr(B + B^T) \, + \, z^T tr(B + B^T) z \tag{5.25}$$

$$d \quad = \quad z^T tr(B + B^T)^2 z \tag{5.26}$$

Then the best starting point λ_0 was shown to be the sum of the sensor weights as the results achieved by Davenport's algorithm proven them to be very close to λ_{max}.

$$\lambda_{max} \quad \approx \quad \lambda_0 \quad = \quad \sum_{i=1}^{n} a_i \tag{5.27}$$

Usually one iteration of Newton-Raphson is sufficient to provide accurate results.
Once the optimal eigenvalue has been calculated, it is necessary to relate it to the respective eigenvector. The easier method is to use the following equations:

$$\hat{\mathbf{q}} \quad = \quad \frac{1}{\sqrt{g^2 \, + \, \gamma^2}} \begin{bmatrix} g \\ \gamma \end{bmatrix} \tag{5.28}$$

Where the g and γ are given by the equations:

$$g \quad = \quad \left[\alpha I + (\lambda - tr(B))S + S^2\right] z \tag{5.29}$$

$$\gamma \quad = \quad \alpha \left[\lambda + tr(B)\right] \, - \, det(S) \tag{5.30}$$

$$\alpha \quad = \quad \lambda^2 \, - \, tr(B)^2 \, + \, tr(adj(S)) \tag{5.31}$$

5.2.3 SVD

Another of the first solutions to Wahba's problem was the Singular Value Decomposition or SVD which uses matrices operations to solve the problem [30]. The guideline to this method is to decompose the B matrix into a product of a matrices U,S and V, where U and V are two orthogonal matrices. Then those matrices can be used to represent the attitude matrix.

$$B \quad = \quad USV^T \tag{5.32}$$

Where S is a diagonal matrix with the values $[s_1, s_2, s_3]$. And it was demonstrated that the matrix A was given by:

$$\hat{A} \quad = \quad U\, diag\left([1 \quad 1 \quad det(U) \cdot det(V)]\right)\, V^T \tag{5.33}$$

The λ_{max} can be calculated through the diagonal of the matrix S and the determinants of U and V.

$$\lambda_{max} \quad = \quad s_1 \,+\, s_2 \,+\, det(U)det(V) \tag{5.34}$$

This method also has a non-unique solution when the two highest eigenvalues aren't distinct.

5.2.4 FOAM with two Observations

Another matrix approach is the Fast Optimal Attitude Matrix (FOAM) presented in [31] developed as an alternative to SVD method in order to avoid the attitude profile matrix, B, decomposition operations. From the results of SVD, it is possible to write the following statements.

$$det(B) \quad = \quad s_1\, s_2\, s'_3 \tag{5.35}$$

$$||B||_F^2 \quad = \quad s_1^2 + s_2^2 + (s'_3)^2 \tag{5.36}$$

$$adj(B^T) \quad = \quad U_+ diag([s_2 s'_3 \quad s'_3 s_1 \quad s_1 s_2])V_+^T \tag{5.37}$$

$$BB^T B \quad = \quad U_+ diag([s_1^2 \quad s_2^2 \quad s'^2_3])V_+^T \tag{5.38}$$

Where U_+ V_+ are given by:

$$U_+ \quad = \quad U\, diag([1 \quad 1 \quad det(U)]) \tag{5.39}$$

$$V_+ \quad = \quad V\, diag([1 \quad 1 \quad det(V)]) \tag{5.40}$$

With this the optimal attitude matrix could be written as:

$$\hat{A} \quad = \quad \frac{(\lambda_{max}^2 + ||B||_F^2)B + 2\lambda_{max} adj(B^T) - 2BB^T B}{\lambda_{max}(\lambda_{max}^2 - ||B||_F^2) - 2det(B)} \tag{5.41}$$

The polynomial characteristic equation as function of λ is then given by:

$$\psi(\lambda) \quad = \quad (\lambda^2 - ||B||_F^2)^2 - 8\lambda det(B) - 4||adj(B||_F^2 \tag{5.42}$$

This algorithm was developed for multiple observation vector, being these equations the generic representation so later another development was presented regarding the specific solution for two observation

vectors which simplify the equations with the following results:

$$det(B) = 0 \tag{5.43a}$$

$$BB^T = a_1^2 b_1 b_1^T + a_2^2 b_2 b_2^T + a_1 a_2 (r_1 \cdot r_2)(b_1 b_2^T + b_2 b_1) \tag{5.43b}$$

$$||B||_F^2 = a_1^2 + a_2^2 + 2a_1 a_2 (r_1 \cdot r_2)(b_1 \cdot b_2) \tag{5.43c}$$

$$adj(B^T) = a_1 a_2 (b_1 \times b_2)(r_1 \times r_2)^T \tag{5.43d}$$

$$||adj(B)||_F = a_1 a_2 ||r_1 \times r_2|| b_1 \times b_2|| \tag{5.43e}$$

These simplifications allow for more simple solution of highest eigenvalue and the optimal attitude matrix.

$$\lambda_{max} = \sqrt{||B||_F^2 + ||r_1 \times r_2|| b_1 \times b_2||} \tag{5.44}$$

$$\begin{aligned} \hat{A} &= \left(\frac{a_1}{\lambda_{max}}\right)\left(b_1 r_1^T + (b_1 \times b_\times)(r_1 \times r_\times)^T\right) \\ &+ \left(\frac{a_2}{\lambda_{max}}\right)\left(b_2 r_2^T + (b_2 \times b_\times)(r_2 \times r_\times)^T\right) + b_\times r_\times \end{aligned} \tag{5.45}$$

As it can be seen this solution is similar to the TRIAD method and actually is the same when $a_2 = 0$.

5.3 Enhanced QUEST

The first dynamic algorithm to be presented in this thesis is the Enhanced Quaternion Estimator [32] which is a simple and quick approach to solve the attitude determination while in eclipse. The method uses an estimator based on the gyro readings to generate a quaternion estimation $\tilde{\mathbf{q}}$ and fuses it with the quaternion $\mathbf{q}_{QUEST}$ obtained through QUEST. Is also possible to take the same approach but with the output of the TRIAD algorithm instead.

$$\hat{\mathbf{q}} = (1 - \beta) \cdot \tilde{\mathbf{q}} + \beta \cdot \mathbf{q}_{QUEST} \tag{5.46}$$

Where $\hat{\mathbf{q}}$ is the resulting quaternion while β is the varying weight that indicates the confidence of the TRIAD/QUEST output quaternion. The weights are dynamically computed in order to reduce the effect of the indeterminate case caused by the co-linearity of the reference vectors.

$$\beta = \left(1 - |r_1 \cdot r_2|^2\right) \beta_0 \tag{5.47}$$

The quaternion estimator is computed using the readings of the gyroscope to linearly propagate the last attitude quaternion reading.

$$\tilde{\mathbf{q}}_k \quad = \quad \Phi_k^q(\tilde{\boldsymbol{\omega}}_k)\ \hat{\mathbf{q}}_{k-1} \tag{5.48}$$

Where Φ_k^q is a $2nd$ order quaternion update matrix written in function of the angular rate, for more information see annex A.4. While in eclipse only the estimator component will be available (QUEST contribution will be disabled) and β will be set to zero in order to fully use the estimator component.

$$Eclipse \quad \Longrightarrow \quad \beta \quad = \quad 0 \Longrightarrow \quad \hat{\mathbf{q}}_k \quad = \quad (1\ -\ \beta) \cdot \tilde{\mathbf{q}}_k \tag{5.49}$$

5.4 Extended Kalman Filter

To address the problem of attitude estimation one of the most popular solutions was presented in 1960 by R.E Kalman [33]. This solution is a two step filter algorithm that uses the observations and their known noise statistical parameters to predict the system behaviour for an underlying linear system. From the original filter Kalman proposed many variations appeared throughout the years and a particular one is well know for it's ability to deal with non-linear systems, the Extended Kalman Filter. This variant linearises the system around a state and uses Jacobian matrices in the dynamic model to predict the evolution from the linearised state and is denoted as Extended Kalman [34]. The discrete system is described with the following equation:

$$\dot{\mathbf{x}}_k \quad = \quad f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \tag{5.50}$$

where u represent the control input and t_k the time at the k^{th} step and $\mathbf{w}_k$ the noise associated to the dynamic motion. The system is then linearised around the point $\hat{x}$ using the partial derivatives of the transition function f with respect to $\mathbf{x}$, $\mathbf{u}$ and $\mathbf{w}$. The Taylor series expansion is used to simplify its computation and representation [34].

$$\Delta\dot{\mathbf{x}}_k \quad = \quad \mathrm{F}\,\Delta\mathbf{x}_k\ +\ \mathrm{B}\,\Delta\mathbf{u}_k\ +\ \mathrm{G}\,\Delta\mathbf{w}_k \tag{5.51}$$

where the partial derivatives F, B and G are the following:

$$\mathrm{F} = \left.\frac{\partial f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k)}{\partial \mathbf{x}_k}\right|_{x_k = \hat{x}} \tag{5.52a}$$

$$\mathrm{B} = \left.\frac{\partial f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k)}{\partial \mathbf{u}_k}\right|_{x_k = \hat{x}} \tag{5.52b}$$

$$\mathrm{G} = \left.\frac{\partial f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k)}{\partial \mathbf{w}_k}\right|_{x_k = \hat{x}} \tag{5.52c}$$

These equations describe the system behaviour but to increase its accuracy is also necessary to include observers model to compare the expected behaviour to and eventually correct it. The observation model in non-linear systems should also be linearised with a similar process in which the observation vector is written through the sensibility function h:

$$\mathbf{y}_k = h_k\left(\hat{\mathbf{x}}_k\,,\, \mathbf{r}_k\right) \tag{5.53}$$

The system is then linearised so that the observation vector variation is also represented with the local derivative H known as the sensibility matrix.

$$\Delta\mathbf{y}_k = \mathrm{H} \cdot \Delta\mathbf{x}_k \tag{5.54a}$$

$$\mathrm{H} = \frac{\partial h(\mathbf{x}_k, \mathbf{r}_k)}{\partial \mathbf{x}_k} \tag{5.54b}$$

5.4.1 MEKF State Equations

The Multiplicative Extended Kalman Filter [5][35] is one of the Kalman Filter variations developed in 1969 for optimal attitude estimation. The concept of the algorithm underlies in the representation of the estimation error with a quaternion which is a result from the quaternion multiplication between the propagated and estimated quaternions. As previously introduced, this quaternion error represents the rotation from the predicted frame to the corrected body frame, i.e.

$$\mathbf{q} = \partial\mathbf{q} \otimes \hat{\mathbf{q}} \tag{5.55}$$

From [5] when the representing small rotations the relation between the quaternion error axis part and the rotation vector error axis can be simplified to the following equation:

$$\partial \mathbf{q} \quad \sim \quad \frac{1}{2}\begin{bmatrix} 2 \\ \partial \boldsymbol{\vartheta} \end{bmatrix} \tag{5.56}$$

The vector $\partial\vartheta$ shall be used as the three first state coordinates with the advantage of having 1 less coordinate than the quaternion representation, thus reducing the computation effort. To correct the gyro drift, a common solution is to include the bias derivative $\dot{\beta}$ in the state vector to later integrate it. From the gyroscope model discussed in the previous chapter the discrete gyro measurements are as follows:

$$\hat{\boldsymbol{\omega}}_k \quad = \quad \boldsymbol{\omega}_k \; + \; \boldsymbol{\beta}_g|_k \; + \; \mathbf{n}_{ARW}|_k \tag{5.57}$$

$$\dot{\boldsymbol{\beta}}_k \quad = \quad \mathbf{n}_{rrw}|_k \tag{5.58}$$

So the full error state vector the algorithm will be updating is:

$$\Delta \mathbf{x} \quad = \quad \begin{bmatrix} \partial\vartheta \\ \Delta\boldsymbol{\beta}_g \end{bmatrix} \tag{5.59}$$

The angular error derivate $\dot{\partial\vartheta}$ can be written through the quaternion derivative and the previous definition of ϑ as explained in [36].

$$\dot{\partial\vartheta} \quad = \quad -\boldsymbol{\omega} \times \partial\vartheta \; - \; \Delta\boldsymbol{\beta}_g \; - \; \mathbf{n}_{arw} \tag{5.60}$$

While the derivate of the second component of the state vector, the bias deviation error, written as $\dot{\Delta\boldsymbol{\beta}}_g$ is given by equation 5.58. The estimated derivative is zero which leaves bias residual error equal to the modelled derivate:

$$\dot{\Delta\boldsymbol{\beta}}_g = \mathbf{n}_{arw} \tag{5.61}$$

These equations can now be written into state transition functions around the linearised point as defined in 5.52a:

$$\mathrm{F}_k \quad = \quad \begin{bmatrix} \mathbf{I}_{3\times3} - \Delta t\,[\boldsymbol{\omega}_{k-1}]_\times & \mathbf{I}_{3\times3}\Delta t \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix} \tag{5.62}$$

For the state modelling to be complete the last step is to linearise the noise model with it's partial derivative G_k. The noise vector $\mathbf{w}_k$ can be divided into two components, one more related to the first state vector, $\mathbf{x}_k$, through the instantaneous angular measurement error $\delta\vartheta$ and the other more related to the gyro bias instability and drift which shall reflect into the other state vector $\Delta\beta$.

$$\mathbf{w}_k = \begin{bmatrix} \mathbf{n}_{arw} \\ \mathbf{n}_{rrw} \end{bmatrix}_k \tag{5.63}$$

Using this formulation with the state vector description made with 5.60 and 5.61 the Jacobian matrix G can now be represented as the following:

$$\mathbf{G}_k = \begin{bmatrix} -\mathbf{I}_{3\times3}\Delta t & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3}\Delta t \end{bmatrix} \tag{5.64}$$

As stated in 4.3.1 both noise vectors are assumed to be uncorrelated and defined as white Gaussian noise distributions which standard deviation is defined as the σ_{arw} and σ_{rrw}.

$$\langle \mathbf{n}_{arw}\mathbf{n}_{rrw} \rangle = 0 \tag{5.65}$$

$$\langle \mathbf{n}_{rrw}\mathbf{n}_{rrw} \rangle = \sigma_{arw}^2 I_{3\times3} \tag{5.66}$$

$$\langle \mathbf{n}_{arw}\mathbf{n}_{arw} \rangle = \sigma_{arw}^2 I_{3\times3} \tag{5.67}$$

This would also mean that the process covariance is simplified into a diagonal matrix representation.

$$W_k = \langle \mathbf{w}_k \mathbf{w}_k^T \rangle = \begin{bmatrix} \sigma_{arw}^2 I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \sigma_{rrw}^2 I_{3\times3} \end{bmatrix} \tag{5.68}$$

5.4.2 MEKF Update Equations

The filter makes predictions about the states until it receives measurements updates upon which it corrects those predictions. In this section it is illustrated how the algorithm uses those informations to update it's predictions, starting by defining the Kalman gain K that corrects each prediction step. It is necessary to compute the gain before each prediction to continuously compensate the estimations using the covariance matrix. The gain K of a k^{th} instant is given by the following equation:

$$K_k = P_k^- \, \mathrm{H}_k^T \left[\mathrm{H}_k \, P_k^- \, \mathrm{H}_k^T + \mathrm{R} \right]^{-1} \tag{5.69}$$

Where P_k^- is the covariance matrix estimation from the previous state, H_k is the derivative of the sensibility matrix and R is the measurement weight matrix which shall vary according to the observations used to update the state as it will be discussed further on section 5.4.3. Upon computing the Kalman gain, it follows the update of the state and covariance predictions which are computed using the previous state and covariance.

$$\begin{aligned} \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + K_k\,[\mathbf{y}_k - H_k\,\hat{\mathbf{x}}_k] = \hat{\mathbf{x}}_k^- + K_k\,\mathbf{z}_k \\ P_k &= [I - K_k\,\mathrm{H}_k]\;P_k^-\;[I - K_k\,\mathrm{H}_k]^T + K_k\,R\,K_k^T \end{aligned} \tag{5.70}$$

Where the $\hat{\mathbf{x}}_k$ and P_k^- represent respectively the previous state estimation and covariance, $\mathbf{z}_k$ represent the measurement residual error which is explained in the following section.

5.4.3 MEKF Observation Model

While building the observation model two possibilities could be considered regarding which measurements should be taken into consideration. The first approach would use the raw sensors data and compare them to the estimated reference values to generate the estimated angular error. Another option is an extension of the previous work in static algorithms, using them to provide a quaternion measurement which will be converted into an angular error when compared to the filter estimate. There obvious advantages in using a quaternion input which ultimately can be resumed in computational speed. However using the raw sensor data to update the filter is more robust while in eclipse and with some adjustments can be made equally light computationally wise. Therefore, it was chosen to use the sensor data as the filter input, based on the work presented in [5] and[37].
Using the sensors measurements directly, the k^{th} residual error model $\mathbf{z}_k$ would be given by the difference between the measurement obtained $\hat{\mathbf{y}}_k$ and the predicted $h(\mathbf{x}_k^-)$:

$$\mathbf{z}_k = \hat{\mathbf{y}}_k - h_k\left(\hat{\mathbf{x}}_k^-\right) \tag{5.71}$$

$$= \begin{bmatrix} \left[\hat{\mathbf{b}}_1\right]_{3\times1} \\ \left[\hat{\mathbf{b}}_2\right]_{3\times1} \end{bmatrix} - \begin{bmatrix} A\left(\hat{\mathbf{q}}_k^-\right)\cdot[\mathbf{r}_1]_{3\times1} \\ A\left(\hat{\mathbf{q}}_k^-\right)\cdot[\mathbf{r}_2]_{3\times1} \end{bmatrix} \tag{5.72}$$

Where $\hat{\mathbf{b}}$ are the measurements vectors and $A\left(\hat{\mathbf{q}}_k^-\right)$ is the attitude matrix build from the last attitude quaternion estimate. This will result in the following measurement sensitivity matrix:

$$\mathrm{H}_k = \begin{bmatrix} \left(A\left(\hat{\mathbf{q}}_k^-\right)\mathbf{r}_1\right)_\times & 0_{3\times3} \\ \left(A\left(\hat{\mathbf{q}}_k^-\right)\mathbf{r}_2\right)_\times & 0_{3\times3} \end{bmatrix} \tag{5.73}$$

As the satellite will use 2 measurements the sensor covariance matrix R described in equation will be a diagonal 6×6 matrix constructed with each sensor covariance:

$$\mathrm{R} = \begin{bmatrix} [\mathrm{R}_1] & 0_{3\times3} \\ 0_{3\times3} & [\mathrm{R}_2] \end{bmatrix} \tag{5.74}$$

$$\mathrm{R}_i = \langle \Delta\mathbf{b}_i \cdot \Delta\mathbf{b}_i^T \rangle \tag{5.75}$$

Where $\Delta\mathbf{b}_i$ is the i^{th} sensor measurement error represented in $\mathcal{B}$. The large size of the matrices is sometimes problematic for low power MCU and while in eclipse only a quarter of the gain matrix contains usable information. To size of the gain matrix is related with the number of observations used, being 3 times the number of directions read. To reduce this Murrel's version of the filter could be implemented, decoupling the measurements and updating individually with each sensor data. This allowed to use a smaller 3×3 gain matrix for each sensor update this will be very helpful when inverting the covariances matrix for equation 5.69 as R was also reduced to a three by three matrix.

5.4.4 MEKF Propagation

While waiting for new observations to update the state with, the attitude quaternion and covariance matrix are propagated throughout the next time steps using state and covariance propagation equations. The state transition matrix for the states are given as a time integration of F which is computed with an exponential matrix as seen before and in this can be expanded into a Taylor Expansion series.

$$\Phi_{\mathbf{x}}(\Delta t) = e^{(\mathrm{F}\Delta t)} \tag{5.76}$$

$$= I_{6\times6} + \mathrm{F}\Delta t + \frac{1}{2!}\mathrm{F}^2\Delta t^2 + \frac{1}{3!}\mathrm{F}^3\Delta t^3 + ... \tag{5.77}$$

This will result in a two part matrix one similar to the one used to integrate the quaternion in EQUEST algorithm with the difference of including the last bias estimation to correct the measured angular rate. The second part which should update the bias in fact maintains it, i.e. assuming there is no evolution of the bias in the propagation step. The state propagation is shortened to a quaternion integration with the corrected angular rate. Another important change is to propagate the covariance matrix which is also done with the transition matrix Φ but also includes the process covariance matrix W_k and integrated into the process noise transitions matrix Q_d as demonstrated in [38]:

$$Q_d(\Delta t) = \int_{t_0}^{t_0+\Delta t} \Phi_{\mathbf{x}}\, \mathrm{G}\, W_k\, \mathrm{G}^T\, \Phi_{\mathbf{x}}^T \tag{5.78}$$

Finally the full propagation step is given by the equations:

$$\hat{\mathbf{q}}_{k+1}^{-} \;=\; \Phi_{k+1}^{q}\left(\hat{\boldsymbol{\omega}}_{k}^{-} - \hat{\boldsymbol{\beta}}\right)\hat{\mathbf{q}}_{k} \tag{5.79}$$

$$P_{k+1}^{-} \;=\; \Phi_{\mathbf{x}}\, P_{k}^{-}\, \Phi_{\mathbf{x}}^{T} \;+\; \mathrm{G}_{k}\, Q\, \mathrm{G}_{k}^{T} \tag{5.80}$$

5.5 Complementary Filter

For a low power MCU, running most Kalman filter variations are a demanding task. But in the work developed in [39] presents a much simpler attitude estimator and later this give origin to the sensor input form known as Explicit Complementary Filter [40]. The model uses the known rotation from $\mathcal{B}$ to $\mathcal{I}$ as the default designation for attitude opposed to the one used here until now.

$$\mathtt{T} \;=\; A^{T} \;=\; \left[R_{\mathcal{B}}^{\mathcal{I}}\right] \tag{5.81}$$

Such that the kinematics of the system can be described by the body angular rate of $\mathcal{B}$ relative to $\mathcal{I}$.

$$\dot{\mathtt{T}} \;=\; \mathtt{T}\left[\boldsymbol{\omega}_{\mathcal{B}}\right]_{\times} \tag{5.82}$$

$$\;=\; \left(\mathtt{T}\boldsymbol{\omega}_{\mathcal{B}}\right)_{\times}\mathtt{T} \tag{5.83}$$

An auxiliary reference frame $\tilde{\mathcal{B}}$ was created to represent the attitude estimation, where the rotation from $\tilde{\mathcal{B}}$ to $\mathcal{I}$ is represented by $\hat{\mathtt{T}}$ allowing to write the estimation error as the rotation matrix $\tilde{\mathtt{T}}$ such that:

$$\tilde{\mathtt{T}} \;=\; \hat{\mathtt{T}}^{T}\mathtt{T} \;=\; \left[R_{\tilde{\mathcal{B}}}^{\mathcal{I}}\right]^{T}\left[R_{\mathcal{B}}^{\mathcal{I}}\right] \tag{5.84}$$

The purpose of the estimator is to converge these two frames such that $\tilde{\mathtt{T}} = I_{3\times 3}$. Based on the rotation dynamics in 5.82 the estimator dynamics general form was written as the derivative of $\hat{\mathtt{T}}$:

$$\dot{\hat{\mathtt{T}}} \;=\; \hat{\mathtt{T}}\left(\hat{\boldsymbol{\omega}}_{\mathcal{B}} - \beta_{g} + k_{p}\,\gamma\right)_{\times} \tag{5.85}$$

where k_p is a positive scalar gain and γ is a correction term generated through $\tilde{\mathtt{T}}$ and there are two different ways to compute this factor. The first option used in the active and passive complementary formulations reconstructs an attitude matrix identified as $\mathtt{T}_y$ from the sensors to then compute $\tilde{\mathtt{T}}$ with 5.84. The disadvantage of this option is the propagation of the error through the reconstruction of $\mathtt{T}_y$ however in the latter paper [[40]] presents a reformulation of the Passive Complementary Filter so that γ can be computed from direct sensors measurements.

$$\gamma \;=\; \sum k_{i}\,\left(\mathbf{b}_{i} \times \hat{\mathtt{T}}^{T}\mathbf{r}_{i}\right) \tag{5.86}$$

where k_i is the positive scalar gain of i sensor, $\mathbf{b}_i$ is the i sensor direction measurement in $\mathbb{B}$ and $\mathbf{r}_i$ is i known direction in $\mathcal{I}$. As explained before there is computational advantage in using the previous quaternion representation of the attitude and using the quaternion dynamics presented in chapter 2 its possible to describe the previous model with quaternion dynamics.

$$\dot{\hat{\mathbf{q}}} \quad = \quad \Omega\left(\hat{\boldsymbol{\omega}}_{\mathcal{B}} - \boldsymbol{\beta}_g + k_p\,\gamma\right) \otimes \hat{\mathbf{q}} \tag{5.87}$$

where $\hat{\mathbf{q}}$ is the representing the rotation from $\mathcal{I}$ to $\mathcal{G}$. Another reason to use this filter is the possibility to include an estimation of the gyroscope bias $\boldsymbol{\beta}_g$, allowing to correct any bias differences using the correction factor γ:

$$\dot{\hat{\boldsymbol{\beta}}}_g \quad = \quad -k_g\,\gamma \tag{5.88}$$

Chapter 6

Control Algorithms

The satellite was designed to be able to behave in different modes in order to handle a few predicted problems/situations, i.e. power loss, these states have different operational demands from the ADCS. These demands can be grouped into two different operational modes. When a major error is diagnosed or when the satellite is initialized the satellite falls into the ”safe” state. In this state the payload is power on thus the only operational requirement is to maintain the spin rate of the satellite under 20^o. This mode is usually named ”detumbling”. In all other modes the payload needs the ADCS to point the satellite to NADIR with a 20^o accuracy, this ADCS operational mode will be named ”pointing”. When in this state if the ADCS finds a problem that precludes fulfilling the mission requirement a report should be sent to induce the satellite transition to the safe mode and consequently the fall back to the ”detumbling” mode. The most usual approach used to design controllers is to compute the torque to bring the system to the desired state, calling it desired toque and then use a function to translate it into the necessary magnetic dipole input through another function.

$$m_b = f(\tau_d) \tag{6.1}$$

In the magnetorquers case this is a problem, as can be seen in equation 4.17, the created torque is always orthogonal to the magnetic dipole and magnetic field vector being impossible through the magnetorquers magnetic dipole to create a torque vector that has a non-null projection onto the magnetic field direction. This is the major disadvantage of using magnetic actuators, an approximation could be made by using a projection of the desired torque vector onto the axis orthogonal as seen in fig. 6.1.

This projection is obtained with a mapping function, using the cross product of the desired torque with the magnetic field direction vector to compute the control magnetic vector.

$$m_b = \frac{\mathbf{B}_\mathcal{B} \times \boldsymbol{\tau}_d}{|\mathbf{B}_\mathcal{B}|^2} \tag{6.2}$$

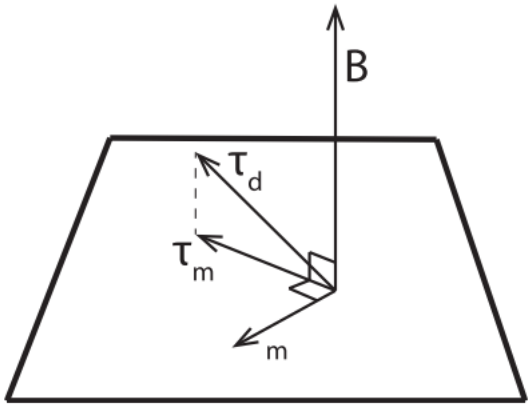

Figure 6.1: Desired torque τ_d projection and resulting torque generated τ_m. Image from [41]

In control is common to use the energy equation into a Lyapunov function to prove the stability of the controller. From the energy point of view the satellite kinematic energy V is described by the sum of the kinetic and potential energy:

$$V_e \quad = \quad V_{pot} \; + \; V_t \; + \; V_{rot} \tag{6.3}$$

Where the potential energy V_{pot} and translational energy V_t are related to the orbit energy while the rotational energy relates to the rotation of $\mathcal{B}$ relative to $\mathcal{I}$. From [42] the energy equations can be specified as:

$$V_{pot} \quad = \quad -\frac{\mu\, m_{sat}}{|\mathbf{r}_{\mathcal{I}}|} \; - \; \frac{1}{2}\,\omega_o^2\,(J_x \; + \; J_y \; + \; J_z) \; + \; \frac{3}{2}\,\omega_o^2\,\hat{\mathbf{o}}^T\,J\,\hat{\mathbf{o}} \tag{6.4a}$$

$$V_t \quad = \quad \frac{1}{2}\,m_{sat}\,\omega_o^2\,|\mathbf{r}_{\mathcal{I}}|^2 \tag{6.4b}$$

$$V_{rot} \quad = \quad \frac{1}{2}\,\boldsymbol{\omega}^T\,J\,\boldsymbol{\omega} \tag{6.4c}$$

6.1 Lyapunov Stability

It is good practice to prove the stability of the controllers proposed before testing it to ensure the solution will converge into a stability state. Among non linear systems Lyapunov's Method is a good approach to prove convergence near the equilibrium state. Lets consider a differential function $V(\mathbf{x} \,:\, \mathbb{D} \to \mathbb{R}$, written

with the dynamic states $\mathbf{x} \in \mathbb{D}$ such that:

$$V(\mathbf{x}_0) \quad = \quad 0 \tag{6.5}$$

$$V(\mathbf{x}) \quad > \quad 0 \quad \forall \quad \mathbf{x} \neq \mathbf{x}_0 \tag{6.6}$$

The system is proven to be asymptotically stable if the built function derivative $\dot{V}$ is negative-definite.

$$\dot{V} \quad \leq \quad 0 \tag{6.7}$$

6.2 Detumbling Algorithm

The first objective the ADCS has to accomplish is to reduce the spinning rate of the satellite until it gets lower than $5^o/s$ relative to the orbit frame. The objective of the detumbling controller will be to create the magnetic moment necessary to lower this rotational energy the minimum.

$$\boldsymbol{\tau}_c : \quad \boldsymbol{\omega} \quad \rightarrow \quad \mathbf{0} \tag{6.8}$$

When considering a $1\,kg$ satellite in $400\,km$ it takes, in average, around six months to fall so the orbital effects are rather slow meaning from the dynamics perspective the orbital energy could be considered constant.

6.2.1 B-dot

Among the magnetic controlled satellites the most widely used solution provided by the literature is the B-dot controller. This controller generates a magnetic dipole based on the rate of change of the Earth magnetic field measured vector but with opposite sign. The magnetic rate is equal change in the magnetic field direction vector and the angular rate of the frame $\mathcal{B}$ w.r.t. $\mathcal{E}$ and eventually this would align the satellite with the magnetic geomagnetic field. As the magnetic field derivative can attain values between $0^o/s$ and $30^o/s$ the constant gain solution sometimes is preferable to use a normalized derivative.

$$m_b \quad = \quad -\,k_b\,\dot{\mathbf{B}}_{\mathcal{B}} \tag{6.9}$$

It is possible to estimate the control torque $\hat{\boldsymbol{\tau}}$ generated by this magnetic dipole with the following equation:

$$\hat{\boldsymbol{\tau}}_c \quad = \quad -\frac{k_b}{||\mathbf{B}_\mathcal{B}||^2}\left(\dot{\mathbf{B}}_\mathcal{B} \times \mathbf{B}_\mathcal{B}\right) \tag{6.10}$$

According to [ref] the global asymptotic stability cannot be proven with this formulation of the problem but the angular rate will be reduced to a value near the orbit angular rate, for the expected $400\,km$ altitude orbit this value would be $0.0655^o/s$. Sometimes a Bang-Bang variant of this controller is used with a constant gain, using the $\dot{\mathbf{B}}_\mathcal{B}$ only to compute the signal of the dipole created.

$$m_b \quad = \quad -m_{\texttt{max}} \cdot \texttt{sign}\left(\dot{\mathbf{B}}_\mathcal{B}\right) \tag{6.11}$$

Where $m_{\texttt{max}}$ will be the maximum dipole generated for the defined gain.

6.2.2 Angular Rate Feedback

Another detumbling method can be used when the satellites have access to the gyroscope data, which is a feedback of the gyro readings into the dipole moment.

$$m_b \quad = \quad -k_b\,(\boldsymbol{\omega}_\mathcal{B} \times \mathbf{B}_\mathcal{B}) \tag{6.12}$$

Where k_b is the controller gain. According to [5] the torque generated with this function could be simplified to:

$$\tau_c \quad = \quad -k\left(I_{3\times 3} - \frac{\mathbf{B}_\mathcal{B} \cdot \mathbf{B}_\mathcal{B}^T}{|\mathbf{B}_\mathcal{B}|^2}\right)\boldsymbol{\omega}_\mathcal{B} \tag{6.13}$$

To prove its stability using a Lyapunov function based on the rotational energy equation of the satellite, with the following form:

$$V \quad = \quad \frac{1}{2}\omega_\mathcal{B}^T\, J\, \omega_\mathcal{B} \quad \geq \quad 0 \tag{6.14}$$

The derivative $\dot{V}$ of the Lyapunov function can be written using the 3.21 and 6.13:

$$\dot{V} \quad = \quad \frac{1}{2}\omega_{\mathcal{B}}^{T}\, k \left(I_{3\times 3} \, - \, \frac{\mathbf{B}_{\mathcal{B}} \cdot \mathbf{B}_{\mathcal{B}}^{T}}{|\mathbf{B}_{\mathcal{B}}|^{2}} \right) \boldsymbol{\omega}_{\mathcal{B}} \quad \leq \quad 0 \tag{6.15}$$

Whenever $\omega_{\mathcal{B}}$ is parallel to $\mathbf{B}_{\mathcal{B}}$, $\dot{V}$ assumes the value 0 making it negative semi-definite which doesn't allow to prove a global asymptotic stability through this method. However according to [43] this isn't an issue as that condition is briefly met and while the satellite doesn't produce torque during that instant it constantly reduces the torque whenever $\omega_{\mathcal{B}}$ isn't parallel to $\mathbf{B}_{\mathcal{B}}$.

6.3 Pointing Algorithm

The second operational mode NADIR pointing allows the ISTsat payload to complete its mission. The first requirement the ADS-B antenna needs is to be pointed to the Earth surface, ideally to the projection point of the satellite position on the earth surface, called NADIR. At NADIR the distance to the satellite is minimum, ensuring the best signal receiving efficiency. The NADIR direction in $\mathcal{E}$ is changing throughout the orbit so much as the position of the satellite itself so the control reference isn't static however in $\mathcal{O}$ the NADIR reference per definition coincides with o_3 meaning it is static. This proves the usefulness of using $\mathcal{O}$ for control.

$$\mathbf{q}_o \quad = \quad q\left(\left[R_{\mathcal{O}}^{\mathcal{B}}\right]\right) \tag{6.16a}$$

$$\quad = \quad \mathbf{q} \otimes q\left(\left[R_{\mathcal{E}}^{\mathcal{O}}\right]^{-1}\right) \tag{6.16b}$$

$$\quad = \quad \mathbf{q} \otimes q\left(\left[R_{\mathcal{O}}^{\mathcal{E}}\right]\right) \tag{6.16c}$$

The attitude quaternion $\mathbf{q}_o$ will represent the satellite attitude w.r.t. $\mathcal{O}$ while the desired attitude quaternion in $\mathcal{O}$ will be represented by $\mathbf{q}_d$.

$$\delta\mathbf{q} \quad = \quad \mathbf{q}_d \otimes \mathbf{q}^{-1} \tag{6.17}$$

The satellite is considered to be correctly pointing to nadir when the quaternion error between is equal to identity quaternion

$$\delta\mathbf{q} \quad \rightarrow \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{6.18}$$

Historically the most used controllers in magnetic actuated cubesats due to their simplicity versus it's performance ratio are the proportional derivative (PD) and the linear quadratic regulator (LQR). However the latter is not recommended because the magnetic field changes throughout the orbit also changing the control matrix being necessary to load the MCU with the solving of Algebraic Riccati Equation or accept a performance reduction by using an orbit averaged control matrix []. Both LQR variations presented higher risks than the already proven PD controller therefore were discarded.

6.3.1 Quaternion Feedback

The controller option chosen to study is a proportional derivative error feedback controller in which the error vector fed is the quaternion rotation from the orbit reference frame to the body reference frame [44].

$$\boldsymbol{\tau}_d \quad = \quad -\left(k_\epsilon\, \delta\mathbf{q}_\epsilon \; + \; k_\omega\, \boldsymbol{\omega}_\mathcal{B} \right) \tag{6.19}$$

Where $\delta\mathbf{q}_\epsilon$ is the vector part of the quaternion error. Combining with eq. 6.2 the magnetic moment to generate is given by the following equation:

$$m_b \quad = \quad -\frac{k_\epsilon}{||\mathbf{B}_\mathcal{B}||^2}\, \left(\mathbf{B}_\mathcal{B} \times\, \delta\mathbf{q}_\epsilon\right) \; - \; \frac{k_\omega}{||\mathbf{B}_\mathcal{B}||^2}\, \left(\mathbf{B}_\mathcal{B} \times \boldsymbol{\omega}_\mathcal{B}\right) \tag{6.20}$$

To study the stability of the controller a equation based on [5] was used to form a Lyapunov function V.

$$V \quad = \quad \frac{1}{4}\, \boldsymbol{\omega}_\mathcal{B}^T\, J\, \boldsymbol{\omega}_\mathcal{B} \, + \, \frac{1}{2}\, k_p\, \left(\delta\mathbf{q}_\epsilon\, {\delta\mathbf{q}_\epsilon}^T \, + \, (1 \, - \, \delta\mathbf{q}_\eta)^2 \right) \quad \geq \, 0 \tag{6.21}$$

Which is always positive except when $\omega = \mathbf{0}$ and $\delta\mathbf{q}_\eta = \mathbf{0}$ where it takes the value 0. The derivative $\dot{V}$ is now described as:

$$\dot{V} \quad = \quad \frac{1}{2}\boldsymbol{\omega}_{\mathcal{B}}^{T}\, J\, \dot{\boldsymbol{\omega}}_{\mathcal{B}} \,+\, k_p\, \left(\delta\mathbf{q}_{\epsilon}\, \delta\dot{\mathbf{q}}_{\epsilon}^{T}\right) \,+\, k_p\, (1 \,-\, \delta\mathbf{q}_{\eta})\, \delta\dot{\mathbf{q}}_{\eta} \quad \leq\, 0 \tag{6.22}$$

Including the equation 3.21 into the previous we arrive at a more detailed equation.

$$\begin{aligned} \dot{V} \quad = \quad -\frac{1}{2}\,\boldsymbol{\omega}_{\mathcal{B}}^{T}\,(\, k_{\epsilon}\,\delta\mathbf{q}_{\epsilon} \,+\, \; k_{\omega}\,\boldsymbol{\omega}_{\mathcal{B}}) \,+\, k_p\,\delta\mathbf{q}_{\epsilon}\,\delta\dot{\mathbf{q}}_{\epsilon}^{T} \\ +\, k_p\,(1 \,-\, \delta\mathbf{q}_{\eta})\,\delta\dot{\mathbf{q}}_{\eta} \end{aligned} \tag{6.23}$$

The equation can be simplified using the kinematic relations of $\dot{\mathbf{q}}_{\eta}$ and $\dot{\mathbf{q}}_{\epsilon}$ presented in chapter 7 of [5] and the previous control law 6.19.

$$\dot{V} \quad = \quad -\frac{1}{2}\,k_{\omega}\,\boldsymbol{\omega}_{\mathcal{B}}^{T}\,\boldsymbol{\omega}_{\mathcal{B}} \tag{6.24}$$

As $\dot{V}$ is always negative while k_w is positive, being negative definite and V is always positive the global asymptotic stability is considered proven.

Chapter 7

Results

To evaluate the performance of the determination and control algorithm solutions, a dynamic model was built in Simulink to simulate the satellite behaviour evolutio. When actuated by the computed magnetorquers duty-cycle as well as to model the noise in the sensors readings output. In this model are included the important equations of motion from chapter 3 and it was subjected to a short validation.
The disturbance torques described in equation 3.20a are critical to tune the satellite controller and evaluate the its performance but in order to do so, the satellite orbit needs to be defined first. The satellite will be launched from ISS and not having propulsion a good initial guess would be the ISS orbit.

Kepler parameter	Value
Semi-Major axis	$6790.76314\,km$
Eccentricity	0.0008434
Inclination	$51.95846^{\,o}$
Argument of Periapsis	$66.91663^{\,o}$
Right Ascending Node	$125.81904^{\,o}$
True Anomaly	$266.60826^{\,o}$

Table 7.1: ISS Kepler parameters taken from NASA website [45]

At $400\,km$ of altitude the air depends on temperature but using the reference density from [1] into equation 3.10 it could be considered to have a mean density around $10^{-14}\,kg/m^3$. For cubesats the center of pressure is around $0.01\,m$ away from the center of mass, while the aerodynamics properties are very difficult to model but according to [11] the ballistic coefficient [1] of a 1U cubesat is between 0.02 and 0.05,

[1] The ballistic coefficient is the product of drag coefficient and section area divided by the body mass $\left(\frac{A \cdot C_d}{m_{sat}}\right)$

allowing to estimate the aerodynamics torque using a value between of 0.03.

$h\ (km)$	400	300	200
$\lvert\mathbf{V}\rvert\ (m/s)$	$7672.54\,m/s$	$7729.83\,m/s$	$7788.43\,m/s$
$\rho\ (kg/m^3)$	1.026×10^{-10}	8.8953×10^{-12}	3.0354×10^{-12}
$\mathbf{F}_d\ (N)$	2.372×10^{-6}	7.0556×10^{-6}	8.2621×10^{-5}
$\boldsymbol{\tau}_d\ (Nm)$	2.372×10^{-8}	7.0556×10^{-8}	8.2621×10^{-7}

Table 7.2: Aerodynamic maximum force and torque at $200\,km$, $300\,km$ and $400\,km$ orbit altitudes

The magnetic residual dipole $\mathbf{m}_r$ is not known until the satellite is fully assembled and even then it will largely depend on which state the satellite is and the tasks which are performing however according to NASA report [46] there is a relation between the satellites mass and the magnetic dipole magnitude of 0.004 to $0.01\ Am^2/kg$ attending the satellite is low power the dipole.

$\langle\lvert\mathbf{m}_r\rvert\rangle$	$5\ mAm^2$
$3\sigma_{m_b}$	$2\ mAm^2$
$\boldsymbol{\tau}_b$	$2.5 \times 10^{-7}\ Nm$

Table 7.3: Magnetic torque maximum value for at $400\,km$

For some of the algorithms is easier to use the normalized sensor covariances instead of the absolute values. The sensors normalized covariance are the directional covariances and besides including the measurement errors should also include the errors due to normalization.

$$\Delta b \quad = \quad \frac{\mathbf{B}}{\mathbf{B}} - \frac{\hat{\mathbf{B}}}{\hat{\mathbf{B}}} \quad = \quad \frac{\mathbf{B}}{|\mathbf{B}|} - \frac{\mathbf{B} + \Delta\mathbf{B}}{|\mathbf{B} + \Delta\mathbf{B}|} \tag{7.1}$$

The matrix representation of the absolute variance in each axis is independent resulting in a diagonal matrix however with normalization there is a small codependency factor is introduced. To check this a Monte Carlo simulation was done with 10^6 points to establish a initial sensor covariance tuning point for the determination algorithms.

	$\langle \Delta b \cdot \Delta b^T \rangle$
Magnetometer (HMC5983)	$\begin{bmatrix} 0.0055 & 0.0018 & 0.0018 \\ 0.0018 & 0.0045 & 0.0015 \\ 0.0018 & 0.0015 & 0.0052 \end{bmatrix}$
Coarse Sun Sensor	$\begin{bmatrix} 0.1669 & -0.0048 & -0.0012 \\ -0.0048 & 0.1668 & 0.0014 \\ -0.0012 & 0.0014 & 0.4093 \end{bmatrix}$

Table 7.4: Initial conditions used for determination algorithms simulations.

After the normalization, the magnetometer covariance matrix is close to covariance built with the normalized absolute error achieved through dividing the specified expected error by the mean magnitude of the magnetic reading. Even so, the $RMSE$ of the magnetometer is $0.56\,^o$.

The coarse sun sensor covariance however is a little more erratic with a specially high covariance associated to the z axis as it only has one solar panel located in the $+z$ face while the other two axes have two panels each facing opposite directions. When looking into the probability density of the error depicted by figure 7.1 the RMSE found to be $22.13\,^o$ which in comparison to the magnetometer is quite high.

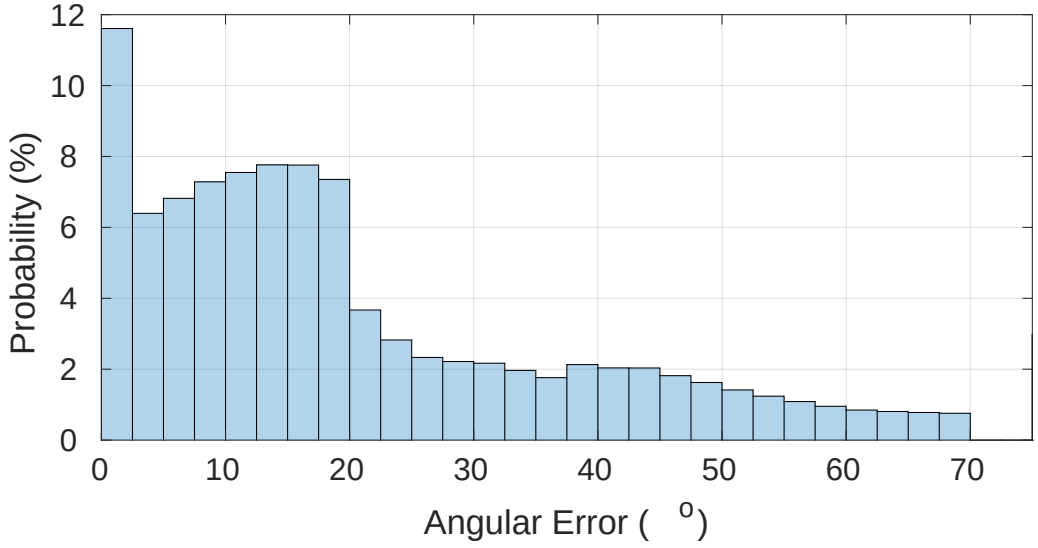

Figure 7.1: Probability density distribution of the coarse sun sensor absolute angular error.

To understand the CSS error better the data was separated according to the number of lighted faces allowing to identify the covariance for each separate lighting state. It was found the error associated to each face and the satellite spherical coverage and presented in the following table.

Lighted Photodiodes	0	1	2	3
$RMSE\,(^o)$	-	34.73	19.23	7.14×10^{-4}
Coverage $(\%)$	2.996	39.65	52.20	5.15

Table 7.5: Sun sensor RMSE and spherical coverage by the number of lighted photodiodes, for more information on pdf for each situation see annex D.1.

The CSS simulated presents good results when three photodiodes are sunlit, unfortunately this only happens for 5% of incident angles. When less than 3 photodiodes are lighted information about one of the tridimensional vector components is lost and the error starts to increases. This would account for almost 92% of the possible angles, most of its flight the satellite will have one or two photodiodes lighted. Finally there is the "blind" incident angles that correspond to angles where the -z face which has no photodiodes is heading towards the sunlight. The major error difference between sensors will be particular visible and important in the tuning of the determination algorithm parameters as well as the number of sunlit solar panels.

7.1 Determination Algorithms

To evaluate the algorithms the most important parameter is the angular error between the estimated quaternion $\hat{\mathbf{q}}$ and the true attitude quaternion $\mathbf{q}$. The error between them can be represented by a quaternion error $\delta\mathbf{q}$ which in this case shall represent the rotation from the estimated body frame $\tilde{\mathcal{B}}$ into the body frame $\mathcal{B}$.

$$\delta\mathbf{q} = \mathbf{q} \otimes (\hat{\mathbf{q}})^{-1} \tag{7.2}$$

$$= \mathbf{q}([R_{\mathcal{I}}^{\mathcal{B}}]) \otimes \mathbf{q}([R_{\tilde{\mathcal{B}}}^{\mathcal{I}}]) \tag{7.3}$$

The angular error $\delta\theta_q$ is given by the transformation from quaternion into rotations vector.

$$\delta\theta_q = 2\cos^{-1}(q_\eta) \tag{7.4}$$

The deterministic algorithms presented in chapter were tested with the two sets of initial conditions al-

though the system dynamics does not directly affect the algorithm performance. The dynamic solutions however depend upon the gyroscope readings and for this test the control was not used causing the angular to change very little. Thus is necessary establish test cases with different angular rates. To get a rough sense of the angular rates which the satellite would be subject to some testing using the controllers algorithms were done, using the true attitude to satellite as an input for the controller. After which the following set of cases were designed, the first case was set to start with a $2^o/s$ angular rate magnitude while the second will be $5^o/s$ which corresponds to the maximum rotation speed allowed before the satellite enters the detumbling mode.

Initial Condition	$\lvert\boldsymbol{\omega}_{\mathcal{B}}\rvert$	$\boldsymbol{\omega}_{\mathcal{B}}$	$\mathbf{q}$	$\hat{\mathbf{q}}$
Case 1	$2^o/s$	$\begin{bmatrix} 1.4664 \\ 1.2373 \\ 0.5647 \end{bmatrix}$	$\begin{bmatrix} 0.8526 \\ 0.0309 \\ 0.3937 \\ 0.3423 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$
Case 2	$5^o/s$	$\begin{bmatrix} -4.4480 \\ 0.0915 \\ 0.1075 \end{bmatrix}$	$\begin{bmatrix} -0.5504 \\ 0.3520 \\ 0.5282 \\ 0.5424 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

Table 7.6: Initial conditions used for determination algorithms simulations.

Lastly for some static algorithms is necessary to define the sensor weight parameters based on their covariance. Although the relative weight equation from equation 7.5 would suggest the best results would appear using $a_1 = 0.999$ and $a_2 = 0.001$ this would not gather enough information from the CSS to have a stable estimation. So it was decided to lower this relation between sigmas to $a_1 = 0.9$ and $a_2 = 0.1$.

$$a_1 = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \tag{7.5}$$

$$a_2 = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} \tag{7.6}$$

Using the first case, 4 orbits were simulated, the error graphics were added to annex D.2, then the static

algorithms were filtered to discard the eclipse instants and then were transcript into table 7.7, where the error evaluation parameters represented are the root mean square error ($RMSE$) and the maximum value found (E_{max}).

	Case 1		Case 2	
Method	$RMSE$ (^{o})	E_{max} (^{o})	$RMSE$ (^{o})	E_{max} (^{o})
TRIAD	34.97	179.997	77.147	180
SVD	22.57	179.992	49.789	180
QUEST	23.33	179.936	38.41	180
FOAM	31.92	179.919	63.03	180

Table 7.7: Deterministic algorithms error results when the satellite is lightened

The table highlights that neither method is within the acceptable error margin more so they all presented errors of 180^o. This can be explained looking at the angle between the reference vectors in figure 7.2(a) which sometimes were separated by less than 15^o and the accuracy the algorithms are able to obtain reduces the closer the vectors are to be parallel. In this graphic a partial plot of the QUEST algorithm error during the first orbit of the first example case is used to emphasize the relation of the angle between reference vectors and the attitude error. It is possible to see the more collinear the vectors are the worst the algorithm performs.

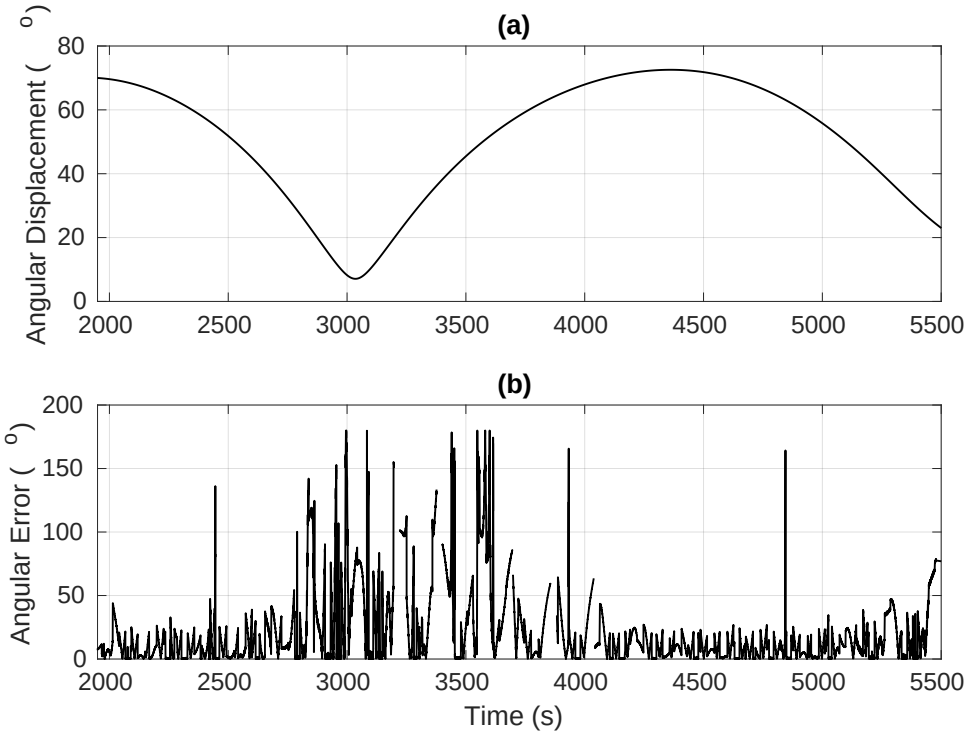

Figure 7.2: First orbit revolution close-up of the arc-degree displacement between magnetometer and CSS read vectors (a) and QUEST attitude error (b) from first case simulation.

The collinear problem together with the CSS performance explains the instability of the solutions however there is a notable performance difference between them. As expected the non-weighted TRIAD was the less stable method of the four a consequence of being the only algorithm that does not use relative weights in the sensor data. The difference obtain with this hardware set, highlights the advantage of using a greater weight for the more accurate magnetometer and a lower for the CSS. QUEST and SVD presented a close error performance, the latter a little better in the first case and worst in the second but both were clearly better than TRIAD and FOAM.

After testing the first group of algorithms, followed the evaluation of the EQUEST, MEKF and ECF. The dynamic algorithms simulations need more tuning for filter gains and covariances used in each method, for the first simulations the best results were granted using the values in table.

7.8.

Solution	Parameters
EQUEST	$\beta = 0.99$
MEKF$_s$	$R_1 = \left(5.5 \times 10^{-3}\right) I_{3\times3} \quad R_2 = diag([0.16, 0.16, 0.45])$ $\sigma_{arw} = 0.003 \quad \sigma_{rrw} = 3.17 \times 10^{-5}$
ECF	$k_1 = 0.95 \quad k_2 = 0.05 \quad k_p = 0.18 \quad k_g = 0.0003$

Table 7.8: Dynamic algorithms default parameters configuration used

The results obtain were unsatisfactory as can be seen in figure 7.3 the angular error of the Explicit Complementary Filter for the first simulation case isn't stable even after the eclipse. This is a problem as the reference signal for attitude controller is influenced by the attitude estimation and if the estimation isn't stable then the controller cannot perform a good attitude control. This problem was also verified in the other two solutions.

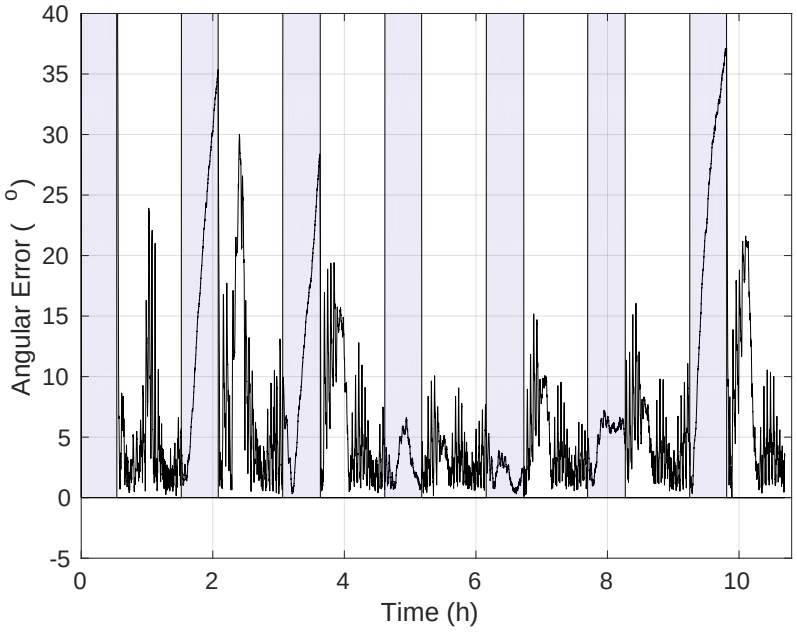

Figure 7.3: Instability of the ECF attitude error during 7 orbits simulation with the initial conditions of the case 1 (shaded regions indicate eclipse instances).

After testing the quaternion propagation, magnetometer and gyroscope data the reason for this was determined to be the corrupted information provided by the CSS. The found solution was to decompose the solution and provide different responses according to the to the number of panels lighted by the sun. This step would extended the tuning period more than was expected even so a stable configuration was obtain for each and is presented next.
The EQUEST method used a configurable base weight β_0 according to the number of panels lighted described in the following table 7.9.

	Number of Panels		
	1	2	3
β_0	0.99	0.95	0.50

Table 7.9: Default dynamic algorithms parameters configuration used

The results of the simulation obtain show that this method due to using the satellite dynamic information is much more stable than QUEST or any of the other deterministic algorithms presented here. A comparison between QUEST and EQUEST for the first case is represented in the figure . Where the deterministic has a much more disperse error in comparison to the continuous error of the EQUEST. In this specific case where the QUEST had an already reported RMSE around 23^o while in the same test period the EQUEST could provide a decrease of the RMSE to 13.58^o.

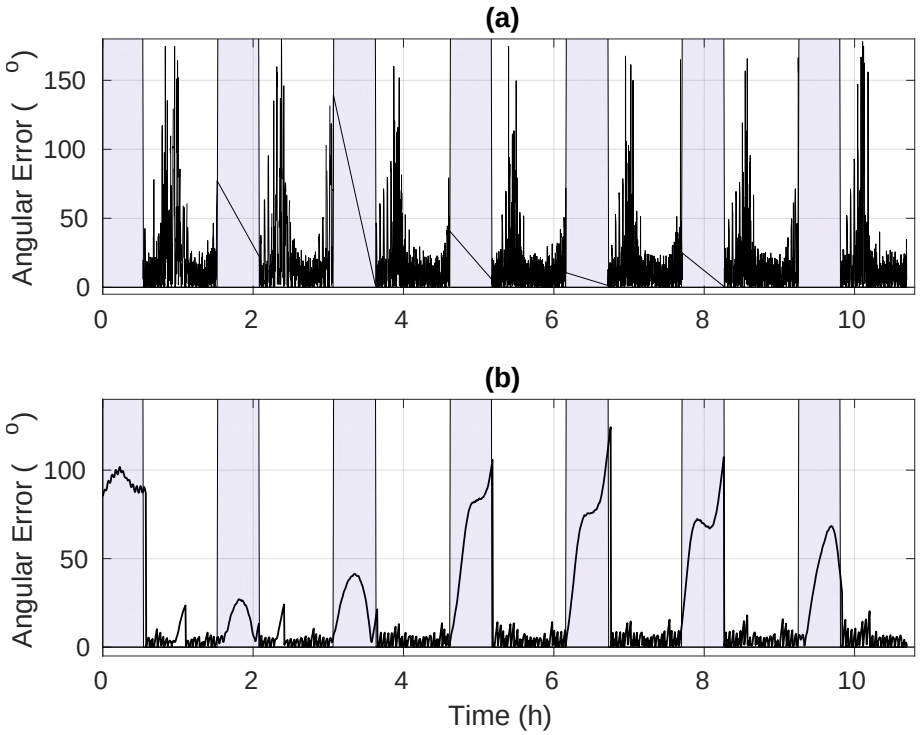

Figure 7.4: Comparison between the angular error of QUEST (a) and EQUEST (b) during 7 orbits simulation with the initial conditions of the case 1 (shaded regions indicate eclipse instances).

The second advantage of this method is that it can provide attitude estimations while the satellite is in eclipse. The satellite propagates the attitude it had right before entering eclipse which doesn't directly mitigate any existent attitude estimation error. The attitude error is propagated and the propagator error is also added to it, this leads to an error in eclipse that varies according to the results obtain right before entering eclipse and the propagator error. The simulations with the different initial conditions revealed the EQUEST can achieve an error up to 124.91^o presenting an overall RMSE of for case 1 and 35.12^o for the second, showing that that results are already better than any of the presented deterministic algorithms so far. The second simulation case showed a similar error during the daylight but in eclipse the error rose much slower which is related to the smaller relative error of the bias at higher angular rates.

	Case 1		Case 2	
Method	$RMSE\ (^o)$	$E_{max}\ (^o)$	$RMSE\ (^o)$	$E_{max}\ (^o)$
EQUEST	33.41	124.91	77.147	180

Table 7.10: EQUEST error simulation results after the satellite first daylight

The MEKF was the next algorithm to test and of all the three dynamic solutions was was the most problematic to tune. The adaptability of the algorithm to the number of lighted solar panels was introduced

in the CSS covariance matrix R_2 which assumes different values for each situation.

	Number of Lighted Panels		
	1	2	3
R_2	$\begin{bmatrix} 0.2213 & -0.0144 & -0.0120 \\ -0.0144 & 0.2283 & -0.0051 \\ -0.0120 & -0.0051 & 0.4916 \end{bmatrix}$	$\begin{bmatrix} 0.1168 & 0.0061 & 0.0044 \\ 0.0061 & 0.1056 & 0.0015 \\ 0.0044 & 0.0015 & 0.2903 \end{bmatrix}$	$\begin{bmatrix} 0.2213 & -0.0144 & -0.0120 \\ -0.0144 & 0.2283 & -0.0051 \\ -0.0120 & -0.0051 & 0.4916 \end{bmatrix}$

Table 7.11: MEKF CSS covariance parameter setup

The results obtain with MEKF solution showed a decrease of the attitude error throughout all orbit instances. During the daylight part of the orbit the first case simulation presented a RMSE of 3.38^o, the overall RMSE error obtained was 24.94^o in case 1 and 10.89^o in case 2 however due to setted initial conditions the satellite starts in eclipse resulting in a high error during that initial period of time. When excluding this initial period of around $1950\,s$ the overall RMSE reduces to 9.19^o for the first case and 7.38^o for the second case. The maximum error obtain after the first eclipse was 42.90^o and 28.76^o for the case 1 and 2 respectively.

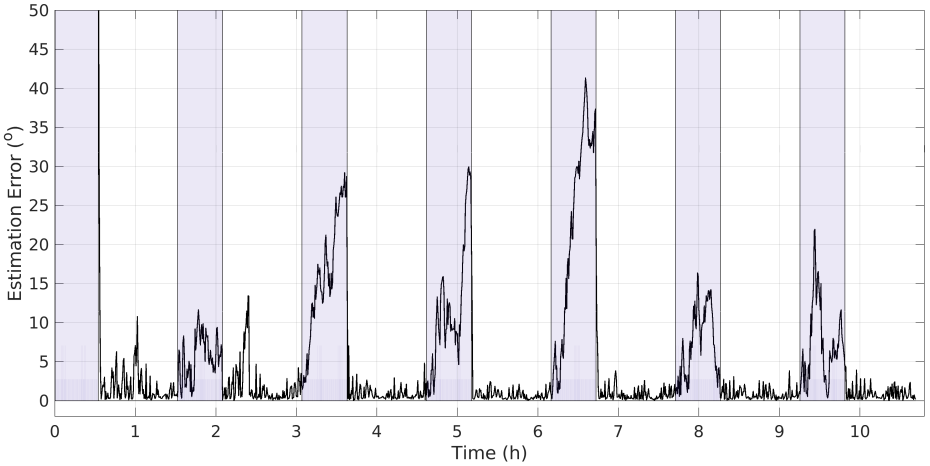

Figure 7.5: Angular error of the MEKF for 7 orbital revolutions simulation with the first case initial conditions (shaded regions indicate eclipse instances).

The greatest advantage the MEKF provides when compared to the EQUEST it's estimation of the angular rate bias error. As seen in the previous figure the attitude estimation error during eclipse is maintained below 43^o during eclipse independent of whatever value the error was before the start of eclipse. The estimation performance improvement during eclipse is due to the drift error decrease during propagation. As seen in the figure 7.6 the estimation after stabilization maintains the bias error well below $50\,mdeg/s$

in each axis after running for two hours, as the $RMSE$ is smaller than $15\,mdeg/s$ for each axis meaning the mean bias error should be even smaller which allows for mitigation of the gyro drift effect through time.

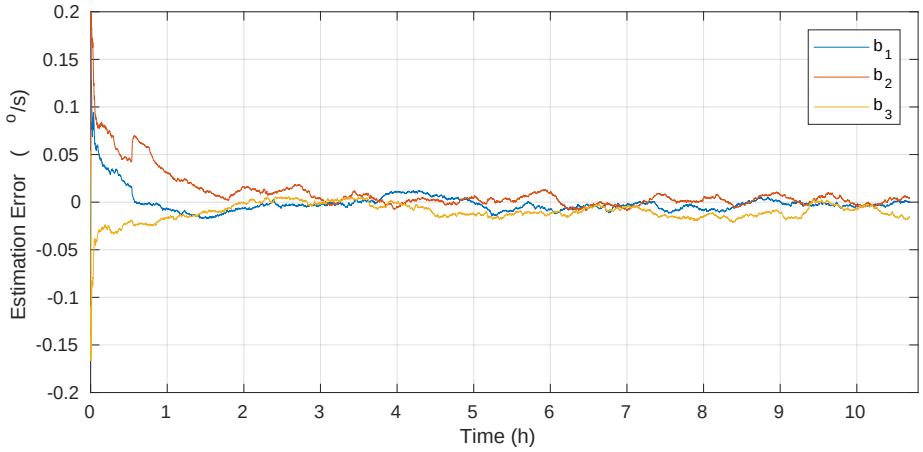

Figure 7.6: MEKF gyro bias estimation error in the 3 body axis (b_1 , b_2 , b_3) for 7 orbital revolutions simulation with the first case initial conditions (shaded regions indicate eclipse instances).

Using the established metrics to evaluate MEKF results the $RMSE$ and E_{max} is represented into the following table 7.12.

	Case 1		Case 2	
Parameter	$RMSE$	E_{max}	$RMSE$	E_{max}
Attitude	$9.19\,^o$	$42.90\,^o$	$7.39\,^o$	$28.76\,^o$
Bias	$10.9\,mdeg/s$	$70.1\,mdeg/s$	$32.1\,mdeg/s$	$40.2\,mdeg/s$

Table 7.12: MEKF attitude error and bias error results for both simulation cases after the first sunlight

At last the ECF was tuned to also have different gains according to the number of lighted panels. Instead of solely adapting the CSS weight k_2 to each situation, all the other gains were tuned as well for each case. The reason for this is as the bias compensation is done directly with the sensor readings and this would allow to correct the gyro drift even in the absence of sunlight. For a conservative approach over the CSS contribution it was chosen to discard the CSS data even if 1 panels is lighted as the expected error for this instance is still quite large.

	Number of Panels			
	0	1	2	3
k_1	0.1	0.1	0.9	0.5
k_2	0.0	0.0	0.1	0.5
k_p	0.01	0.001	0.06	0.20
k_g	0.000001	0.000001	0.00003	0.00003

Table 7.13: Explicit Complementary Filter decoupled parameter setup

The simulations of the two example cases revealed the ECF algorithm has a performance comparable to the MEKF but generally more stable although with significant differences between cases. The overall $RMSE$ over seven revolutions after receiving the first valid data set from CSS was 6.53^o for the first setup, for the second case the $RMSE$ was 6.33. The maximum error is also very different between examples, the first presented a maximum of 24.89^o after stabilization while the second revealed a maximum of . 18.58^o.

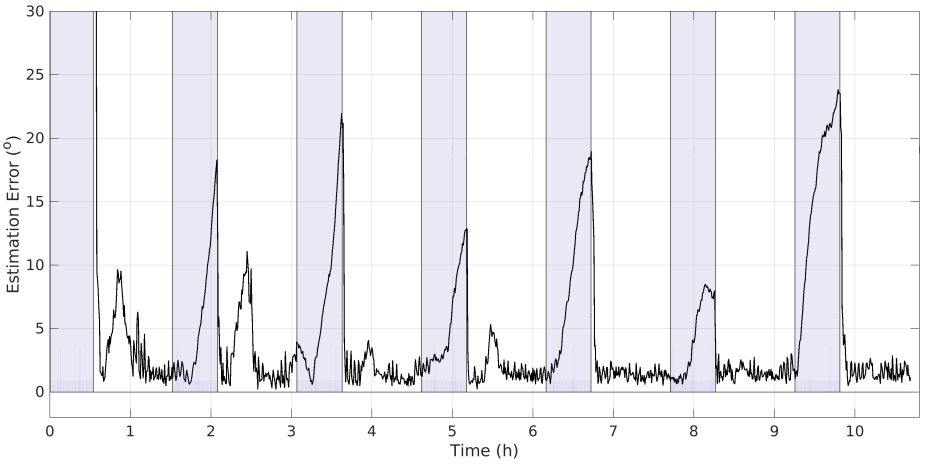

Figure 7.7: Angular error of the ECF for 7 orbital revolutions simulation with the first case initial conditions (shaded regions indicate eclipse instances).

The gyro bias estimation obtained with the ECF showed good results for the first case converging the error to zero in the initial period and maintaining the bias error in each axis below $40\,mdeg/s$ after stabilizing. On the other hand the second case simulation reveal the estimator error wasn't converging to zero in the initial period although it after stabilizing the error was below $40\,mdeg/s$ in each axis. For stability of the bias estimation, it was chosen a small integral gain resulting in a lower convergence rate when compared to the MEFK estimator and this is clear in the initial periods of both cases.

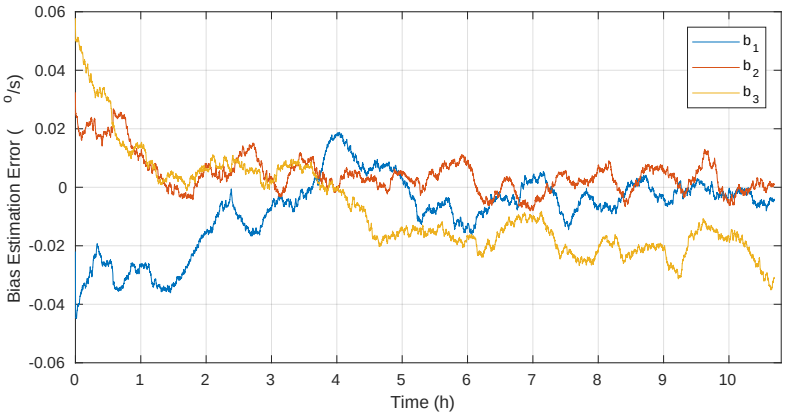

Figure 7.8: ECF gyro bias estimation error in the 3 body axis (b_1 , b_2 , b_3) for seven orbital revolutions simulation with the first case initial conditions (shaded regions indicate eclipse instances).

Using the same metrics to evaluate ECF results the $RMSE$ and E_{max} is summarized into the following table 7.12.

	Case 1		Case 2	
Parameter	$RMSE$	E_{max}	$RMSE$	E_{max}
Attitude	$6.53\,^o$	$24.89\,^o$	$6.33\,^o$	$20.90\,^o$
Bias	$12.6\,mdeg/s$	$26.2\,mdeg/s$	$32.1\,mdeg/s$	$60.3\,mdeg/s$

Table 7.14: ECF attitude error and bias error results for both simulation cases after the first sunlight

7.2 Detumbling Results

For the detumbling algorithm test the cases chosen used three different initial conditions, one low rate for a quick assessment of the most probable case. Three other with the established limit rate magnitude although with different directions of rotations and also different initial attitudes. All the first three test cases assume an orbit equal to the ISS and despite not expecting the orbit to be much different from it in the beginning the method may be used latter if the satellite gets into safe mode and the orbit is no longer similar to the ISS. To count for such situation the fourth and last test case was set to begin in an orbit with a different inclination even if in reality this orbital parameter is expected to change very little throughout the mission it is the one that provokes the most changes the magnetic field. This modification will induce an immediate change in the orientation of the field relative to the orbit and how quickly the magnitude changes, in this case as a lower inclination was selected, 15^o, the magnetic field magnitude will be more regular. This last test would try to prove the magnetic derivative algorithms reliability even in the case of a slower magnetic field rate.

Parameter	Case 1	Case 2	Case 3	Case 4
$\lvert\boldsymbol{\omega}_{\mathcal{B}}\rvert$ $(^o/s)$	15	30	30	30
$\boldsymbol{\omega}_{\mathcal{B}}$ $(^o/s)$	$\begin{bmatrix} 7.9410 \\ 8.7150 \\ 9.2730 \end{bmatrix}$	$\begin{bmatrix} 18.3270 \\ 0.9990 \\ 23.7300 \end{bmatrix}$	$\begin{bmatrix} 9.8010 \\ 19.6650 \\ 20.4270 \end{bmatrix}$	$\begin{bmatrix} 27.5400 \\ 10.3710 \\ 5.8260 \end{bmatrix}$
Orbit Inclination $(^o)$	51.95846	51.95846	51.95846	15.4387
$\mathbf{q}$	$\begin{bmatrix} 0.1190 \\ 0.4984 \\ 0.9597 \\ 0.3404 \end{bmatrix}$	$\begin{bmatrix} 0.5789 \\ 0.2214 \\ 0.7431 \\ 0.2523 \end{bmatrix}$	$\begin{bmatrix} 0.8575 \\ 0.2172 \\ 0.2339 \\ 0.4035 \end{bmatrix}$	$\begin{bmatrix} 0.6879 \\ 0.2080 \\ 0.6662 \\ 0.1992 \end{bmatrix}$

Table 7.15: Detumbling controller example conditions

To minimize energy consumption these cases were used to tune the minimum gain necessary for each algorithm to reach the threshold angular rate before completing one revolution. The B-dot seem to work best with $k = -0.00025$, the bang bang version of the B-dot settled with $k = -0.1$ and the gyro feedback controller used $k = -3500$. As the bang bang b-dot works with normalized vectors it limits the controller output to -0.1 however the other two controllers used fully use the magnetorquers drivers input range. Assuming the magnitude of the magnetic field changes between $20\,\mu T$ and $50\,\mu T$ the saturation of the controllers corresponding to the 80% duty-cycle can be achieved. The saturation of the gyro feedback is expected to be around $4.5^o/s$, the saturation of the b-dot was verified to be around $3200\,nT/s$. After running some simulations, the results were presented into the table 7.16 and as can be seen they remain consistent throughout the cases except for the case 2 simulations of the gyro feedback and b-dot as well. In both these simulations of case 2 the spacecraft was constantly aligning the angular rate axis with the magnetic field. This alignment caused the derivative of the magnetic field to be almost zero and the cross product of angular with the magnetic field to be zero, resulting in both cases in a near zero control torque which is not. As discussed in section 6.2.2 these are the points where the system doesn't converge into the design equilibrium point but also doesn't diverge as can be seen in 7.9 graphic where the angular rate in one axis was periodically constant for both algorithms.

Case	Parameter	B-dot	Bang Bang B-dot	Gyro-feedback
1	$\Delta t\ (s)$	217	1006	230
	$\Delta E\ (J)$	50	38	47
2	$\Delta t\ (s)$	17182	2094	7170
	$\Delta E\ (J)$	774	82	485
3	$\Delta t\ (s)$	390	4030	392
	$\Delta E\ (J)$	161	150	133
4	$\Delta t\ (s)$	332	2255	313
	$\Delta E\ (J)$	132	84	105

Table 7.16: Detumbling controllers simulation results

In some cases the gyro-feedback was less than 5% slower than the b-dot while revealing an improved energy consumption by saving up to 18% relative to the b-dot. When the inclination orbit changed it clearly showed the disadvantage of b-dot for small inclination parameter in comparison with the gyro-feedback which was around 5% faster while consuming less 18% of energy. The lower output limit of the bang-bang is evident in cases 1,3 and 4 where it was slower to reach the threshold than the other two by around 500% in case 1 and 1000% in case 3 and 4. Although it took more time to reach the objective rate, the energy consumption for all the cases was better than other two controllers. It showed less than 20% energy consumption in case 1 and 4, in case 3 was on pair with the other two showing a consumption higher than the gyro-feedback but less than the b-dot and finally for case 4 the difference got to an extreme less, than 80%, due to the bad performance presented by the other two controllers explained earlier.

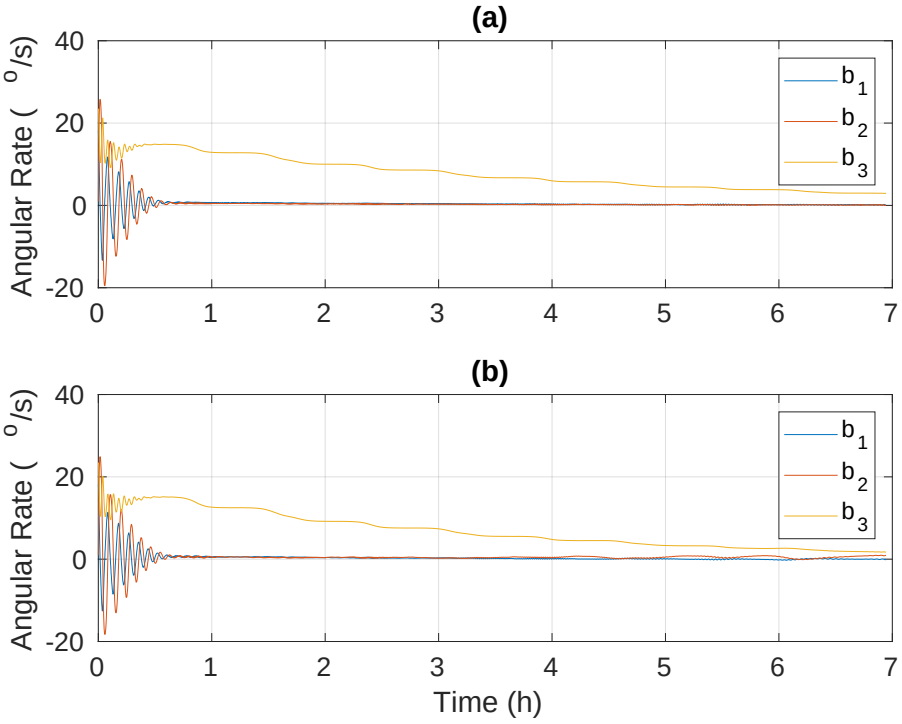

Figure 7.9: Angular rate in the three body axis for the second simulation case using the b-dot (a) and gyro feedback (b) algorithms.

7.3 Pointing results

The last part of this chapter will focus on the simulation results obtained with the PD algorithm which should allow the satellite to properly carry out the mission. Contrary to the attitude estimation the control simulations take much more time to complete because the actuator response has to be included into the simulation. This makes impossible to use the same data set for all the simulations as was done in the attitude estimation, for the detumbling controller this posed a small problem as the actuator response was sampled at 0.5 Hz and the the threshold angular rate was in most cases achieved in less than $1\,h$ of the simulated time. However for the pointing controller is much more important to evaluate the stability after achieving the desired values which extend the simulation time frame by a few orbit periods. Also for the controller to be effective a smaller $0.5\,s$ sampling time was imposed as the $2\,s$ used in detumbling would allow for much external torque to induce a considerate rotation and also avoids controller overshoot problems. The detumble controllers simulated reduce the angular rate until they reach their low limit of $0.8\,^o/s$ from this point on they can't reduce much more due to noise contributions and due to the small gain which produces a smaller actuation signal than the magnetorquers drivers can produce. Taking this into consideration the chosen pointing start condition chosen was $1\,^o/s$ to be near this minimum angular rate. The orbit initial conditions used in this test were already described in 7.1 and the remaining initial conditions for this test are described in the following table.

$\boldsymbol{\omega}_{\mathcal{B}}\ (^o/s)$	$\mathbf{q}$	k_ϵ	k_ω
$\begin{bmatrix} 0.76 \\ -0.64 \\ 0.12 \end{bmatrix}$	$\begin{bmatrix} 0.6073 \\ -0.3896 \\ 0.0589 \\ -0.6899 \end{bmatrix}$	-12760	-1500

Table 7.17: Initial angular rate, attitude and controller gains used in for the simulation.

In this test it was necessary to feed the position as reference to compute the desired orientation and it was assumed to be exempt of errors when in reality it will be fed by a propagator which introduces some errors. However the modulation of this error would need a position/orbit estimator study drifting from this thesis focus. Apart from this assumption the remaining errors will be included, as will occur in the real flight the satellite will use the results from the attitude estimator to compute the actuation signal based upon it's position relative to the Earth, the interference of the atmosphere, gravity and residual magnetic dipole were also included. The simulation was set to run for 7 complete orbit revolutions, equivalent to a total time of $10h30min$.

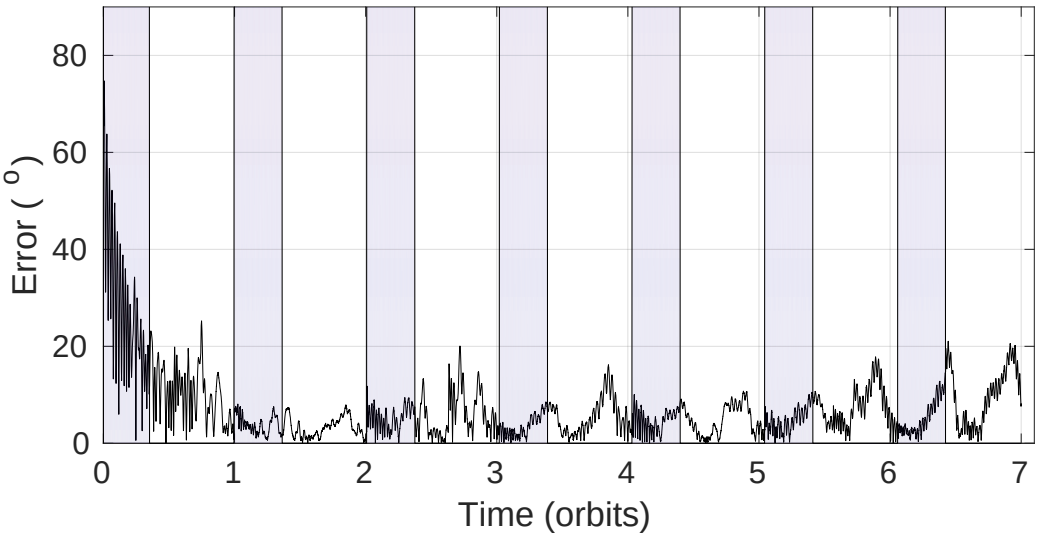

Figure 7.10: Pointing controller simulation angular error throughout 7 orbits (shaded regions indicate eclipse instances).

The estimator used in this test was the ECF configured as previously tested and the initial guess fed was forced to be to the corrected satellite attitude to avoid the period where the estimator has not yet converged into the correct attitude. It is possible to avoid this situation because the ADCS state transition dictates the satellite will always pass from a detumbling phase first before starting the pointing control

and it can be assumed that in detumbling the satellite can also be doing attitude estimation entering the pointing state with a near converged attitude estimation.

In this simulation the controller took around half orbit to get the z axis within 20^o of the NADIR align position and after the first orbit it spend $99.7\,\%$ of the time inside that error limit. The more difficult belt below the 10^o pointing error counts for $87.45\,\%$ after the satellite first orbit revolution and $54.30\,\%$ of the time within the 5^o limit.

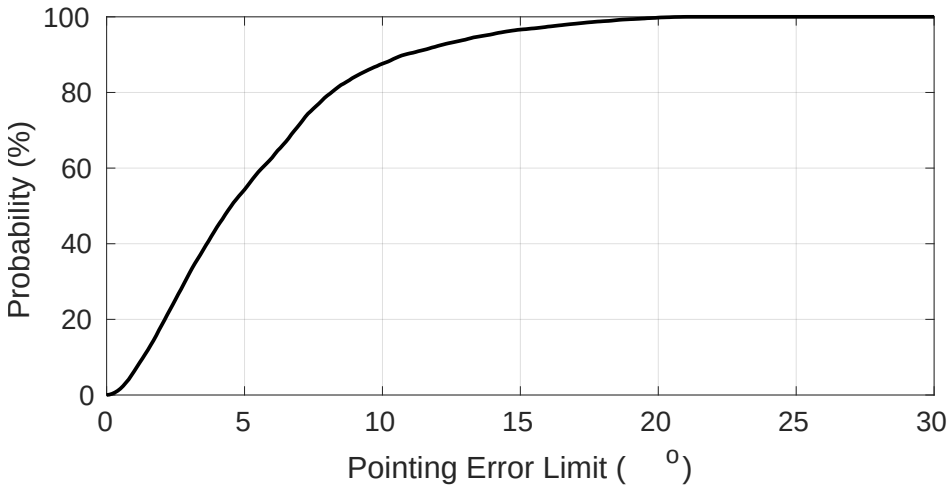

Figure 7.11: Cumulative probability function of the pointing controller error.

The last parameter to evaluate the pointing controller by is the electric energy consumption of the magnetorquers which should not surpass the power budget and limits imposed by the EPS. The budget counted for an average power consumption of $39\,mW$ from the ADCS magnetorquers. The maximum power the magnetorquers demand from this results is $233.7\,mW$ which is equivalent to the maximum power one magnetorquer alone is capable of consuming. More importantly the average power consumption after stabilization (around 1 orbit time) is $6.1\,mW$. and does increase when compared to overall average of $10.5\,mW$ during the 7 orbit period.

$\langle P \rangle$	max(P)	$\langle P \rangle_{2:7}$
$10.5\,mW$	$233.7\,mW$	$6.1\,mW$

Table 7.18: Power consumption statistics of the pointing controller for the 7 orbit simulation.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

The first conclusion we get from the results of this work is that the CSS configuration used provides good sun direction vectors when three photodiodes are lighted, if only two photodiodes are lighted the error increases greatly but is still usable. Lastly for only one lighted photodiode the probability of having a great error is too high to be usable. Due to this, the static attitude estimators QUEST, TRIAD, FOAM and SVD by themselves are not adequate for this satellite as they cannot ensure a stable attitude estimation which is necessary to perform the attitude control. However these algorithms when incorporated into the EQUEST method remain subjectively simple, needing low computational power and provides very good attitude estimations while the satellite is in the day time part of the orbit. The problem emerges when the solution propagates the attitude estimation without observations for very long periods of time, e.g. eclipse, due to the gyroscope bias the attitude propagation is drifting faster from the actual attitude. This disadvantage limits the use of EQUEST to a particular set of scenarios: if the gyroscope bias is very stable then through tests is possible to compensate for it or if it was imposed to only operate during the day time region of the orbit. The latter however would arise a few complications in the the control response time, even so neither EQUEST usability scenarios are compatible with the ISTsat-1 mission and hardware.

The solutions studied to solve the bias problem were the MEKF and the ECF, both provide a good attitude estimations throughout every point of the orbit. With MEKF the attitude results are better than EQUESTs while suffering the same problem, during the eclipse the error increased although not as much as EQUEST. The source of this problem was not in the bias as the MEKF can provide a very good bias estimation converging to the actual value in every situation. The algorithm allows fine tuning through the covariance matrices and due to this eclipse error increase it is possible to tune these matrices to achieve even better results for these regions. The biggest disadvantage of the MEKF when compared to the EQUEST or ECF is exactly related to these matrices, as even after the Murrell's formulation reducing the algorithm largest operations to 3 by 3 matrix operations it still uses more matrix operations in general than the other two options, specially to compute the Kalman gain (see equation 5.69). For a bigger and

more powerful satellite would be preferable to use this algorithm as the tuning possibilities and space legacy make it the most secure option. However for the low power OBC inside the ISTsat-1 the high number of matrix operations can pose a problem and even if it was capable of computing an attitude estimation with this algorithm there is a chance it could not do it at the desired sampling time.
The results from ECF are much more stable than any of the other tested algorithms and provides a comfortable error for the attitude controller to work with. Although the gyroscope bias estimation is not as good as MEKF, its estimations are always close to the real value. This is the most adequate algorithm for the ISTsat-1 because the computational complexity is very low while the attitude estimation is good enough to ensure NADIR pointing with less than $150\,km$ error.
From the detumbling results two different performance groups can be defined according to their performance: the gyro and b-dot form the first both quick at reducing the satellite angular rate, the second is the Bang-Bang B-dot which by definition is limited and as such has a slower response. From these two groups the slower response is the more efficient and it spends energy at a slower pace allowing the EPS to replenish its battery and keep it charged for most of the time. Concerning safety and robustness during the detumbling the bang-bang variation of the b-dot would be the advisable controller.
Lastly, the pointing controller achieves an accuracy in pointing at NADIR below 10^{o} for 87% of the simulated time translating into less than $70\,km$ pointing error. These results already account for the error ECF introduces in estimating the satellite attitude, therefore the combo estimator plus controller in spite of their errors and disturbance torques are able to meet the the mission requirements. A good safety margin was also introduced by maintaining the control actuation time at $0.5\,s$ meaning it was chosen to use a slow sampling time even though the satellite may be able to use a quicker control sampling time to ensure all other OBC tasks are run in real-time. Another important note is that after the start of the controller the satellite will take less than half an orbit to be within 20^{o} of NADIR depending on the initial attitude. The slow control suits the ISTsat-1 discrete actuation as the low proportional gain also results in less overshoot near the NADIR reference axis and offers a more tight control over the angular rate avoiding excessive rotation problems.

8.2 Future Work

From the hardware perspective the ISTsat-1 could make use of a more detailed magnetorquer , to characterize their hysteresis and compensate the inductive impedance effects in actuation inaccuracy if needed. Still in sensor topic the CSS model could also be more detailed, if it were to include an Earth albedo model. This could also allow to enhance the CSS measurements if the S/C used the same model for computing the sun direction vector.
From the operational perspective a few more tests are necessary, the most important is to test the OBC capability of computing a set of solutions in real-time without interfering with the other tasks.
Lastly there are some performance tests to be ran like using a Helmholtz chamber to check if the detumbling time is within the simulated values. Another more complex test is necessary in order to validate the simulated performance, to do this an air-bearing should be used to enable three rotational degrees

of freedom, as well as a Helmholtz chamber and a light bulb. This will allow to generate and control all the ADCS reference vectors while simulating a situation closer to the orbit conditions.

Bibliography

[1] J. R. Wertz. *Spacecraft Attitude determination and control*. Kluwer Academic Publishers, 1978.

[2] G. Xu and Y. Xu. GPS: Theory, algorithms and applications, third edition. In *GPS: Theory, Algorithms and Applications, Third Edition*, pages 1–489. 2016. ISBN 9783662503676. doi: 10.1007/978-3-662-50367-6.

[3] J. Georgy, A. Noureldin, and T. B. Karamat. *Fundamentals of inertial navigation, satellite-based positioning and their integration*, volume 136. 2013. ISBN 9783642304651.

[4] J. Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors. Technical report, 2006. URL `ftp://sbai2009.ene.unb.br/Projects/GPS-IMU/George/arquivos/Bibliografia/79.pdf`.

[5] F. L. Markley and J. L. Crassidis. *Fundamentals of Spacecraft Attitude Determination and Control*. Springer-Verlag New York Inc., 2014. ISBN 978-1-4939-0801-1. doi: 10.1007/978-1-4939-0802-8.

[6] N. Dantam. Quaternion Computation, 2014.

[7] J. Solà. Quaternion kinematics for the error-state Kalman filter. 2017. URL `http://arxiv.org/abs/1711.02508`.

[8] O. Montenbruck and E. Gill. *Satellite Orbits*. 2000. ISBN 978-3-540-67280-7. doi: 10.1007/978-3-642-58351-3.

[9] V. L. Pisacane. *The Space Environment And Its Effects On Space Systems*. American Institute of Aeronautics and Astronautics, 1st edition, 2008. ISBN 9781563479267.

[10] J. Davis. Mathematical Modeling of Earth ' s Magnetic Field. Technical report, 2004.

[11] M. R. Haneveer. *Orbital Lifetime Predictions - An assessment of model-based ballistic coefficient estimations and adjustment for temporal drag coefficient variations*. PhD thesis, TU Delft, 2017.

[12] C. M. Reynerson. Aerodynamic disturbance force and torque for spacecraft and simple shapes using finite plate elements - Part II: Aerodynamic force & torque. *IEEE Aerospace Conference Proceedings*, pages 1–9, 2012. ISSN 1095323X. doi: 10.1109/AERO.2012.6187257.

[13] W. E. Wiesel. *Spaceflight Dynamics*. Irwin McGraw-Hill, 2nd edition, 1989.

[14] S. Lee, A. Hutputanasin, A. Toorian, W. Lan, R. Munakata, J. Carnahan, D. Pignatelli, and A. Mehrparvar. CubeSat Design Specification. Technical report, California Polythenchnic State University, 2014.

[15] F. Beer, E. R. J. Jr., and E. Eisenberg. *Vector Mechanics for Engineers: Statics*. McGraw-Hill, 11th edition, 2015. ISBN 978-0073224473.

[16] R. F. Afonso. Sistema de Gestão e Determinação de Atitude do Engenharia Electrónica Júri. page 74, 2016.

[17] S. Merhav. *Aerospace Sensor Systems and Applications*, volume 9. 1996. ISBN 9780387946054. doi: 10.1088/0957-0233/9/1/023. URL http://books.google.com/books?id=GSyNye99dz0C.

[18] C. C. M. N. Armenise. *Advances in Gyroscope Technology*. 2010. ISBN 9783642154935. doi: 10.1007/978-3-642-15494-2.

[19] InvenSense. MPU9250 Product Specification Revision 1.1. Technical Report 408, 2016.

[20] Q. Lam, N. Stamatakos, C. Woodruff, and S. Ashton. Gyro Modeling and Estimation of Its Random Noise Sources. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, (June), 2003. doi: 10.2514/6.2003-5562. URL http://arc.aiaa.org/doi/10.2514/6.2003-5562.

[21] Osi Optoelectronics. Photodiode Characteristics. *Osi Optoelectronics*, page 6, 2009. URL https://www.google.com/?ion=1{&}espv=2{#}q=osioptoelectronicsphotodiodecharacteristics.

[22] EnduroSat. 1U Solar Panel – Specification. Technical report.

[23] Y. Winetraub, S. Bitan, U. Dd, and A. Heller. Attitude Determination – Advanced Sun Sensors for Pico-satellites. *Tel-Aviv University*, pages 1–10.

[24] S. Theil, P. Appel, and A. Schleicher. Low Cost , Good Accuracy - Attitude Determination Using Magnetometer and Simple Sun Sensor. *Annual AIAA/USU Conference on Small Satellites*, 17, 2003.

[25] Honeywell. 3-Axis Digital Compass IC HMC5883L. Technical report, 2010.

[26] H. D. Black. A passive system for determining the attitude of a satellite. *AIAA Journal*, 2(7):1350–1351.

[27] F. Granziera Jr., R. V. F. Lopes, and M. C. Tosin. The attitude determination problem from two reference vectors- a description of the triad algorithm and its attitude covariance matrix. *Ciências Exatas e Tecnológicas*, 28(1):21–35, 2007.

[28] F. Markley and D. Mortari. Quaternion attitude estimation using vector observations. *Journal of the Astronautical Sciences*, 48(2):359–380, 2000. URL http://www.researchgate.net/publication/257944313{_}JASpaper/file/72e7e5266b18652266.pdf.

[29] M. D. Shuster and S. D. Oh. Three-axis attitude determination from vector observations. *Journal of Guidance, Control, and Dynamics*, 1981. ISSN 0731-5090. doi: 10.2514/3.19717.

[30] F. L. Markley. Attitude Determination using Vector Observations and the Singular Value Decomposition. *The Journal of the Astronautical Sciences*, 36(3):245–258, 1988. URL `http://www.control.auc.dk/{~}tb/best/aug23-Bak-svdalg.pdf`.

[31] F. L. Markley. Attitude determination using vector observations - A fast optimal matrix algorithm. *The Journal of the Astronautical Sciences*, 41(2):261–280, 1993. ISSN 00219142.

[32] J. L. Crassidis, S. F. Andrews, F. L. Markley, and K. Ha. Contingency designs for attitude determination of TRMM. Technical report, NASA, 1995.

[33] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[34] D. Simon. *Optimal State Estimation: Kalman, H∞, and Nonlinear Approaches*. 2006. ISBN 0471708585. doi: 10.1002/0470045345.

[35] J. L. Crassidis, F. L. Markley, and Y. Cheng. Survey of Nonlinear Attitude Estimation Methods. *Journal of Guidance, Control, and Dynamics*, 30(1):12–28, 2007. ISSN 0731-5090. doi: 10.2514/1.22452. URL `http://arc.aiaa.org/doi/10.2514/1.22452`.

[36] N. Trawny and S. I. Roumeliotis. Indirect Kalman filter for 3D Attitude Estimation. *... , Dept. of Comp. Sci. & Eng., ...*, (2005-002):1–25, 2005. doi: 10.2514/6.2005-6052. URL `http://www-users.cs.umn.edu/{~}trawny/Publications/Quaternions{_}3D.pdf`.

[37] J. W. Murrell. Precision attitude determination for multimission spacecraft. 02 1978. doi: 10.2514/6.1978-1248.

[38] P. S. Maybeck. *Stochastic Models, Estimation, and Control*, volume 1. Academic Press, 1979. ISBN 0124807011.

[39] R. Mahony, T. Hamel, and J. M. Pflimlin. Complementary filter design on the special orthogonal group SO(3). *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, CDC-ECC '05*, 2005(1):1477–1484, 2005. ISSN 00189286. doi: 10.1109/CDC.2005.1582367.

[40] R. Mahony, T. Hamel, P. Morin, and E. Malis. Nonlinear complementary filters on the special linear group. *International Journal of Control*, 85(10):1557–1573, 2008. ISSN 00207179. doi: 10.1080/00207179.2012.693951.

[41] Z. Tudor. Design and Implementation of Attitude Control for 3-axes Magnetic Coil Stabilization of a Spacecraft. (May), 2011.

[42] P. C. Hughes. *Spacecraft Attitude Dynamics and Control*. DOVER PUBLICATIONS, INC., Mineola, New York, 1st edition, 2004. ISBN 0471818429. doi: 10.1017/CBO9781107415324.004.

[43] G. Avanzini and F. Giulietti. Magnetic detumbling of a rigid spacecraft. *Journal of Guidance, Control, and Dynamics*, 35:1326–1334, 07 2012. doi: 10.2514/1.53074.

[44] B. Wie and P. M. Barba. Quaternion feedback for spacecraft large angle maneuvers. *Journal of Guidance, Control, and Dynamics*, 8(3):360–365. ISSN 0731-5090. doi: 10.2514/3.19988.

[45] ISS Trajectory Data. `https://spaceflight.nasa.gov/realdata/sightings/SSapplications/Post/JavaSSOP/orbit/ISS/SVPOST.html`. Accessed: 2019-03-13, 14h08.

[46] NASA. Spacecraft Magnetic Torques. Technical report, NASA, 1969.

Appendix A

Math Properties

A.1 Vector Operators

Let v be a random $\mathbb{R}^3$ vector the matrix $[v]_\times$ is defined as the cross product matrix form with another $\mathbb{R}^3$ random vector u.

$$v \times u = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \cdot u = [v]_\times \cdot u \tag{A.1}$$

$$= \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \cdot v = [u]_\times^T \cdot v \tag{A.2}$$

A.2 Quaternion Operations

The quaternion product is generally identified with the symbol $\otimes$ and can be expressed as two different matrix notations:

$$\mathbf{q} \otimes \mathbf{p} = \begin{bmatrix} q_\eta \cdot p_\eta - \mathbf{q}_\epsilon \cdot \mathbf{p}_\epsilon \\ q_\eta \cdot \mathbf{p}_\epsilon + p_\eta \cdot \mathbf{q}_\epsilon - \mathbf{q}_\epsilon \times \mathbf{p}_\epsilon \end{bmatrix} \tag{A.3}$$

$$= \begin{bmatrix} q_\eta & -\mathbf{q}_\epsilon^T \\ \mathbf{q}_\epsilon & q_\eta \mathbf{I} + [\mathbf{q}_\epsilon]_\times \end{bmatrix} \cdot \mathbf{p} = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{bmatrix} \cdot \mathbf{p} \tag{A.4}$$

$$= \begin{bmatrix} p_\eta & -\mathbf{p}_\epsilon^T \\ \mathbf{p}_\epsilon & p_\eta \mathbf{I} + [\mathbf{p}_\epsilon]_\times \end{bmatrix} \cdot \mathbf{q} = \begin{bmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & -p_3 & p_2 \\ p_2 & p_3 & p_0 & -p_1 \\ p_3 & -p_2 & p_1 & p_0 \end{bmatrix} \cdot \mathbf{q} \tag{A.5}$$

The quaternion $\mathbf{q}$ has its conjugate represented as $\mathbf{q}^*$ and have a symmetric imaginary axis vector.

$$\mathbf{q}^* = \begin{bmatrix} \eta \\ -\boldsymbol{\epsilon} \end{bmatrix} \tag{A.6}$$

This means the quaternion product of a quaternion by its conjugate it's equal to the identity quaternion which is a property of the inverse making the conjugate by definition equal to the inverse and a useful method to compute the inverse.

$$\mathbf{q} \otimes \mathbf{q}^* = \mathbf{q}^* \otimes \mathbf{q} = \mathbf{q} \otimes \mathbf{q}^{-1} = \mathbf{q}^{-1} \otimes \mathbf{q} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{A.7}$$

A.3 Quaternion Derivative

From the definition the quaternion derivative is written as the limit of the instant difference divided by the time difference.

$$\dot{\mathbf{q}} = \lim_{t \to t_0} \frac{\mathbf{q}_t - \mathbf{q}_{t_0}}{t - t_0} \tag{A.8}$$

But for small angles the derivative of the quaternion can be represented by a rotation from $\mathcal{I}$ to $\mathcal{B}$ multiplicating the quaternion by the angular rate pure quaternion measured in $\mathcal{B}$. So that the derivative

can be written as a function of pure quaternion Ω.

$$\dot{\mathbf{q}} \approx \frac{1}{2}\Omega(\boldsymbol{\omega})\,\mathbf{q} \tag{A.9}$$

The pure quaternion is the vector transformed into quaternion and due to the angular rate special use the matrix form of its pure quaternion is represented with Ω.

$$\Omega(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\boldsymbol{\omega}^T \\ \boldsymbol{\omega} & -[\boldsymbol{\omega}]_\times \end{bmatrix} \tag{A.10}$$

A.4 Quaternion Discrete Integration

The attitude quaternion can be propagated using it's derivative in a direct integration method as explicited with equation A.11 although this is only possible in a time-continuous platform like Simulink. For the onboard MCU the algorithm will run in discrete time line so the problem will have to be adapted for discrete time in order to correctly represent the real behavior of the system. The solution to the differential equation A.9 is adapted from continuous time to discrete:

$$\mathbf{q}(t) = e^{\left(\frac{\Delta t}{2}\,\Omega\right)}\,\mathbf{q}_0 \tag{A.11}$$

$$\mathbf{q}_{k+1} = \Phi^q_{k+1}\cdot\mathbf{q}_k \tag{A.12}$$

Where the discrete version of the transition matrix $^{\left(\frac{\Delta t}{2}\,\Omega\right)}$ from instant k to the next instant $k+1$ is represented by the matrix Φ. From here the Taylor series expansion can be used to avoid compute the matrix exponential.

$$\Phi^q_{k+1} \approx \mathbf{I}_{4\times4} + \left(\frac{\Delta t}{2}\right)\cdot\Omega(\boldsymbol{\omega}) + \frac{1}{2!}\left(\frac{\Delta t}{2}\cdot\Omega(\boldsymbol{\omega})\right)^2 + \frac{1}{3!}\left(\frac{\Delta t}{2}\cdot\Omega(\boldsymbol{\omega})\right)^3 + \ldots \tag{A.13}$$

If the angular rate wouldn't change between each k^{th} steps, a derivative of the quaternion would be sufficient to describe quaternion state transition matrix Φ using a zeroth order quaternion integrator equation:

$$\Phi^q_{k+1} = = I_{4\times4} + \left(\frac{\Delta t}{2}\right)\cdot\Omega(\boldsymbol{\omega}_{k+1}) \tag{A.14}$$

But if the angular rate is approximated to an average between each step a first order quaternion integrator could then be described following the results demonstrated in [1].

$$\langle\boldsymbol{\omega}_{k+1}\rangle = \frac{\boldsymbol{\omega}_{k+1} + \boldsymbol{\omega}_k}{\Delta t} \tag{A.15}$$

After using the averaged angular rate into the transition and the Taylor series expansion the following equation to compute Φ emerges:

$$\Phi_{k+1}^{q} \quad = \quad \mathbf{I}_{4\times 4} \; + \; \left(\frac{\Delta t}{2}\right) \cdot \Omega\left(\boldsymbol{\omega}_{k+1}\right) \; + \; \left(\frac{\Delta t^2}{48}\right) \cdot \left[\Omega\left(\boldsymbol{\omega}_{k+1}\right)\Omega\left(\boldsymbol{\omega}_{k}\right) \; - \; \Omega\left(\boldsymbol{\omega}_{k}\right)\Omega\left(\boldsymbol{\omega}_{k+1}\right)\right] \tag{A.16}$$

Appendix B

Algorithm Summary

In summary the filter algorithms can be described by the steps presented in the following tables.

B.1 Enhanced QUEST

Propagate	$\tilde{q}_k = \Phi_k^q(\tilde{\omega}_k)\,\hat{q}_{k-1}$
Gain	$\beta = \left(1 \;-\; |r_1 \cdot r_2|^2\right)\beta_0$ $\beta = 0$ (*when in eclipse*)
Update	$\hat{q}_k = (1 \;-\; \beta)\cdot\tilde{q}_k \;+\; \beta \cdot q_{k_{QUEST}}$

Table B.1: EQUEST steps summary

B.2 MEKF Sensor Input

Initialize	$\hat{x}_k^- = x_0$
	$P_k^- = P_0$
Gain	$H_k = \begin{bmatrix} \left(A\left(\hat{q}_k^-\right) r_1\right)_\times & 0_{3\times3} \\ \left(A\left(\hat{q}_k^-\right) r_2\right)_\times & 0_{3\times3} \end{bmatrix}$
	$S_k = H_k\, P_k^-\, H_k^T + R$
	$K_k = P_k^-\, H_k\, S_k^{-1}$
Update	$P_k^+ = [I - K_k H_k]\, P_k^-\, [I - K_k H_k]^T$
	$z_k = \begin{bmatrix} \left[\tilde{b_1}\right]_{3\times1} \\ \left[\tilde{b_2}\right]_{3\times1} \end{bmatrix} - \begin{bmatrix} A\left(\hat{q}_k^-\right) \cdot [r_1]_{3\times1} \\ A\left(\hat{q}_k^-\right) \cdot [r_2]_{3\times1} \end{bmatrix}$
	$\hat{x}_k^+ = \hat{x}_k^- + K_k z_k$
	$\hat{x}_k^+ = \begin{bmatrix} \delta\hat{\vartheta}_k^{+T} & \hat{\beta}_k^{+T} \end{bmatrix}^T$
	$\hat{q}_k^* = \delta q_k(\delta\hat{\vartheta}_k^+) \otimes \hat{q}_k^-$
	$\hat{q}_k^+ = \frac{\hat{q}_k^*}{\|\|\hat{q}_k^*\|\|}$
Propagation	$\hat{\omega}_k = \omega_k - \hat{\beta}_k$
	$\hat{q}_{k+1} = \Phi_{k+1}^q(\hat{\omega}_k)\, \hat{q}_k^+$
	$P_{k+1}^- = \Phi_{\mathbf{x}}\, P_k^-\, \Phi_{\mathbf{x}}^T + \mathrm{G}_k\, Q\, \mathrm{G}_k^T$

Table B.2: MEKF summary using sensors input

B.3 Explicit Complementary Filter

Observation	γ	$= \sum k_i \, (\mathbf{b}_i \times A(\hat{q}) \, \mathbf{r}_i)$
Update	$\dot{\hat{\mathbf{q}}}$	$= \Omega \left(\hat{\boldsymbol{\omega}}_{\mathcal{B}} - \boldsymbol{\beta}_g + k_p \, \gamma \right) \otimes \hat{\mathbf{q}}$
	$\dot{\hat{\boldsymbol{\beta}}}_g$	$= - k_g \, \gamma$
Propagation	$\hat{\omega}_k$	$= \omega_k - \hat{\beta}_k$
	$\hat{q}_{k+1}$	$= \Phi^q_{k+1}(\hat{\omega}_k) \, \hat{q}^+_k$

Table B.3: ECF summary

Appendix C

Simulink Model

After building the simulink model it was necessary to check if the it's errors were acceptable and to do so a set of intial conditions were selected whose results can be compared to a reference. The set of initial conditions necessary to test the translation were the intial position, velocity and date of the satellite in the $\mathcal{I}$ frame as well as the satellite mass. For the body rotation movement the intial conditions needed to set were the attitude relative to $\mathcal{I}$ and angular rate in the $\mathcal{B}$ relative to $\mathcal{I}$.

Condition	**Value**
Date	8^{th} Jun of 2017
$x_{\mathcal{I}}\,(m)$	-4.2706×10^6
$y_{\mathcal{I}}\,(m)$	-5.2571×10^6
$z_{\mathcal{I}}\,(m)$	5.6095×10^3
$\dot{x}_{\mathcal{I}}\,(m/s)$	3.7068×10^3
$\dot{y}_{\mathcal{I}}\,(m/s)$	-2.9891×10^3
$\dot{z}_{\mathcal{I}}\,(m/s)$	6.0162×10^3
$\boldsymbol{\omega}_{\mathcal{B}}\ (^o/s)$	$[1\,1\,1]^T$
$\mathbf{q}$	$[1\,0\,0\,0]^T$
S/C_{mass}	1.0
$S/C_{inertia}$	$1.6667 \times 10^{-5} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
$\partial t\ (s)$	0.1

Table C.1: Inital conditions used for model validation simulation

C.1 Position

The moviments of translation and rotation can be considedered as independent of each other and as such they were decouple to test each separatly. The default configuration of the solver simulink uses

is a variable step which was manually limited to 1 second and run for 10 orbits which is equivalent to almost $15\,h$. The simulation results were compared to NASA tool GMAT (General Mission Analysis Tool) simulation of the same orbit.

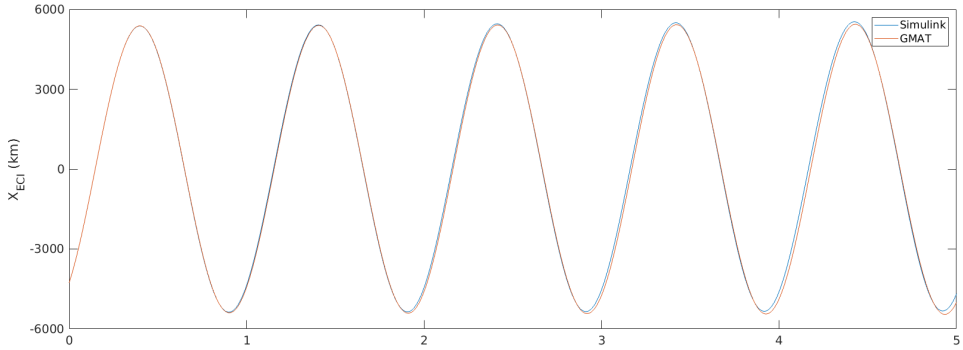

Figure C.1: X coordenate obtained with GMAT and simulink model throuhout 5 orbits

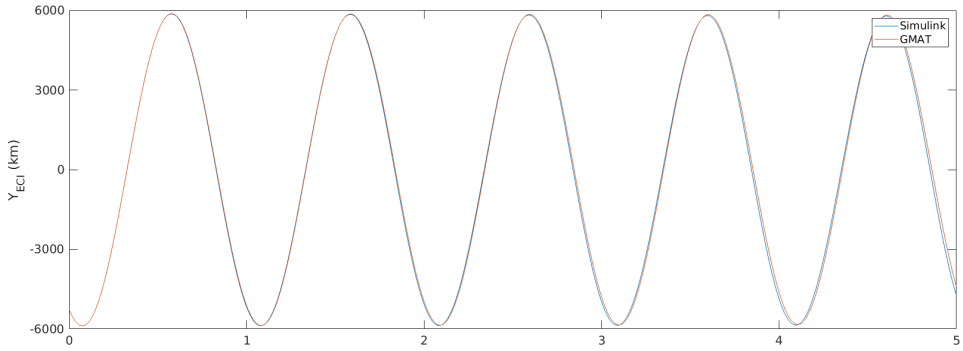

Figure C.2: Y coordenate obtained with GMAT and simulink model throuhout 5 orbits

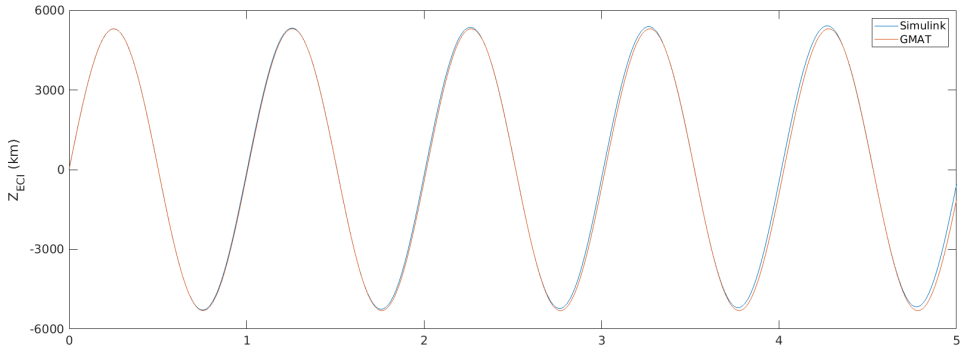

Figure C.3: Z coordenate obtained with GMAT and simulink model throuhout 5 orbits

The error rises constantly through time reveling a correspondig steady error in the position propagator although near the end of the fifth orbit it the coordinates starts to deviate more rapidly from the GMAT reference. From the following table it is possible to check that most of the time the error under $300\,km$.

Orbits	5	10
$\bar{E}_x\,(km)$	98.7	317.9
$\bar{E}_y\,(km)$	109.8	311.2
$\bar{E}_z\,(km)$	105.8	321.4

Table C.2: Simulator positional error compared to GMAT reference

C.2 Atitude

For the simulator attitude validation the approach was used instead of comparing to GMAT, a known case was setup up for the simulator to work on. The setup of the simulink solver was maintain as setted previoulsy but the the computation of the attitude is expected to be more demanding a smaller step may be expected. Another inital assumption was made regarding the inertia matrix of the tensor if the body is assumed symmetric in all axis then the inertia matrix is diagonal which consequently will nullify one of the dynamic equation (eq. **??**) components.

$$(\boldsymbol{\omega} \times I\boldsymbol{\omega}) \;=\; 0 \quad \leftarrow I_{11} \;=\; I_{22} \;=\; I_{33} \tag{C.1}$$

The first test will assed the integration errors in the attitude computation and will consist in setting an intial angular rate and compare its evolution over a period of time against the expected results. The simpliest test was to impose a $10\,^o/s$ rate in one axis and read the results of a $36\,s$ simulation to check if in the final position was equal to the intial and the middle points are coherent. To recreate this kind of movement the body must not be subject to any torques leading to temporarily turn off the external and control torques. The second test was similar but imposing a $1.0\,^o/s$ rate in all axes simultaneously. The expect result according to the rotation angle representation is a revolution with a $207.8461\,s$ period.

$$|\boldsymbol{\omega}| \quad = \quad \sqrt{3} \quad \rightarrow \quad T \quad = \quad \frac{360}{|\boldsymbol{\omega}|} \quad = \quad 207.8461\,s \tag{C.2}$$

Using the the identity attitude $[1\,0\,0\,0]^T$ as the inital condition is possible to see final attitude value is actually symmetric to the initial value. Remebering the quaternion rules, symmetric quaternions represent the same rotation, the results are what was expected and do return to the initial value after the $207.8461\,s$.

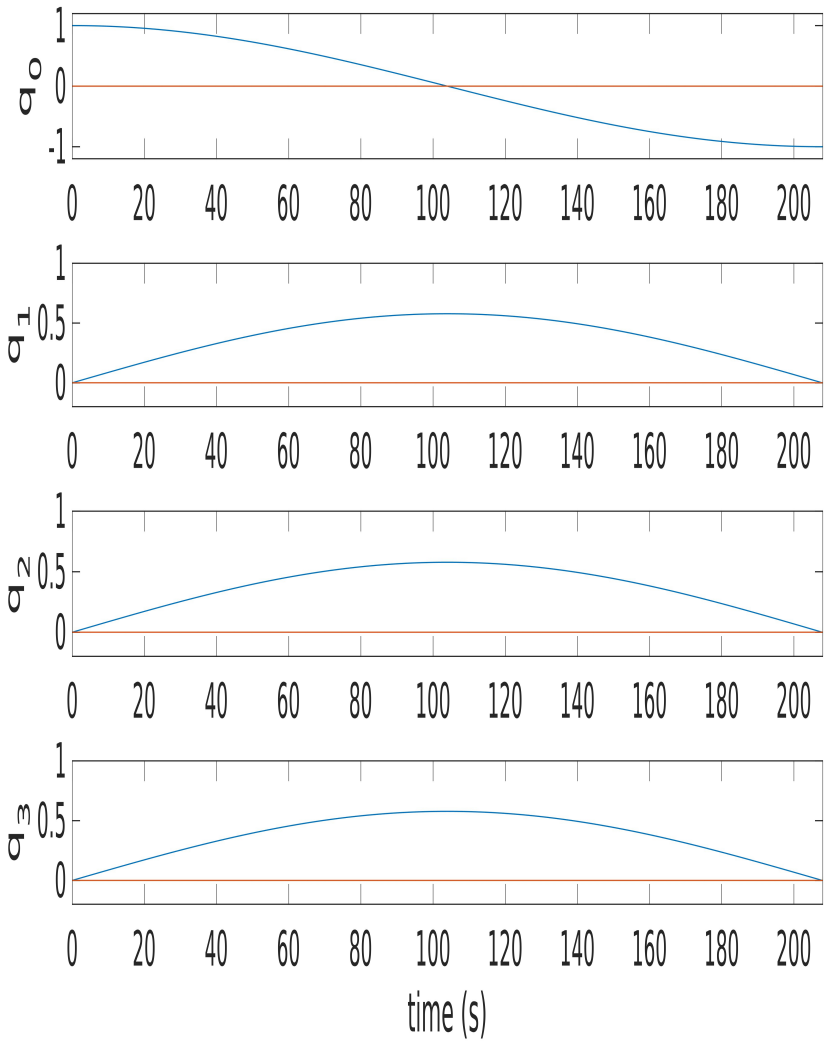

Figure C.4: Attitude simulation validation test results

Lastlty, a third test was done using another known result. This time, with applying a sinusoidal external torque with the folling form:

$$\tau(t) = A \cdot \cos(2\pi f t) \tag{C.3}$$

Where A is the torque amplitude and f the torque sinusoidal frequency. Using the dynamic model established in chapter 3 the angular derivative is established as:

$$\dot{\omega} = I^{-1} \left(\tau_{actuadores}\right) \tag{C.4}$$

Consequently the resulting angular rate velocity is represented by the integration:eguinte forma:

$$\omega(t) = \int \tau \, dt = I^{-1} \cdot \frac{A}{2\pi f} \cdot \sin(2\pi f t) + a_1 \tag{C.5}$$

The simulation will impose a zero angular rate in the satellite consequently the initial condition set by the coefficient a_1 will also be zero. To simplify the angular displacement validation the torque was applied along the first principal axis of the satellite. Using the previous equation the rotation equation can be reduced to:

$$\theta_i(t) = -\frac{I_{i,i}^{-1} \cdot A}{(2\pi f)^2} \cdot \cos(2\pi f t) + b_1 \tag{C.6}$$

The expected behaviour is a sinusoidal function with the an amplitude coefficient equal to set by the torque amplitude, frequency and inertia along the first principal axis. The equation condition is setted by the coefficient b_1 which shall be setted to zero by equalling it to the sinusoidal amplitude component.

$$0 = -\frac{I_{i,i}^{-1} \cdot A}{(2\pi f)^2} + b_1 \rightarrow b_1 = \frac{I_{i,i}^{-1} \cdot A}{(2\pi f)^2} \tag{C.7}$$

Initially, we can assume the test will simulate 720 seconds therefore to have 2 cycles it needs a frequency of $f \geq \frac{1}{360} = 0.0028\,\text{Hz}$. To avoid more than half revolution to both sides, the amplitude must be limited to $A = 9.5698 \times 10^{-6}$. Lastly, it results in the following equation:

$$\tau = \begin{bmatrix} A\cos(2\pi f t) \\ 0 \\ 0 \end{bmatrix} \tag{C.8}$$

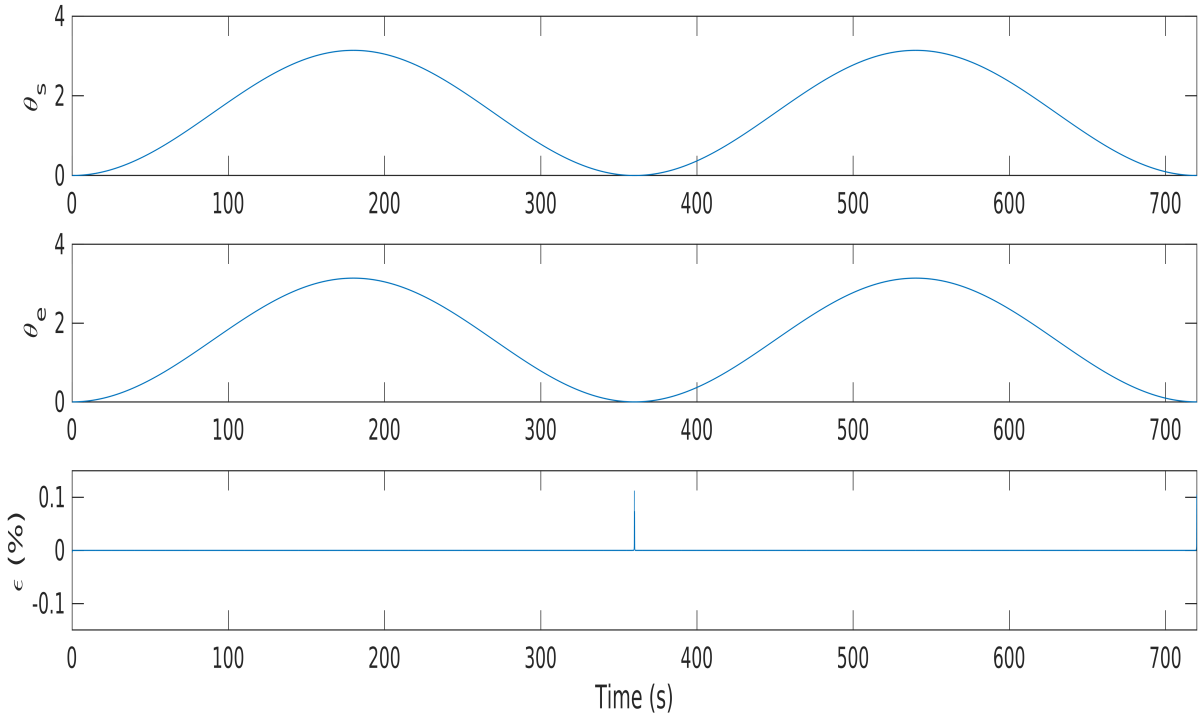

Figure C.5: Angular displacement simulated, θ_s, predicted θ_e and relative error ϵ through 720 seconds

The difference from the simulator to expected results have a maximum relative error of 0.12% after 720 seconds. However this maximum represents a spike as most 99% of the time the error is below 0.001%. These results validated the simulator dynamic model for external torques applied in one axis.

Appendix D

Simulation Plots

D.1 Sensor Angular Error Plots

In this section the coarse sun sensor error cumulative probability is represented for the three situations possible.

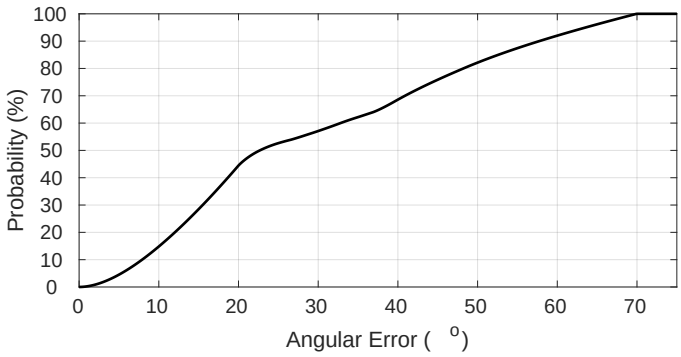

Figure D.1: Cumulative probability function of the coarse sun sensor when only 1 photodiode is lighted

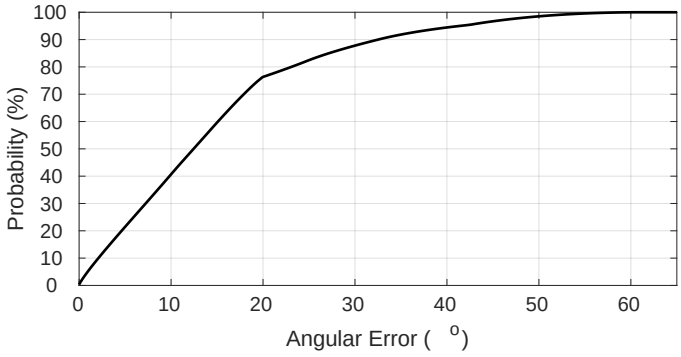

Figure D.2: Cumulative probability function of the coarse sun sensor when 2 photodiode is lighted

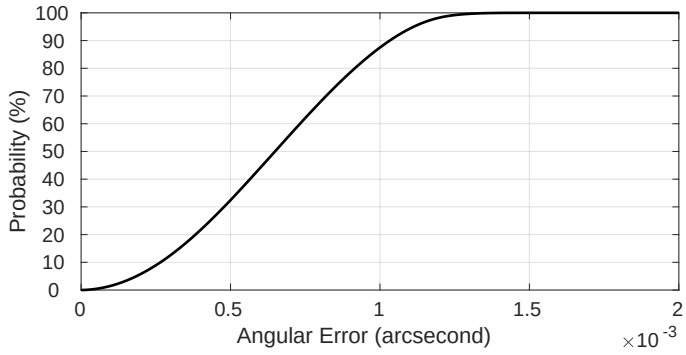

Figure D.3: Cumulative probability function of the coarse sun sensor when 3 photodiode is lighted

D.2 Attitude Determination Plots

In the present section the detail results of the attitude determination are presented through a set of graphic data.

D.2.1 TRIAD

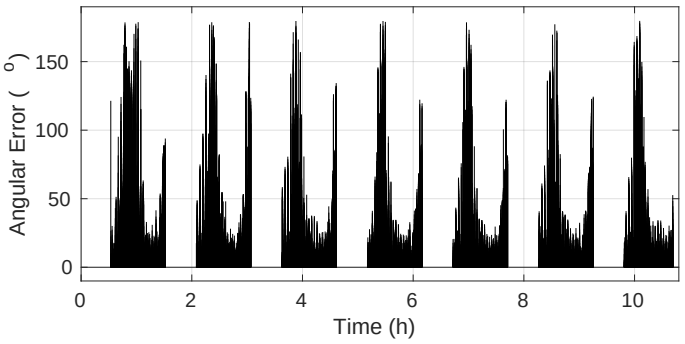

Figure D.4: TRIAD attitude error for the first case during a seven orbit simulation.

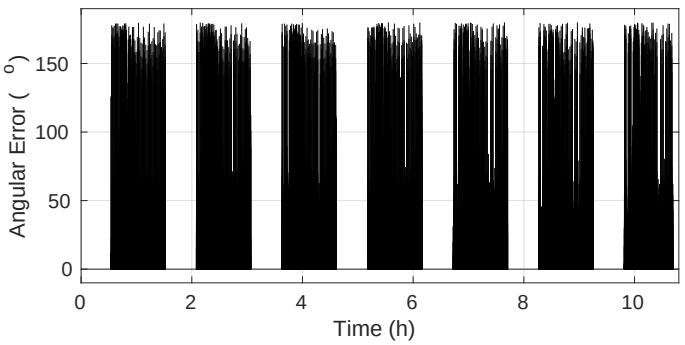

Figure D.5: TRIAD attitude error for the second case during a seven orbit simulation.

D.2.2 QUEST

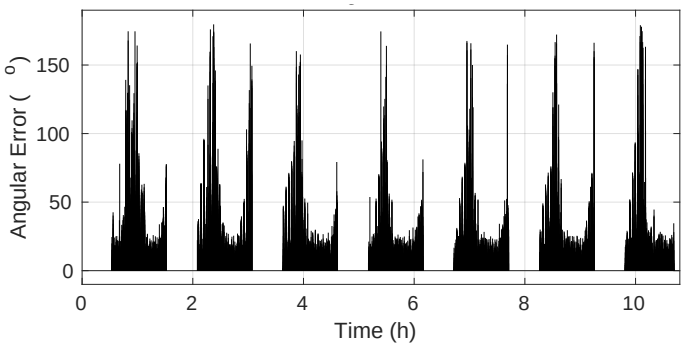

Figure D.6: QUEST attitude error for the first case during a seven orbit simulation.

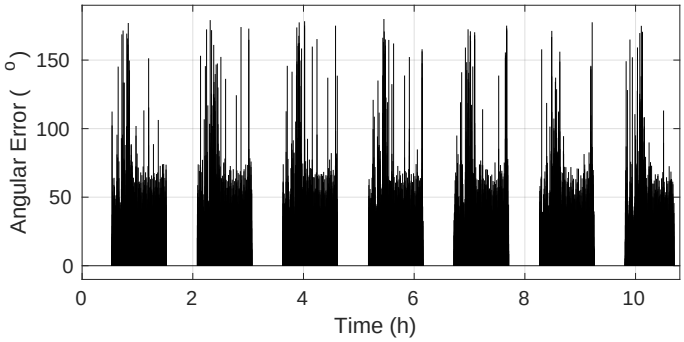

Figure D.7: QUEST attitude error for the second case during a seven orbit simulation.

D.2.3 SVD

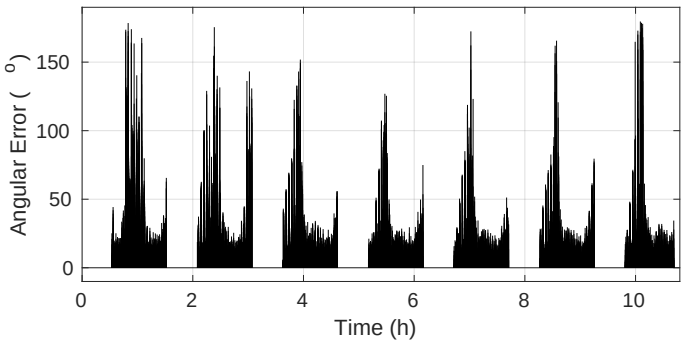

Figure D.8: SVD attitude error for the first case during a seven orbit simulation.

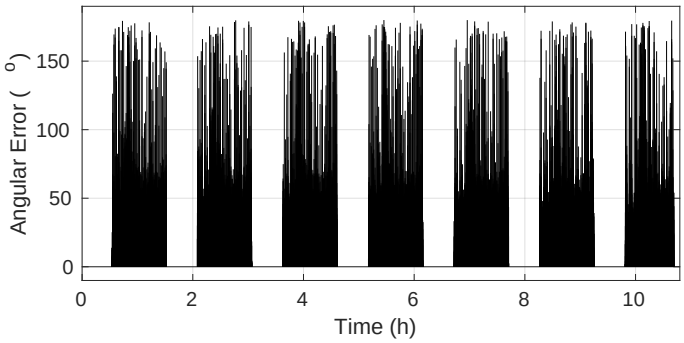

Figure D.9: SVD attitude error for the second case during a seven orbit simulation.

D.2.4 FOAM

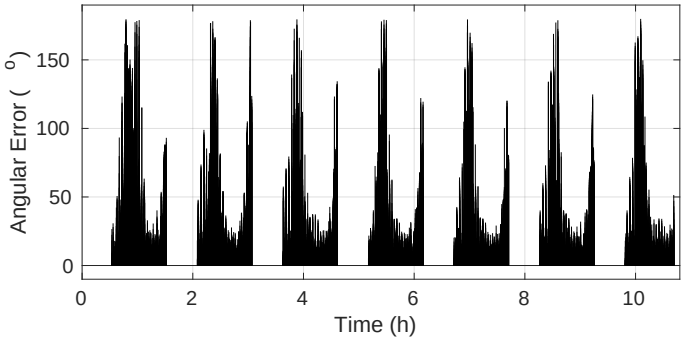

Figure D.10: FOAM attitude error for the first case during a seven orbit simulation.

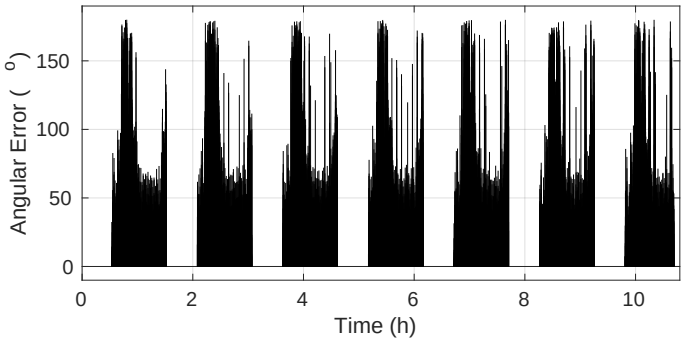

Figure D.11: FOAM attitude error for the second case during a seven orbit simulation.

D.2.5 EQUEST

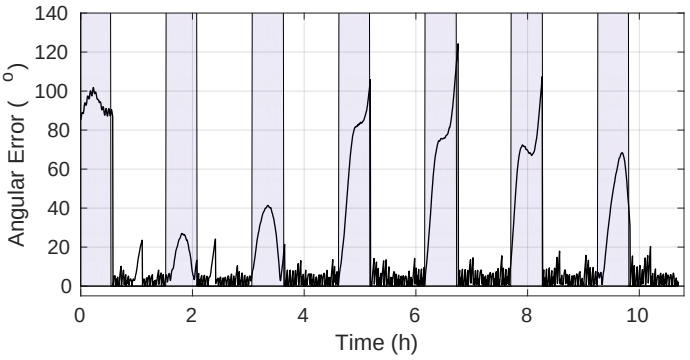

Figure D.12: EQUEST attitude error for the first case during a seven orbit simulation.

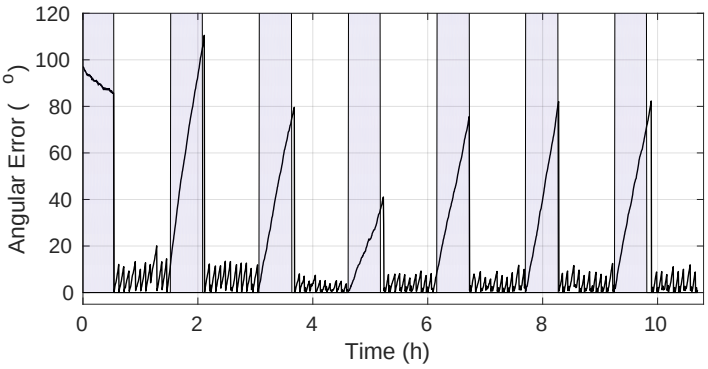

Figure D.13: EQUEST attitude error for the second case during a seven orbit simulation.

D.2.6 MEKF

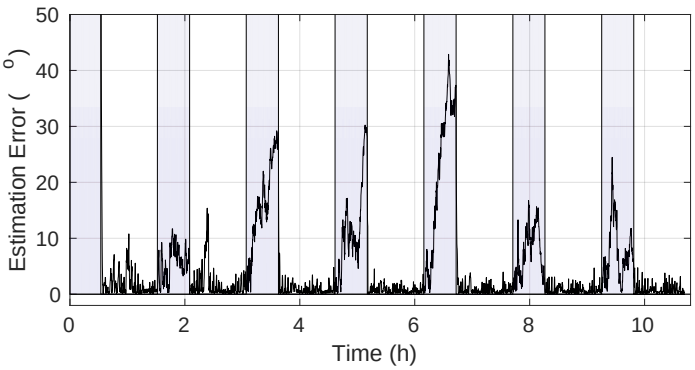

Figure D.14: MEKF attitude error for the first case during a seven orbit simulation.

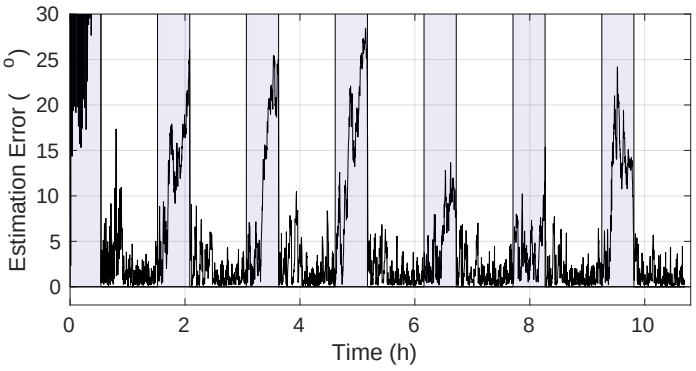

Figure D.15: MEKF attitude error for the second case during a seven orbit simulation.

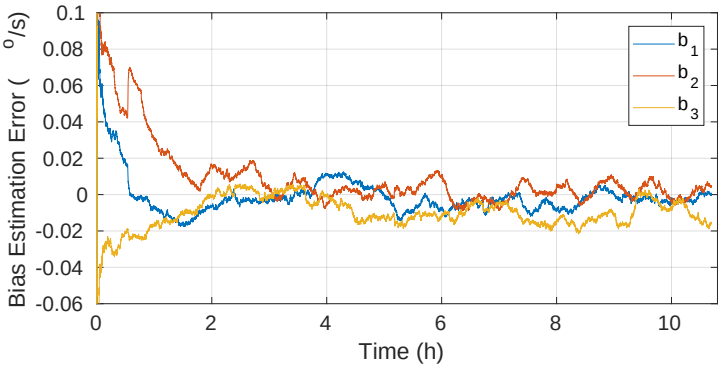

Figure D.16: MEKF gyro bias estimation error for the first case during a seven orbit simulation.

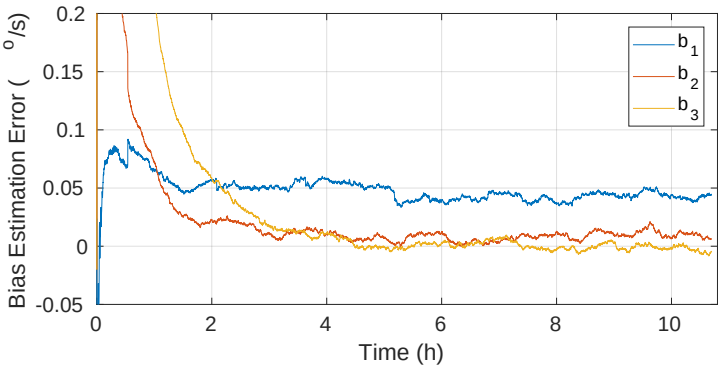

Figure D.17: MEKF gyro bias estimation error for the second case during a seven orbit simulation.

D.2.7 ECF

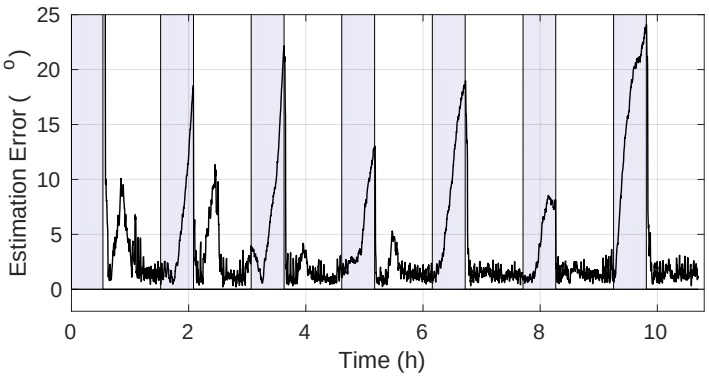

Figure D.18: ECF attitude error for the first case during a seven orbit simulation.

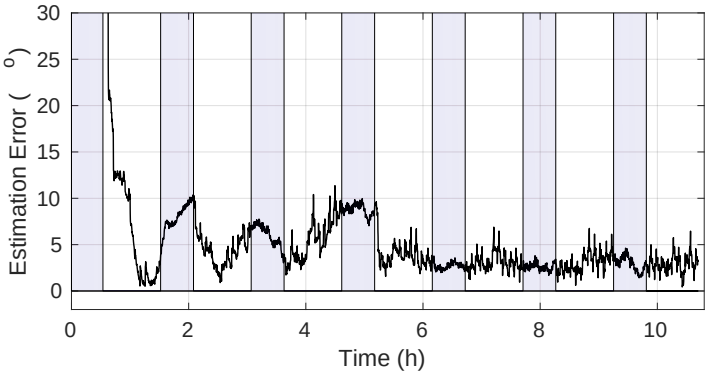

Figure D.19: ECF attitude error for the second case during a seven orbit simulation.

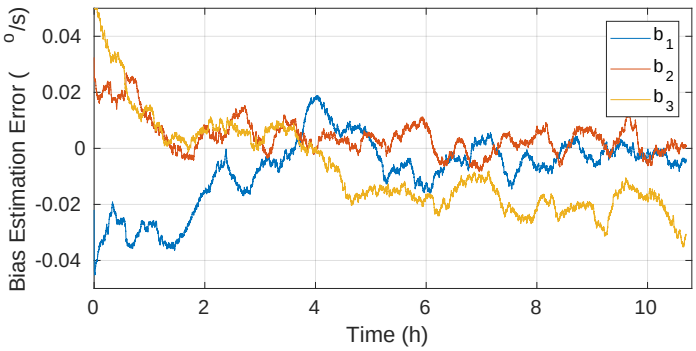

Figure D.20: ECF gyro bias estimation error for the first case during a seven orbit simulation.

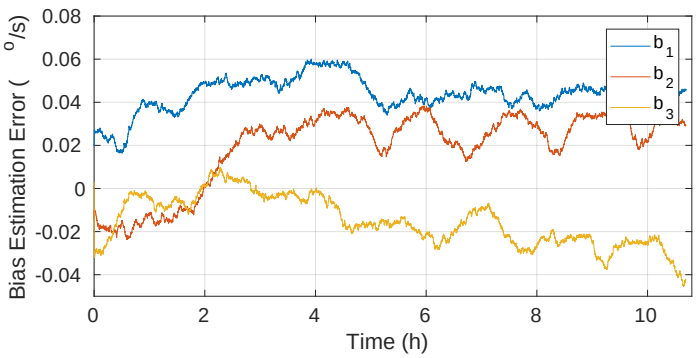

Figure D.21: ECF gyro bias estimation error for the second case during a seven orbit simulation.

D.3 Detumbling Angular Rate Plots

D.3.1 B-dot

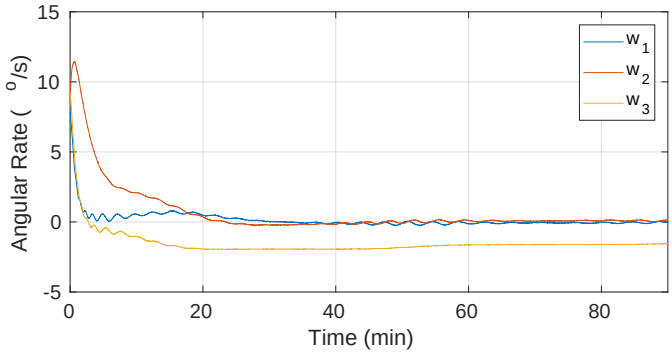

Figure D.22: B-dot Case 1 Angular Rate.

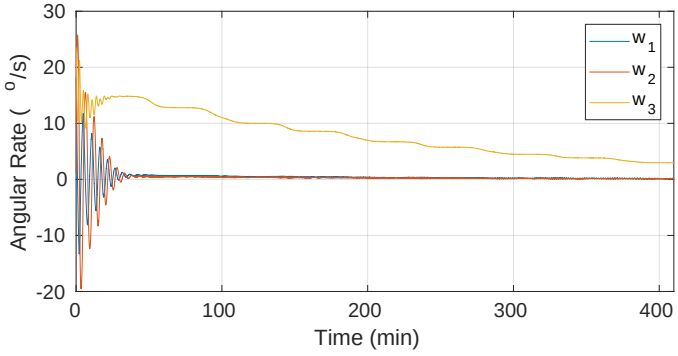

Figure D.23: B-dot Case 2 Angular Rate.

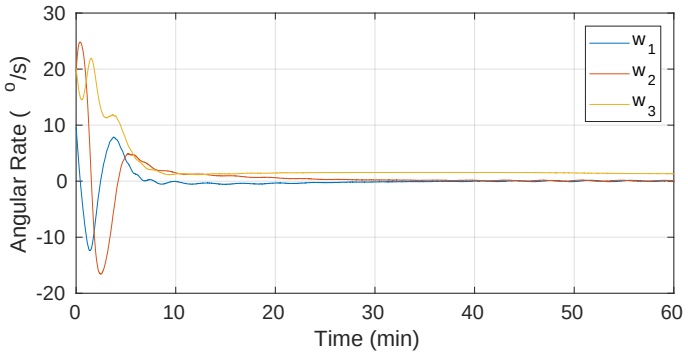

Figure D.24: B-dot Case 3 Angular Rate.

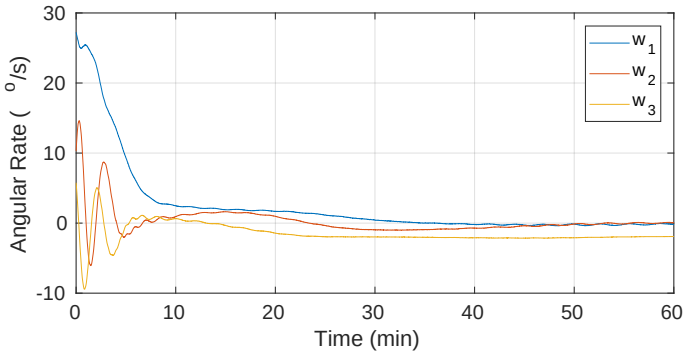

Figure D.25: B-dot Case 4 Angular Rate.

D.3.2 Bang Bang B-dot

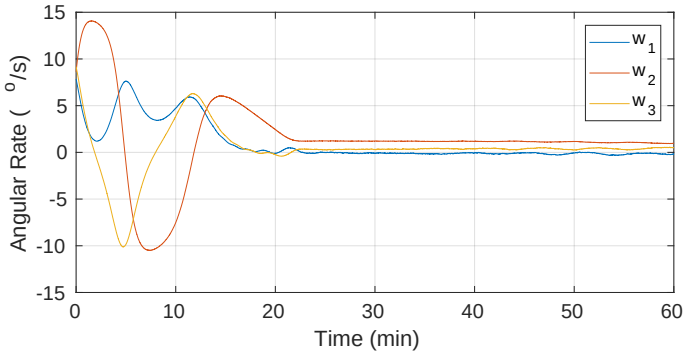

Figure D.26: Angular rate in each body axis using the Bang-Bang B-dot in the simulation of case 1.

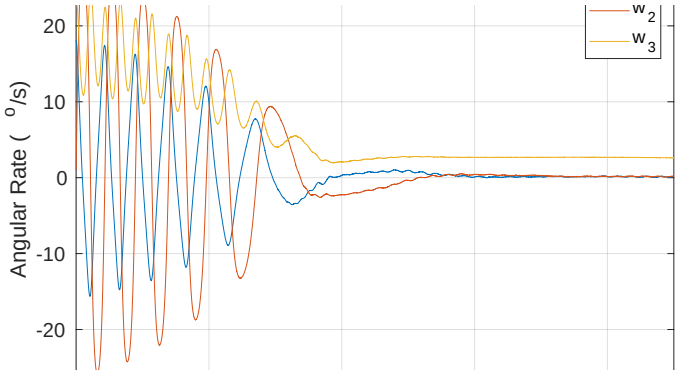

Figure D.27: Angular rate in each body axis using the Bang-Bang B-dot in the simulation of case 2.

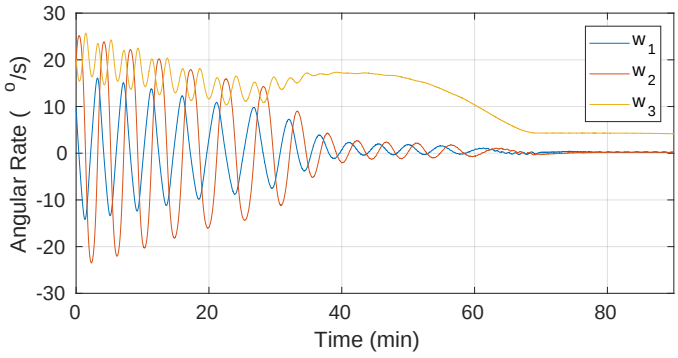

Figure D.28: Angular rate in each body axis using the Bang-Bang B-dot in the simulation of case 3.

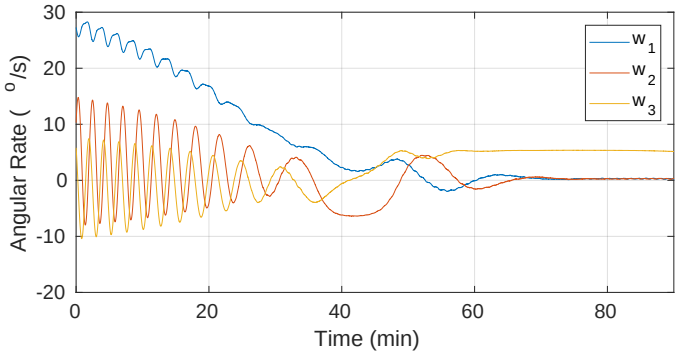

Figure D.29: Angular rate in each body axis using the Bang-Bang B-dot in the simulation of case 4.

D.3.3 Gyro Feedback

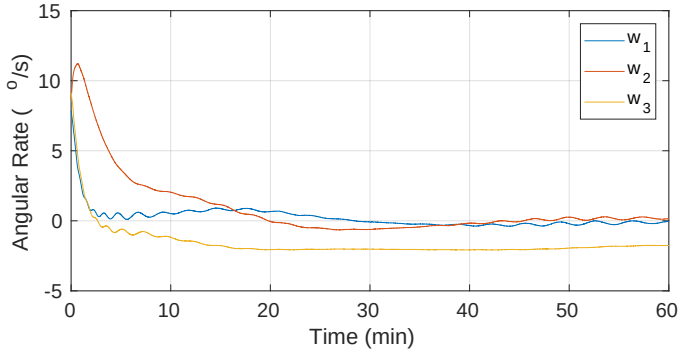

Figure D.30: Angular rate in each body axis using the Gyro feedback for the simulation of case 1.

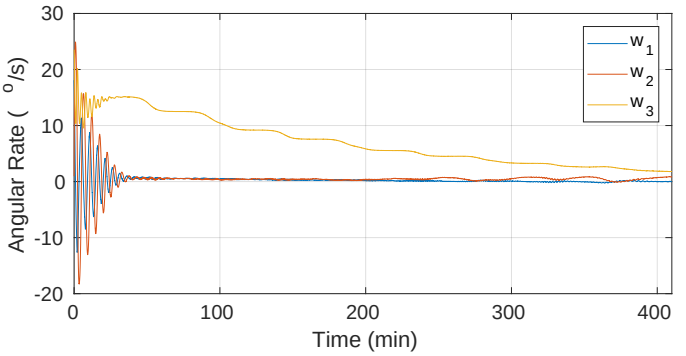

Figure D.31: Angular rate in each body axis using the Gyro feedback for the simulation of case 2.

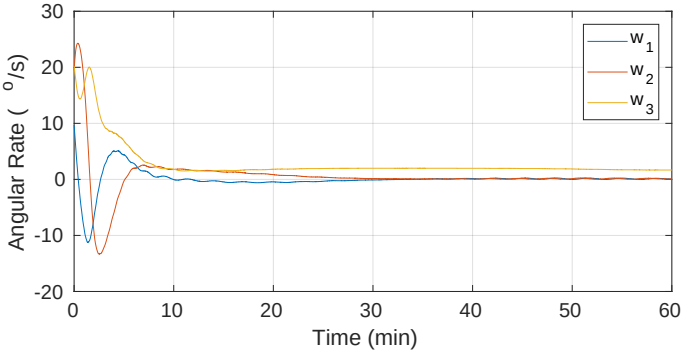

Figure D.32: Angular rate in each body axis using the Gyro feedback for the simulation of case 3.

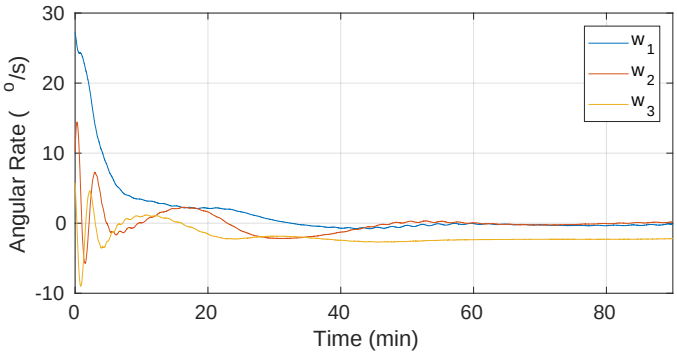

Figure D.33: Angular rate in each body axis using the Gyro feedback for the simulation of case 4.