TÉCNICO LISBOA

Design and development of the ECOSat's Attitude Determination and Control System (ADCS) onboard software

João Miguel Vitorino Castro Lousada

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Examination Comittee

Chairperson:	Prof. Doutor Fernando José Parracho Lau (DEM)
Supervisor:	Prof. Doutor Afzal Suleman (DEM)
Members of the Committee:	Prof. Doutora Maria Alexandra Aguiar Gomes (DEM)

July 2013

Dedicated to my family for the unconditional support...

Acknowledgments

The author would like to thank Dr. Afzal Suleman in particular, who provided this great opportunity that was performing a MSc. thesis at the University of Victoria and supervised the whole development of this work.

Special thanks also goes to the members of the ECOSat team, that provided a great working environment and excellent resources: Bernard Lambrechts, Kris Dolberg and Justin Saukarookoff, who helped the whole ADCS system to come together; Robin Novlesky, for the great knowledge and advises; Nigel Syrotuck and Justin Curran who coordinated everyone very efficiently and made the whole project succeed.

Acknowledgement also goes Dr. Nikolas Trawny from the NASA Jet Propulsion Laboratory and Dr. Yoshi Hashida from Surrey Satellite Technology Ltd., whose insightful advices helped the development of this thesis very greatly. Their knowledge is vast and the author appreciate the insights and tips they provided.

Resumo

O principal objectivo desta tese foi o desenvolvimento e teste de um *software* para o sistema de determinação e control de attitude (ADCS) do projecto ECOSat.
Inicialmente, decidiu-se diferenciar entre os sistemas de determinação de atitude (ADS) e controlo de atitude (ACS). O ADS foi desenvolvido primeiro uma vez que a sua informação é necessária para o desenvolvimento e funcionamento do ACS.
Apesar de, no desenho de veículos espacias, o sistema de navegação e orientação (GNS) ser, por vezes, independente do ADCS, foi decidido acoplar os dois sistemas. A opção, no desenho de missão, de não efectar correcções orbitais, reduziu os requesitos do GNS a simplesmente determinar e monitorizar os parametros orbitais do satélite. Para isso, são utilizadas medições de GPS, pesadas e comparadas com a previsão dos parâmetros orbitais dada por um algoritmo desenvolvido para o efeito.
A determinação da atitude do satélite envolve um algoritmo mais complexo que tem em conta medições de instrumentos com elevados níveis de ruído, como magnetometros e dados dos paneís solares (usados como sensor de Sol analógico). Um algoritmo com filtro de Kalman foi desenvolvido para este efeito. Este efectua uma previsão da atitude futura do satélite através de modelo da dinâmica de quaterniões e operações destes, aos quais está associada uma certa incerteza. O filtro de Kalman estima esta incerteza em todos os momentos, de modo a compará-la com as medições que vão sendo tomadas. O algoritmo de determinação de atitude representa a parte principal do *software*, uma vez que o algoritmo de controlo está altamente dependente de uma determinação de atitude adequada a cada momento.
As operações do ACS começam com o *de-tumbling* do satélite. Um algoritmo foi desenvolvido para lidar com esta parte da missão do satélite, que realiza todas as correcções necessárias levar o satélite de um estado imprevisível de *tumbling*, induzido após o lançamento, até um estado controlado. Este algoritmo tem que ser bastante independente de medições, uma vez que, nesta fase da missão, os dados do GPS e, consequentemente, do sensor de Sol, não estão disponíveis.
O resto do ACS actua durante a missão do satélite e fornece os ajustes de atitude necessários para o cumprimento da missão. Esta foi definida, nos requesitos internos da equipa, como uma missão de *payload* constantemente apontado para a Terra e uma rotação em torno do seu eixo de aproximadamente $0.06°/s$, ou seja, uma revolução por orbita para a altitude da missão de $800km$.
O *software* desenvolvido foi então testado através de uma serie de simulações de modo a averiguar a sua performance. Foram testadas as capacidades de determinção e de controlo de attitude, terminando com um teste onde as condições *worst case scenario* de missão foram simuladas. Verificou-se, para este caso, um tempo de convergência total de cerca de $2.3 \times 10^4 s$ (correspondendo a cerca de 3 períodos orbitais) até atingir a attitude nominal.

Palavras-chave: nanosatélite, ADCS, software, magnetometro, magnetorquer, filtro de Kalman.

Abstract

The main focus of this thesis is the development and testing of a fully functional Attitude Determination and Control System (ADCS) Software for the ECOSat project.

Initially it was decided to differentiate between the two subsystems of Attitude Determination System (ADS) and Attitude Control System (ACS). The ADS was developed firstly as its information was required for the development and operation the ACS.

Although in spacecraft design the Guidance and Navigation System (GNS) is, sometimes, independent from the ADCS, it was decided to have it as a part of the ADCS. ECOSat's mission design option of not performing orbital corrections reduced the GNS's requirements to solely the determination and monitoring of the satellite's orbital parameters. For this, GPS measurements are taken into account and weighted against the predicted orbital parameters, given by an orbit propagation algorithm developed for this purpose.

The determination of the satellite's orientation involves a more complex algorithm that takes into account measurements from highly noisy instruments, such as the magnetometers and the data from the solar arrays (used as an analog sun sensor). A Kalman filter algorithm was developed for this. It performs a prediction of the satellite's future orientation through a dynamic model and a variety of quaternion operations, that have a certain inaccuracy. The Kalman filter estimates this inaccuracy at all times, in order to weight it against recurrent measures being taken. This orientation determination algorithm represents the core of the whole software as the control algorithm is highly dependent on having accurate data of the satellite's current orientation.

The ACS operation starts with the de-tumbling of the satellite. A specific algorithm was developed for this part of the satellite's mission, which performs all the necessary attitude corrections in order to bring the satellite from the unpredictable tumbling rates induced after launch to the required attitude. It needs to be very "measurement-independent" as, in this phase of the mission, GPS and, consequently, correct sun sensor data will be unavailable.

The rest of the ACS acts throughout the satellite's mission time and provides the necessary attitude adjustments that allow it to perform its mission. This is defined, from internal team requirements, to be a constant nadir pointing attitude (i.e., with the payload constantly pointing towards the Earth) and a angular rate of approximately $0.06°/s$ about its longitudinal axis, giving a full rotation per orbit at the mission's defined altitude of $800km$.

Once completely developed, the complete ADCS algorithm was put through a series of tests and simulations to access its performance. The algorithm was firstly tested on its attitude determination and control capabilities. Finally, both attitude and control were tested, in a simulation that mimicked worst case scenario conditions of the real mission. The results showed that the ADCS system is capable of recovering from the tumbling situation within the first $1.1 \times 10^4 s$, (approximately 3 hours) and converges to the proper attitude within $2.3 \times 10^4 s$ (a little over 6 hours) from the mission's beginning.

Keywords: nanosatellite, ADCS, software, magnetometer, magnetorquer, Kalman filter.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

1.1.1 The Canadian Satellite Design Competition

When Geocentrix first proposed the Canadian Satellite Design Challenge (CSDC) back in September 2010, it called out to universities across Canada, challenging students to design and manufacture an operational nanosatellite in what would be the first competition of this type in the world [1].
During the CSDC teams are challenged to create their own innovative satellite design, including a mission definition and subsystem design and manufacturing, not only giving students a better understanding of the practical aspects of a real satellite designing process but also possibly leading to significant advances in related technologies. Once constructed all satellites undergo a complete space environmental and launch qualification testing, leading up to the actual launch of the winner's satellite. [1]

1.1.2 The ECOSat Project

To answer this competition, the University of Victoria created the ECOSat team. The team is made up of 20 to 30 students from different backgrounds and it handles not only all the technical aspects of the satellite design, but also tries to be involved in the community by actively working with local schools and small organizations [2].
Though small in size, the nanosatellite has to have all the critical functionality of a larger satellite like the power, payload, attitude determination and control, communication, and operations. It is an extensive but very rewarding challenge.
The ECOSat Team defined as part of mission design, to a have a Sun synchronous circular orbit with 800 km and an orbital period of 6042 s (100 min). The requirements for the Attitude Determination and Control Subsystem (ADCS) called for an Earth pointing attitude and a rotation about the satellite's longitudinal axis for optimum energy absorption through the solar arrays.
The author of this paper had the good fortune of being presented with an opportunity to lead the design-

ing of this ADCS which he gladly accepted. This MSc. Thesis consists of the work developed during the author's time with the ECOSat team in developing the complete ADCS Software.

1.2 Thesis Overview

1.2.1 State of the Art

It cannot be said that the idea of performing the attitude control of a satellite by means of a magnetic based set of torquers is an innovative one. As a matter of fact, concepts for a magnetic control system have long been proposed by Wang [3] and Wisni [4], amongst others. However, the relatively small torques provided, the dependence on the highly variable intensity of the Earth's magnetic field and the complexity of the control problem that arises from it, have kept these systems from a largely spread usage in current satellite designs.

The Kalman filter dates even further back and was originally developed as a tool for linear estimation [5], [6], [7] and soon developed into nonlinear applications in the Apollo program [8], [9]. Later, Farrell, [10], [11], studied the possibility of using Kalman filtering for obtaining an estimation of a satellite's attitude. In this early work, Ferrell considered a torque free motion and made use of Euler angles. In [12], Cherry and O'Connor incorporated disturbance torques caused by the propulsion system into their attitude estimation. Also Potter and Vander Velde [13], performed research in applying Kalman filtering to attitude determination.

However, it was only after 1969 that research started to show relevant results and interesting applications [14],[15],[16],[17]. By 1971, quaternions started to be considered for attitude estimation in general [20] and later for kalman filter attitude equation by Lefferts *et al.* [18]. Since then, a large number of applications of the simple nonlinear extended Kalman filter (EKF) have been presented (e.g. [19],[21]).

Recently a large number of variations of the EKF have been proposed. These include the multiplicative extended Kalman filter (MEKF) [22], the additive EKF [23],[24], the backwards-smoothing EKF [25] or the deterministic EKF-like estimators [26],[27]. The work developed by Crassidis *et al.* [28] provides an extensive overview of current proposed methods of nonlinear attitude estimation, including EKF and its variations.

1.2.2 Thesis Motivation

Although their usage is still far from other systems of attitude control, magnetorques provide some very intersting features. These attitude actuators composed of 3 orthogonal (or solenoids are more energy efficient, lightweight and simple the majority of other systems of attitude control. They also have a higher lifetime (since they consume no propellant) and absence of moving parts, which greatly improves their reliability [29]. These characteristics make such a system a very promising candidate for performing the attitude control of a nanosatellite with no extremely high pointing accuracy requirements.

Since these systems have a simple design and high reliability, they can be easily developed in a labo-

ratory environment with a few specific materials. This is particularly useful for a student driven project such as the ECOSat as it greatly reduces the overall costs of the system.
On the software side, the lower computation requirements, higher speed of the resulting applications and easiness to be integrated with hardware [30] indicated *C* as the most suitable programming language for the development of the proposed algorithm.

1.3 Coordinate Systems

There are a multitude of coordinate systems used in satellite attitude determination and orbital mechanics and it is fundamental to understand these different coordinate system and how they relate to each other. In this section, the most used coordinate systems will be explained. The transformation of coordinates between these systems is detailed in the work of Cai *et. al* [31].

Earth-Centered Inertial (ECI) Coordinate System

The Earth-Centered Inertial frame has its origin in the center of the Earth, so it moves it in its orbit around the Sun, but does not rotate with the Earth. It is a *pseudo inertial* frame, meaning, it is not completely inertial as it is accelerating in its orbit about the Sun around the Galaxy's center, but can be considered inertial for most applications within the vicinity of the Earth with an almost insignificant discrepancy. [32]
The z axis of the ECI frame is constantly point towards the North Pole, while the x axis points towards the Vernal Equinox. The y axis is defined as to complete an orthogonal coordinate system.

Earth-Centered Earth Fixed (ECEF) Coordinate System

Very similar to the ECI coordinate system, the ECEF shares the origin and z axis of the ECI frame. However it has the particularity of rotating with the Earth about its axis, in such a way that its x and y axis (contained in the equatorial plane) will always intersect that same point on the Earth's surface. More specifically, the x axis points towards the intersection of the equator and the Greenwich Meridian (or Prime Meridian), while the y completes the orthogonal coordinate system. This rotation makes the coordinate system not inertial.
To perform a transformation from the ECEF frame to the ECI it is only necessary to perform a rotation about the z axis given by the difference between the Greenwich Meridian and the Vernal Equinox at the time of the transformation.
This frame is crucial for the unfolding of this MSc. Thesis as it constitutes a intermediate step from any of the following coordinate system to the inertial ECI.

Geographic Coordinate System

This coordinate system, also called Latitude, Longitude and Altitude (LLA) coordinate system, is fully defined by two angles (latitude and longitude) and a vertical distance from the Earth's surface (geodetic

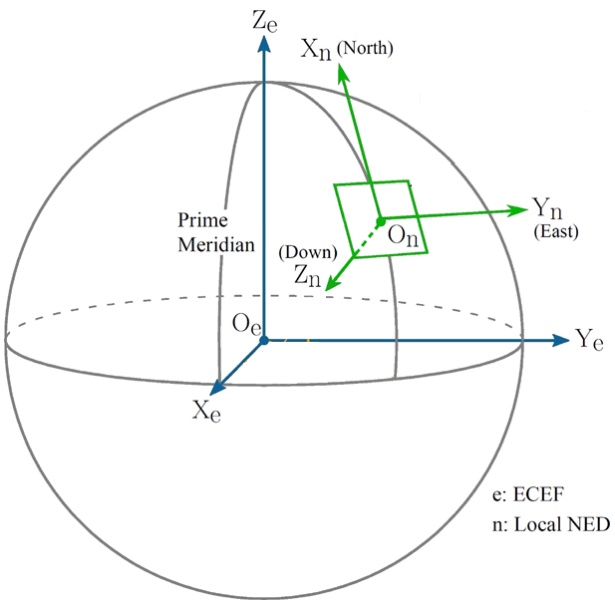

Figure 1.1: ECEF (in blue) and LNED (in green) coordinate systems. [31]

height or altitude).

It is particularly important, as GPS measurements will be given within it, having later to be transformed to the ECI frame in order to determine the orbit's parameters (which can only be done in an inertial frame). If we first convert the vertical altitude into distance to the Earth's center the conversion to the ECEF frame becomes simple. Care is needed, though, as this transformation is time dependent.

Body-fixed Coordinate System

The Body-fixed frame is also a non inertial frame, meaning that Newton's laws are not valid within it. However it is a very useful coordinate system since most attitude determination instruments will give results in it, making it the starting point of any attitude determination system.

It has the origin in the center of mass of the satellite and it's axis are aligned with the satellite's principal moments of inertia frame. It is body fixed, i.e., it rotates with the satellite, sharing its angular rates and orientation.

Local North East Down (LNED) Coordinate System

The LNED frame has the same origin has the Body-fixed frame, that is, the center of mass of the satellite. However its axis are defined differently. The z always points towards the center of the Earth, while the x and y axis point towards the local North and East, respectively, thus completing the orthogonal coordinate system. [31]

The importance of LNED frame becomes evident when we remember that the predicted models of the Earth's magnetic field use it as reference. These models have to be transformed to ECEF and, later, to ECI in order to be compared with the satellite's magnemeter readings. Figure 1.1 shows both the ECEF and LNED coordinate systems.

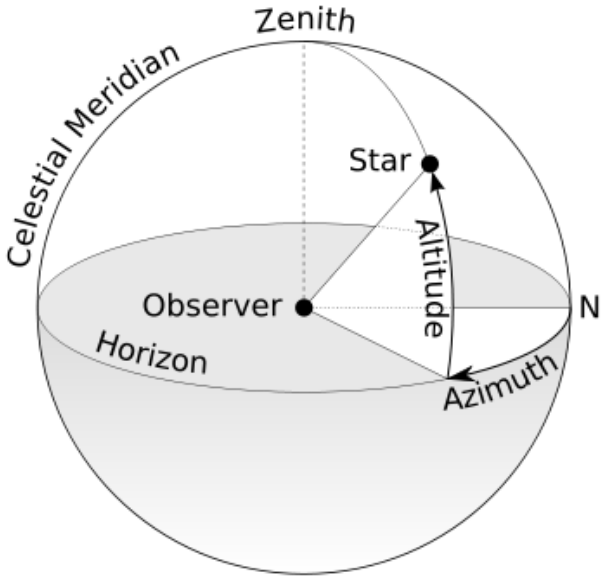

Figure 1.2: Representation of the Horizontal coordinate system [33]

Horizontal Coordinate System

This coordinate system is often used for celestial observation and is particularly useful as it used in prediction models of the Sun's position, that are used by the attitude determination system as comparison reference. It does, however, need a variety of transformations before it can be compared with the solar array's data, starting with a conversion to the LNED frame.
The Horizontal coordinate system has its origin in the observer, in this case, the satellite, and is defined by two angles, as it can be better visualized in Figure 1.2. The azimuth is the angle of the observed object in relation to the local North, measured clockwise, while the elevation (or altitude) is angle between the horizontal plane (local horizon) and the object observed.

1.4 Quaternions

1.4.1 Quaternions Introduction

Quaternions were first devised by William Rowan Hamilton in [34] and are, in essence, a representation of the attitude of a body in relation to a certain reference frame. Although more complex than other attitude representation concepts, quaternions don't suffer from the some of the problems the affect these other representations, such as the *gimbal lock* [35], making them more appropriate and reliable for an attitude determination system.
Quaternions are basically a four element object defined as

$$q = q_4 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} \tag{1.1}$$

where **i**, **j** and **k** represent imaginary components:

$$\mathbf{i^2} = \mathbf{j^2} = \mathbf{k^2} = -1,$$

$$-\mathbf{ij} = \mathbf{ji} = \mathbf{k}, -\mathbf{jk} = \mathbf{kj} = \mathbf{i}, -\mathbf{ki} = \mathbf{ik} = \mathbf{j} \tag{1.2}$$

This notation is in accordance with the JPL Proposed Standard Conventions [36], rather than the notations firstly presented by Hamilton [34], back in 1866.
The imaginary part of the quaternion is given by $q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$ and is also called the vector part (designated as **q**) while q_4 represents the real (or scalar) part of the quaternion. A quaternion can, therefore, also be represented as

$$q = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T \tag{1.3}$$

1.4.2 Quaternion Operations

In this section a brief explanation of the most commonly used quaternion operations is presented.

The *conjugate* of a quaternion defined as in Equation (1.1) is simply given by

$$q^* = q_4 - \mathbf{q} = q_4 - q_1\mathbf{i} - q_2\mathbf{j} - q_3\mathbf{k} \tag{1.4}$$

and it has some interesting properties such as [37]

$$(pq)^* = q^*p^* \tag{1.5}$$

and

$$(p^*q)^* = q^*p. \tag{1.6}$$

However *conjugation* can also be used to extract the scalar and vector parts of a quaternion q. The scalar part of q can be determined as

$$q_4 = \frac{q + q^*}{2} \tag{1.7}$$

and the vector part as

$$\mathbf{q} = \frac{q - q^*}{2}. \tag{1.8}$$

The *norm* of a quaternion is denoted as $|q|$ and can also be obtain from the *conjugate* as

$$|q| = \sqrt{qq^*} = \sqrt{q^*q} \tag{1.9}$$

or, more simply, as

$$|q| = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2}. \tag{1.10}$$

This *norm* will be of particular importance as it allows for the *normalization* of quaternions impeding the filter from diverging. *Normalization* of a quaternion is done as follows

$$\mathbf{U}q = \frac{q}{|q|} \tag{1.11}$$

where $\mathbf{U}q$ represents the *normalized* quaternion, also denominated *unit quaternion*.
The *inverse* of a quaternion has, by definition, the property of $q^{-1}q = qq^{-1} = 1$. It can be obtain from the *conjugate* quaternion as

$$q^{-1} = \frac{q^*}{|q|^2} \tag{1.12}$$

which, in the case of *unit quaternion*, becomes simply:

$$q^{-1} = q^*. \tag{1.13}$$

Furthermore, the algebraic operations of *multiplication* and *addition* are commonly used in the algorithm. *Addition* of two quaternion p and q is simply given by

$$p + q = p_4 + q_4 + (p_1 + q_1)\mathbf{i} + (p_2 + q_2)\mathbf{j} + (p_3 + q_3)\mathbf{k}. \tag{1.14}$$

As for multiplication, Hamilton [34] first devised it as

$$\begin{aligned} p \otimes q = p_4q_4 - p_1q_1 - p_2q_2 - p_3q_3 \\ +(p_4q_1 + p_1q_4 + p_2q_3 - p_3q_2)\mathbf{i} \\ +(p_4q_2 - p_1q_3 + p_2q_4 + p_3q_1)\mathbf{j} \\ +(p_4q_3 + p_1q_2 - p_2q_1 + p_3q_4)\mathbf{k} \end{aligned} \tag{1.15}$$

which can also be written, in a simpler way, in terms of scalar and vector parts of the two quaternion p and q as

$$p \otimes q = (p_4,\, \mathbf{p}) \otimes (q_4,\, \mathbf{q}) = (p_4q_4 - \mathbf{p} \cdot \mathbf{q},\; p_4\mathbf{q} + q_4\mathbf{p} + \mathbf{p} \times \mathbf{q}) \tag{1.16}$$

where $\cdot$ and $\times$ represent the dot and cross products, respectively. This multiplication, however, is not commutative and so, one can must care that $p \otimes q \neq q \otimes p$.

1.4.3 Quaternions as Attitude Representation

If defined as an unit quaternion, i.e.,

$$|q| = \sqrt{|\mathbf{q}|^2 + q_4^2} = 1, \tag{1.17}$$

a quaternion can represent a rotation. These quaternions are called rotational quaternions [38] and define a rotation according to

$$\mathbf{q} = \begin{bmatrix} k_x \sin(\theta/2) \\ k_y \sin(\theta/2) \\ k_z \sin(\theta/2) \end{bmatrix} = \hat{\mathbf{k}} \sin(\theta/2), \tag{1.18}$$

$$q_4 = \cos(\theta/2)$$

A quaternion defined this way will represent a rotation around the axis described by $\hat{\mathbf{k}}$ by the rotation angle θ [22]. It is also noteworthy that such a definition leads to q representing the same final orientation as $-q$, being the only difference the path of the rotation taken, with the shortest path always represented by positive q_4. [36]

For the development of a model for the quaternion dynamics required for the Attitude Determination algorithm, it is important to, firstly, introduce the concepts of the *skew-symmetric matrix operator*:

$$\Big[\mathbf{q}\times\Big] = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \tag{1.19}$$

and the Ω matrix:

$$\begin{aligned} \Omega(\omega) &= \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} \\ &= \begin{bmatrix} -[\omega\times] & \omega \\ -\omega^T & 0 \end{bmatrix} \end{aligned} \tag{1.20}$$

This Ω matrix will be particularly important in the determination of the quaternion derivative. Also the powers of these two matrices will be necessary later on, and are presented in Appendix A. For obtaining these powers, it was important to consider the *skew-symmetric matrix operator* property of *Anti-Commutatively*:

$$[\omega\times] = -[\omega\times]^T \tag{1.21}$$

and *Langrage's Formula* [22]:

$$[a\times][b\times] = ba^T - (a^T b)\, I_{3\times 3} \tag{1.22}$$

1.4.4 Quaternion Dynamics

If we consider an object defined in the local coordinate frame $\{L\}$ and moving in with respect to the global reference frame $\{G\}$, we can compute the rate of change (or *time derivative*) of its attitude quaternion

by

$$ {}^{L(t)}_{G}\dot{q}(t) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} \left({}^{L(t+\Delta t)}_{G}q - {}^{L(t)}_{G}q \right) \tag{1.23} $$

where the quaternion ${}^{L(t+\Delta t)}_{G}q$ can also be written as a product of two quaternions:

$$ {}^{L(t+\Delta t)}_{G}q = {}^{L(t+\Delta t)}_{L(t)}q \otimes {}^{L(t)}_{G}q, \tag{1.24} $$

with,

$$ {}^{L(t+\Delta t)}_{L(t)}q = \begin{bmatrix} \hat{\mathbf{k}} \sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix} \tag{1.25} $$

Note that the quaternion ${}^{L(t+\Delta t)}_{L(t)}q$ describes the rotation of the reference frame $\{L(t)\}$ to the reference frame $\{L(t+\Delta t)\}$ in terms of the rotation angle θ and the axis of rotation $\hat{\mathbf{k}}$ (which is expressed in the $\{L(t+\Delta t)\}$ reference frame). Since when $\Delta t \to 0$, the angle of rotation will be very small, the following approximations can be done:

$$ {}^{L(t+\Delta t)}_{L(t)}q = \begin{bmatrix} \hat{\mathbf{k}} \sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix} \approx \begin{bmatrix} \hat{\mathbf{k}} \cdot \theta/2 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \cdot \delta\theta \\ 1 \end{bmatrix} \tag{1.26} $$

where $\delta\theta$ has the direction of the axis of rotation and the magnitude of the angle of rotation. Dividing it by Δt will yield, in its limit, the rotational velocity.

$$ \omega = \lim_{\Delta t \to 0} \frac{\delta\theta}{\Delta t} \tag{1.27} $$

Given this, it is now possible to obtain the quaternion time derivative as

$$ \begin{aligned} {}^{L(t)}_{G}\dot{q}(t) &= \lim_{\Delta t \to 0} \frac{1}{\Delta t} \left({}^{L(t+\Delta t)}_{G}q - {}^{L(t)}_{G}q \right) \\ &= \lim_{\Delta t \to 0} \frac{1}{\Delta t} \left({}^{L(t+\Delta t)}_{L(t)}q \otimes {}^{L(t)}_{G}q - {}^{L(t)}_{G}q \right) \\ &\approx \lim_{\Delta t \to 0} \frac{1}{\Delta t} \left(\begin{bmatrix} \frac{1}{2} \cdot \delta\theta \\ 1 \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \right) \otimes {}^{L(t)}_{G}q \\ &= \frac{1}{2} \begin{bmatrix} \omega \\ 0 \end{bmatrix} \otimes {}^{L(t)}_{G}q \\ &= \begin{bmatrix} -[\omega\times] & \omega \\ -\omega^T & 0 \end{bmatrix}_{G}^{L(t)} q \\ &= \frac{1}{2} \mathbf{\Omega}(\omega) {}^{L(t)}_{G}q \end{aligned} \tag{1.28} $$

where $\mathbf{\Omega}(\omega)$ is given by Equation (1.20).

Introducing now the *quaternion integrator matrix* $\mathbf{\Theta}(t, t_k)$, one can obtain the attitude quaternion at

current time t from a previous attitude quaternion at time t_k by:

$$ {}^{L}_{G}q(t) = \mathbf{\Theta}(t, t_k){}^{L}_{G}q(t_k) \tag{1.29} $$

Assuming a constant angular velocity ω over the time interval of the integration one can obtain the *zeroth order quaternion integrator* [22]:

$$ \mathbf{\Theta}(\Delta t) = \cos\left(\frac{|\omega|}{2}\Delta t\right) \cdot I_{4\times 4} + \frac{1}{|\omega|}\sin\left(\frac{|\omega|}{2}\Delta t\right) \cdot \mathbf{\Omega}(\omega) \tag{1.30} $$

with the integration interval $\Delta t = t - t_k$.
When $\omega \to 0$, making use of L'Hôpital's rule, we obtain

$$ \lim_{|\omega|\to 0} \mathbf{\Theta}(\Delta t) = I_{4\times 4} + \frac{\Delta t}{2}\mathbf{\Omega}(\omega) \tag{1.31} $$

However, a more accurate approximation is to consider the angular velocity ω as a linear evolution within the interval of integration. This yields the *first order quaternion integrator*:

$$ \mathbf{\Theta}(t, t_k) = \exp\left(\frac{1}{2}\mathbf{\Omega}(\bar{\omega})\right) + \frac{1}{48}\left(\mathbf{\Omega}(\omega(t))\mathbf{\Omega}(\omega(t_k)) - \mathbf{\Omega}(\omega(t_k))\mathbf{\Omega}(\omega(t))\right)\Delta t^2 \tag{1.32} $$

where $\bar{\omega}$ is simply the arithmetic mean of $\omega(t)$ and $\omega(t_k)$ given by

$$ \bar{\omega} = \frac{\omega(t) - \omega(t_k)}{2}. \tag{1.33} $$

Chapter 2

Kalman Filter

2.1 Kalman Filter

There are two commonly used types of algorithm for spacecraft attitude estimation: batch filtering and recursive filtering. Batch filtering makes use of all attitude measurements to update the spacecraft's attitude, including measurements taken after the moment at which attitude information is being calculated. This means that batch filters can only be applied as *a posteriori* attitude information. Although recursive state filters tend to be more unstable than batch filters (since they are more sensitive to individual data points [39]), they use only the current and previous sets of measurements, for the attitude estimation. This gives the possibility to use them for *real-time* attitude estimation.

A *Kalman filter* is a type of recursive filter that takes into account a series of measurements, their inherent noise models and other inacuracies to give a best estimation the current state (the satellite's attitude).

This filter makes use of a system's dynamics model (e.g. satellite attitude dynamics), known control inputs to that system (e.g. the torques applied), and multiple sequential measurements (e.g. form the satellite's attitude sensors) to form an estimate of the system's varying quantities (its *state*). This estimate is better than the one obtained by using a single measurement.

However, noisy sensor data, approximations in the equations that define the system's dynamics, and external factors that are not accounted for introduce some uncertainty on the estimated values for a system's state [40]. The Kalman filter takes into account both a prediction of a system's state based on the system's dynamics and a new measurement taken using a weighted average. The purpose of the weighted average over the arithmetic one is that values with better (i.e., smaller) estimated uncertainty are "trusted" more. The weights are calculated from the *covariance*, a measure of the estimated uncertainty of the prediction of the system's state. The result of the weighted average is a new state estimate that lies in between the predicted and measured state, and has a better estimated uncertainty than either alone. This process is repeated every time step, with the new estimate and its covariance informing the prediction used in the following iteration. This means that the Kalman filter works recursively and requires only the last "best guess", rather than the entire history of a system's state to calculate a new

state.
It is, in essence, a two-step algorithm. In the first step, the *prediction step*, the algorithm will develop a prediction of the future satellite's attitude state, based on the current state. For this, the algorithm makes use of a well defined dynamic model for the satellite's attitude based on the current state.
In the second step, the *update step*, a new set of measurements is available and the algorithm will take into account these measurements, the modeled noise of those measurements and the prediction obtained in the previous step to then perform the weighted estimation of the current satellite's attitude.

2.2 Kalman Filter Equations

In this section, an overview of the general equations that are used in Kalman filters is given. These equations will be (in Sections 2.3 through 2.5) adapted to the specific problem of satellite attitude determination. The equations here presented are based on the work of Welch and Bishop [41]. However a more intuitive notation is used, in order to facilitate further development of the equations.
Firstly it is important to introduce the following matrices:

- $\mathbf{F}_k$, the state-transition model matrix;
- $\mathbf{H}_k$, the observation/measurement model matrix;
- $\mathbf{Q}_k$, the covariance matrix of the process noise;
- $\mathbf{R}_k$, the covariance matrix of the observation/measurement noise;
- $\mathbf{B}_k$, the control input model matrix.

In these, the subscript k corresponds to the time step. In each time step the subscripts $k-1$, k, and $k+1$ correspond to the previous, the current and the next time step, respectively. In a general approach, given the previous state vector of the system, $\mathbf{x}_{k-1}$, a prediction can be made as to the current state $\mathbf{x}_k$ by:

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k \tag{2.1}$$

This prediction makes use of the systems dynamic model, defined by the matrix $\mathbf{F}_k$, applied to the previous state $\mathbf{x}_{k-1}$ and control input model matrix $\mathbf{B}_k$, applied to the input control vector $\mathbf{u}_k$. $\mathbf{w}_k$ corresponds to the process noise vector. This noise vector is assumed to follow a gaussian distribution with zero mean and it is defined by the covariance matrix of the process noise $\mathbf{Q}_k$. This $\mathbf{Q}_k$ needs to accurately estimate the process noise.
For the next steps, the notation $\hat{\mathbf{x}}_{n|m}$ is introduced. This represents the estimation of the state vector $\mathbf{x}$ at time n, based on the state and measurements up until time m. This implies that $m \leq n$. For fully defining a state, one needs the current (at k) state estimation vector $\hat{\mathbf{x}}_{k|k}$ based on all the information available (up until k) and the error covariance matrix $\mathbf{P}_{k|k}$. This matrix $\mathbf{P}_{k|k}$ defines how accurate the $\hat{\mathbf{x}}_{k|k}$ estimation is.
For further analysis, it is easier to consider to two distinct steps mentioned in Section 2.1.

2.2.1 Prediction Step

For each time step, the prediction step shall be the first part of the algorithm operations. In it, the state vector is estimated based on the system's dynamic model and the control inputs, as in:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} \tag{2.2}$$

Note the differences to Equation (2.1), e.g. the control model input model and its vector are no longer the current ones, as the necessary control vector cannot be determined prior to the state estimation. Note also that the process noise vector $\mathbf{w}_k$, defined by the covariance matrix $\mathbf{Q}_k$, is no longer considered. This comes naturally as the noise is not a part of the measurements used by the algorithm and is considered to have a zero mean. It is, however, considered in the calculations of the state estimation's covariance matrix:

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1}\mathbf{F}_k^T + \mathbf{Q}_k \tag{2.3}$$

2.2.2 Update Step

In the update step, a measurement (or observation) vector, $\mathbf{z}_k$, is now available. This measurement is compared to a predicted estimation of the measurement vector, $\hat{\mathbf{z}}_{k|k-1}$. It is defined by the measurement model matrix $\mathbf{H}_k$ and the estimated state vector, $\hat{\mathbf{x}}_{k|k-1}$, determined in the prediction step.

$$\hat{\mathbf{z}}_{k||k-1} = \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \tag{2.4}$$

Comparing the measured and predicted measurement vector, we obtain the innovation (or measurement residual),

$$\mathbf{r}_k = \mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1} \tag{2.5}$$

and with it, the innovation covariance matrix

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k \tag{2.6}$$

where $\mathbf{R}_k$ is the covariance matrix of the observation/measurement noise.
Given the innovation covariance matrix, it is now possible to determine the optimal Kalman gain [7]:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T \mathbf{S}_k^{-1} \tag{2.7}$$

Finally, we are now in condition to determine the state estimate vector,

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \mathbf{r}_k \tag{2.8}$$

and its covariance matrix,

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_{k|k-1}, \tag{2.9}$$

thus fully defining the current state. The algorithm would then move forward in time with:

$$\hat{\mathbf{x}}_{k-1|k-1} = \hat{\mathbf{x}}_{k|k} \tag{2.10}$$

$$\mathbf{P}_{k-1|k-1} = \mathbf{P}_{k|k} \tag{2.11}$$

and then repeat the whole process, starting with the new prediction step.

2.3 Extended Kalman Filter

Usually, the first approach in a Kalman filter algorithm design is to consider that the physical processes can be approximated to a linear function. This simplifies the filter implementation and reduces calculation times. However its assumptions can be very inaccurate in some occasions. [42]
In the extended Kalman filter (EKF), the state transition and observation models are no longer considered to be linear functions of the state. This yields a more correct approach and more accurate results for the estimation of a satellite's attitude information.
The state vector estimate $\hat{\mathbf{x}}_{k|k-1}$, previously given by Equation (2.2), is more accurately define as:

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}) \tag{2.12}$$

where f represents the differentiable function used to compute the predicted state from the previous estimate. However, f cannot be applied to the covariance $\mathbf{P}_{k|k-1}$ directly. Instead a matrix of partial derivatives (the Jacobian) is computed. The same approach is used for the measurement model matrix $\mathbf{H}_k$ with the differentiable function h (used for computing the predicted measurement vector from the predicted state vector). This implies than, for Equations (2.3) through (2.7), the matrices $\mathbf{F}_k$ and $\mathbf{H}_k$ are given by:

$$\mathbf{F}_k = \left[\frac{\delta f}{\delta x}\right]_{\hat{\mathbf{x}}_{k|k-1}, \mathbf{u}_{k-1}} \tag{2.13}$$

$$\mathbf{H}_k = \left[\frac{\delta h}{\delta x}\right]_{\hat{\mathbf{x}}_{k|k-1}} \tag{2.14}$$

Also Equation (2.9), is now given (from [22]) by:

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T \tag{2.15}$$

2.4 The 7-state Extended Kalman Filter

The 7-state Extended Kalman filter is a reliable control filter commonly used for the ADCS task. It uses relatively simpler mathematical operations, when compared to other EKFs. [43]
As its name indicates, the 7-state EKF uses a state vector of 7 elements. These are the 4 attitude quaternion elements, with respect to the inertial frame, and the 3 angular rates, also defined in the

inertial frame:

$$\mathbf{x} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & \omega_x & \omega_y & \omega_z \end{bmatrix}^T \tag{2.16}$$

During the time propagation of the state vector, the quaternion and the angular rate components are considered separately. The attitude quaternion is propagated forward using the quaternion derivative equation introduced in Equation (1.28) and the angular rates are propagated using Euler's Moment Equations [44]:

$$I\dot{\omega} = \mathbf{T} - \omega \times I\omega \tag{2.17}$$

where T represents the torque vector acting on the system and *I* the moment of inertia tensor. Not considering disturbance torques, $\mathbf{T}$ corresponds to the input control torque vector.

2.4.1 Prediction Step

The state vector estimate $\hat{\mathbf{x}}_{k|k-1}$ is still given by Equation (2.2). However, now that the state vector is defined (by Equation (2.16)), we can define the state transition model matrix, $\mathbf{F}_k$, as:

$$\mathbf{F}_k = \mathbf{I}_{7\times 7} + \begin{bmatrix} \frac{\delta \dot{q}}{\delta q} & \frac{\delta \dot{q}}{\delta \omega} \\ \frac{\delta \dot{\omega}}{\delta q} & \frac{\delta \dot{\omega}}{\delta \omega} \end{bmatrix}_{7\times 7} \cdot \Delta t \tag{2.18}$$

where the index 7×7 indicates the dimensions of the matrices and Δt is the time step at which the algorithm will operate.

The state covariance matrix, $\mathbf{P}_{k|k-1}$, is still given by Equation (2.3), with $\mathbf{F}_k$ now defined by Equation (2.18).

Using Equations (1.28) and (2.17) we are able to propagate the two components of the state vector:

$$q_{k|k-1} = q_{k-1|k-1} + \frac{1}{2}\int_{t_{k-1}}^{t_k} (\mathbf{\Omega} q_{k-1|k-1}) dt \tag{2.19}$$

$$\omega_{k|k-1} = \omega_{k-1|k-1} + \int_{t_{k-1}}^{t_k} I^{-1}\left[\mathbf{T} - \omega \times I\omega\right] dt \tag{2.20}$$

2.4.2 Update Step

Within the update step, almost all of the kalman filter equations remain the same, using the results from the prediction step. One exception is the optimal Kalman gain, which is now, more accurately given (from [43]) by:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T \left[\mathbf{H}_k \mathbf{P}_{k|k-1}\mathbf{H}_k + \mathbf{R}_k\right]^{-1} \tag{2.21}$$

with the measurement model matrix, now given by:

$$\mathbf{H}_k = \frac{\delta z}{\delta x} = \begin{bmatrix} \frac{\delta z}{\delta q_1} & \frac{\delta z}{\delta q_2} & \frac{\delta z}{\delta q_3} & \frac{\delta z}{\delta q_4} & \frac{\delta z}{\delta \omega_x} & \frac{\delta z}{\delta \omega_y} & \frac{\delta z}{\delta \omega_z} \end{bmatrix} \tag{2.22}$$

Note that the filter uses only the previous state for estimating the next state, rather than all previous data available. This greatly reduces the total stored data of the algorithm and the computation time. It is also noteworthy the fact that the estimated state (Equations (2.19) and (2.20)) is obtained by addiction operations from the previous state. This is a simplistic approach to the quaternion operations presented in section 1.4.2 that may lead to accumulated inaccuracies.

2.5 The 6-state Extended Kalman Filter

The 7-state EKF presents a good way of attitude determination for small satellites. However, the 7-state EKF involves many 7 x 7 matrices, placing a fairly heavy computational demand upon the onboard computer of the typical low power and low cost nanosatellite. Hence, any means of significantly reducing the computational needs of the EKF while not sacrificing operability is highly desirable. [45]

The 6-state EKF, first theorized by Lefferts [18] aims at reducing the computational load of the 7-state EKF by reducing the attitude quaternion to a 3 element attitude vector within the filter computations. This needs to be performed with caution in order not to loose the advantages of using quaternions over 3D vectors for attitude representation.

In order not to fall back into 3 element vector attitude representation, a more complex set of calculations is used. This will, by itself, increase computational load, however, the reduced dimension of the matrices being calculated more than makes up for it. [43],[45]

Firstly, it is important to introduce the notions of quaternion and angular rate differential errors δq and $\delta\omega$. These are the main characteristic of the 6-state EKF and represent, in essence, small deviations in the attitude quaternion and body rates. These can be used to propagate the variables in time by:

$$q_k = \delta q \otimes q_{k-1} \tag{2.23}$$

$$\omega_k = \omega_{k-1} + \delta\omega \tag{2.24}$$

where the $\otimes$ operator represents the quaternion multiplication, previously defined in Section 1.4.2.

While Equation (2.24) is somewhat intuitive and simply understood, Equation (2.23) does not come so naturally. Basically δq represents a deviation in the attitude quaternion. It is, itself a quaternion, but it does **not** correspond to the spacecraft's attitude. Considering all the quaternion components

$$\delta q = \begin{bmatrix} \delta q_1 & \delta q_2 & \delta q_3 & \delta q_4 \end{bmatrix}^T \tag{2.25}$$

and quaternion properties (specifically, their unitary norm), it comes that

$$\delta q_4 = \sqrt{1 - \delta q_1^2 - \delta q_2^2 - \delta q_3^2} \tag{2.26}$$

Since δq_4 can be defined from the remaining properties, its use it redundant, and so, the 6-state EKF state vector is defined as

$$\mathbf{y} = \begin{bmatrix} \delta q_1 & \delta q_2 & \delta q_3 & \delta \omega_x & \delta \omega_y & \delta \omega_z \end{bmatrix}^T \tag{2.27}$$

This state vector is considered the *auxiliary* state vector and is the one used in the filter computations for the 6-state EKF. Also a *true* state vector, similar to the one defined in Equation (2.16) is used, in order to represent the attitude of the satellite. This *true* state vector will introduce the notion of angular rate bias, $\mathbf{b}$, that gives the difference between the measured and the estimated angular rates:

$$\hat{\omega} = \omega_m - \mathbf{b} \tag{2.28}$$

For very accurate measurements and estimations, $\mathbf{b} \approx \mathbf{O}_{3\times1}$, a 3×1 null matrix (or vector), thus facilitating calculations. The *true* state vector is, then, given by:

$$\mathbf{x} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & \mathbf{b}_x & \mathbf{b}_y & \mathbf{b}_z \end{bmatrix}^T \tag{2.29}$$

The auxiliary state vector, can also be further optimized [22] into:

$$\mathbf{y} = \begin{bmatrix} \delta\theta_1 & \delta\theta_2 & \delta\theta_3 & \Delta\mathbf{b}_x & \Delta\mathbf{b}_y & \Delta\mathbf{b}_z \end{bmatrix}^T \tag{2.30}$$

where

$$\delta q_i = k_i \sin(\delta\theta/2) \approx \frac{1}{2}\delta\theta_i \tag{2.31}$$

with k_i corresponding to the normalized vector part of the quaternion

$$k_i = \frac{q_i}{|\mathbf{q}|} \tag{2.32}$$

and with

$$\mathbf{b}_k = \mathbf{b}_{k-1} + \Delta\mathbf{b} \tag{2.33}$$

2.5.1 Prediction Step

The state transition matrix, $\mathbf{F}_k$ is now obtained from [22]:

$$\mathbf{F}_k = \begin{bmatrix} \mathbf{\Theta} & \mathbf{\Psi} \\ \mathbf{O}_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix} \tag{2.34}$$

The matrix $\mathbf{\Theta}$ is, in fact, a rotational matrix with ω as the axis and $|\hat{\omega}|\Delta t$ as the angle:

$$\mathbf{\Theta} = \cos(|\hat{\omega}|\Delta t) \cdot \mathbf{I}_{3\times3} - \sin(|\hat{\omega}|\Delta t) \cdot \left[\frac{\hat{\omega}}{|\hat{\omega}|}\times\right] + (1 - \cos(|\hat{\omega}|\Delta t)) \cdot \frac{\hat{\omega}}{|\hat{\omega}|} \cdot \left[\frac{\hat{\omega}}{|\hat{\omega}|}\right]^T \tag{2.35}$$

It can be further simplified, for small values of $|\hat{\omega}|$, to:

$$\mathbf{\Theta} = \mathbf{I}_{3\times3} - \Delta t\,[\hat{\omega}\times] + \frac{\Delta t^2}{2}\,[\hat{\omega}\times]^2 \tag{2.36}$$

The matrix $\mathbf{\Psi}$ is given by:

$$\mathbf{\Psi} = -\mathbf{I}_{3\times3}\Delta t + \frac{1}{|\hat{\omega}|^2}\left(1-\cos(|\hat{\omega}|\Delta t)\right)[\hat{\omega}\times] - \frac{1}{|\hat{\omega}|^3}\left(1-\sin(|\hat{\omega}|\Delta t)\right)[\hat{\omega}\times]^2 \tag{2.37}$$

Which can also be approximated, for small values of $|\hat{\omega}|$, by:

$$\mathbf{\Psi} = -\mathbf{I}_{3\times3}\Delta t + \frac{\Delta t^2}{2}[\hat{\omega}\times] - \frac{\Delta t^3}{6}[\hat{\omega}\times]^2 \tag{2.38}$$

The $\mathbf{O}_{3\times3}$ and $\mathbf{I}_{3\times3}$ matrices represent a null and an identity 3×3 matrices, respectively.
The state transition matrix $\mathbf{F}_k$ is fully defined, and is used, according to Equation (2.3), to determine the state covariance matrix.

2.5.2 Update Step

In relation to the 7-state EKF, most equations from the update step, remain the same in the 6-state EKF, with the biggest different being the dimension of the matrices. A few calculations need to be introduced, though, to cope with the different state vector being used.
The final estimated state vector is no longer obtained from (2.8), but rather:

$$\mathbf{y}_{k|k} = \begin{bmatrix} \delta\theta \\ \Delta\mathbf{b} \end{bmatrix} = \mathbf{K}_k \cdot \mathbf{r} \tag{2.39}$$

where $\mathbf{r}$ represents the *innovation* given by the difference between the measured and the predicted measurement vectors.
Finally, the attitude quaternion and the angular rate (*i.e.* the true state vector) can than be updated according to Equations (2.23),(2.24),(2.31) and (2.33).

Chapter 3

ECOSat Attitude Determination System

For the correct determination of attitude of the ECOSat, it important to know beforehand its location in space. This is relevant because the Sun's relative position and the local magnetic field depend on the position at that time. Given the corrent orbital elements, we can obtain the expected values for the local magnetic field and Sun position and compare them against instrument data for attitude determination.

3.1 Orbit Determination and Propagation

Orbits around a primary body (in this case, the Earth) describe an ellipse, according to Kepler's laws, with the primary body in one of the focus. This orbit, assuming no disturbances or thrusting maneuvers, shall remain constant in time, for an inertial frame.

In the case of the ECOSat, this is true: the mission makes use of no thrusting systems and therefore is not capable of inducing orbital changes. However, the same cannot be said for disturbance forces. These include the gravity variation caused by Earth's equatorial bulge, aerodynamic drag caused by residual atmosphere, gravitational disturbances induced by the moon and sun, solar radiation pressure, amongst others.

These disturbances will make the an orbit prediction based on these Keplarian laws increasingly incorrect over time. Therefore, it is necessary to incorporate the GPS measurements into the algorithm to correct the deviations from prediction that are induced over time.

Models for most of these disturbance torques have been presented with various degrees of complexity (ranging from simple linear approximations to high order complex mathematical models). For example, the Earth's equatorial bulge effect could be approximated to a precession of the orbital plane around the rotation axis of the Earth described by:

$$\omega_p = -\frac{3}{2}\frac{R_E^2}{\left(a(1-e^2)\right)^2} J_2 \omega \cos i \tag{3.1}$$

with R_E corresponding to the Earth's equatorial radius, a to the semi-major axis of the satellite orbit (which will be further explained in Section 3.1.1), e and i to the orbital eccentricity and inclination, respectively (both detailed in Section 3.1.1) and J_2 described by:

$$J_2 = \frac{2\epsilon_E}{3} - \frac{R_E^3 \omega_E^2}{3GM_E} \tag{3.2}$$

where ϵ_E is the Earth's oblateness, ω_E is the Earth's rotational rate and M_E is the Earth's mass.
The J_2, denominated Earth's second dynamic form factor, has a constant value (1.08×10^{-3}) and accounts for the second (with the first being the simple sphere) of the spherical harmonics that model the Earth's shape. [46]
However, these models of the disturbance forces yield an increased demand in computational power and time. If the GPS data is precise and properly integrated in the simple Keplarian orbital propagation, one can dismiss these models and obtain a lightweight algorithm from the computational point of view, with no significant loss in precision. That was the approach followed for the case of the ECOSat.

3.1.1 Orbital Elements

The orbital elements are the parameters required to uniquely identify a specific orbit around the Earth (or any other central body). There are multiple ways to choose these parameters, but, for any set of parameters used, exactly 6 different parameters will be required for uniquely defining the orbit and position within it. This comes naturally from the 6 degrees of freedom of the problem (the 3 position dimensions (e.g. x, y, z in a cartesian coordinate frame) and the 3 velocity dimension).
Although commonly used in other problems, a cartesian coordinate frame is not the most practical approach for the case of orbital mechanics. The classical keplerian orbital elements, on the other hand, present a much more intuitive set of parameters. These are represented in Figure 3.1

Eccentricity (*e*)

The eccentricity (the only orbital element not pictured in Figure 3.1), basically, describes how elongated the ellipse that defines the orbit is. It ranges from 0 (a non elongated ellipse, *i.e.* a circular orbit) to 1 (the maximum elongation of the ellipse, *i.e.* a parabola). Higher values are also used, however, eccentricities bigger than 1 imply that the orbit is an hyperbolic one. This means that the orbit has enough orbital energy to leave the system, *e.g.* the gravity swing-by maneuvers often performed in Jupiter by deep space probes.
The eccentricity can be obtain for any elliptical orbit by:

$$e = \sqrt{1 + \frac{2\epsilon h^2}{\mu^2}} \tag{3.3}$$

where ϵ is the specific orbital energy, h the angular momentum and μ the standard gravitational param-

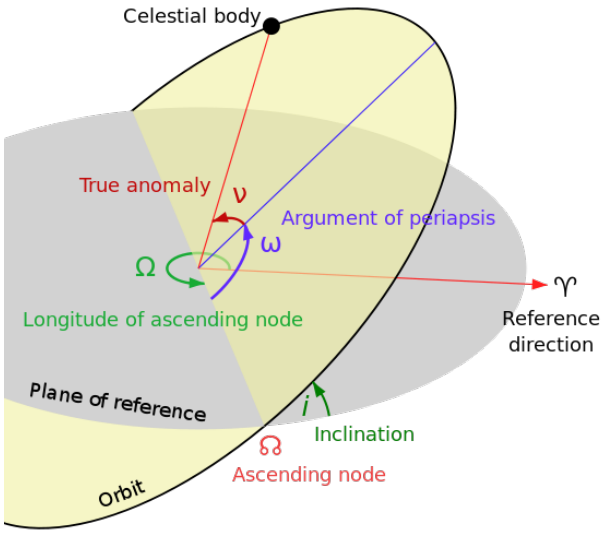

Figure 3.1: Classical Orbital Elements. [47]

eter.

Semi-major Axis (*a*)

The semi-major axis, as the name indicates, is half of the length of the major axis that defines the orbital ellipse. A representation of the semi-major axis and semi-minor axis of an orbit is present in Figure 3.2.

The perigee (closest point of the orbit to the Earth) radius, r_p and the apogee (furthest point of the orbit in relation to the Earth) radius, r_a allow for the trivial calculation of the semi-major axis:

$$a = \frac{r_p + r_a}{2} \tag{3.4}$$

and also relate to the eccentricity by:

$$e = \frac{r_a - r_p}{r_a + r_p} \tag{3.5}$$

These two elements (semi-major axis and eccentricity) fully define the size and shape of the orbit in a bi-dimensional plane. According to Kepler's laws, the orbit will always be contained in this plane. It now follows that we define this orbit in space.

Inclination (*i*)

The inclination, represented in green in Figure 3.1, defines the angle by which the orbital plane is tilted with respect to the reference plane (usually, the equatorial plane in the case of satellite's orbiting the Earth). This angle can be measured when the object orbiting the central body intercepts the reference frame. It's velocity is always contained in the orbital plane and, as such, can be used for obtaining the inclination angle.

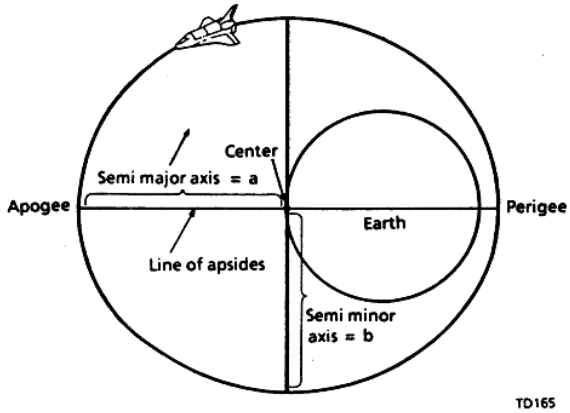

Figure 3.2: Bi-dimensional representation of an orbit. Notice that the Earth is not at the center of the ellipse but rather on one of the focus. [48]

Longitude of the ascending node (Ω)

The longitude of the ascending node defines the angular position of the ascending node, *i.e.*, the point where the orbiting object passes the reference plane (Earth's equatorial plane, in our case) in the upwards direction. As seen in Figure 3.1, the longitude of the ascending node is measured in respect to a reference direction. For a geocentric orbit, with Earth's equatorial plane as the reference plane, the reference direction is usually the First Point of Aries (*i.e.* the direction of the sun during the Vernal Equinox in the Northern Hemisphere) as the origin of longitude. The angle is measured eastwards from the First Point of Aries to the ascending node.

These two elements (inclination and longitude of the ascending node) fully define the equatorial plane in space with respect to the central body.

Argument of the periapsis (ω)

The argument of the periapsis defines how the orbit is orientated within its plane. It gives the angular distance of the periapsis with respect to ascending node. With this final element the orbit is fully defined in space. It is represented in Figure 3.1, in blue.

Mean anomaly at epoch (M_0)

Finally, the mean anomaly defines the position of the orbiting body in its orbit, at a specific time (the "epoch").

The mean anomaly is a representative "angle" that varies linearly with time and corresponds to $360°$ after one orbital period. However, it does not correspond to a real geometric angle over time, as the orbital velocity is not constant over the different parts of the orbit (with the expeption of circular orbits). The true anomaly (ν), shown in red in Figure 3.1, does represent the real geometric angle in the plane of the orbit, between perigee and the position of the orbiting object at any given time. It can be obtained

from the mean anomaly making use of the eccentric anomaly (*E*) and the following equations:

$$\cos\nu = \frac{\cos E - e}{1 - e \cdot \cos E} \tag{3.6}$$

$$M = E - e \cdot \sin E \tag{3.7}$$

where *M* corresponds to the mean anomaly at time *t* after the Epoch, given by:

$$M = M_0 + nt \tag{3.8}$$

with *n* corresponding to the mean angular motion in orbit and *t* the time passed since the orbiting object was at the perigee.

3.1.2 GPS Data

In this section, the process of obtaining all the orbital elements introduced in the previous section, making use of the GPS measurements is presented. For defining the orbit of the ECOSat, at least two sets of GPS measurements are necessary. Each set of measurements has the following information:

- **Latitude** (*Lat*): angular distance from the equator as defined in the Geographic Coordinate System (in degrees);
- **Longitude** (*Lon*): angular distance from the Greenwich meridian as defined in the Geographic Coordinate System (in degrees);
- **Altitude** (*h*): vertical distance from the Earth's surface as defined in the Geographic Coordinate System (in meters);
- **Date** (*d, m, y*): date at which measurements were taken (including day, month and year, in UTC),
- **Time** (*t*): the time at which the measurements were taken (in seconds, since 00:00:00 UTC).

Firstly, we can obtain the current orbital radius (with respect to the Earth's center) from the measurements of altitude. For that we first need to note that, due to the non-spherical shape of the Earth, the latitude given by the GPS measurements (the geodetic latitude) does not correspond exactly to the latitude measured from the Earth's center, as represented in Figure 3.3. This will lead to incorrections in the orbital radius obtained. Therefore, one must first convert the GPS latitude into geocentric latitude. This requires the calculation of the geocentric latitude (λ_S) of the surface point (point Q of Figure 3.4) as

$$\lambda_S = \arctan\left((1-f)^2 \tan\mu\right) \tag{3.9}$$

where *f* is a general geometrical parameter that represents the Earth's flatness and μ to the geodetic latitude (both as given by the GPS measurements). Its value has become more and more precise as methods for determining it have improved over the years. The value used in this project was the one

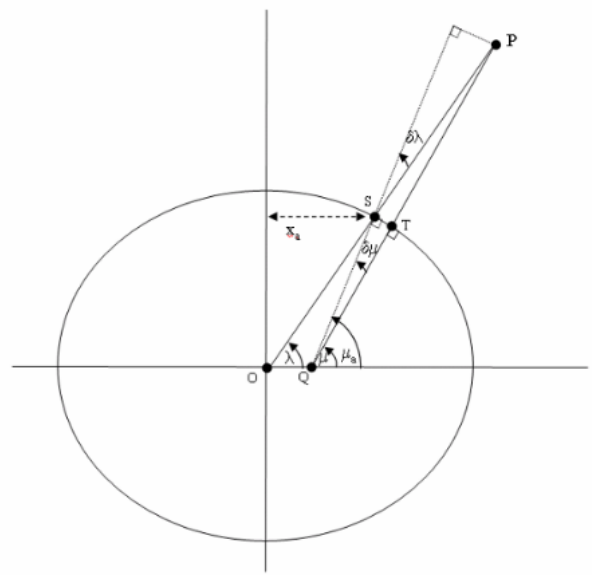

Figure 3.3: Representation of the difference between geodetic latitude (μ) and geocentric latitude (λ) of a point P.

estimated in the WGS84 [49].

The local radius of the Earth's geoid is also obtained from the WGS84 as:

$$r_S = \sqrt{\frac{r_e^2}{1 + \left(\frac{1}{(1-f)^2} - 1\right)\sin^2\lambda_S}} \tag{3.10}$$

where r_e corresponds to the Earth's equatorial radius.

One can then obtain the geocentric latitude (λ),

$$\lambda = \arctan\left(\frac{h\sin\mu + r_S\sin\lambda_S}{h\cos\mu + r_S\cos\lambda_S}\right) \tag{3.11}$$

with *h* corresponding to the altitude with respect to the surface.

The current orbital radius can then be obtained from the trigonometry of non-right triangles:

$$r_O = \sqrt{h^2 + r_S^2 - 2hr_S\cos\alpha} \tag{3.12}$$

where α, as represented in Figure 3.5, can be obtained from

$$\begin{aligned} \alpha &= 180^\circ - \beta - \gamma \\ &= 180^\circ - (\mu - \lambda) - (\lambda - \lambda_S) \\ &= 180^\circ - \mu + \lambda_S \end{aligned} \tag{3.13}$$

Differentiating the measurements from the GPS, we can then, obtain the horizontal and vertical components of the velocity, v_h and v_v, respectively. These allow us to compute the semi-major axis and

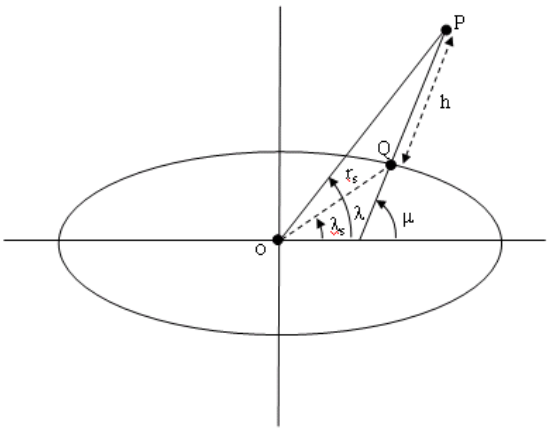

Figure 3.4: Representation of the process used to obtain the geocentric latitude (λ) of an object in point *P*. The altitude over the surface point Q (as measured by the GPS) is represented by *h*, while r_s represents the Earth's radius at that point, λ_S the geocentric latitude of the surface point *Q* and μ the geodetic latitude (as given by the GPS).

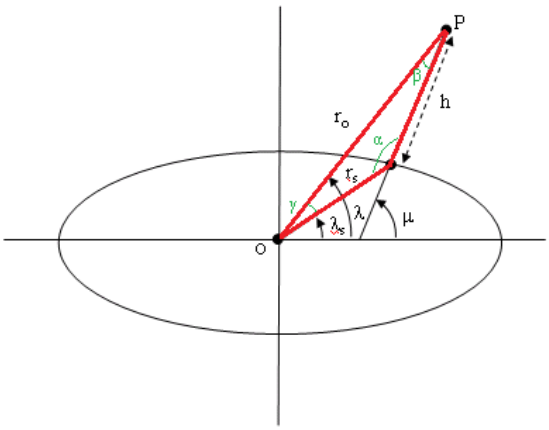

Figure 3.5: Representation of the process used to obtain the orbit radius (*OP*) as according to Equation (3.12).

the eccentricity by considering the *vis viva* equation [50]:

$$-\frac{\mu}{2a} = \frac{v^2}{2} - \frac{\mu}{r} \tag{3.14}$$

and solving it with respect to a:

$$\begin{aligned} a &= \frac{\mu}{\frac{2\mu}{r} - v^2} \\ &= \frac{r\mu}{2\mu - rv^2} \\ &= \frac{r}{2 - \frac{rv^2}{\mu}} \end{aligned} \tag{3.15}$$

On the other hand, the eccentricity can be obtained from the flight path angle

$$\gamma = \arctan\left(\frac{v_v}{v_h}\right) \tag{3.16}$$

by:

$$e = \sqrt{\left(\frac{rv^2}{\mu}\right)^2 \cos^2\gamma + \sin^2\gamma} \tag{3.17}$$

Considering the two measurements of the GPS, their time tags and the current date, the orbital plane in the Earth Centered Inertial (ECI) frame can be obtained by mathematical methods of plane definition from 3 non collinear points (the two GPS measurement points and the Earth's center). This allows us to compute the inclination of the orbit in relation to the Earth's equatorial plane simply by

$$i = \arccos\left(\frac{n_o \cdot n_e}{|n_o| \times |n_e|}\right) \tag{3.18}$$

where n_o and n_e represent two vector normal to the orbital and equatorial plane respectively.
For the computation of the ascending node, the algorithm analyses whether (between the two GPS measurements) the satellite is moving upwards or downards (with respect to the equatorial plane). This allows for the later distinction between the descending and ascending nodes. These are simply obtained from the intersection of the orbital and equatorial planes.
For the case of nearly circular orbits (such as the one defined for the ECOSat's mission) the remaining orbital elements need not to be determined as they add no significant increase in position's determination accuracy. The argument of the periapsis and mean anomaly at epoch were, therefore, set to $0\,\mathrm{deg}$ (coincinding with the Greenwich Meridian) for the initial conditions of the algorithm. The mean anomaly will, naturally, increase as the satellite goes in its orbit.

3.1.3 Orbit Prediction

According to Kepler's laws of planetary motion [51] all orbital elements of an orbiting body, with the exception of the mean anomaly are constants. The mean anomaly can be computed from the mean

angular orbital rate (n) by

$$M_t = M_{t-1} + n\Delta t \tag{3.19}$$

where Δt corresponds to the time interval between the current and previous mean anomalies, M_t and M_{t-1}, respectively. The mean angular rate is also constant throughout the orbit and can be computed as:

$$n = \sqrt{\frac{\mu}{a^3}} \tag{3.20}$$

Following these laws, the algorithm makes a prediction of the orbital parameters for the defined time step (1 second). After this, a new set of measurements is available and a new orbit obtained. The two sets of orbital parameters (predicted and computed) are then weighted and averaged for obtaining the algorithm's orbit estimation. This happens iteratively throughout the satellite's mission.

3.2 Attitude Determination

3.2.1 Solar Array Data

For the estimation of the sun's position at instant of the orbit the USA's National Renewable Energy Laboratory's (NREL) Solar Position Algorithm (SPA) [52] was integrated in this project's algorithm.
The NREL's algorithm was developed for Earth based applications such as energy gathering arrays. However, it is perfectly valid for space based applications since it takes into account different atmospheric variables that provide additional accuracy according to each Earth's region. These variables need to be adjusted for the space environment, *e.g.*, setting the atmospheric refraction and atmospheric pressure to 0. This adjustment of variables for space environment, in most cases, leads to a simplification of the equations used in the original algorithm. However, these approximations are valid and lead to a lighter algorithm with insignificant loss in accuracy. This makes the final algorithm lighter and faster which is important if it is to be used iteratively.
Other inputs of the NREL's algorithm include the position variables (longitude, latitude and altitude) and time, which are directly obtained from the GPS measurements. Given all the inputs, the algorithm estimates the sun's position in the horizontal coordinate system. This represents the SPA's output and it then needs to be converted into the ECI coordinate system. This yields the algorithm's prediction on the sun's position and is, later, compared with the data obtained from the solar arrays. Simply put, the solar arrays, disposed on 3 orthogonal sides of the satellite, compare the power being generated at each moment on each of the 3 sides in order to triangulate the Sun's position. If the levels are low the algorithm discards the data of the solar arrays in the determination, in order to avoid inaccuracies during eclipse times.

3.2.2 Magnetometer Data

Like in the case of the solar position determination, an external algorithm was adapted and integrated for the task of Earth's magnetic field prediction. USA's National Geophysical Data Center (NOAA) World

Magnetic Model (WMM) [53] consists of a degree and order 12 spherical-harmonic main (i.e., core-generated) field model comprised of 168 spherical-harmonic Gauss coefficients and a degree and order 12 spherical-harmonic Secular-Variation (SV) (core-generated, slow temporal variation) field model. The WMM source code is public domain and not licensed or under copyright and, like the SPA, it is available in C source code. This makes it perfectly suitable for the purposes this project.
The WMM algorithm inputs are:

- Latitude
- Longitude
- Altitude
- Date

These can all be directly obtained for the GPS measurements. The outputs of the algorithm are:

- North Component of the geomagnetic field (X)
- East Component of the geomagnetic field (Y)
- Vertical Component of the geomagnetic field (Z)
- Total Intensity of the geomagnetic field (F)
- Horizontal Intensity of the geomagnetic field (H)
- Geomagnetic Inclination (DIP)
- Geomagnetic Declination (Magnetic Variation) (DEC)

Out of these, only the first three (X, Y and Z) are used in the ADS algorithm. They form a vector that corresponds to the direction (and intensity) of the geomagnetic field, given in the LNED coordinate system. The vector is then normalized, since the intensity of the field is not relevant for the task of attitude determination. This forms the algorithm's prediction of the Earth's magnetic field. The data measurements from the magnetometer's are then taken and compared with the prediction to obtain the most accurate estimation of the ECOSat's attitude.

3.3 Algorithm Implementation

The algorithm implemented was firstly developted in simulation mode, *i.e.* not attached to the hardware. The algorithm has non noisy models of the instruments and can determine the actual attitude of the satellite. Simulated noise is then added and the determination attitude algorithm tests against the reference *true* attitude.
The determination of the orbital parameters and attitude of the satellite is based on the 6 state Kalman filter, previously defined in Section 2.5. The flowchart presented in Figure 3.6 represents the structure

of this algorithm.

In the first step, the variables for the simulation are set. These include the initial time and date of the simulation, the position and the attitude of the satellite, *i.e.*, the initial attitude quaternion and angular velocity vector. These variables correspond to the true attitude and position of the satellite that will be estimated by the attitude determination algorithm. They change throughout the iteration according to satellite dynamics and serve as reference for the later incorporation of the instrument measurements. Also the number of iterations for the simulation is defined here.

The initial conditions of the Kalman filter are then defined. These provide starting point of the filter. From this starting point the filter will try to converge to the real attitude of the satellite. Since the the initial attitude of a real satellite is impossible to guess, the definition of the variables can be done rather arbitrary. Performance of the filter with different values for the starting conditions is presented in Chapter 5. Also the expected noise of the different instruments is defined at this point. This is dependent on the type and precision of the instrument and was defined according to each instrument's manufacturer's specifications.

Other initiation tasks of this simulation algorithm include the creation of all the Kalman filter variables to be used in the attitude prediction and update and the set up of external data files where the output parameters of the filter are stored. Also the time step of each iteration and the number of iteration to be simulated is defined in this step.

In the second step, already included inside the iteration phase of the algorithm, the time frame of the iteration is defined based on the date and time and time step set previously (if it is the first iteration, the initial values are considered. This allows for the computation of the sun's position for that specific time frame, among other time dependent variables. Following that, the orbital elements are propagated forward according to the initial conditions.

On the third step, the instrument measurements are simulated. The process for the simulation of these measurements takes into account the true position, attitude and angular rates of the satellite and is completed with the addition of random white noise with a norm as defined in step one. The simulated measurements include gyroscope's data, electromagnetic field, GPS data and the Sun's position at the iteration time.

The forth step consists on the estimation of the orbital parameters. This takes into account the Keplerian prediction of the orbit and the current (noisy) GPS measurements. Since the satellite is not provided with a thrusting system, it is expected that the Keplerian prediction is a fairly accurate one. The algorithm then takes a weighted average (where more weight is given to the prediction) in order to best estimate the current orbital parameters.

The fifth step is already part of the Kalman filter algorithm that estimates the attitude and angular rates of the satellite. In this step, the propagation step of the Kalman filter, the angular rate and the attitude quaternion are propagated according to Equations 2.28 and 2.2, respectively. In Equation 2.2, $\mathbf{F}_k$ is given by Equation 2.34. Also the state covariance matrix, $\mathbf{P}_{k|k-1}$, is calculated according to Equation 2.3, where the noise covariance matrix, $\mathbf{Q}_k$, is given by

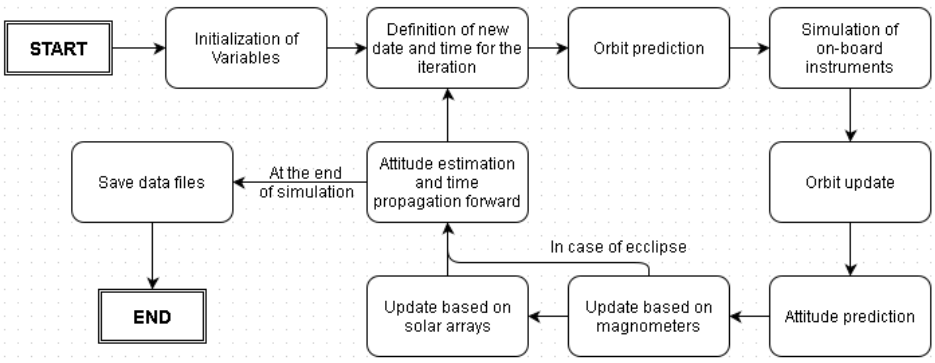

Figure 3.6: Flowchart of the Attitude Determination Algorithm

$$\mathbf{Q}_k = \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{12}^T & \mathbf{Q}_{22} \end{bmatrix} \tag{3.21}$$

with

$$\mathbf{Q}_{11} = \sigma_r^2 \cdot I_{3\times3} + \sigma_w^2 \cdot \left(I_{3\times3}\frac{\Delta t^3}{3} + \frac{\frac{(|\hat{\omega}|\Delta t)^3}{3} + 2\sin(|\hat{\omega}|\Delta t) - 2|\hat{\omega}|\Delta t}{|\hat{\omega}|^5} \cdot [\hat{\omega}\times]^2 \right) \tag{3.22}$$

$$\mathbf{Q}_{12} = -\sigma_w \cdot \left(I_{3\times3}\frac{\Delta t^2}{2} - \frac{|\hat{\omega}|\Delta t - \sin(|\hat{\omega}|\Delta t)}{|\hat{\omega}|^3} \cdot [\hat{\omega}\times] + \frac{\frac{(|\hat{\omega}|\Delta t)^2}{2} + \cos(|\hat{\omega}|\Delta t) - 1}{|\hat{\omega}|^4} \cdot [\hat{\omega}\times]^2 \right) \tag{3.23}$$

$$\mathbf{Q}_{22} = \sigma_w^2 \Delta t \cdot I_{3\times3} \tag{3.24}$$

The sixth step corresponds to update step of the Kalman filter. Within it, the measurement matrix, $\mathbf{H}_k$ is extracted from the simulated measurements followed by the computation of the Kalman gain matrix, $\mathbf{K}_k$, according to Equation 2.21. The state vector of the 6 state kalman filter can then be obtained from Equation 2.39. Finally, the attitude quaternion and the angular rate (*i.e.* the true state vector) can than be updated according to Equations (2.23),(2.24),(2.31) and (2.33).

Note that, in this filter, two different update steps exist (corresponding to steps six and seven of the algorithm). The first accounts for the sun position measurements while the second takes into account magnetometer measurements. This method allows us to consider two different sets of measurements in the same iteration. Note also that the satellite's designed orbit will contain moments when the satellite is eclipsed by the Earth and, thus, there will be occasions with no measurements of the Sun's position available. This situation was also accounted for in the simulation, leading to use of only magnetometer measurements for 27% of the satellite's orbit. Since it is a Sun synchronous orbit, we can consider this value to be constant for simulation porpuses.

The eighth and final step contained in the iteration process corresponds simply to the setting of the Kalman filter initial variables according to the final estimation of the iteration. This will be necessary to start the iteration process once more.

Finally, after all the iterations defined in the first step are completed, the algorithm stores all the data

obtained for later analysis. These files are later analysed using the *Matlab* software (see Chapter 5 for the results).

Chapter 4

ECOSat Attitude Control System (ESACS)

4.1 Magnetorquers equations

Magnetorquers are, in essence, a group of three solenoids (or more, if the satellite design requires some degree of redundancy) disposed in different orthogonal direction. These solenoids, when having a current I passing through them, create a magnetic field of intensity

$$B = \mu_0 \frac{NI}{2r}, \tag{4.1}$$

being μ_0 the constant magnetic permeability in vacuum, *N* the number of loops of the solenoid and *r* the radius of each loop.
A magnetic moment m will also be created

$$m = NI\pi r^2 \tag{4.2}$$

The magnetic field created by the solenoid will, then, interact with the Earth's magnetic field inducing a torque given by the cross product

$$\tau = m \times B_e \tag{4.3}$$

where B_e represents the Earth's local magnetic field.
Using Equation 4.2 and the intensity of the Earth's magnetic field, the different parameters of the magnetorquers can be adjusted according to the designed torque requirements. [54, 39]

4.2 Types of Control Algorithms

In systems control there are two basic types of controllers: feedback and feed-forward. A feedback controller measures the variables that are being controlled and performs the necessary adjustments to the

system in order to obtain the desired values of these variables. This type of controllers has a delayed response as there needs to be a deviation of the controlled variable and consequently, a brief moment when the value of the variable(s) is not the desired one. On the other hand, the feed-forward controller can have a much faster response as it accounts for disturbances before these take effect on the system. It is, so to say, a preventive controller. A good example of this type of controller is a house's thermostat that takes into account the fact that a door or a window is open. The controller then predicts the eminent decrease in temperature and makes the necessary adjustments to avoid it.
Within the feedback type of control, there are a multitude of different controllers. There simple controllers such as the on-off controller or the proportional controller, where the controller output is directly proportional to the error in the measured variable. However, a more complex and more accurate variation of the proportional controller is the well-known and widely used proportional-integral-derivative (PID) controller. As its name indicates, the PID controller involves three separate terms: the proportional term, the integral term and the derivative term. This type of controller has historically been used more than other controllers and has been generally accepted as the best performance controller for the majority of applications. [55] It is also characterised by low computational loads which makes it ideal for the purpose of this project.

4.3 Normal mission attitude control algorithm

As mentioned in the previous section, a PID controller was selected for the task of attitude control of this project. The PID controller, with the three terms (proportional, integral and derivative) is the most commonly used controller. However, for some applications only one or two terms are required to provide an appropriate control of the system. These are designated PI, PD, P, or I controllers, according to the terms used.
From a general point of view, a PID controller, with all it's terms, is given by:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{d}{dt} e(t) \tag{4.4}$$

where $u(t)$ represents the controller's output and $e(t)$ the controller's input, i.e., the variable's error in relation to the reference value. K_p, K_i and K_d represent the proportional, integral and derivative gains, respectively. These gains need to be adjusted as incorrect values can lead to poor controller performance or even instability of the controller. A series tests yielded the values presented in Table 4.1. These results allow us to conclude the better performance of the PD for this application in particular. The exact values obtain for its parameters are presented in Table 4.2.

For the case of a satellite's attitude control based on quaternions, Equation 4.4 takes the form of

$$T(t) = K_p q_e(t) + K_i \int_0^t q_e(\tau)\, d\tau + K_d \omega_e(t) \tag{4.5}$$

Table 4.1: Convergence times for both static and non static reference attitudes of different variations of the PID controller.

Type of controller	Time of convergence (min)	
	Static reference	Non static reference
PID	174	231
PI	342	387
PD	167	198
P	153	507
D	–	–
I	–	–

Table 4.2: Best values obtained for the gains of the normal mission controller.

Gains	Best Value Found
K_p	0.00015
K_d	0.2
K_i	–

where q_e represents the error in the attitude quaternion in relation to the reference attitude quaternion and ω_e, the error in the angular rates (also in relation to the reference values).

During these tests of Table 4.1, it was found that controller performed best with only the proportional and derivative terms. The use of the integral term not only led to a higher settling time, but also implied higher computational loads. The controller was, thus, reduced to a PD controller defined by:

$$\begin{aligned} T(t) &= K_p q_e(t) + K_d \omega_e(t) \\ &= 0.00015 q_e(t) + 0.2 \omega_e(t) \end{aligned} \tag{4.6}$$

Equation 4.6 yields the necessary torque for correcting the attitude given the reference and the real values of the attitude variables. However, it does not account for limitations of the actuation system (the magnetorquers). For the sake of correctness, the simulation software takes into account the local magnetic field and the maximum magnetic moment of each coil (established, from the hardware design, as $1A.m^2$) in order to calculate the maximum achievable torque at each time. This leads to a more accurate simulation of the real mission conditions.

4.4 De-tumbling algorithm

During the initial tumbling phase of the mission, correct attitude estimations cannot be made since GPS data (and, consequently, correct magnetometer and solar array data) will be unavailable - due to the initial incorrect pointing of the GPS antenna. The only available data corresponds to the angular rate measurements from the gyroscopes.

Due to the lack of attitude information, the de-tumbling algorithm needs to be very measurement independent. One commonly used option is the B-dot controller [56]. This controller only makes use of

Table 4.3: Best values obtained for the gains of the de-tumbling controller.

Gains	Best Value Found
K_p	-0.005
K_d	$-$
K_i	$-$

the derivative of the magnetic field (and not the locally dependent magnetic field measurements themselves), so it can be used even without GPS data. However, an even simpler and faster control strategy is to take the momentary angular rates measured by the gyroscopes and apply an inversely proportional controller to them. A series of simulations was performed to obtain the best value for the gain.

Note that, like for the normal mission controller, the simulation algorithm verifies at all times that the torques being simulated never surpass the imposed physical limit of the hardware.

The de-tumbling controller is used in the initial phase, when the unpredictable and elevated angular rates make it impossible to obtain GPS data. However, once the angular rates are low enough (the threshold was defined as a total angular rate of $0.001 rad/s$), the control algorithm will change to the normal mission "mode", where all attitude information is available and more accurate control can be performed. In the following chapter the performance of the algorithm in the tasks of attitude determination and control is analysed.

Chapter 5

Results

In this chapter the developed algorithm was tested to assess its performance. Tests were made to its attitude determination and attitude control capabilities separately.

5.1 Attitude Determination Performance

The performance in the determination of attitude of the satellite was tested first. Different attitudes and initial conditions were simulated and their result analysed. The simulations were divided into tests with a static attitude quaternion, *i.e.*, with null angular rates and dynamic attitude quaternions (with constant non-null angular rates).

5.1.1 Static Attitude Determination

In order to initially test the attitude determination filter's stability a simulation of with a real fixed attitude quaternion $q_{real} = [1, 0, 0, 0]^T$ was performed. The initial conditions of the filter were set to the real ones, *i.e.*, $q_k = q_{real} = [1, 0, 0, 0]^T$ and $\omega_k = [0, 0, 0]^T$ [1]. The results of the filter for this simulation are presented in Figures 5.1 and 5.2. An initial spike in the angular rates is visible in the angular rate (Figure 5.2). This lead to a deviation of the estimated attitude quaternion from the the real quaternion (Figure 5.2). These initial deviations can be explained by the initial uncertainty of the filter (high values of the covariance matrix). As the simulation continued, the certainty of the estimations increased and the filter quickly converged to the real values. Note that the main characteristic we wanted to demonstrate, *i.e.* the stability of the filter for static values was proven.

In this first simulation, the initial values of the filter were set to coincide with the real values of the satellite's attitude. This is, however, a very unlikely situation as the initial conditions of the satellite upon the beginning of its mission are unpredictable (due to the uncertainties of the initial tumbling rates). The subsequent simulation defined the initial conditions of the filter intentionally different from the real values. In it, the same real values from the previous test were used but the initial conditions were now set to

[1] All angular rates are given in SI units (rad/s).

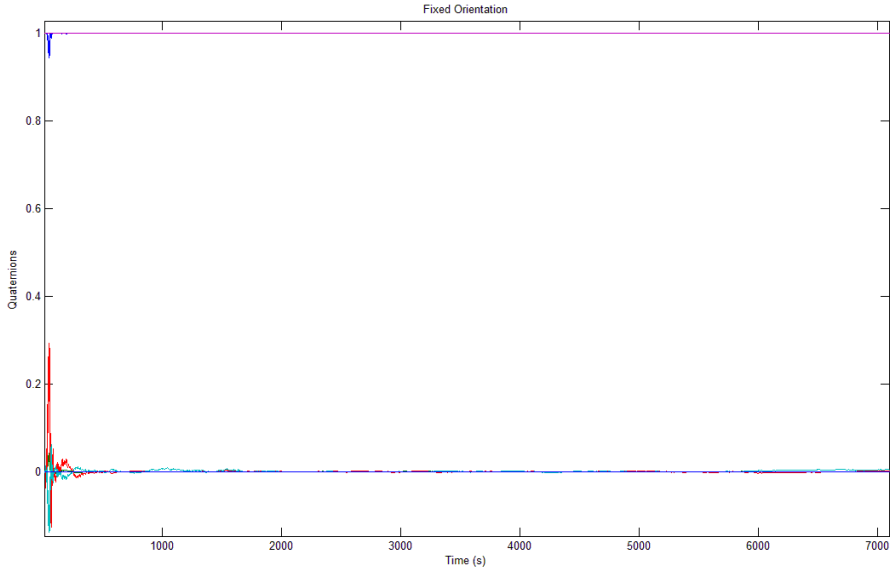

Figure 5.1: Behavior of the estimated attitude quaternion for initial conditions $q_k = [1, 0, 0, 0]^T$ and $\omega_k = [0, 0, 0]^T$ and real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0, 0, 0]^T$. Colors are defined as $q_k = [dark - blue, red, green, teal]^T$ and $q_{real} = [purple, black, yellow, dark - blue]^T$.

$q_k = [0, 1, 0, 0]^T$ and $\omega_k = [0, 0, 0]^T$. Note the difference in the initial values for the attitude quaternion. This represents an initial error of the filter in the attitude direction of $180\,\mathrm{deg}$. The behavior of the filter for this simulation is represented in Figures 5.3 and 5.4. Note that the filter converged quickly to the correct values even though it was set initially to the wrong values. It is true that it takes longer than in the previous test. But this was already expected given the initial error. Note also that the convergence time is approximately $1800s$ $(30min)$ which represents a very small value when compared to the expected mission time.

5.1.2 Dynamic Attitude Determination

The previous simulations demonstrated the filter's capability of correct attitude determination for a constant attitude quaternion, even given an initial error of its values. However, for the purposes of this mission, a constant attitude quaternion was not desirable. In the following simulation a non-null angular rate was considered. The initial real conditions of the simulation were $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0.01, 0, 0]^T$. The real attitude quaternion was no longer constant as described by the quaternion dynamics (Section 1.4.4). The initial conditions of the filter were set, in this simulation, to the exact same values of the real. Figures 5.5 and 5.6 clearly show that the filter presents no problem given these conditions. It has an initial deviation in both the attitude quaternion and angular rates but quickly converges again to the correct values. It's also visible that the filter follows the variations of the attitude quaternion perfectly after converging.

In the following test the real conditions of the simulation were maintained, *i.e.* $q_{real} = [1, 0, 0, 0]^T$

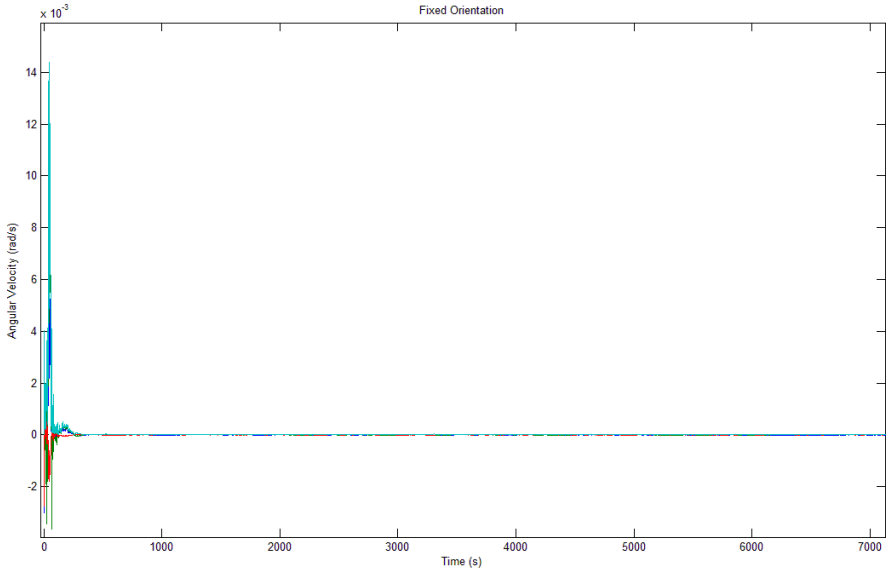

Figure 5.2: Behavior of the estimated angular rates for initial conditions $q_k = [1, 0, 0, 0]^T$ and $\omega_k = [0, 0, 0]^T$ and real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0, 0, 0]^T$. Colors are defined as $\omega_k = [dark - blue, red, green]^T$ and $|\omega|$ in teal.

and $\omega_{real} = [0.01, 0, 0]^T$. However, an error in the initial conditions of the filter was induced. The initial attitude quaternion was set to $q_k = [0, 0, 1, 0]^T$. As shown in Figures 5.7 and 5.8, the filter takes longer to converge to the correct values, as it was expected. But it does converge after approximately $1500s$. The simulation that followed induced an error also in the initial angular rate, in addition to the previous error in the attitude quaternion. An initial angular rate, $\omega_k = [0, 0, 0.5]^T$ was defined. Figures 5.9 and 5.10 represent the filter's response to this situation. It is clear an initial instability of the filter, quickly changing it's estimations. This is explained by the relatively high and inaccurate angular rates (approximately a 500 times higher than the expected mission angular rates). But even given this initial instability and inaccuracy, the filter converged quickly and followed the real values correctly afterwards.

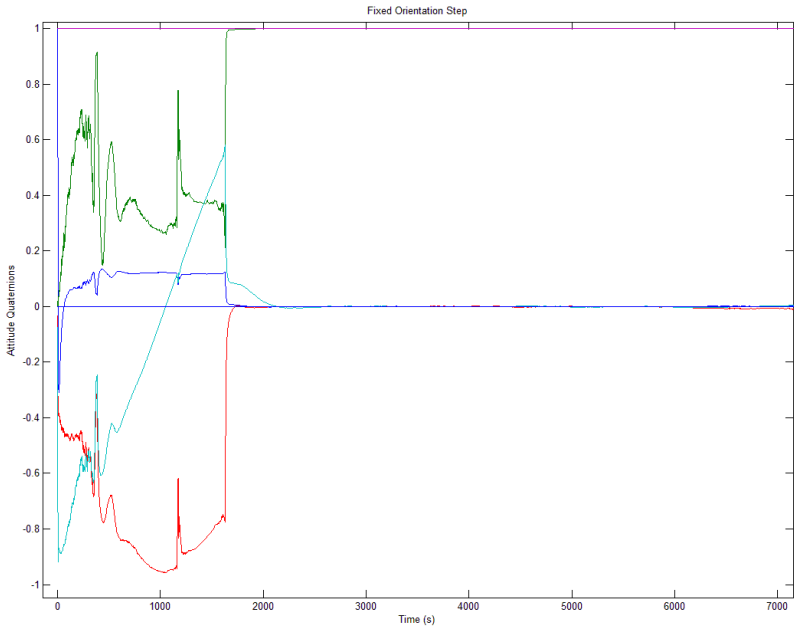

Figure 5.3: Behavior of the estimated attitude quaternion for initial conditions $q_k = [0, 1, 0, 0]^T$ and $\omega_k = [0, 0, 0]^T$ and real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0, 0, 0]^T$. Colors are defined as $q_k = [gree, dark - blue, red, teal]^T$ and $q_{real} = [purple, black, yellow, dark - blue]^T$.

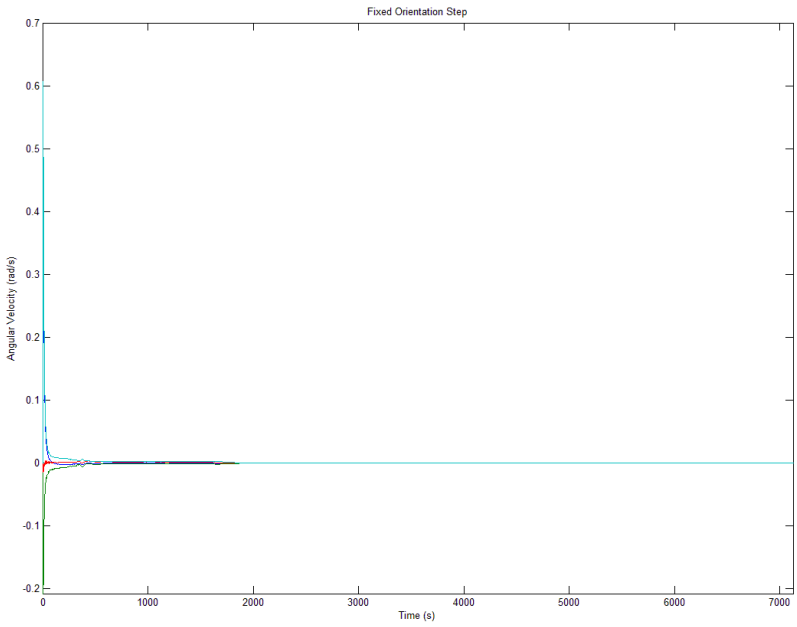

Figure 5.4: Behavior of the estimated angular rates for initial conditions $q_k = [0, 1, 0, 0]^T$ and $\omega_k = [0, 0, 0]^T$ and real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0, 0, 0]^T$. Colors are defined as $\omega_k = [dark - blue, red, green]^T$ and $|\omega|$ in teal.

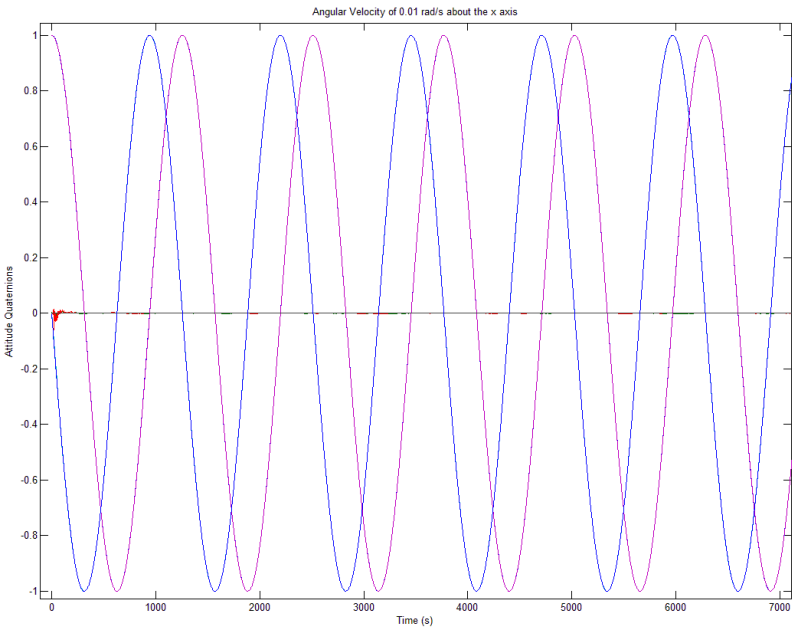

Figure 5.5: Behavior of the estimated attitude quaternion for initial conditions $q_k = [1, 0, 0, 0]^T$ and $\omega_k = [0.01, 0, 0]^T$ and real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0.01, 0, 0]^T$. Colors are defined as $q_k = [dark - blue, red, green, teal]^T$ and $q_{real} = [purple, black, yellow, dark - blue]^T$.

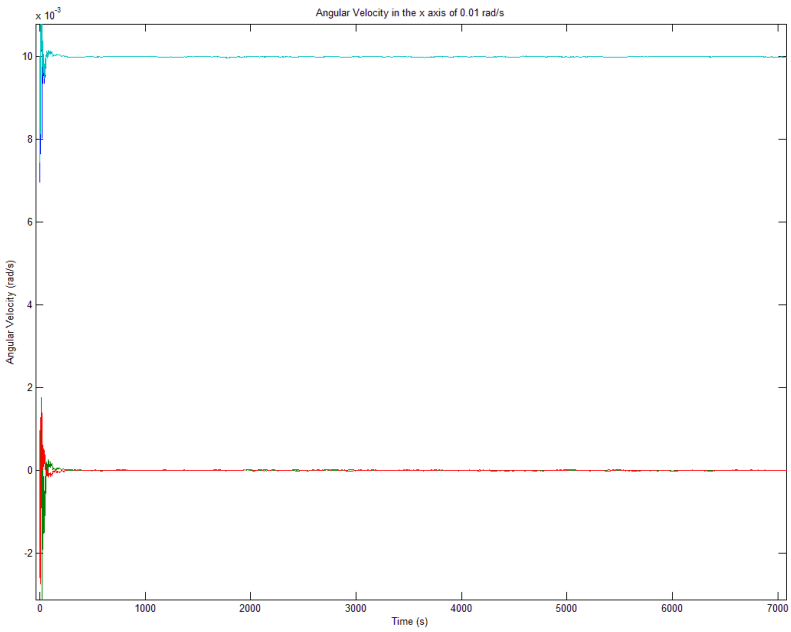

Figure 5.6: Behavior of the estimated angular rates for initial conditions $q_k = [1, 0, 0, 0]^T$ and $\omega_k = [0.01, 0, 0]^T$ and real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0.01, 0, 0]^T$. Colors are defined as $\omega_k = [dark - blue, red, green]^T$ and $|\omega|$ in teal.

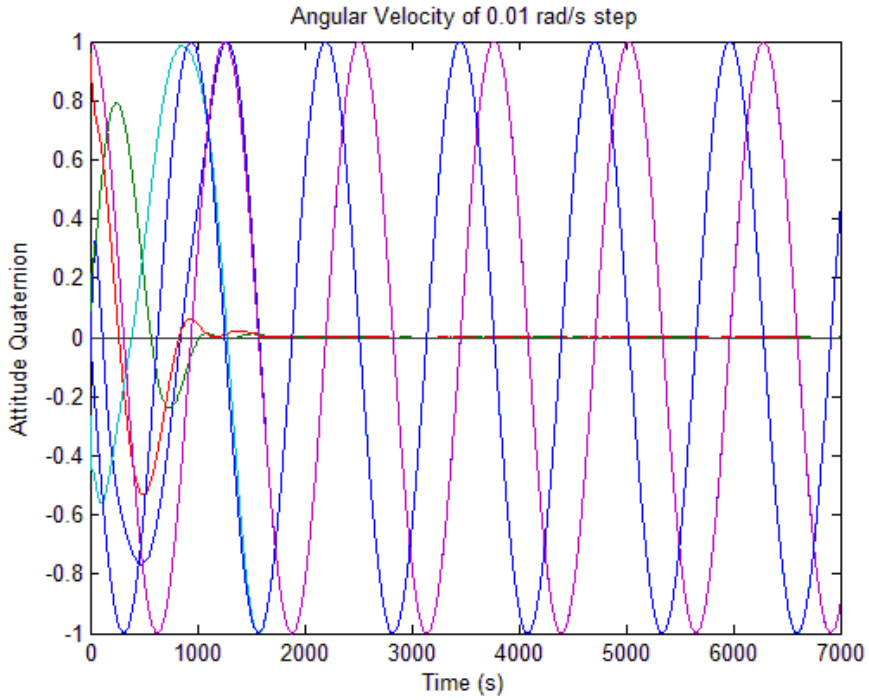

Figure 5.7: Behavior of the estimated attitude quaternion for initial conditions $q_k = [0, 0, 1, 0]^T$ and $\omega_k = [0.01, 0, 0]^T$ and real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0.01, 0, 0]^T$. Colors are defined as $q_k = [dark - blue, red, green, teal]^T$ and $q_{real} = [purple, black, yellow, dark - blue]^T$.

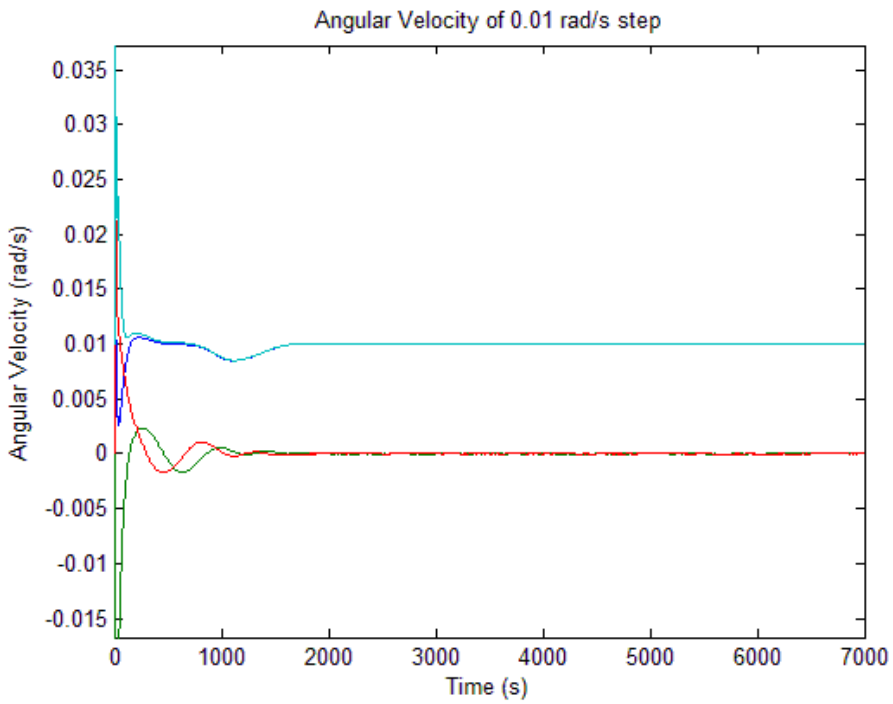

Figure 5.8: Behavior of the estimated angular rates for initial conditions $q_k = [0, 0, 1, 0]^T$ and $\omega_k = [0.01, 0, 0]^T$ and real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0.01, 0, 0]^T$. Colors are defined as $\omega_k = [dark - blue, red, green]^T$ and $|\omega|$ in teal.

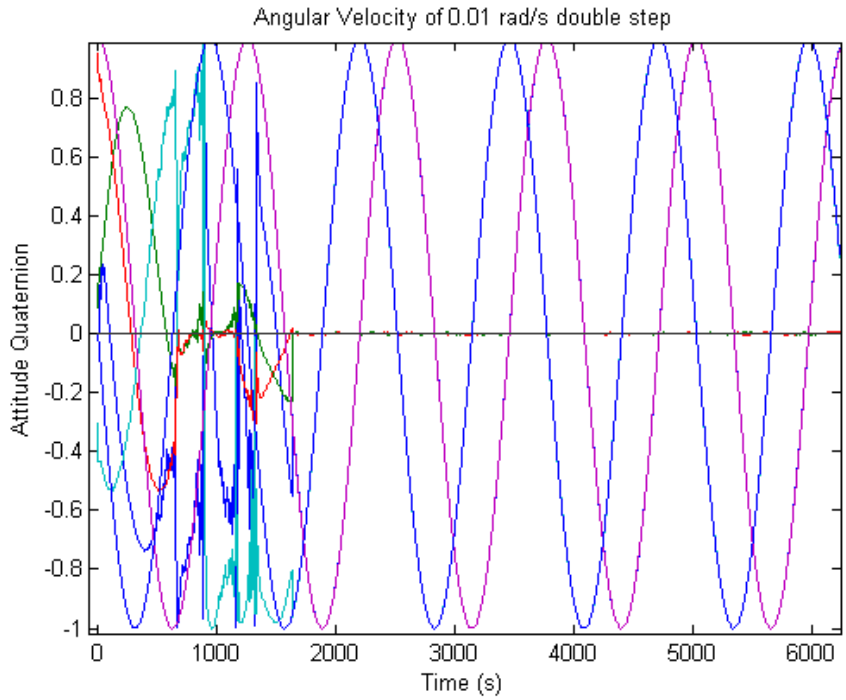

Figure 5.9: Behavior of the estimated attitude quaternion for initial conditions $q_k = [0, 0, 1, 0]^T$ and $\omega_k = [0, 0, 0.5]^T$ and real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0.01, 0, 0]^T$. Colors are defined as $q_k = [dark - blue, red, green, teal]^T$ and $q_{real} = [purple, black, yellow, dark - blue]^T$.

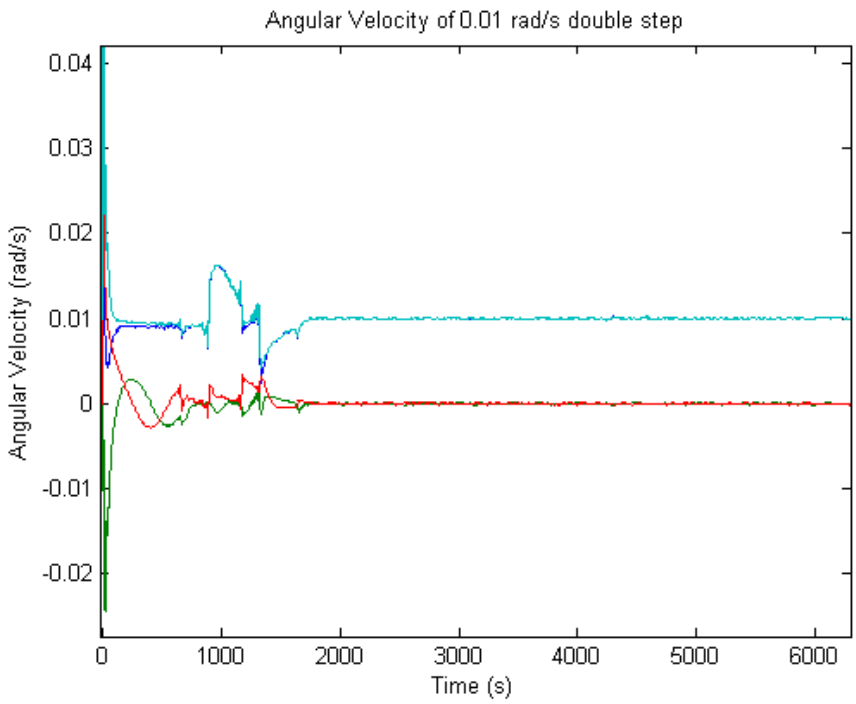

Figure 5.10: Behavior of the estimated angular rates for initial conditions $q_k = [0, 1, 0, 0]^T$ and $\omega_k = [0, 0, 0]^T$ and real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0, 0, 0]^T$. Colors are defined as $\omega_k = [dark - blue, red, green]^T$ and $|\omega|$ in teal.

5.2 Attitude Control Performance

In the previous section, the performance of the algorithm for the task of attitude determination was demonstrated for a multitude of scenarios. This involved tests for different attitude situations and with different induced errors. Once demonstrated the attitude determination capacities of the filter, the attitude control part of the algorithm was fully tested.

5.2.1 Static Reference

Initially, a simple simulation of the control's algorithm performance for a fixed reference (*i.e.*, desired) attitude quaternion was performed. The filter was set to an initial angular rate of $\omega_k = [0.01, 0, 0]^T$. The reference conditions were $q_{ref} = [1, 0, 0, 0]^T$ and $\omega_{ref} = [0.01, 0, 0]^T$. Figures 5.11, 5.12 and 5.13 correspond to the case of correct initial values, *i.e.*, no initial error. Figure 5.13 represents the torques being applied to the system by the control system. These are always limited according to the hardware's physical limitations and are a function of the magnetorquers' maximum power and the external electro magnetic field, which depends on the position in orbit. Figures 5.14, 5.15 and 5.16 account for an uncertainty in the initial conditions ($q_k = [0, 1, 0, 0]^T$ and $q_{real} = [1, 0, 0, 0]^T$). Note the slower response of the control system for this case. This was expected since the filter first needs to converge to the correct value. Note also, that filter follows a fixed reference of $q_{ref} = [-1, 0, 0, 0]^T$ rather than $q_{ref} = [1, 0, 0, 0]^T$. This might seem like an error at first glance, but if we recall Section 1.4.3, we realize that $q = [-1, 0, 0, 0]^T$ and $q = [1, 0, 0, 0]^T$ correspond to the same exact attitude.

5.2.2 Nominal Mission

Once demonstrated the algorithm's control capability for a given constant reference attitude value, its performance while following the mission's attitude requirements was tested. This is no longer a fixed attitude but rather an attitude constantly changing and defined by constant angular rates.

The mission design defined a first attitude requirement of constantly pointing of the payload towards the Earth. Other attitude requirement refered the need to also rotate about the longitudinal axis (one full rotation per orbit). This requirement is due to optimized pointing needs of the solar panels estimated by the Power Subsystem team. But, on a first test, only the rotation of $\omega_{ref} = [0.001056, 0, 0]^T$ (corresponding to one rotation in the ECI frame per orbit) was considered. Figures 5.17 to 5.19 represent the performance of the control system for the initial angular rates of $\omega_{real} = [0.01, 0, 0]^T$.

The results show a quick and controlled convergence of the attitude towards the desired attitude. Extremely high angular rates were also considered and the results are presented in Figures 5.20 to 5.22. In that simulation initial angular rates were set to $\omega_{real} = [0.1, 0.1, 0.1]^T$. Although slower, the control system has no problem on bringing the satellite's attitude to the desired one. Note that these tests already account for an uncertainty in the initial attitude quaternion ($q_{real} = [1, 0, 0, 0]^T$ and $q_k = [0, 1, 0, 0]^T$).

Following this simplified nominal attitude. The rotation on the second axis was included, making the desired angular rates $\omega_{ref} = [0.0007467, 0.0007467, 0]^T$, maintaining the $|\omega_{ref}| = 0.001056$, corresponding

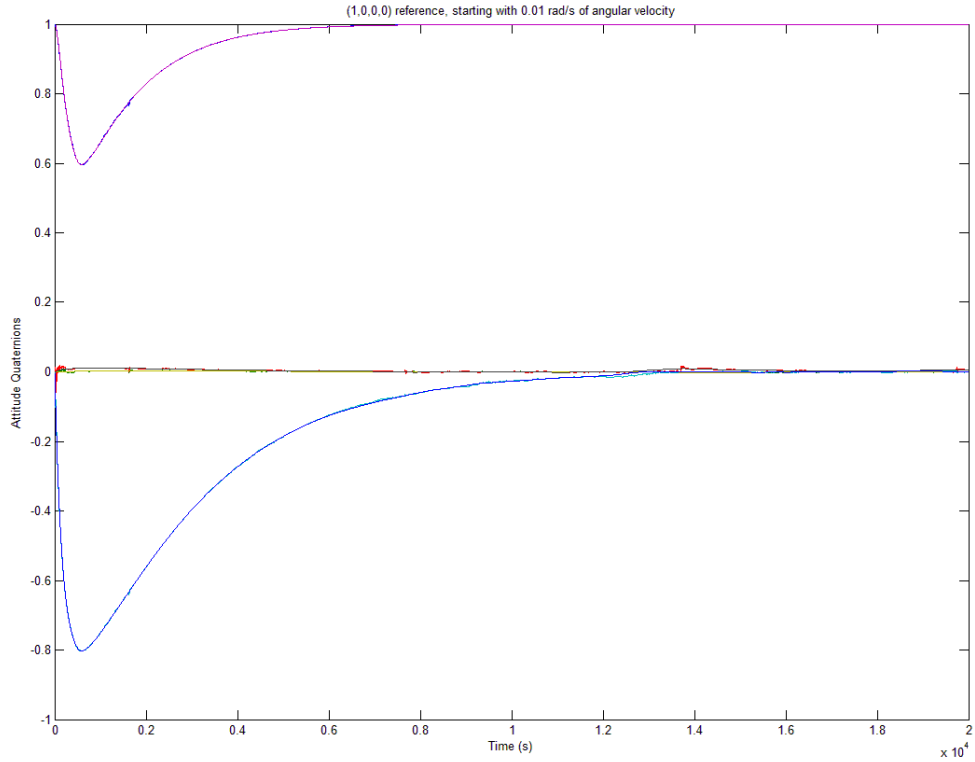

Figure 5.11: Behavior of the estimated attitude quaternion for initial conditions $q_k = [1, 0, 0, 0]^T$ and $\omega_k = [0.01, 0, 0]^T$, real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0.01, 0, 0]^T$ and reference conditons $q_{ref} = [1, 0, 0, 0]^T$ and $\omega_{ref} = [0, 0, 0]^T$. Colors are defined as $q_k = [dark - blue, red, green, teal]^T$ and $q_{real} = [purple, black, yellow, dark - blue]^T$. The reference is not represented.

to a full rotation per orbit about an intermediate axis. Figures 5.23 to 5.25 correspond to the performance of the control algorithm for this reference attitude with an initial angular rates of $\omega_{real} = [0.05, 0.05, 0.05]^T$. The following test, represented by Figures 5.26 to 5.28, corresponds to the same situation but accounts also for the initial uncertainty of the filter, *i.e.*, considers an initial attitude angular rates completely different than the real ones. This leads to an initial period of high uncertainty and high angular rates (approximately 3 hours). Afterwards, the attitude estimation filter converges to real values and the control algorithm is able to controllably bring the satellite's attitude to the desired one, within approximately 3 more hours.

This last simulation corresponds to the closest approach to the real mission's conditions: high tumbling angular rates in all directions and high uncertainty in the initial phase of the mission. The algorithm developed has demonstrated to be able to handle this situation and bring the satellite to it's nominal position within a total of 6 hours (21600 s) since the beginning of mission.

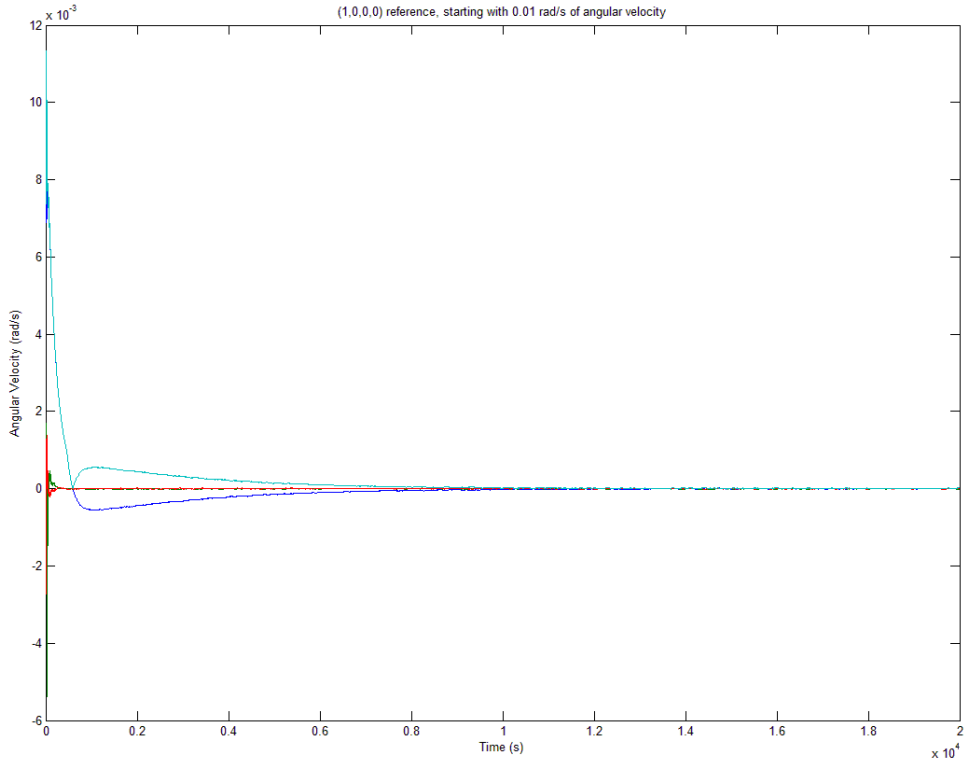

Figure 5.12: Behavior of the estimated angular rates for initial conditions $q_k = [1, 0, 0, 0]^T$ and $\omega_k = [0.01, 0, 0]^T$, real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0.01, 0, 0]^T$ and reference conditons $q_{ref} = [1, 0, 0, 0]^T$ and $\omega_{ref} = [0, 0, 0]^T$. Colors are defined as $\omega_k = [dark - blue, red, green]^T$ and $|\omega|$ in teal.

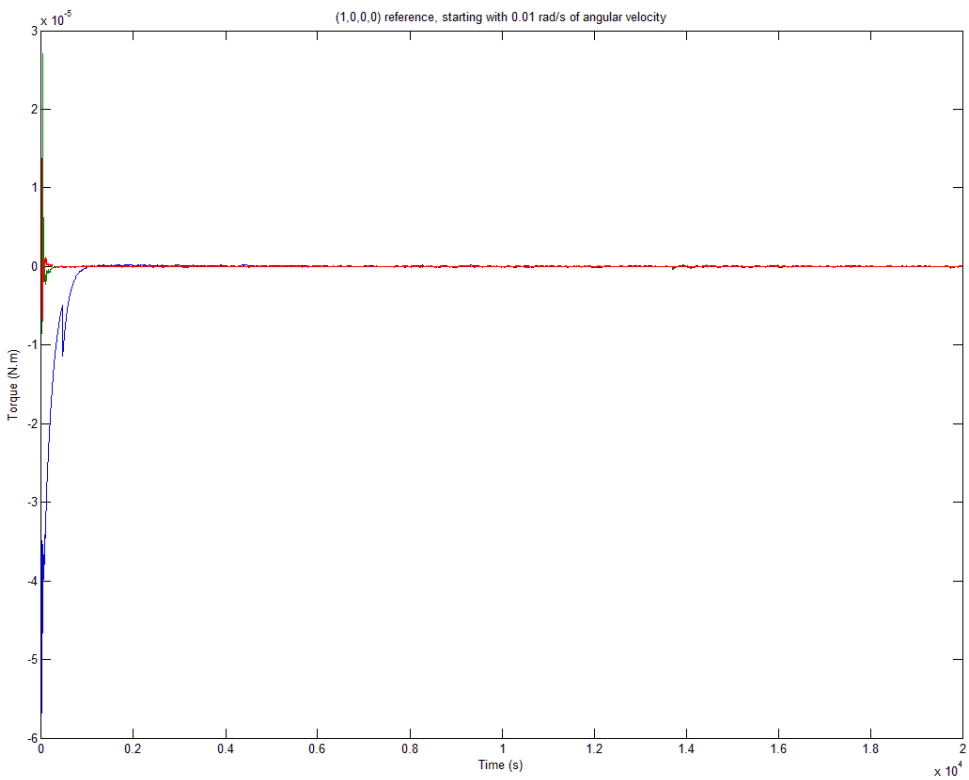

Figure 5.13: Torques applied by the control system for initial conditions $q_k = [1, 0, 0, 0]^T$ and $\omega_k = [0.01, 0, 0]^T$, real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0.01, 0, 0]^T$and reference conditons $q_{ref} = [1, 0, 0, 0]^T$ and $\omega_{ref} = [0, 0, 0]^T$. Colors are defined as $T_{(x,y,z)} = [dark - blue, red, green]^T$.

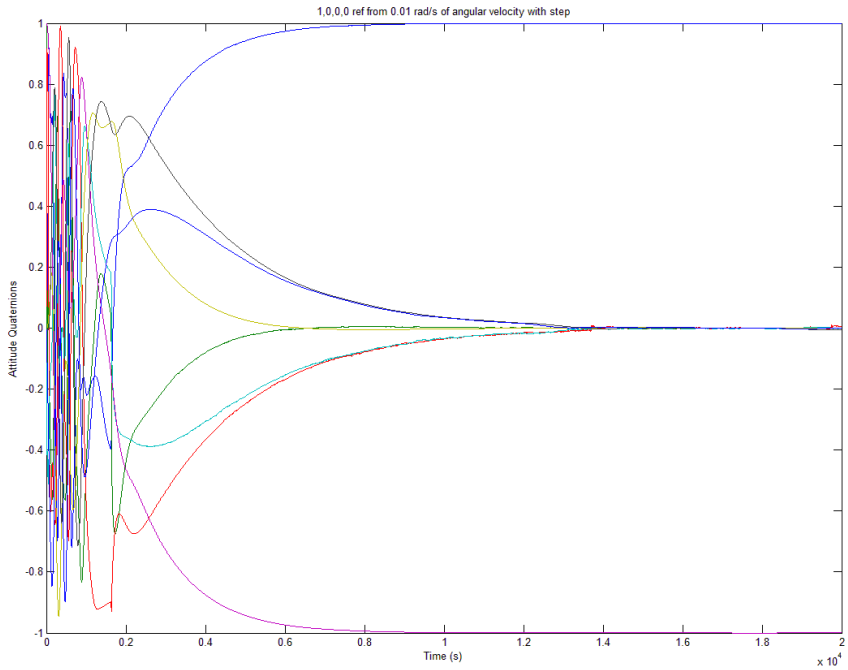

Figure 5.14: Behavior of the estimated attitude quaternion for initial conditions $q_k = [0, 1, 0, 0]^T$ and $\omega_k = [0.01, 0, 0]^T$, real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0.01, 0, 0]^T$ and reference conditons $q_{ref} = [1, 0, 0, 0]^T$ and $\omega_{ref} = [0, 0, 0]^T$. Colors are defined as $q_k = [dark - blue, red, green, teal]^T$ and $q_{real} = [purple, black, yellow, dark - blue]^T$. The reference is not represented.

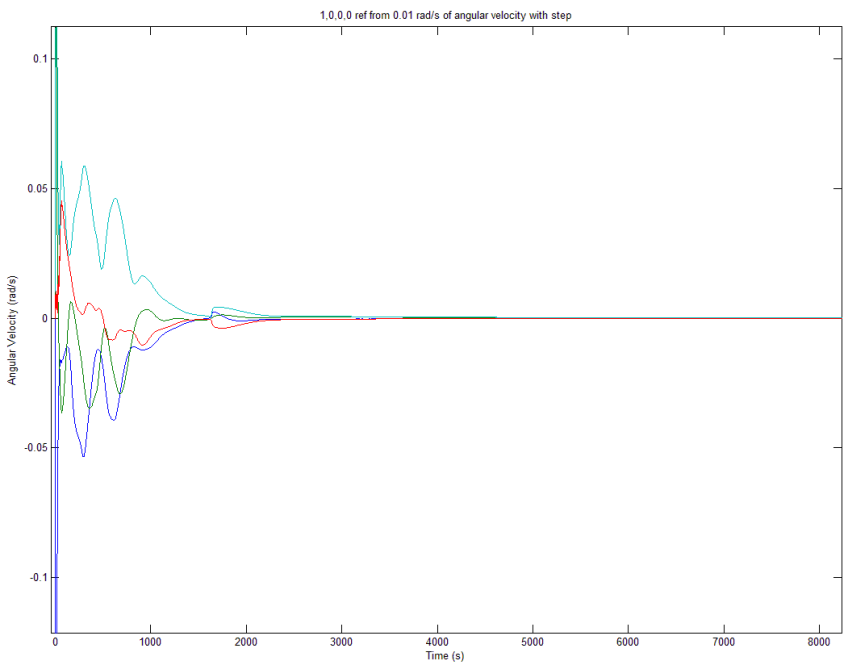

Figure 5.15: Behavior of the estimated angular rates for initial conditions $q_k = [0, 1, 0, 0]^T$ and $\omega_k = [0.01, 0, 0]^T$, real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0.01, 0, 0]^T$ and reference conditons $q_{ref} = [1, 0, 0, 0]^T$ and $\omega_{ref} = [0, 0, 0]^T$. Colors are defined as $\omega_k = [dark - blue, red, green]^T$ and $|\omega|$ in teal.

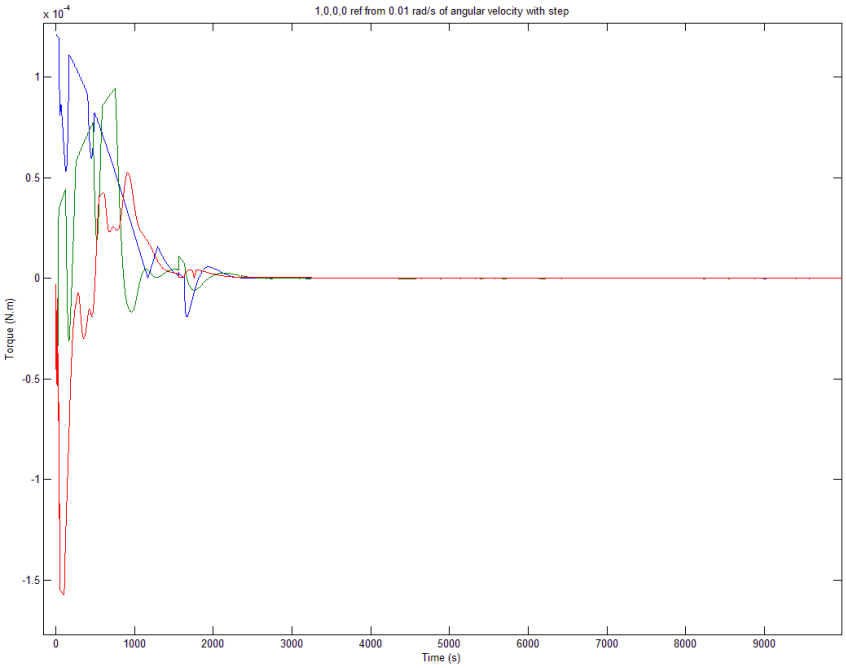

Figure 5.16: Torques applied by the control system for initial conditions $q_k = [0,1,0,0]^T$ and $\omega_k = [0.01,0,0]^T$, real initial conditions $q_{real} = [1,0,0,0]^T$ and $\omega_{real} = [0.01,0,0]^T$and reference conditons $q_{ref} = [1,0,0,0]^T$ and $\omega_{ref} = [0,0,0]^T$. Colors are defined as $T_{(x,y,z)} = [dark-blue, red, green]^T$.

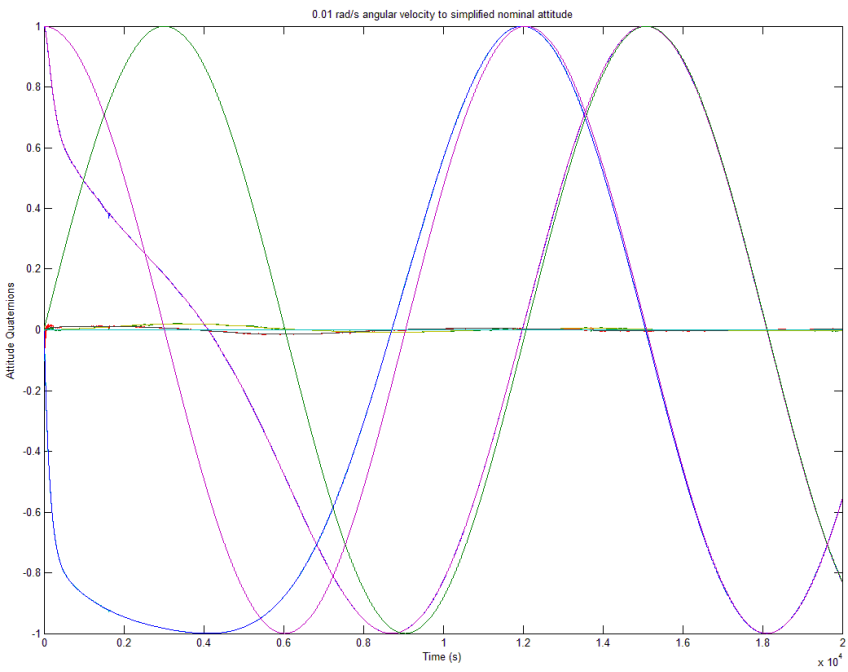

Figure 5.17: Behavior of the estimated attitude quaternion for initial conditions $q_k = [0,1,0,0]^T$ and $\omega_k = [0.01,0,0]^T$, real initial conditions $q_{real} = [1,0,0,0]^T$ and $\omega_{real} = [0.01,0,0]^T$ and reference conditons $q_{ref} = [1,0,0,0]^T$ and $\omega_{ref} = [0.001056,0,0]^T$. Colors are defined as $q_k = [dark-blue, red, green, teal]^T$, $q_{real} = [purple, black, yellow, dark-blue]^T$ and $q_{ref} = [purple, teal, red, green]^T$.

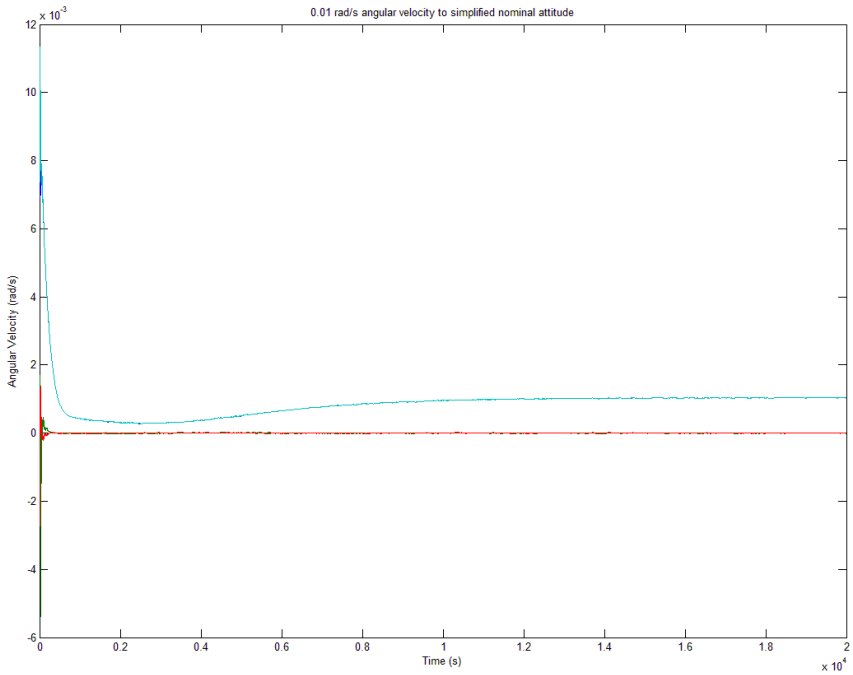

Figure 5.18: Behavior of the estimated angular rates for initial conditions $q_k = [0, 1, 0, 0]^T$ and $\omega_k = [0.01, 0, 0]^T$, real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0.01, 0, 0]^T$ and reference conditons $q_{ref} = [1, 0, 0, 0]^T$ and $\omega_{ref} = [0.001056, 0, 0]^T$. Colors are defined as $\omega_k = [dark - blue, red, green]^T$ and $|\omega|$ in teal.

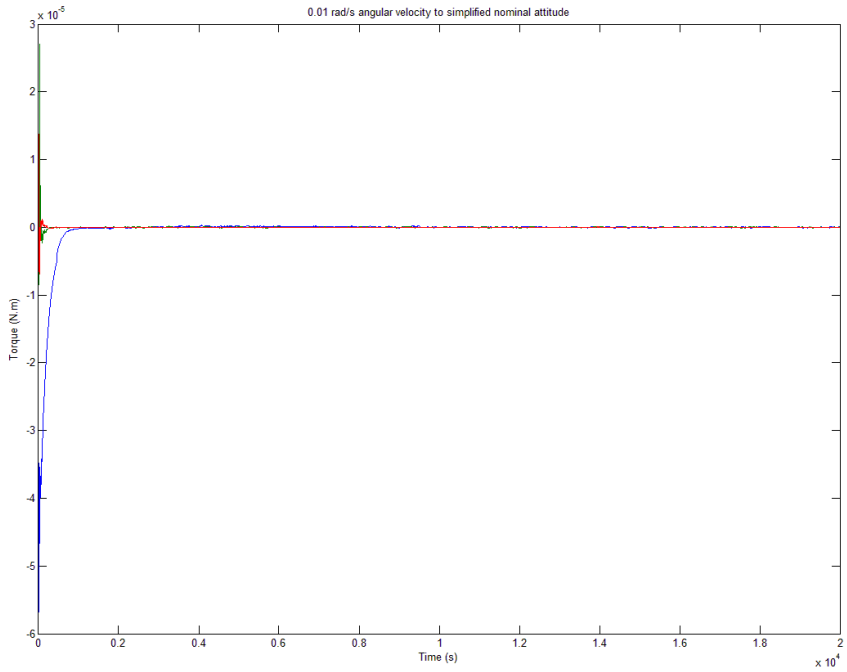

Figure 5.19: Torques applied by the control system for initial conditions $q_k = [0, 1, 0, 0]^T$ and $\omega_k = [0.01, 0, 0]^T$, real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0.01, 0, 0]^T$ and reference conditons $q_{ref} = [1, 0, 0, 0]^T$ and $\omega_{ref} = [0.001056, 0, 0]^T$. Colors are defined as $T_{(x,y,z)} = [dark - blue, red, green]^T$.

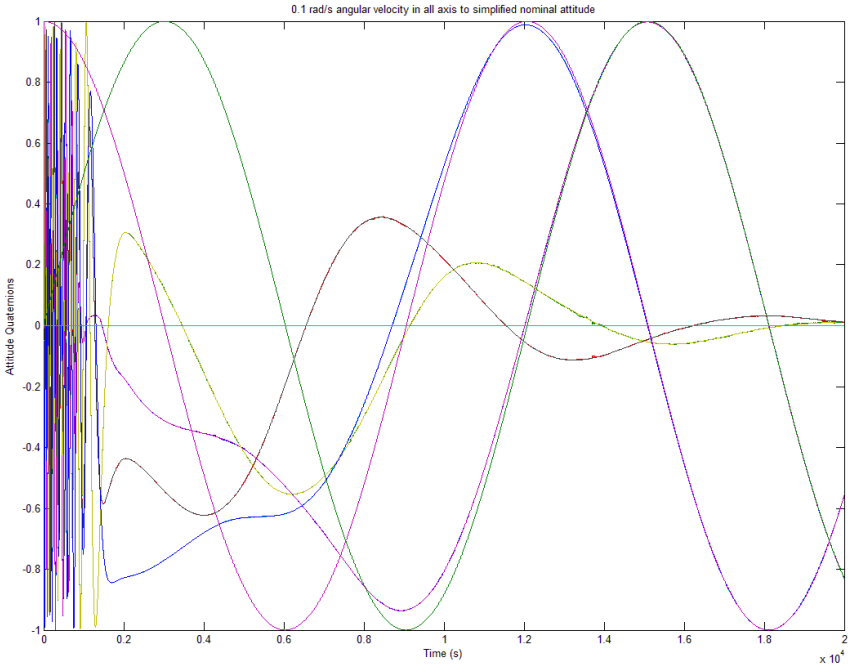

Figure 5.20: Behavior of the estimated attitude quaternion for initial conditions $q_k = [0,1,0,0]^T$ and $\omega_k = [0.1, 0.1, 0.1]^T$, real initial conditions $q_{real} = [1,0,0,0]^T$ and $\omega_{real} = [0.1, 0.1, 0.1]^T$ and reference conditons $q_{ref} = [1,0,0,0]^T$ and $\omega_{ref} = [0.001056, 0, 0]^T$. Colors are defined as $q_k = [dark-blue, red, green, teal]^T$, $q_{real} = [purple, black, yellow, dark-blue]^T$ and $q_{ref} = [purple, teal, red, green]^T$.

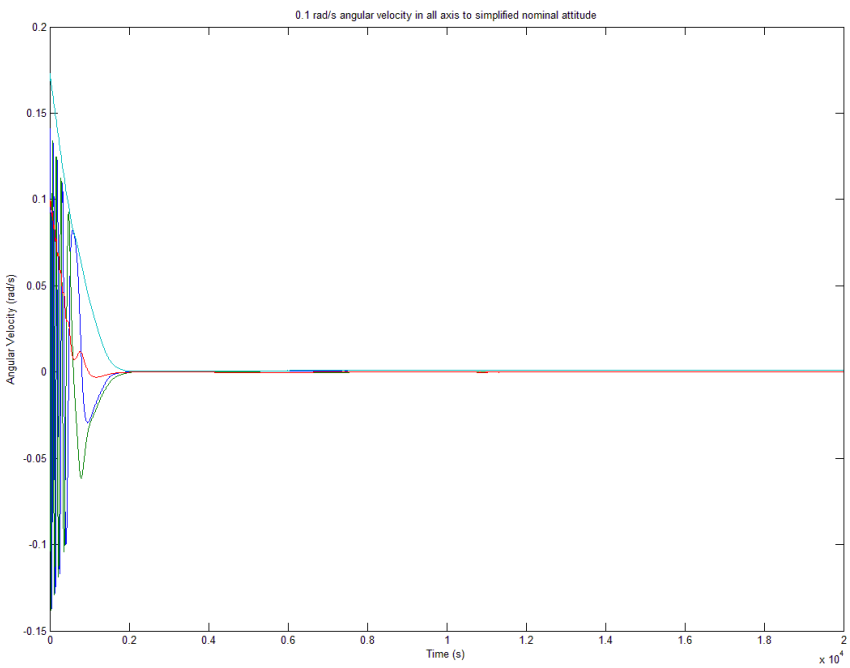

Figure 5.21: Behavior of the estimated angular rates for initial conditions $q_k = [0,1,0,0]^T$ and $\omega_k = [0.1, 0.1, 0.1]^T$, real initial conditions $q_{real} = [1,0,0,0]^T$ and $\omega_{real} = [0.1, 0.1, 0.1]^T$ and reference conditons $q_{ref} = [1,0,0,0]^T$ and $\omega_{ref} = [0.001056, 0, 0]^T$. Colors are defined as $\omega_k = [dark-blue, red, green]^T$ and $|\omega|$ in teal.

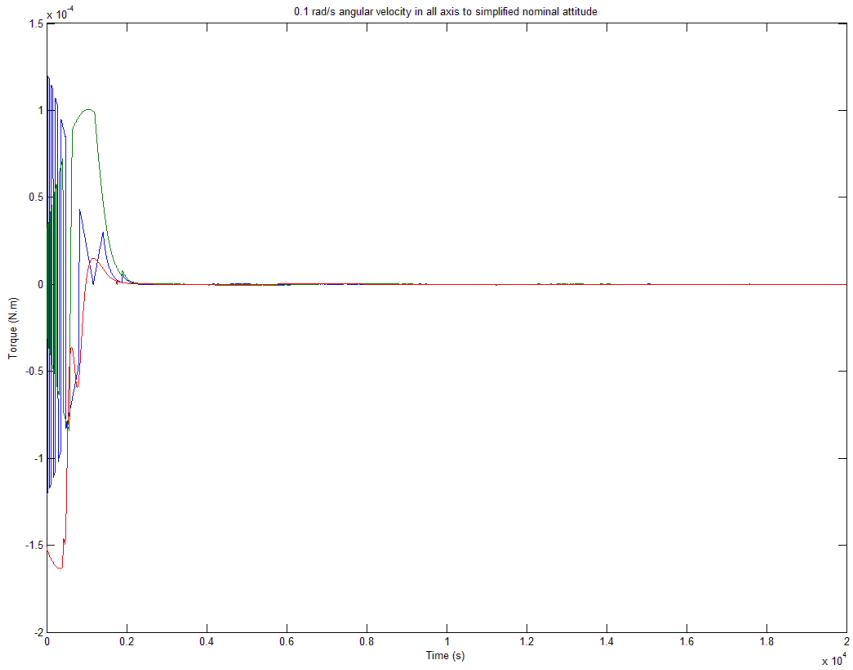

Figure 5.22: Torques applied by the control system for initial conditions $q_k = [0, 1, 0, 0]^T$ and $\omega_k = [0.1, 0.1, 0.1]^T$, real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0.1, 0.1, 0.1]^T$ and reference conditons $q_{ref} = [1, 0, 0, 0]^T$ and $\omega_{ref} = [0.001056, 0, 0]^T$. Colors are defined as $T_{(x,y,z)} = [dark - blue, red, green]^T$.

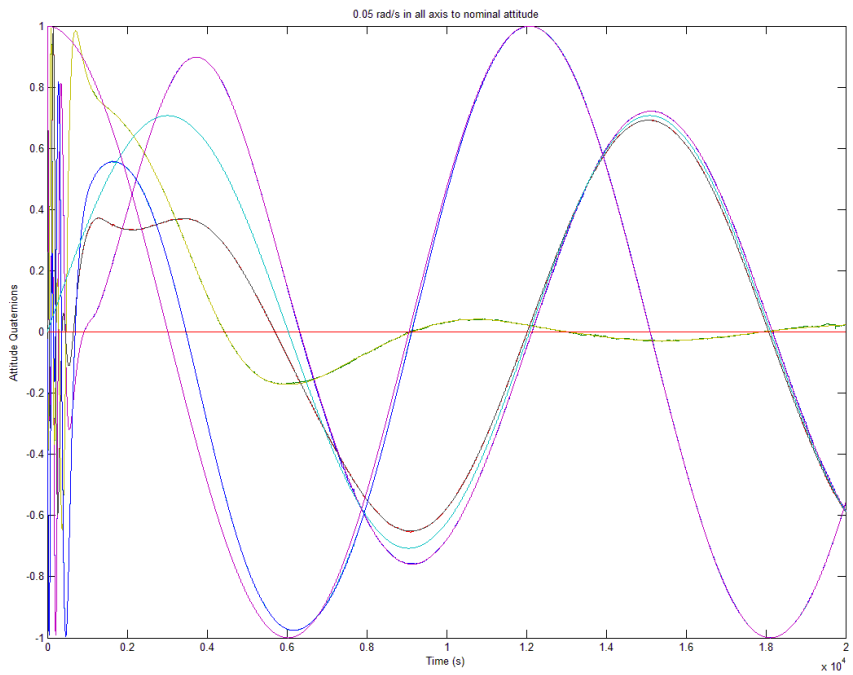

Figure 5.23: Behavior of the estimated attitude quaternion for initial conditions $q_k = [1, 0, 0, 0]^T$ and $\omega_k = [0.05, 0.05, 0.05]^T$, real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0.05, 0.05, 0.05]^T$ and reference conditons $q_{ref} = [1, 0, 0, 0]^T$ and $\omega_{ref} = [0.0007467, 0.0007467, 0]^T$. Colors are defined as $q_k = [dark - blue, red, green, teal]^T$, $q_{real} = [purple, black, yellow, dark - blue]^T$ and $q_{ref} = [purple, teal, red, green]^T$.

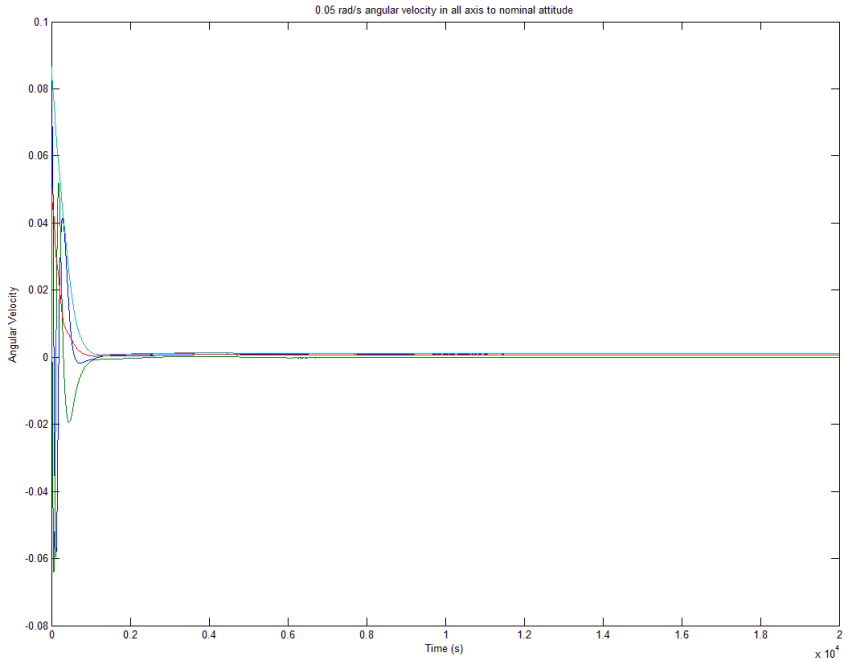

Figure 5.24: Behavior of the estimated angular rates for initial conditions $q_k = [1,0,0,0]^T$ and $\omega_k = [0.05, 0.05, 0.05]^T$, real initial conditions $q_{real} = [1,0,0,0]^T$ and $\omega_{real} = [0.05, 0.05, 0.05]^T$ and reference conditons $q_{ref} = [1,0,0,0]^T$ and $\omega_{ref} = [0.0007467, 0.0007467, 0]^T$. Colors are defined as $\omega_k = [dark - blue, red, green]^T$ and $|\omega|$ in teal.

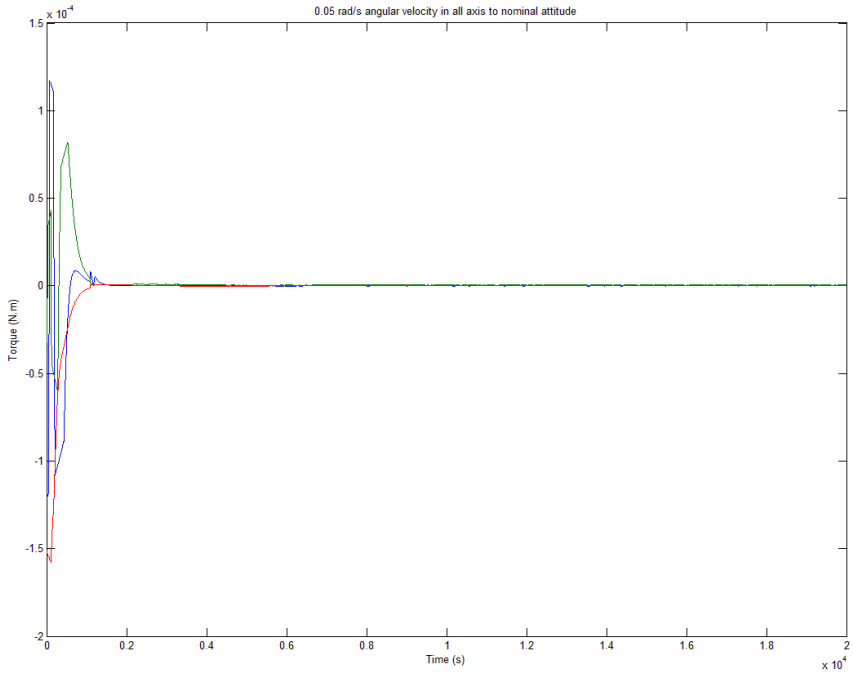

Figure 5.25: Torques applied by the control system for initial conditions $q_k = [1,0,0,0]^T$ and $\omega_k = [0.05, 0.05, 0.05]^T$, real initial conditions $q_{real} = [1,0,0,0]^T$ and $\omega_{real} = [0.05, 0.05, 0.05]^T$ and reference conditons $q_{ref} = [1,0,0,0]^T$ and $\omega_{ref} = [0.0007467, 0.0007467, 0]^T$. Colors are defined as $T_{(x,y,z)} = [dark - blue, red, green]^T$.

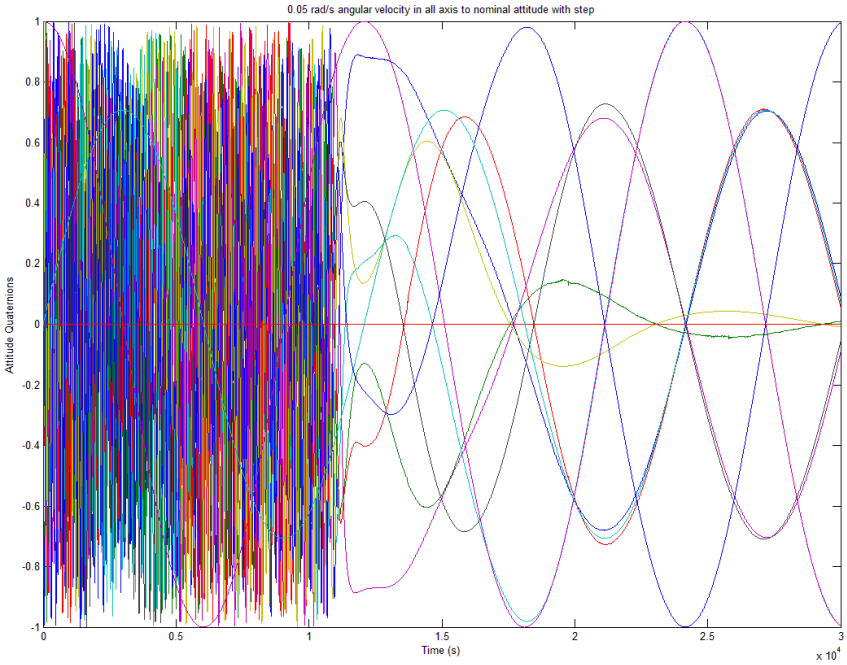

Figure 5.26: Behavior of the estimated attitude quaternion for initial conditions $q_k = [0,1,0,0]^T$ and $\omega_k = [0,0,0]^T$, real initial conditions $q_{real} = [1,0,0,0]^T$ and $\omega_{real} = [0.05, 0.05, 0.05]^T$ and reference conditons $q_{ref} = [1,0,0,0]^T$ and $\omega_{ref} = [0.0007467, 0.0007467, 0]^T$. Colors are defined as $q_k = [dark-blue, red, green, teal]^T$, $q_{real} = [purple, black, yellow, dark-blue]^T$ and $q_{ref} = [purple, teal, red, green]^T$.

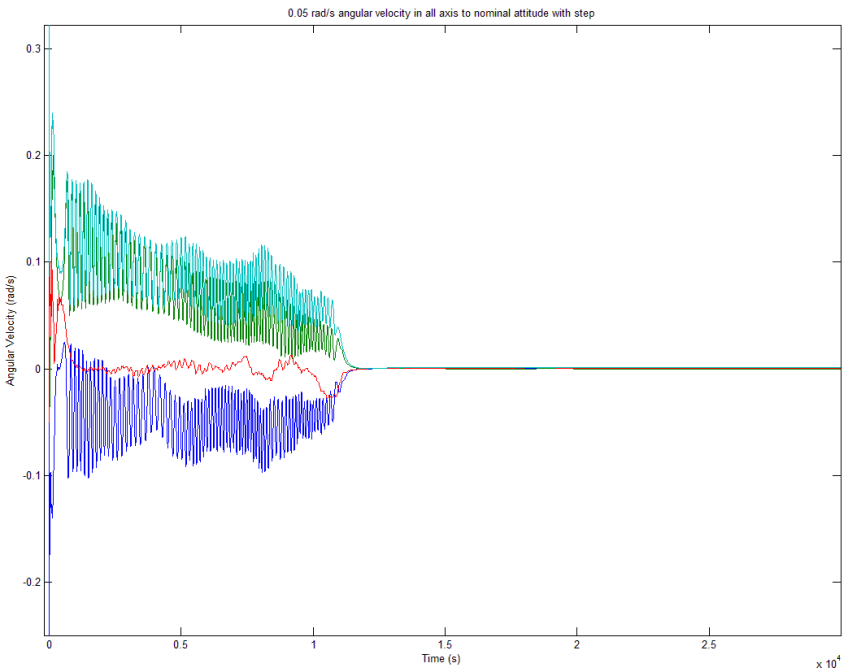

Figure 5.27: Behavior of the estimated angular rates for initial conditions $q_k = [0,1,0,0]^T$ and $\omega_k = [0,0,0]^T$, real initial conditions $q_{real} = [1,0,0,0]^T$ and $\omega_{real} = [0.05, 0.05, 0.05]^T$ and reference conditons $q_{ref} = [1,0,0,0]^T$ and $\omega_{ref} = [0.0007467, 0.0007467, 0]^T$. Colors are defined as $\omega_k = [dark-blue, red, green]^T$ and $|\omega|$ in teal.

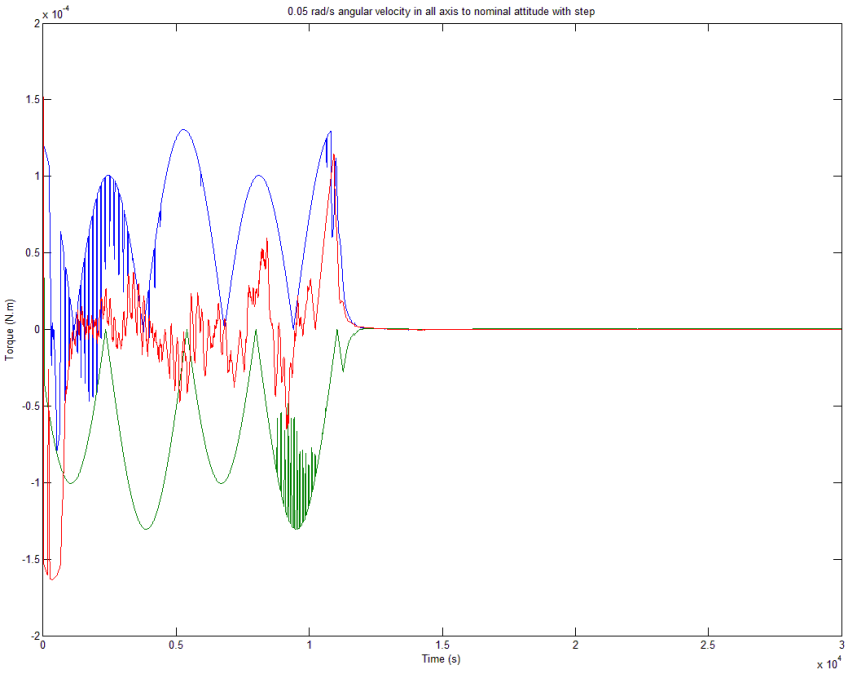

Figure 5.28: Torques applied by the control system for initial conditions $q_k = [0, 1, 0, 0]^T$ and $\omega_k = [0, 0, 0]^T$, real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0.05, 0.05, 0.05]^T$ and reference conditons $q_{ref} = [1, 0, 0, 0]^T$ and $\omega_{ref} = [0.0007467, 0.0007467, 0]^T$. Colors are defined as $T_{(x,y,z)} = [dark - blue, red, green]^T$.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

A fully functional Attitude Determination and Control System (ADCS) software was developed and tested. For this, notions of quaternion's operations and attitude representation, different coordinate systems and Kalman filtering were introduced.

The software was developed in two different steps. Firstly the algorithm for the correct attitude determination of the attitude was designed. This algorithm makes use of a 6-state Kalman filter, that is, in turn, separated into 3 different steps. A prediction step where the algorithm predicts its next state based on satellite attitude dynamics and two different update steps where different attitude measurements are taken into account and weighted against the prediction. The algorithm integrates measurements from GPS, gyroscopes, magnemeters and solar arrays to obtain the best possible attitude estimation.

Given the correct determination of attitude, the attitude control algorithm was developed. This algorithm has two different functionalities: de-tumbling control and normal mission control. In the first part of the mission, the satellite is tumbling about it's axes unpredictably and the GPS antenna is not facing its proper position. Therefore the GPS data is not available, making it impossible to correctly determine the satellite's attitude. The de-tumbling algorithm uses only the information from the gyroscopes to reduce the angular rates. Once these are low enough, GPS data can be obtained and the normal mission functionality can start. In it, a PD control system is used as it proved to have a better performance than the initial PID control system.

Once completely developed, the complete ADCS algorithm was put through a series of tests and simulations to access its performance. The algorithm was firstly tested on its attitude determination capacity, where it had no problems converging to the true attitude even for non constant true attitude and a totally inaccurate initial attitude. Afterwards test were made on the control part of the system. Given static or changing reference attitudes, the algorithm had no difficulties following them. Finally, both attitude and control were tested, in a simulation that mimicked the conditions expected in the real mission. The simulation was set with high initial angular rates in all axis ($\omega = [0.05, 0.05, 0.05]$ corresponding to the highest expected tumbling rates) and purposely inaccurate initial attitude estimation. This would be a

”worst case scenario” simulation. The results showed that the ADCS system is capable of recovering from the tumbling situation within the first $1.1 \times 10^4 s$, (approximately 3 hours) and converges to the proper attitude within $2.3 \times 10^4 s$ (a little over 6 hours) from the mission’s beginning. Afterwards, the filter follows the reference attitude without problems.

6.2 Future Work

This thesis was performed within UVic’s ECOSat project. A fully functional ADCS software was developed and fully tested. Results showed its proper performance and its readiness to be integrated into the hardware as soon as possible.

The software was developed to make this integration the easiest possible. An extensive ”readme” file was included in the software’s folder. This file covers all the changes that need to be made in order to integrate the software properly within the hardware. Also, comments were included in the code to facilitate the reading and understanding of its critical parts. It is expected that anyone with a basic knowledge of the C programming language and satellite attitude will have no problem understanding and working with this software.

One aspect that could still be improved would be the de-tumbling algorithm. Due to the tight deadlines usual for this type of projects, the B-dot de-tumbling control system was not explored in full detail. The simpler angular rate proportional control system presented a better performance in the initial testing and was, therefore, selected for the de-tumbling task of the filter. However, the author believes that, given more time, the B-dot controller could have been fine tuned into a better performance.

Bibliography

[1] Geocentrix Technologies Ltd. (2011). *Canadian Satellite Design Challenge*. Retrieved November 8, 2012, from: `www.geocentrix.ca/index.php?option=com_content&view=article&id=2&Itemid=2`.

[2] Anubhav Mishra. (2012). *UVic's ECOSat team*. Retrieved November 8, 2012, from: `http://www.csdc.uvic.ca/home.php`.

[3] Wang, P.; Shtessel, Y. B.; Wang, Y. (1998). *Satellite Attitude Control Using only Magnetorquers*. IEEE American Control Conference.

[4] Wiśniewski, R.; Blanke, M. (1997). *Three-axis Satellite Attitude Control Base on Magnetic Torquing*. 13th IFAC World Congress.

[5] R.E. Kalman (1960) *A New Approach to Linear Filter Prediction Problems*. Journal of Basic Engineering, Vol. 82, pp. 35-45.

[6] R.E. Kalman; R.S. Bucy (1961) *New Results in Linear Filtering and Prediction Theory*. Journal of Basic Engineering, Vol. 83, pp. 95-108.

[7] Zarchan, P.; Musoff, H. (2005). *Fundamentals of Kalman Filtering: A Practical Approach*. AIAA.

[8] G.L. Smith; S.F. Schmidt (1961) *The Application of Statistical Filter Theory to Optimal Trajectory Determination Onboard a Circumlunar Vehichle*. presented at AAS Meeting, Aug. 1-3, 1961.

[9] S.F. Schmidt (1961) *The Kalman Filter: Its Recognition and Development for Aerospace Applications*. Journal of Guidance and Control, Vol. 4, pp. 4-7.

[10] J.L. Farrell (1964) *Attitude Determination by Kalman Filtering*. Vol. I, NASA-CR-598.

[11] J.L. Farrell (1970) *Attitude Determination by Kalman Filtering*. Automatics, Vol. 6, pp. 419-430.

[12] G.W. Cherry; J. O'Connor (1965) *Design Principles of the Lunar Excursion Module Autopilot*. MIT Rept. R-499.

[13] J.E. Potter; W.E. Vander Velde (1968) *Optmal Mixing of Gyroscope and Star Tracker Data*. Journal of Spacecraft and Rockets, Vol. 5, pp. 536-540.

[14] E.G. Foudriat (1969) *A Limited Memory Attitude Determination System Using Simplified Equations of Motion*. Symposium on Spacecraft Attitude Determination, Aerospace Corp. Vol I, pp. 15-27.

[15] D.C. Pauling; D.B. Jackson; C.D. Brown (1969) *SPARS Algorithms and Simulation Results*. Symposium on Spacecraft Attitude Determination, Aerospace Corp. Vol I, pp. 293-317.

[16] D.B. Jackson (1969) *Applications of Nonlinear Estimation Theory to Spacecraft Attitude Determination Systems*. Symposium on Spacecraft Attitude Determination, Aerospace Corp. Vol I, pp. 89-111.

[17] S. Kau; K.S.P. Kumar; G.B. Granley (1969) *Attitude Determination via Nonlinear Filtering*. IEEE Transaction on Aerospace and Electronic Systems, Vol. AES-5, pp. 906-911

[18] E.J. Lefferts; F.L Markley; M.D. Shuster (1982) *Kalman Filter for Spacecraft Attitude Estimation*. Journal of Guidance, Control and Dynamics, Vol. 5, No. 5, pp. 417.

[19] J.L. Crassidis; J.L Junkins (2004) *Optimal Estimation of Dynamic Systems*. Chapman Hall/CRC, Boca Raton, FL, USA.

[20] A. Edwards (1971) *The State of Strapdown Inertial Guidance and Navigation*. Navigation, Vol. 18, pp.386-401.

[21] F.L. Markley (2003) *Attitude Error Representations for Kalman Filtering*. Journal of Guidance, Control and Dynamics, Vol. 63, pp. 311-317.

[22] Trawny, N.; Roumenliotis S. I. (2005). *Indirect Kalman Filter for 3D Attitude Estimation*. Multiple Autonomous Robotic Systems Laboratory, Technical Report no. 2005-002, Rev. 57.

[23] M.L. Psiaki; E.M. Kintner; S.P. Powell (2002) *Attitude Estimation or for a Flexible Spacecraft in an Unstable Spin*. Journal of Guidance, Control and Dynamics, Vol. 25, No. 1, pp. 88-85.

[24] F.L. Markley (2002) *A Novel Quaternion Filter*. AIAA Guidance, Navigation and Control Conference, Monterey, CA, USA.

[25] M.L. Psiaki (2005) *Backward-Smoothing Extended Kalman Filter*. Journal of Journal of Guidance, Control and Dynamics, Vol. 28, No. 5, pp. 885-894.

[26] F.L. Markley (1994) *Deterministic EKF-Like Estimator for Spacecraft Attitude Estimation*. Proceedings of the American Control Conferece, Baltimore, MD, USA, pp. 247-251.

[27] R.H. Smith (1995) *An H_∞-Type Filter for GPS-Based Attitude Estimation*. AAS/AIAA Spaceflight Mechanics Conference, Albuquerque, NM, USA, AAS-95-134.

[28] J.L. Crassidis; F.L. Markley; Y. Cheng (2007) *A Survey of Nonlinear Attitude Estimation Methods*. University of Buffalo, NY, USA.

[29] Ovchinnikov, M. Yu.; Penkov, V. I.; Ilyin, A. A.; Selivanoc, S. A. (2005). *Magnetic Attitude Control Systems of the Nanosatellite TNS-Series*. 5th IAA Symposium on Small Satellites for Earth Observation, Berlin.

[30] Banahan, M.; Brady, D.; Doran, M. (1991). *The C Book*. Second Edition, Addison Wesley.

[31] Cai, G.; Chen, B. M.; Lee, T. H. (2011). *Unmanned Rotorcraft Systems*. Advances in Industrial Control, ©Springer-Verlag London Limited.

[32] DiSalle, R. (2002). *Space and Time: Inertial Frames*. The Standford Encyclopedia of Philosophy.

[33] Cesa, J.; (2011). *Azimuth and Altitude*. Self-published work.

[34] Hamilton, W. R. (1866). *Elements of Quaternions*. Longmans, Green & Co., London.

[35] Hoag, D. (1963). *Apollo Guidance and Navigation - Considerations of Apollo IMU Gimbal Lock*. MIT, Instrumental Laboratory Document E-1344.

[36] Breckenridge, W. G. (1999). *Quaternions - Proposed Standard Conventions*. JPL, Tech. Rep. INTEROFFICE MEMORANDUM IOM 343-79-1199.

[37] Kuipers, J. (1999). *Quaternions and Rotation*. Department of Mathematics, Cavin College, Grand Rapids, USA.

[38] Shuster, M. D. (1993). *A survey of attitude representations*. Journal of the Astronautical Sciences, vol. 41, no. 4, pp. 439-517.

[39] Werz, J. R.; Larson, W. J. (2005). *Space Mission Analysis and Design*. Third Edition, Space Technologies Library.

[40] Peter S. Maybeck (1979) *Stochastic models, estimation and control, vol. I*. Department of Electrical Engineering, Air Force Institute of Technology, Wright-Patterson Air Force Base, Academic Press, Ohio, USA.

[41] Greg Welch; Gary Bishop (2001) *An Introduction to the Kalman Filter*. Department of Computer Science, University of North Carolina at Chapel Hill, USA.

[42] Julier, S. J.; Uhlmann, J. K. (1997). *A new extension of the Kalman filter to nonlinear systems*. International Symposium on Aerospace/Defense Sensing, Simulation and Controls 3.

[43] M.J. Hale; P. Vergez; M.J. Meerman; Y. Hashida (2004) *Kalman Filtering and the Attitude Determination and Control Task*. Department of Astronautics, US Air Force Academy, CO, USA

[44] R.M. Plessis (1997) *Poor Man's Explanation of Kalman Filtering or How I Stopped Worrying and Learned to Love Matrix Inversion*. Rockwell International, CA, USA.

[45] Y. Hashida (2004) *ADCS for Future UoSat Standard Platform: REVISION*. SSTL Internal Technical Note, SSTL, Guilford, Surrey, United Kingdom.

[46] C. Rithmayr (2004) *Contributions of Spherical Harmonics to Magnetic and Gravitational Fields*. NASA/TM–2004–213007.

[47] Wikipedia user Lasunncty (2007) *Orbital Elements* http://en.wikipedia.org/wiki/File:Orbit1.svg, Wikipedia, retrived May, 2012

[48] NASA (2001) *Orbital Mechanics 101* http://www.isset.org, retrived May, 2012

[49] NIMA Technical Report TR8350.2 *Department of Defense World Geodetic System 1984, Its Definition and Relationships With Local Geodetic Systems* Third Edition, National Geospatial-Intelligence Agency

[50] Tom Logsdon (1998) *Orbital Mechanics: theory and applications*. John Wiley & Sons

[51] Robin Green (1985) *Spherical Astronomy*. Cambridge University Press

[52] I. Reda; A. Andreas (2003) *Solar Position Algorithm for Solar Radiation Applications*. NREL Report No. TP-560-34302

[53] S. Maus; S. Macmillan (2010) *The US/UK Worl Magnetic Model for 2010-2015*. NOAA National Geophysical Data Center, British Geological Survey Geomagnetism Team

[54] Montgomery, D. B. (1969). *Solenoid Magnet Design: The Magnetic and Mechanical Aspects of Resistive and Superconducting Systems*. New York, Wiley-Interscience.

[55] Stuart Bennett (1993) *A history of control engineering*. 1930-1955, IET, pp. 48

[56] Thomas Flatley, Wendy Morgenstern, Alan Reth, Frank Bauer (1997) *A B-Dot Acquistion Controller for the RADARSAT Spacecraft* NASA Conference Publication, NASA

[57] Jacobs, O.L.R. (1993). *Introduction to Control Theory*. Second Edition, Oxford University Press.

[58] F.L. Markley (2004) *Attitude Estimation or Quaternion Estimarion?*. Journal of Astronautical Sciences, Vol. 51, No. 1, pp. 75-101.

Appendix A

Powers of the $[\omega\times]$ and Ω matrices

The powers of the *skew-symmetric matrix operator* ($[\omega\times]$) are given by

$$[\omega\times]^2 = \omega\omega^T - |\omega|^2 \cdot \mathbf{I}_{3\times 3} \tag{A.1}$$

$$\begin{aligned} [\omega\times]^3 &= (\omega\omega^T - |\omega|^2 \cdot \mathbf{I}_{3\times 3})\,[\omega\times] \\ &= -|\omega|^2 \cdot [\omega\times] \end{aligned} \tag{A.2}$$

$$\begin{aligned} [\omega\times]^4 &= [\omega\times]^3 \cdot [\omega\times] \\ &= -|\omega|^2 \cdot [\omega\times]^2 \end{aligned} \tag{A.3}$$

$$\begin{aligned} [\omega\times]^5 &= [\omega\times]^3 \cdot [\omega\times]^2 \\ &= |\omega|^4 \cdot [\omega\times] \end{aligned} \tag{A.4}$$

$$\begin{aligned} [\omega\times]^6 &= [\omega\times]^5 \cdot [\omega\times] \\ &= |\omega|^4 \cdot [\omega\times]^2 \end{aligned} \tag{A.5}$$

$$\begin{aligned} [\omega\times]^7 &= [\omega\times]^5 \cdot [\omega\times]^2 \\ &= -|\omega|^6 \cdot [\omega\times] \end{aligned} \tag{A.6}$$

and so on, while the powers of the Ω matrix are given by:

$$\begin{aligned}
\Omega(\omega)^2 &= \begin{bmatrix} [\omega\times]^2 - \omega\omega^T & -[\omega\times]\omega \\ \omega^T[\omega\times] & -\omega^T\omega \end{bmatrix} \\
&= \begin{bmatrix} -|\omega|^2 \cdot \mathbf{I}_{3\times3} & \mathbf{0}_{3\times1]} \\ \mathbf{0}_{1\times3} & -|\omega|^2 \end{bmatrix} \\
&= -|\omega|^2 \cdot \mathbf{I}_{4\times4}
\end{aligned} \tag{A.7}$$

$$\Omega(\omega)^3 = -|\omega|^2 \cdot \Omega_{4\times4} \tag{A.8}$$

$$\Omega(\omega)^4 = |\omega|^4 \cdot \mathbf{I}_{4\times4} \tag{A.9}$$

$$\Omega(\omega)^5 = |\omega|^4 \cdot \Omega_{4\times4} \tag{A.10}$$

$$\Omega(\omega)^6 = -|\omega|^6 \cdot \mathbf{I}_{4\times4} \tag{A.11}$$

and so on.