TÉCNICO
LISBOA

Attitude Determination for the NANOSTAR Project

Hugo Miguel Martins Marques

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisor: Prof. Paulo Jorge Coelho Ramalho Oliveira

Examination Committee

Chairperson: Prof. José Fernando Alves da Silva
Supervisor: Prof. Paulo Jorge Coelho Ramalho Oliveira
Member of the Committee: Prof. José Raul Carreira Azinheira

November 2019

Acknowledgments

First and foremost, I would like to thank my advisor Professor Paulo Oliveira for encouraging me to embrace this endeavor, providing guidance and insightful discussions, fomenting autonomy and providing all the necessary facilities and equipment for the research and writing of this dissertation.

I would also like to thank my fellow aerospace engineering colleagues from Instituto Superior Técnico, with whom I had the pleasure to take this five years journey.

Lastly, I wish to express my gratitude to my family, friends and girlfriend, for their unconditional love and support, and for cheering me whenever needed.

Resumo

O Sistema de Determinação e Controlo de Atitude (ADCS) de um nanosatélite é um subsistema chave no fornecimento preciso de atitude, algo necessário para apontar instrumentos e realizar manobras. O seu design está bastante condicionado em termos de massa, volume, tamanho, custo e potência. O objetivo principal desta dissertação é fornecer o projeto NANOSTAR com um estudo fundamentado em termos dos algoritmos de determinação de atitude e sensores que podem ser utilizados nas missões desenhadas no âmbito do projeto. Para isso, foi desenvolvida uma plataforma de simulação que descreve realisticamente o ambiente do nanosatélite, permitindo geração e propagação de órbitas, e ao mesmo tempo fornecimento de dados ao ADCS. De seguida, três algoritmos de determinação de atitude representativos, nomeadamente o Estimador de Quaternião (QUEST), o Filtro de Kalman Estendido Multiplicativo (MEKF) e um recentemente desenvolvido Observador Não Linear de Atitude Globalmente Exponencialmente Estável, foram estudados e implementados na plataforma desenvolvida, utilizando medidas de um star tracker, sensor solar, magnetómetro e giroscópio. Finalmente, foi feita uma comparação dos três algoritmos em termos da eficiência de recursos computacionais usados, desempenho em estado estacionário e desempenho em caso de falhas, usando simulações realistas. Os resultados obtidos providenciam ao projeto uma importante análise das vantagens, desvantagens, complexidade, eficiência de recursos computacionais e desempenho dos três algoritmos, que permitirá futuras tomadas de decisão fundamentadas no design do ADCS.

Palavras-chave: Determinação de Atitude, Nanosatélite, Métodos Determinísticos, Filtro de Kalman, Observador Não Linear

Abstract

The Attitude Determination and Control System (ADCS) of a nanosatellite is a key subsystem to provide precise attitude knowledge and pointing for the on-board payload and necessary maneuvers. Its design has serious constraints in terms of mass, volume, size, cost and power. The main goal of this dissertation is to provide the NANOSTAR project with a grounded study in terms of attitude determination algorithms and sensors that can be employed in the missions designed under the scope of the project. For that, a simulation platform that realistically describes the nanosatellite environment, allowing orbit generation and propagation, as well as data creation to feed the ADCS was developed. Then, three representative attitude determination algorithms, namely the Quaternion Estimator (QUEST), the Multiplicative Extended Kalman Filter (MEKF), and a recently developed Globally Exponentially Stable Cascade Attitude Nonlinear Observer, were studied and implemented on the platform developed, using vector measurements provided by a star tracker, Sun sensor, magnetometer and rate gyroscope. Finally, the comparison of the three algorithms in terms of computational resources efficiency, steady-state performance and performance in the case of faults is done, using realistic simulation scenarios. The results obtained provide meaningful insight on the advantages, disadvantages, complexity, computational resources efficiency and performance of the three algorithms, providing the project with a grounded analysis that can be used for future decision making in terms of the ADCS design.

Keywords: Attitude Determination, Nanosatellite, Deterministic Methods, Kalman Filter, Nonlinear Observer

Contents

List of Tables

List of Figures

List of Symbols

Functions and Operators

$\hat{\square}$ Normalized value.

$\widehat{\square}$ Estimated value.

$adj(\square)$ Matrix adjoint.

$blkdiag(\square, \square)$ Block diagonal matrix.

$det(\square)$ Matrix determinant.

$diag(\square, \square)$ Diagonal matrix.

$\exp\{\square\}$ Matrix exponential.

$E\{\square\}$ Expectation.

$max(\square, \square)$ Maximum.

$tr(\square)$ Trace.

$\left[\square\times\right]$, $\mathbf{S}(\square)$ Cross product matrix of some vector $\square$.

$\mathbf{\Xi}(\mathbf{q})$, $\mathbf{\Psi}(\mathbf{q})$ Quaternion matrices.

$\otimes$, $\odot$ Quaternion product operations.

Greek symbols

α_i Nonlinear observer parameter.

$\boldsymbol{\beta}$ Gyroscope bias.

$\boldsymbol{\chi}_1$ Column representation of vector measurements and bias.

$\boldsymbol{\chi}_2$ Column representation of a rotation matrix.

$\boldsymbol{\epsilon}$ Residual.

$\boldsymbol{\eta}$ Sensor noise vector.

$\mathbf{\Gamma}$ Discrete input matrix.

$\boldsymbol{\omega}$ Short for $\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{BI}}$.

$\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{BI}}$ Angular velocity of frame $\mathcal{B}$ with respect to frame $\mathcal{I}$, mapped in frame $\mathcal{B}$.

$\boldsymbol{\omega}_m$ Measured angular velocity.

$\boldsymbol{\omega}_{\oplus_{\mathcal{I}}}$ Earth's rotation velocity.

$\boldsymbol{\Phi}$ State transition matrix.

$\boldsymbol{\Upsilon}$ Discrete process noise distribution matrix.

$\boldsymbol{\upsilon}$ Measurement residual.

$\boldsymbol{\vartheta}$ Rotation vector attitude parameterization.

γ_i Nonlinear observer parameter.

λ Eingenvalue.

λ_0 Sum of Wahba's weights a_i.

λ_{max} Largest eigenvalue of $\mathbb{K}$.

$\mu_\oplus$ Earth's gravitational parameter.

ϕ, θ, ψ Euler angles associated with certain axes.

ρ Local atmospheric density.

σ Standard deviation.

τ_{GMST} Greenwich Mean Sidereal Time (GMST) angle.

ϑ Angle of rotation around axis.

Roman symbols

$\mathbf{0}_{m\times n}$ $m \times n$ null matrix.

$\mathbf{A}$ Attitude rotation matrix, or DCM.

$\mathbf{a}$ Acceleration.

$\mathbf{A}(\Box)$ Attitude rotation matrix formed from other parameterization of attitude $\Box$.

$\mathbf{a}^p$ Perturbations acceleration.

$\mathbf{A}_{\mathcal{BI}}$ Attitude rotation matrix from $\mathcal{I}$ to $\mathcal{B}$ reference frame.

a_i Wahba's problem weights.

$\mathbb{B}$ Attitude profile matrix.

$\mathcal{B}$ Spacecraft body frame of reference.

$\mathbf{B}$	Input matrix.
$\mathbf{b}$	Measurement represented in frame $\mathcal{B}$.
C_D	Coefficient of drag.
$\mathbf{D}$	Direct transmission matrix.
$\mathcal{E}$	Earth-Centered/Earth-Fixed Frame.
$\mathbf{e}$	Rotation axis.
$\boldsymbol{f}$	Force.
$\mathbf{F}$	State matrix.
$\mathbf{f}$	Differentiable function.
$F_{\odot}$	Solar constant.
$\mathbf{G}$	Process noise distribution matrix.
G	Universal gravitational constant.
$\boldsymbol{H}$	Angular momentum.
$\boldsymbol{H}^0$	Angular momentum with respect to the origin.
$\mathbf{H}$	Observation matrix.
$\mathbf{h}$	Differentiable function that relates **x** with **y**.
$\mathcal{I}$	Earth-Centered Inertial Frame of reference.
$\mathbf{I}_q$	Identity quaternion.
$\mathbf{I}_n$	Identity matrix of size $n \times n$.
$\boldsymbol{J}$	Moment of inertia tensor.
$\boldsymbol{J}^c$	Moment of inertia tensor about the center of mass.
J_n	Zonal coeficient.
$\mathbb{K}$	Davenport's q method symmetric traceless matrix.
$\mathbf{K}$	Kalman gain matrix.
$\boldsymbol{L}$	Net torque.
$\boldsymbol{L}^0$	Net torque about the origin.
L	Wahba's problem loss function.
$\mathbf{m}$	Magnetic dipole.

m	Nanosatellite mass.
$M_\oplus$	Mass of the Earth.
$\mathbf{n}$	Outward normal unit vector.
$\mathbf{P}$	Error covariance matrix.
$\mathbf{p}$	Arbitrary vector.
$P_\odot$	Pressure of solar radiation.
$\mathtt{Q}$	Nonlinear observer parameter.
$\mathbf{Q}$	Spectral density matrix.
$\mathbf{q}$	Quaternion.
$\mathbf{q}(\square)$	Quaternion formed from other parameterization of attitude $\square$.
$\mathbf{q}^*$	Quaternion conjugate.
$\boldsymbol{r}$	Position of the spacecraft with respect to the center of mass of the Earth.
$\mathbf{R}$	Measurement covariance matrix.
$\mathbf{r}$	Measurement represented in frame $\mathcal{I}$.
r	norm of $\boldsymbol{r}$.
$R_\oplus$	Earth's radius.
S	Area.
t	Time.
T_0	Number of Julian centuries elapsed from the J2000 epoch.
$\mathbf{u}$	Input to a system.
$\boldsymbol{v}$	Velocity.
$\mathbf{v}$	Measurement noise vector.
$\boldsymbol{V}$	Magnetic field.
$\mathbf{w}$	Process noise vector.
$\mathbf{x}$	State vector.
$\mathbf{y}$	Observation vector.

Subscripts

a	Aerodynamic.

g	Gravity.
0	Initial or at time 0.
k	Quantity at time k.
m	Magnetic.
mag	Magnetometer.
s	Solar radiation pressure.
ss	Sun sensor.
$star$	Star tracker.
t	Third-body.

Superscripts

T	Transpose.
$true$	True value of some variable.
+	Updated.
-	Prior to update.

Glossary

ADCS	Attitude Determination and Control System.
AEKF	Additive Extended Kalman Filter.
AIM	Asteroid Impact Mission.
COTS	Commercial-of-the-shelf.
CSD	CubeSat Design Specification.
CSS	Coarse Sun Sensor.
CVG	Coriolis Vibratory Gyroscope.
DCM	Direction Cosine Matrix.
ECEF	Earth-Centered/Earth-Fixed.
ECI	Earth-Centered Inertial.
EKF	Extended Kalman Filter.
ERDF	European Regional Development Fund.
ESA	European Space Agency.
ESOQ	Estimator of the Optimal Quaternion.
FLOPs	Floating Point Operations per second.
FLOP	Floating Point Operation(s).
FOAM	Fast Optimal Attitude Matrix.
FOG	Fiber Optic Gyroscope.
FOV	Field of View.
GAS	Globally Asymptotically Stable.
GES	Globally Exponentially Stable.
GMST	Greenwich Mean Sidereal Time.
GPS	Global Positioning System.
GTO	Geostationary Transfer Orbit.
ICRF	International Celestial Reference Frame.
IC	Integrated Circuit.
JPL	Jet Propulsion Laboratory.
LEO	Low-Earth Orbit.
LOS	Line-of-Sight.
MEKF	Multiplicative Extended Kalman Filter.

MEMS	Micro Electromechanical Systems.
MOI	Moment of Inertia Tensor.
MRP	Modified Rodrigues Parameters.
MarCO	Mars Cube One.
NASA	National Aeronautics and Space Administration.
NATO	North Atlantic Treaty Organization.
QMM	QUEST Measurement Model.
QUEST	Quaternion Estimator.
RAAN	Right Ascension of the Ascending Node.
RG	Rate Gyroscope.
RIG	Rate Integrating Gyroscope.
RMS	Root Mean Square.
S/C	Spacecraft.
SO(3)	Special Orthogonal group of dimension 3.
SRP	Solar Radiation Pressure.
SVD	Singular Value Decomposition.
TRIAD	TRIaxial Attitude Determination.
VLBI	Very Long Baseline Interferometry.
WMM	World Magnetic Model.

Chapter 1

Introduction

The trend in satellites, present in the last decades, to do more for less cost, having smaller, cheaper, faster and better space missions, has led to the decrease of spacecraft sizes. Small satellites, ranging from 750 kg to less than 1 kg, have been responsible for greatly reducing the time needed to obtain science and technology results, reducing the overall costs at the same time.

The segment of nanosatellites, concerning masses ranging from 1 to 10 kg [1], is of particular interest, showing an increasing popularity and growth, with the number of projects increasing. From these, CubeSats, following The CubeSat Design Specification (CSD) [2], should be highlighted. The CSD was created in 1999 with the purpose of providing a standard for design of nanosatellites, reducing cost, development time and increasing accessibility, and boosted the development of nanosatellites. A CubeSat is a 10 cm cube with a mass of up to 1.33 kg.

According to a recent survey of worldwide nanosatellite missions [3], they are build for a relatively short lifetime and mission objectives range from technology demonstration and operational use, to university programs providing an important ”hands-on” experience. Although education and technology demonstration are the primary objects in most programs, some demanding applications related with Earth Observation or science domains, have also appeared [4], with examples such as NASA's In-Sight/MarCO mission or ESA's AIM.

The decrease in satellite sizes was accompanied by strict mass, volume, power and cost constraints. Only the miniaturization of satellite technologies and systems, possible by technology advances enabling the decrease in electronics size, with increased capability and low power consumption, made possible to fulfill those constraints. The reduced time and cost of the projects also enabled commercial-of-the-shelf (COTS) hardware to become of frequent use.

From the several key subsystems composing a nanosatellite, surveyed and analyzed in [3], the Attitude Determination and Control System (ADCS) is an important subsystem to provide precise pointing for the on-board payloads and maneuvers.

Early nanosatellites had low pointing requirements, or none at all, hence or there was no ADCS, or it was composed of passive attitude stabilization schemes, such as spinning stabilization, gravity gradient stabilization or passive magnetic stabilization [5]. Eventually, active attitude control schemes started to be adopted, from active magnetic attitude control, which provided low precision, to momentum-biased attitude control, which provided medium-precision, and finally reaction wheels-based attitude control, which is a fine precision scheme and is dominant for nanosatellites with fine-precision pointing requirements.

Nowadays, many nanosatellites requiring precise pointing and stabilization have its ADCS composed of miniaturized and novel precise attitude sensors and proper attitude determination algorithms while the desired pointing and stabilization is achieved by using proper miniaturized and novel attitude actuators and active control schemes.

1.1 The NANOSTAR Project

NANOSTAR [6] is a collaborative project, funded by the Interreg Sudoe Programme through the European Regional Development Fund (ERDF), among 7 universities, 2 aerospace clusters and 3 ESA Business Incubation Centers, in Portugal, France and Spain. It is a network among universities, the regional industry and the scientific ecosystem, in order to create a leading platform on nanosatellites in Europe. The project aims at providing students with the experience of a real space engineering process that includes all stages, from conception and specifications, to design, assembly, integration, testing and documentation. In this way, students from the Southwest of Europe can prepare a high level of skills in space engineering and project engineering, so that they are the future main players in the field of nanosatellites.

The training on technology of nanosatellites is to be provided through several design, development and testing student challenges. To date, two nanosatellite space mission preliminary design challenges were launched. The first (from now on referred as Mission 1), launched in February 2019, consists in creating the preliminary design of a CubeSat fly-by mission to the Moon, motivated by the possible presence of water ice in cold, permanently shadowed, craters at the Moon's poles. The satellite must be equipped with an optical camera as scientific payload, in order to perform observations and measurements of the Moon's surface, while executing a close-distance fly by. Among other requirements [7], a pointing accuracy of the main scientific payload (optical camera) of $0.5°$ must be ensured during the payload operation. More recently, in September 2019, a new space mission preliminary design challenge was also launched that consists in a mission (from now on referred as Mission 2) orbiting Earth, with a payload that will monitor the metabolism of a marine photosymbiotic species of worms and their efficiency for urea and air recycling. Essentially, 2 missions can then be distinguished, one that consists in an orbit around the Moon, and the other around the Earth.

A challenge for the detailed design and testing of nanosatellite components, subsystems and facilities for the missions stated above, was also recently launched.

1.2 Literature Review

The attitude of a spacecraft is its orientation in space with respect to a specific reference frame and its knowledge is essential for most missions success, playing for a long time an essential role in unmanned air vehicles such as aircrafts and satellites. Attitude estimation is a problem with a rich historical background that continues to attract intensive research.

Some of the earliest solutions to address this problem, referred in this work as deterministic methods or point-to-point approaches, essentially provide algebraic solutions based on measurements taken at the same time. One of the first of these methods is the TRIAD algorithm [8], able to determine the attitude of a spacecraft using two and only two measurements. Fighting the limitations of TRIAD, a year after, Grace Wahba formulated a general criterion for attitude determination using two or more vector measurements [9], that would lead to several alternative solutions based on different attitude parameterizations [10–12], such as the Davenport's q-method, the QUEST algorithm and the ESOQ algorithm, providing solutions on quaternions, and the SVD algorithm and the FOAM algorithm, providing solutions in matrix form.

Other solutions try to find the best estimate of the true attitude using dynamical models and measurements, both corrupted by noise and uncertainties. These type of approaches can be used to simultaneously estimate a quantity and filter out noise at the same time. From these approaches, the ones related to Kalman filtering, originally developed as a tool in linear estimation theory [13, 14], may be highlighted. The first application of Kalman filtering to determine the attitude of a spacecraft used an Euler angles parameterization [15]. Since the dynamic and measurement models involved are nonlinear, the Extended Kalman Filter (EKF) is used to address nonlinear estimation problems [16]. The EKF earned then a reputation, becoming the most used method to determine the attitude of a spacecraft. EKF solutions come in several forms, such as the Additive EKF (AEKF) or the Multiplicative EKF (MEKF) [17]. However being a very powerful estimation technique, the EKF estimators do not give guarantees of optimality, stability or convergence.

Other nonlinear approaches rose in the attempt to overcome poor performance or even divergence arising from the linearization implicit in the EKF. Some of those alternatives are well documented in [18]. Some examples are methods based on QUEST, such as filter QUEST [19], or recursive QUEST [20], however showing sub-optimal performance characteristics in comparison to the EKF. Other alternatives appear in the form of unscented filters [21, 22], showing some advantages over the standard EKF such as a lower expected error and the possibility of application to non-differentiable functions.

Particle filters, the two-step optimal estimator [23, 24], predictive and adaptive approaches, and nonlinear observers can also be used to estimate attitude. Nonlinear observers have emerged as an attractive alternative given the possibility to establish convergence bounds and provide stability guarantees, with an associated lower computational cost, and a more straightforward tuning of the design parameters.

1.3 Motivation and Goals

The two missions under the scope of the NANOSTAR project need to guarantee the pointing of its payload in order for the success of their scientific missions. The detailed design and testing challenge, recently launched, motivates then the design of an ADCS as a key subsystem for guaranteeing the pointing requirements of the two missions.

The development of the ADCS can essentially be divided into a part that consists in determining the attitude, through precise determination algorithms, and to a part that consists in controlling that attitude, through precise control schemes. It is on the first part of the problem that this dissertation focuses.

To develop the attitude determination part of the ADCS, a suite of sensors must therefore be chosen, taking into account the size and mass constraints in designing a nanosatellite, and also mission constraints. Then, a simulator must be developed that realistically simulates the spacecraft's environment, allowing orbit generation and propagation, as well as data creation to feed the ADCS sensors. This simulator will then be available for future developments and studies under the scope of the project. Then the study, implementation and testing of attitude estimation algorithms must be done, documenting their theoretical background, implementation characteristics and simulation results in realistic scenarios.

Therefore, under the scope of the NANOSTAR project, motivated by the design of the attitude determination part of the ADCS, this dissertation goals are summarized as follows:

I Select a sensor suite adequate for both missions.

II Create a reliable simulation platform that realistically simulates the spacecraft's environment, allowing orbit generation and propagation, as well as data creation to feed the ADCS.

III Study, adapt and implement three representative attitude estimation algorithms: the Quaternion Estimator (QUEST), the Multiplicative Extended Kalman Filter (MEKF), and a recent Globally Exponentially Stable Cascade Attitude Nonlinear Observer.

IV Analyze and compare the three algorithms in terms of computational resources efficiency and performance in various realistic simulation scenarios.

Ultimately, this work must use the estimation algorithms to solve the problem of attitude estimation in conditions representative of the two specified missions, giving meaningful insight on the advantages, disadvantages, computational resources efficiency and performance of the three algorithms, providing the project with a grounded analysis that can be used for decision making in terms of the ADCS system design.

1.4 Dissertation Outline

This dissertation is organized as follows:

Chapter 2 provides an overview of the underlying theory about the topic of spacecraft attitude determination, namely the most representative attitude representations, the frames of reference used throughout the work, and the spacecraft mechanics fundamentals needed to characterize the orbital and rotational motion of the spacecraft.

Chapter 3 gives an overview of different characteristic sensors and actuators for spacecraft attitude determination and control, briefly explaining their operation. A sensor suite is selected and their modeling is explained.

Chapter 4 documents the theory underlying the attitude determination methods that are the focus of this work.

Chapter 5 discusses the development of a simulation environment that realistically describes the environment and motion conditions of the spacecraft, enabling the generation of realistically data to feed the sensors and properly test the algorithms implemented. It also describes the integration and implementation of the attitude determination algorithms of chapter 4. Finally, it analyzes the operational efficiency of the three algorithms.

Chapter 6 describes the simulation case scenarios used for testing the attitude determination algorithms, presents the results obtained and provides a discussion of those results.

Chapter 7 sums up the work performed with a conclusion, giving a few ideas for future work.

Chapter 2

Theoretical Background

The attitude of a spacecraft is its orientation in space with respect to a specific reference frame. As the motion of a rigid spacecraft is not only specified by its position and velocity, but also by its rotational motion about its center of mass, attitude knowledge is essential for most space applications. The area of attitude determination is basically the study of methods for estimating the attitude, typically involving the use of several types of sensors and sophisticated data processing algorithms.

The knowledge of a spacecraft attitude can either be single-axis attitude knowledge or three-axis. Single-axis attitude is the specification of the orientation of a single spacecraft axis in inertial space, ordinarily the spin axis, and it requires two parameters. However, the complete spacecraft orientation is not fixed by knowing the orientation of a single axis, as the rotation of the spacecraft about that axis is still unknown. Knowing that extra parameter, three-axis attitude knowledge is obtained, fixing the orientation of the three orthogonal spacecraft axes in inertial space.

This chapter provides an overview of the underlying theory about the topic of spacecraft attitude determination, namely the most representative attitude representations in section 2.1, the frames of reference used throughout the work in section 2.2, and the spacecraft mechanics fundamentals needed to characterize the orbital and rotational motion of the spacecraft in section 2.3.

The reader is suggested to start his reading with Appendix A, before addressing this chapter, as a review of some mathematical notions and conventions adopted in this work is carried out.

2.1 Attitude Representations

Considering $\mathcal{B} = \{\mathfrak{b}_1, \mathfrak{b}_2, \mathfrak{b}_3\}$, an orthogonal, right-handed basis whose axis and origin are fixed in the spacecraft body frame and the reference system $\mathcal{I} = (\mathfrak{i}_1, \mathfrak{i}_2, \mathfrak{i}_3)$, also orthogonal and right handed, the spacecraft attitude is the orientation of the first basis $\mathcal{B}$, with respect to the reference system $\mathcal{I}$. This orientation takes many forms, the so called attitude representations or parameterizations, ranging from vectors of three and four components to matrices. This section presents the most representative ones, discussing overall advantages and disadvantages. The overview done in this section is based on references [11, 25, 26].

2.1.1 Rotation Matrix

As shown in annex A.1, the 3 components of the body coordinate frame, $\mathcal{B}$, can be represented along the 3 axes of the reference frame $\mathcal{I}$. To do this, nine parameters are required, forming a 3×3 matrix that rotates the vectors from the reference frame to the body frame. This matrix is called the Direction Cosine Matrix (DCM):

$$\mathbf{A}_{\mathcal{B}\mathcal{I}} = \begin{bmatrix} \mathfrak{b}_1 \cdot \mathfrak{i}_1 & \mathfrak{b}_1 \cdot \mathfrak{i}_2 & \mathfrak{b}_1 \cdot \mathfrak{i}_3 \\ \mathfrak{b}_2 \cdot \mathfrak{i}_1 & \mathfrak{b}_2 \cdot \mathfrak{i}_2 & \mathfrak{b}_2 \cdot \mathfrak{i}_3 \\ \mathfrak{b}_3 \cdot \mathfrak{i}_1 & \mathfrak{b}_3 \cdot \mathfrak{i}_2 & \mathfrak{b}_3 \cdot \mathfrak{i}_3 \end{bmatrix} \tag{2.1}$$

Being a DCM, $\mathbf{A}_{\mathcal{B}\mathcal{I}}$ is a proper real orthogonal matrix preserving the length of vectors and the angles between them, and thus representing a rotation. The group of matrices that have these properties constitute the Special Orthogonal *group* SO(3).

An arbitrary vector $\mathbf{p}$ in frame $\mathcal{I}$, $\mathbf{p}_{\mathcal{I}}$, is rotated to frame $\mathcal{B}$ by:

$$\mathbf{p}_{\mathcal{B}} = \mathbf{A}_{\mathcal{B}\mathcal{I}}\mathbf{p}_{\mathcal{I}} \tag{2.2}$$

where $\mathbf{p}_{\mathcal{B}}$ is the representation of $\mathbf{p}$ in the $\mathcal{B}$ frame, and $\mathbf{A}_{\mathcal{B}\mathcal{I}}$ represents the rotation matrix from $\mathcal{I}$ to $\mathcal{B}$.

The attitude matrix or rotation matrix $\mathbf{A}_{\mathcal{B}\mathcal{I}}$ is the fundamental representation of the spacecraft's attitude. This representation is a popular choice as it defines uniquely the attitude, it does not suffer from singularities, has no need of evaluating trigonometric functions, and it is continuously differentiable. However, it implies working with 9 different parameters, six of which are redundant due to the orthogonality and symmetry requirements.

2.1.2 Euler Axis/Angle

Euler's theorem states that any rotation represented by a rotation matrix $\mathbf{A}$ is a rotation about a fixed axis, specified by a unit vector $\mathbf{e}$. To completely specify the matrix, the angle of rotation ϑ about this axis needs to be considered. $\mathbf{e}$ and ϑ are known as the Euler axis and Euler angle of rotation, respectively.

The Euler axis/angle parameterization of an attitude matrix is given by:

$$\mathbf{A}(\mathbf{e}, \vartheta) = (\cos\vartheta)\mathbf{I}_3 - \sin\vartheta \left[\mathbf{e}\times\right] + (1 - \cos\vartheta)\mathbf{e}\mathbf{e}^T \tag{2.3}$$

where $\left[\mathbf{e}\times\right]$ is the cross product matrix formed from vector $\mathbf{e}$ as follows from annex A.2.

Although the attitude matrix seems to depend on four parameters, as $||\mathbf{e}|| = 1$, there are only three independent parameters. Equation 2.3 shows that the attitude matrix is a periodic function of the rotation angle over an unlimited range with period 2π.

The Euler rotation angle can be expressed in terms of the elements of the rotation matrix as:

$$\cos\vartheta = \frac{1}{2}\left(tr(\mathbf{A}(\mathbf{e}, \vartheta)) - 1\right) \tag{2.4}$$

If $\sin\vartheta \neq 0$, the components of **e** are given by:

$$\mathbf{e} = \frac{1}{2\sin\vartheta}\begin{bmatrix} A(\mathbf{e},\vartheta)_{23} - A(\mathbf{e},\vartheta)_{32} \\ A(\mathbf{e},\vartheta)_{31} - A(\mathbf{e},\vartheta)_{13} \\ A(\mathbf{e},\vartheta)_{12} - A(\mathbf{e},\vartheta)_{21} \end{bmatrix} \tag{2.5}$$

Equation 2.4 has two solutions, differing on the sign. Those different solutions imply axis vectors **e** in opposite directions, expressing the fact that a rotation about **e** by an angle ϑ, is equivalent to a rotation about $-\mathbf{e}$ and $-\vartheta$.

This representation has the clear advantage that it has a intuitive physical interpretation. However, it requires working with one redundant parameter, leading with trigonometric functions, and the axis **e** is undefined when $\sin\vartheta = 0$.

It is also convenient to combine the Euler axis and angle into a three component rotation vector $\boldsymbol{\vartheta}$:

$$\boldsymbol{\vartheta} \equiv \vartheta\mathbf{e} \tag{2.6}$$

which is useful for representing small rotations.

2.1.3 Quaternion

Following the quaternion convention and associated operations reviewed in appendix A.3, the quaternion may be defined as:

$$\mathbf{q}(\mathbf{e},\vartheta) = \begin{bmatrix} \mathbf{e}\sin(\vartheta/2) \\ \cos(\vartheta/2) \end{bmatrix} \tag{2.7}$$

with the quaternion representation of the attitude matrix as:

$$\mathbf{A}(\mathbf{q}) = \left(q_4^2 - ||\mathbf{q}_{1:3}||^2\right)\mathbf{I}_3 - 2q_4\left[\mathbf{q}_{1:3}\times\right] + 2\mathbf{q}_{1:3}\mathbf{q}_{1:3}^T \tag{2.8}$$

The quaternion, being a four-element parameterization, has the lowest dimensionality possible for a global non-singular representation of $SO(3)$, also without the need of evaluating trigonometric or other complicated functions. The quaternions are more efficient for specifying rotations than the attitude matrix itself, and obey only to one constraint, the unity norm constraint. It has, however, one redundant degree of freedom and no obvious physical interpretation.

A rotation of a three-component vector **p** is implemented by the quaternion product:

$$\mathbf{q}\otimes\mathbf{p}\otimes\mathbf{q}^* = \left[\mathbf{q}\odot\right]^T\left[\mathbf{q}\otimes\right]\begin{bmatrix}\mathbf{p}\\0\end{bmatrix} = \begin{bmatrix}\Xi^T(\mathbf{q})\Psi(\mathbf{q})\mathbf{p}\\0\end{bmatrix} \tag{2.9}$$

Quaternions suffer from a major drawback as **q** and $-\mathbf{q}$ represent the same rotation and attitude estimation based on quaternions may exhibit unwinding phenomenon, where the estimate may start arbitrarily close to the true attitude and yet rotate through large angles before achieving the final desired attitude.

2.1.4 Euler Angles

Euler angles define a rotation in terms of three successive rotations, about fixed body axes. The notation used in this work gives a rotation as:

$$\mathbf{A}_{ijk}(\phi,\theta,\psi) = \mathbf{A}(\mathbf{e}_k,\psi)\mathbf{A}(\mathbf{e}_j,\theta)\mathbf{A}(\mathbf{e}_i,\phi) \tag{2.10}$$

using rotation axes selected from the set:

$$\mathbf{e}_1 = \begin{bmatrix}1\\0\\0\end{bmatrix}, \quad \mathbf{e}_2 = \begin{bmatrix}0\\1\\0\end{bmatrix}, \quad \mathbf{e}_3 = \begin{bmatrix}0\\0\\1\end{bmatrix} \tag{2.11}$$

In an asymmetric Euler angle sequence, the three angles are often referred to as roll, pitch and yaw, usually perceiving a screwing motion, a motion that points the vehicle up or down and a motion that points the vehicle left or right, respectively.

The asymmetric Euler angle 3-2-1, often used to describe the attitude of an Earth-pointing spacecraft, was chosen to be the Euler angles sequence of reference, and is described by:

$$\mathbf{A}_{321}(\phi,\theta,\psi) = \mathbf{A}(\mathbf{e}_1,\psi)\mathbf{A}(\mathbf{e}_2,\theta)\mathbf{A}(\mathbf{e}_3,\phi) \tag{2.12}$$

with:

$$\mathbf{A}(\mathbf{e}_1,\psi) = \begin{bmatrix}1 & 0 & 0\\0 & \cos\psi & \sin\psi\\0 & -\sin\psi & \cos\psi\end{bmatrix} \tag{2.13a}$$

$$\mathbf{A}(\mathbf{e}_2,\theta) = \begin{bmatrix}\cos\theta & 0 & -\sin\theta\\0 & 1 & 0\\\sin\theta & 0 & \cos\theta\end{bmatrix} \tag{2.13b}$$

$$\mathbf{A}(\mathbf{e}_3,\phi) = \begin{bmatrix}\cos\phi & \sin\phi & 0\\-\sin\phi & \cos\phi & 0\\0 & 0 & 1\end{bmatrix} \tag{2.13c}$$

Euler angles have clear physical interpretation in many cases and no redundant parameters. However, these are not as convenient for numerical computations as the quaternions, imply working with trigonometric functions and have singularities for some values of θ.

Some other less representative attitude parameterizations, not discussed in this work, such as the Rodrigues parameters or Gibbs vector and the Modified Rodrigues Parameters (MRP) may also be employed to represent attitude.

2.1.5 Attitude Error Representations

Using the example of $\mathbf{A}_{\mathcal{B}\mathcal{I}}$, the attitude estimation error can be represented as a small rotation $\mathbf{A}_{\mathcal{B}\widehat{\mathcal{B}}}$ between $\mathcal{B}$ and an estimated body frame $\widehat{\mathcal{B}}$:

$$\mathbf{A}_{\mathcal{B}\mathcal{I}} = \mathbf{A}_{\mathcal{B}\widehat{\mathcal{B}}}\mathbf{A}_{\widehat{\mathcal{B}}\mathcal{I}} \tag{2.14}$$

being $\mathbf{A}_{\widehat{\mathcal{B}}\mathcal{I}}$ the estimated attitude. The errors can also be represented as a small rotation between $\mathcal{I}$ and an estimated reference frame $\widehat{\mathcal{I}}$ in a similar way.

Following [11], the attitude errors can be represented in terms of the rotation vector of equation 2.6 and its small angle approximation as:

$$\mathbf{A}(\boldsymbol{\delta\vartheta}) = \exp\left(-\left[\boldsymbol{\delta\vartheta}\times\right]\right) \approx \mathbf{I}_3 - \left[\boldsymbol{\delta\vartheta}\times\right] + \frac{1}{2}\left[\boldsymbol{\delta\vartheta}\times\right]^2 \tag{2.15}$$

where $\exp(\,)$ represents the matrix exponential. The attitude errors can also be represented in terms of the quaternion and its small angle approximation as:

$$\mathbf{A}(\boldsymbol{\delta}\mathbf{q}) \approx \mathbf{I}_3 - 2\left[\boldsymbol{\delta}\mathbf{q}_{1:3}\times\right] + 2\left[\boldsymbol{\delta}\mathbf{q}_{1:3}\times\right]^2 \tag{2.16}$$

It is notable that these representations are equivalent through second order considering:

$$\boldsymbol{\delta\vartheta} = 2\boldsymbol{\delta}\mathbf{q}_{1:3} \tag{2.17}$$

Other representations may be used to represent attitude errors, such as the Gibbs vector or the MRPs.

2.2 Reference Frames

A reference frame is specified by its origin and the orientation of its coordinate axes. Any of the attitude parameterizations of section 2.1 represents the components of a coordinate frame along the axes of some other specific reference frame. The reference frames with a special interest are presented in this section.

2.2.1 Inertial Reference Frame

An inertial reference frame is one in which Newton's laws of motion are valid. Any frame moving without rotation and at constant velocity with respect to an inertial frame can also be considered inertial.

The standard inertial reference frame, as of this writing, is the third realization of the International Celestial Reference Frame (ICRF 3), with its axes fixed with respect to precise equatorial coordinates of extragalactic radio sources in Very Long Baseline Interferometry (VLBI) programmes [27]. This frame has its origin at the center-of-mass of the solar system, its Z axis aligned with the Earth's North pole, and the X axis with the vernal equinox, the intersection of Earth's equatorial plane with the ecliptic plane, in the direction of Sun's position relative to the Earth on the first day of spring. Both the polar axis and the ecliptic plane are not inertially fixed, so this system axes are defined as mean orientations at some fixed epoch time.

For attitude analysis of spacecraft orbiting the Earth, the approximately inertial Earth-Centered Inertial Frame (ECI), denoted by $\mathcal{I} = \{\mathfrak{i}_1, \mathfrak{i}_2, \mathfrak{i}_3\}$ is used. It has its origin at the center of mass of the Earth and the axes aligned with the mean North Pole and mean vernal equinox at some epoch. This frame has a linear acceleration since Earth is orbiting around the Sun, but that may be neglected for attitude analysis [11]. The ECI frame and the J2000 epoch are used in this work.

2.2.2 Earth-Centered/Earth-Fixed Frame

The Earth-Centered/Earth-Fixed (ECEF) Frame, designated by $\mathcal{E} = \{\mathfrak{e}_1, \mathfrak{e}_2, \mathfrak{e}_3\}$ rotates with the Earth. It has its $\mathfrak{e}_3$ axis equal to $\mathfrak{i}_3$, its $\mathfrak{e}_1$ axis pointing in the direction of the Earth's prime meridian and $\mathfrak{e}_2$ axis completing the right-handed system. The rotation angle between the ECI and the ECEF frames is known as the Greenwich Mean Sidereal Time (GMST) angle, denoted by τ_{GMST}. τ_{GMST} is given, in units of second, by:

$$\begin{aligned}\tau_{GMST} = 24110.54841 + 8640184.812866 T_0 + 0.093104 T_0^2 \\ -6.2 \times 10^{-6} T_0^3 + 1.002737909350795\ (3600h + 60m + s)\end{aligned} \tag{2.18}$$

with T_0 the number of Julian centuries elapsed from the J2000 epoch and h, m and s, respectively the hour, minute and seconds of the day. This quantity has to be reduced to a range from 0 to 86 400 s and then converted to degrees.

The transformation of a vector in the ECI frame, to its ECEF representation is mapped by the rotation matrix $\mathbf{A}_{\mathcal{E}\mathcal{I}}$:

$$\mathbf{A}_{\mathcal{E}\mathcal{I}} = \begin{bmatrix} \cos(\tau_{GMST}) & \sin(\tau_{GMST}) & 0 \\ -\sin(\tau_{GMST}) & \cos(\tau_{GMST}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.19}$$

2.2.3 Spacecraft Body Frame

The spacecraft body frame considered in this work, denoted by $\mathcal{B} = \{\mathfrak{b}_1, \mathfrak{b}_2, \mathfrak{b}_3\}$, has its origin at the center of mass of the satellite and its axes rotating with the spacecraft. The $\mathfrak{b}_3$ axis is aligned in the direction of the main scientific payload, $\mathfrak{b}_1$ is aligned with the normal to the bottom plate of the satellite, and $\mathfrak{b}_2$ completes the right-handed system.

2.3 Spacecraft Mechanics

This section presents the fundamentals needed to characterize the orbital and rotational motion of the spacecraft. The idea of Keplerian orbits is firstly introduced and then the attitude dynamics and kinematics are studied. Spacecraft perturbations that affect both the orbital and rotational motion are described in the end of this section. The background provided by this section is based on references [11, 28–31].

2.3.1 Keplerian Motion

Johannes Kepler, in the 17^{th} century, proposed his three laws of planetary motion, defining the elliptic shape of planetary orbits, the time required for a planet to complete an orbit and the velocity at which these planets travel around the Sun. Later, Isaac Newton provided a mathematical solution for Kepler's laws, combining his Second Law of Motion, applied to a constant mass system, with his Law of Universal Gravitation.

Newton's law of gravitation states that any two bodies attract each other with a force proportional to the product of their masses and inversely proportional to the square of the distance between them. Mathematically, this statement applied to the motion of a satellite around Earth is given by:

$$\boldsymbol{f}_g = -GM_\oplus m\frac{\boldsymbol{r}}{r^3} = -\mu_\oplus m\frac{\boldsymbol{r}}{r^3} \tag{2.20}$$

where $\boldsymbol{f}_g$ is the gravitational force applied on the satellite, G is the universal gravitation constant, $M_\oplus$ is Earth's mass, m is the satellite mass, $\boldsymbol{r}$ is the vector from the center of the Earth to the satellite, $r = ||\boldsymbol{r}||$, and $\mu_\oplus$ is the Earth's gravitational parameter.

Combining equation 2.20 with Newton's second law, the equation for the acceleration vector of the satellite can be obtained:

$$\ddot{\boldsymbol{r}} = -\frac{\mu_\oplus}{r^3}\boldsymbol{r} \tag{2.21}$$

Equation 2.21, called the two body equation of motion, is the relative equation of motion of a satellite position vector as the satellite orbits the Earth. In its derivation, is assumed that the gravity is the only force acting on the satellite, the Earth is spherically symmetric, the Earth's mass is much greater than the satellite's mass, and the Earth and the satellite are the only two bodies in the system. Additionally, the two bodies are being represented as point masses, which is satisfactory for analyzing the translation motion.

Perturbation forces, discussed in section 2.3.3, such as the gravity differential force due to Earth's oblateness, aerodynamic drag and solar radiation pressure, are often added to the right side of equation 2.21, in order to better describe the motion of the satellite. The modified equation including the sum of these perturbations, in terms of their accelerations, $\mathbf{a}^p$, is:

$$\ddot{\boldsymbol{r}} = -\frac{\mu_\oplus}{r^3}\boldsymbol{r} + \mathbf{a}^p_{\mathcal{I}} \tag{2.22}$$

where $\mathbf{a}^p_{\mathcal{I}}$ is the representation of $\mathbf{a}^p$ in frame $\mathcal{I}$.

Classical Orbital Elements

The most convenient characterization of an orbit is not the Cartesian position and velocity vectors but the six classical Keplerian orbital elements which give a more satisfying physical characterization of the orbit. These are the:

- Semimajor axis (size of the orbit).
- Eccentricity (shape of the orbit).
- Initial mean anomaly (related to the spacecraft's initial position in the orbit).
- Inclination (angle between orbit plane and reference plane).
- Right ascension of the ascending node (RAAN) (angle between vernal equinox direction and the line of nodes).
- Argument of periapsis or perigee (angle between the ascending node direction and periapsis or perigee direction).

The orbit perturbations included in equation 2.22 cause a deviation from the ideal Keplerian orbit of equation 2.21, which means that, instead of the theoretically constant classical orbital elements for a Keplerian orbit, these are, in truth, time-varying.

2.3.2 Attitude Kinematics and Dynamics

Attitude kinematics covers the aspects of rotational motion that can be analyzed without consideration of forces and torques. The study of how forces and torques influence that motion is in the realm of attitude dynamics.

Instead of the point mass model assumed in section 2.3.1, for the study of rotational motion a rigid body model is used to characterize the satellite. A rigid body is defined by the existence of a reference body frame, $\mathcal{B}$, specified in section 2.2.3, in which all the distance between any two given points remain constant.

Attitude kinematics analyses mainly how one of the attitude parameterizations of section 2.1 changes in time according to the angular velocity, $\boldsymbol{\omega}$, irrespective of the forces or torques that cause the motion. In this case, the attitude relating the spacecraft body frame, $\mathcal{B}$, and the inertial frame of reference, $\mathcal{I}$, is the object of study and the kinematic relations for the attitude matrix and quaternions representations are analyzed.

Denoting $\mathbf{A}_{\mathcal{BI}}$ as the rotation matrix from the frame $\mathcal{I}$ to frame $\mathcal{B}$, $\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{BI}}$ as the angular velocity of frame $\mathcal{B}$ with respect to frame $\mathcal{I}$, mapped in frame $\mathcal{B}$, then, the fundamental equation of attitude kinematics is given by:

$$\dot{\mathbf{A}}_{\mathcal{BI}} = -\left[\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{BI}} \times\right] \mathbf{A}_{\mathcal{BI}} \tag{2.23}$$

where $\dot{\mathbf{A}}_{\mathcal{BI}}$ is the time derivative of $\mathbf{A}_{\mathcal{BI}}$, and $\left[\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{BI}} \times\right]$ is computed according to appendix A.2.

The kinematic relation of equation 2.23 may also be written in terms of the quaternion. The quaternion kinematic relationship is given by:

$$\dot{\mathbf{q}}_{\mathcal{BI}} = \frac{1}{2}\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{BI}} \otimes \mathbf{q}_{\mathcal{BI}} \equiv \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{BI}})\mathbf{q}_{\mathcal{BI}} \equiv \frac{1}{2}\boldsymbol{\Xi}(\mathbf{q})\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{BI}} \tag{2.24}$$

where $\mathbf{q}_{\mathcal{BI}}$ is the quaternion representation of the rotation $\mathbf{A}_{\mathcal{BI}}$, $\boldsymbol{\Xi}(\mathbf{q})$ is computed according to appendix A.3, and:

$$\boldsymbol{\Omega}(\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{BI}}) \equiv \begin{bmatrix} -\left[\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{BI}}\times\right] & \boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{BI}} \\ -(\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{BI}})^T & 0 \end{bmatrix} \tag{2.25}$$

Attitude dynamics analyses the forces and torques applied to the spacecraft in order to determine the angular velocity $\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{BI}}$ necessary to compute the attitude from the kinematic equation.

The angular momentum plays a fundamental role in the study of attitude dynamics. Considering a spacecraft as a collection of n point masses, its angular momentum, with respect to the origin of an inertial coordinate frame, is defined in terms of the masses m_i, positions $\boldsymbol{r}^{i0}$ of each point mass, and velocities $\boldsymbol{v}^{i0} = \dot{\boldsymbol{r}}^{i0}$ by:

$$\boldsymbol{H}^0 \equiv \sum_{i=1}^{n} \boldsymbol{r}^{i0} \times m_i \boldsymbol{v}^{i0} \tag{2.26}$$

Using Newton's second law in an inertial frame, we may derive:

$$\dot{\boldsymbol{H}}_{\mathcal{I}}^0 = \sum_{i=1}^{n} \boldsymbol{r}^{i0} \times \boldsymbol{f}_{\mathcal{I}}^{i_{ext}} \equiv \boldsymbol{L}_{\mathcal{I}}^0 \tag{2.27}$$

where the $\boldsymbol{f}_{\mathcal{I}}^{i_{ext}}$ component refers to the external forces, the law of action and reaction was used to ignore the internal forces and $\boldsymbol{L}_{\mathcal{I}}^0$ is the net torque about the origin exerted on the collection of mass points by all the external forces.

By proving that the motion of the center of mass and the motion of the mass points about their center of mass are uncoupled, and using the last two equations, the fundamental equation of attitude dynamics can be derived:

$$\dot{\boldsymbol{H}}_{\mathcal{I}}^c = \boldsymbol{L}_{\mathcal{I}}^c \tag{2.28}$$

which holds even if the center of mass undergoes acceleration but only in a non-rotating frame.

Let's introduce the notion of the moment of inertia tensor (MOI), represented in a general frame as:

$$\boldsymbol{J}^c \equiv -\sum_{i=1}^{n} m_i [\boldsymbol{r}^{ic}\times]^2 \tag{2.29}$$

As the MOI is constant in the body frame, it is almost always represented in that frame.

The angular momentum in the body frame is given by:

$$\boldsymbol{H}_{\mathcal{B}}^c = \boldsymbol{J}_{\mathcal{B}}^c \boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{BI}} \tag{2.30}$$

To summarize, and assuming the attitude is parameterized by a quaternion, the following equations are enough to model the rotational motion of the satellite rigid body:

$$\dot{\boldsymbol{H}}^{c}_{\mathcal{I}} = \boldsymbol{L}^{c}_{\mathcal{I}} \tag{2.31a}$$

$$\boldsymbol{H}^{c}_{\mathcal{B}} = \mathbf{A}(\mathbf{q}_{\mathcal{B}\mathcal{I}})\boldsymbol{H}^{c}_{\mathcal{I}} \tag{2.31b}$$

$$\boldsymbol{\omega}^{\mathcal{B}\mathcal{I}}_{\mathcal{B}} = (\boldsymbol{J}^{c}_{\mathcal{B}})^{-1}\boldsymbol{H}^{c}_{\mathcal{B}} \tag{2.31c}$$

$$\dot{\mathbf{q}}_{\mathcal{B}\mathcal{I}} = \frac{1}{2}\boldsymbol{\omega}^{\mathcal{B}\mathcal{I}}_{\mathcal{B}} \otimes \mathbf{q}_{\mathcal{B}\mathcal{I}} \tag{2.31d}$$

As external torques are more easily computed in the body frame, and a reduction in the number of equations simplifies the process, the equations 2.31a, 2.31b and 2.31c may be substituted by:

$$\dot{\boldsymbol{\omega}}^{\mathcal{B}\mathcal{I}}_{\mathcal{B}} = (\boldsymbol{J}^{c}_{\mathcal{B}})^{-1}\left[\boldsymbol{L}^{c}_{\mathcal{B}} - \boldsymbol{\omega}^{\mathcal{B}\mathcal{I}}_{\mathcal{B}} \times (\boldsymbol{J}^{c}_{\mathcal{B}}\boldsymbol{\omega}^{\mathcal{B}\mathcal{I}}_{\mathcal{B}})\right] \tag{2.32}$$

The external torques $\boldsymbol{L}^{c}_{\mathcal{B}}$ are composed by intentional control torques and by the disturbance torques described in section 2.3.3.

2.3.3 Spacecraft Perturbations

Spacecraft perturbations may be divided into perturbation forces, which induce accelerations, and perturbation torques. They contribute respectively to the terms $\mathbf{a}^{p}_{\mathcal{I}}$ and $\boldsymbol{L}^{c}_{\mathcal{B}}$ of equations 2.22 and 2.32. These perturbations cause unwanted time-variation in the orbital motion and in the angular velocity of the satellite. Description of the most representative perturbation forces and torques and models for their computation are available in this section.

Perturbation Forces

A spacecraft has to deal with disturbing forces, leading to disturbing accelerations that perturb the orbit. The four most important and representative sources for these forces are the gravity due to a non-spherical Earth, $\mathbf{a}_g$, forces due to bodies other than the central body, $\mathbf{a}_t$, aerodynamic drag, $\mathbf{a}_a$, and solar radiation pressure, $\mathbf{a}_s$, all represented in inertial coordinates. Equation 2.22 takes the form:

$$\ddot{\boldsymbol{r}} = -\frac{\mu_{\oplus}}{r^3}\boldsymbol{r} + \mathbf{a}^{p}_{\mathcal{I}} = -\frac{\mu_{\oplus}}{r^3}\boldsymbol{r} + \mathbf{a}_g + \mathbf{a}_t + \mathbf{a}_a + \mathbf{a}_s \tag{2.33}$$

Non-spherical Earth Perturbation

The gravity field, depending directly on the mass, reflects the non-uniformity of the mass distribution caused by the Earth not being a perfect sphere.

The most used approach to model non-spherical gravity uses a spherical harmonic expansion. In this work, a 6^{th} order spherical harmonic geopotential model including only zonal harmonics, presented in [11], is used. Each zonal harmonic is associated with a zonal coefficient J_n, being the perturbing acceleration given by:

$$\mathbf{a}_g = \sum_{i=2}^{6} \mathbf{a}_{J_i} \tag{2.34}$$

The term arising from J_2 is over 400 times bigger than the one arising from J_3, and even bigger than the other higher order terms. For the purpose of this work, only J_2 is considered and the final perturbing acceleration is given by:

$$\mathbf{a}_g = \mathbf{a}_{J_2} \equiv -\frac{3}{2} J_2 \left(\frac{\mu_\oplus}{r^2}\right) \left(\frac{R_\oplus}{r}\right)^2 \begin{bmatrix} \left(1 - 5\left(\frac{r_z}{r}\right)^2\right) \frac{r_x}{r} \\ \left(1 - 5\left(\frac{r_z}{r}\right)^2\right) \frac{r_y}{r} \\ \left(3 - 5\left(\frac{r_z}{r}\right)^2\right) \frac{r_z}{r} \end{bmatrix} \tag{2.35}$$

with $J_2 = 1.08262668355 \times 10^{-3}$ and $R_\oplus = 6378137\,m$, the radius of the Earth.

Third-body Forces

The Keplerian motion is a two-body problem. Any other body entering into the system of forces is considered a third body. Considering the motion of a body with mass m_2 at position $\boldsymbol{r}_2$ about a body with mass m_1 at position $\boldsymbol{r}_1$, under the influence of $n-2$ other bodies with masses m_i at positions $\boldsymbol{r}_i$, their accelerations take the form:

$$\ddot{\boldsymbol{r}}_1 = -\frac{Gm_2}{||\boldsymbol{r}_1 - \boldsymbol{r}_2||^3}(\boldsymbol{r}_1 - \boldsymbol{r}_2) - \sum_{i=3}^{n} \frac{Gm_i}{||\boldsymbol{r}_1 - \boldsymbol{r}_i||^3}(\boldsymbol{r}_1 - \boldsymbol{r}_i) \tag{2.36a}$$

$$\ddot{\boldsymbol{r}}_2 = -\frac{Gm_1}{||\boldsymbol{r}_2 - \boldsymbol{r}_1||^3}(\boldsymbol{r}_2 - \boldsymbol{r}_1) - \sum_{i=3}^{n} \frac{Gm_i}{||\boldsymbol{r}_2 - \boldsymbol{r}_i||^3}(\boldsymbol{r}_2 - \boldsymbol{r}_i) \tag{2.36b}$$

Subtracting equation 2.36a from 2.36b gives the perturbed motion, with $\boldsymbol{r} \equiv \boldsymbol{r}_2 - \boldsymbol{r}_1$, $r \equiv ||\boldsymbol{r}||$, $\mu \equiv G(m_1 + m_2)$, and $\mu_i \equiv Gm_i$ for $i \geq 3$:

$$\ddot{\boldsymbol{r}} = -\frac{\mu}{r^3}\boldsymbol{r} - \sum_{i=3}^{n} \mu_i \left(\frac{\boldsymbol{r}_1 - \boldsymbol{r}_i + \boldsymbol{r}}{||\boldsymbol{r}_1 - \boldsymbol{r}_i + \boldsymbol{r}||^3} - \frac{\boldsymbol{r}_1 - \boldsymbol{r}_i}{||\boldsymbol{r}_1 - \boldsymbol{r}_i||^3} \right) \tag{2.37}$$

The terms in the sum can be treated as small perturbations if all the other bodies are much farther away from m_1 than is m_2.

Aerodynamic Drag

The atmospheric drag is the perturbation with the biggest impact for objects in low-Earth orbits. For the effects of computing aerodynamic drag and torque, it is typical to model the geometry of the satellite as a collection of N flat plates of area S_i and outward normal unit vector $\mathbf{n}^i_{\mathcal{B}}$ expressed in body-frame coordinates. The inclination of the i^{th} plate to the relative velocity in inertial coordinates is given by:

$$\cos\gamma^i_a = (\mathbf{A}^T_{\mathcal{BI}}\mathbf{n}^i_{\mathcal{B}})^T \left(\frac{\boldsymbol{v}_{rel_{\mathcal{I}}}}{||\boldsymbol{v}_{rel_{\mathcal{I}}}||} \right) \tag{2.38}$$

where $\boldsymbol{v}_{rel_{\mathcal{I}}}$ is the spacecraft relative velocity with respect to the atmosphere in frame $\mathcal{I}$.

The drag force with this model, expressed in inertial coordinates, is:

$$\boldsymbol{f}_a = -\frac{1}{2}\rho C_D ||\boldsymbol{v}_{rel_{\mathcal{I}}}|| \boldsymbol{v}_{rel_{\mathcal{I}}} \sum_{i=1}^{N}{}' S_i \cos\gamma_a^i \tag{2.39}$$

where the prime on the sum indicates that only plates with $\cos\gamma_a^i > 0$ are included in the summation, ρ is the local atmospheric density and C_D is a dimensionless drag coefficient, usually in the range between 1.5 and 2.5.

As the atmosphere is not stationary in the inertial frame, the relative velocity can be computed assuming the atmosphere co-rotates with Earth:

$$\boldsymbol{v}_{rel_{\mathcal{I}}} = \boldsymbol{v}_{\mathcal{I}} + \left[\boldsymbol{\omega}_{\oplus\mathcal{I}} \times\right] \boldsymbol{r}_{\mathcal{I}} \tag{2.40}$$

where $\boldsymbol{r}_{\mathcal{I}}$ and $\boldsymbol{v}_{\mathcal{I}}$ are, respectively, the position and velocity of the spacecraft expressed in frame $\mathcal{I}$ and $\boldsymbol{\omega}_{\oplus\mathcal{I}}$ is given by $\boldsymbol{\omega}_{\oplus\mathcal{I}} = \omega_{\oplus} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ with $\omega_{\oplus\mathcal{I}} = 0.000072921158553\ rad/s$.

Solar Radiation Pressure

Solar Radiation Pressure (SRP) affects the orbit of a spacecraft through momentum exchange between the spacecraft and photons incident on the spacecraft, hence only present when the spacecraft is not in shadow. It outweighs drag in higher altitude orbits. Using the flat plate model, SRP can be characterized by:

$$\boldsymbol{f}_s = -P_{\odot} c_s \sum_{i=1}^{N}{}' \mathbf{S}^i \cos\gamma_s^i \tag{2.41}$$

where the prime on the sum indicates that only plates with $\cos\gamma_s^i > 0$ are included in the summation, $P_{\odot}$ is the pressure of solar radiation and the constant c_s defines how the incident radiation interacts with the spacecraft and for most practical problems takes a value between 1 and 2.

The inclination of the i^{th} plate to the spacecraft Sun vector is given by:

$$\cos\gamma_s^i = \left(\mathbf{A}_{\mathcal{BI}}^T \mathbf{n}_{\mathcal{B}}^i\right)^T \mathbf{d}_{sat\odot} \tag{2.42}$$

where $\mathbf{d}_{sat\odot}$ is a unit vector directed from the spacecraft to the center of the Sun, expressed in Inertial coordinates.

The pressure of solar radiation, $P_{\odot}$, at the position of the spacecraft is given by:

$$P_{\odot} = \frac{F_{\odot}}{c\, \boldsymbol{r}_{sat\odot}^2} \tag{2.43}$$

where $F_{\odot}$ is the solar constant, the flux densitiy of solar radiation at a distance of 1 AU from the Sun, c is the speed of light and $\boldsymbol{r}_{sat\odot}$ is the position vector from the satellite to the Sun.

Perturbation Torques

External undesirable torques change the overall momentum of the spacecraft. The most representative perturbation torques are the gravity-gradient torque, $\boldsymbol{L}_g$, magnetic torque, $\boldsymbol{L}_m$, aerodynamic torque, $\boldsymbol{L}_a$, and solar radiation pressure torque, $\boldsymbol{L}_s$, all represented in body frame coordinates.

Gravity-gradient Torque

Any non-symmetrical rigid body in a gravity field is subject to a gravity-gradient torque. Approximating the gravity field as spherically symmetric for computing this torque, it can be written that:

$$\boldsymbol{L}_g^c = \frac{3\mu_\oplus}{r^3}\mathbf{o} \times (\boldsymbol{J}^c\mathbf{o}) \tag{2.44}$$

where $\boldsymbol{r}$ is the radius vector from the center of the central body to the satellite and $r \equiv ||\boldsymbol{r}||$, $\mathbf{o}$ is the body frame representation of a nadir-pointing unit vector, and $\boldsymbol{J}^c$ is the moment of inertia tensor about the center of mass.

Magnetic Torque

The torque generated by a magnetic dipole **m** in a magnetic field ***V*** is given by:

$$\boldsymbol{L}_m = \mathbf{m} \times \boldsymbol{V} \tag{2.45}$$

Near Earth, the magnitude of the Earth's magnetic field is still high enough for it to be used by magnetic control torquers, and to create magnetic disturbance torques from undesirable magnetic dipoles onboard. In higher orbits, the Earth's magnetic field and its effects become negligible.

Aerodynamic Torque

The aerodynamic forces also translate in undesired aerodynamic torques.

With the inclination of the i^{th} plate to the relative velocity in $\mathcal{B}$ frame, being given by:

$$\cos\gamma_a^i = \frac{\mathbf{n}_{\mathcal{B}}^i . \boldsymbol{v}_{rel_{\mathcal{B}}}}{||\boldsymbol{v}_{rel_{\mathcal{B}}}||} \tag{2.46}$$

The aerodynamic force on the i^{th} plate is:

$$\boldsymbol{f}_a^i = -\frac{1}{2}\rho C_D ||\boldsymbol{v}_{rel_{\mathcal{B}}}|| \boldsymbol{v}_{rel_{\mathcal{B}}} S_i \max(\cos\gamma_a^i, 0) \tag{2.47}$$

The aerodynamic torque on the spacecraft is then given by:

$$\boldsymbol{L}_a^i = \sum_{i=1}^{N} \boldsymbol{r}^i \times \boldsymbol{f}_a^i \tag{2.48}$$

where $\boldsymbol{r}^i$ is the vector from the spacecraft center of mass to the center of pressure of the i^{th} plate.

Solar Radiation Pressure Torque

Following the same logic as the SRP acceleration, SRP torque only affects the spacecraft when not in shadow.

The angle between the Sun vector and the normal to the i^{th} plate is given by:

$$\cos\gamma_s^i = \mathbf{n}_{\mathcal{B}}^i . \mathbf{d}_{sat\odot_{\mathcal{B}}} \tag{2.49}$$

where $\mathbf{d}_{sat\odot_{\mathcal{B}}}$ is the spacecraft to Sun unit vector represented in frame $\mathcal{B}$.

The SRP force on the i^{th} plate is given by:

$$\boldsymbol{f}_s^i = -P_{\odot} c_s S^i \max(\cos\gamma_s^i, 0) \tag{2.50}$$

and, finally, the SRP torque on the spacecraft is:

$$\boldsymbol{L}_s = \sum_{i=1}^{N} \boldsymbol{r}^i \times \boldsymbol{f}_s^i \tag{2.51}$$

where $\boldsymbol{r}^i$ is the vector from the spacecraft center of mass to the center of pressure of the SRP on the i^{th} plate.

Chapter 3

Attitude Hardware

One of the fundamental parts of the ADCS of a spacecraft is the suite of sensors and actuators on board. Sensors are generally employed to carry out attitude related measurements i.e. quantities that contain information relative to the orientation of the satellite, as its angular rate or the direction of the magnetic field, the Sun or other stars. Actuators are used to stabilize or maneuver the attitude of the satellite.

The rapid development of micro-electronics, integrated circuit (IC) technologies and micro electromechanical systems (MEMS) enabled the important reduction in size and mass of historically and typically used attitude hardware. These miniaturized sensors and actuators, although sometimes less reliable, are smaller, lighter, cheaper and consume less power, making them appropriate to the employment in nanosatellites. Experimentation and development continues, with this kind of sensors and actuators showing great improvement.

This chapter provides an overview of different characteristic sensors and actuators for spacecraft attitude determination and control, in sections 3.1 and 3.2 respectively, briefly explaining their operation. A sensor suite is selected in section 3.3 and their modeling is explained in section 3.4.

3.1 Sensors

Attitude sensors can be divided into two main categories, reference sensors and inertial sensors. Reference sensors measure in body coordinates the direction and, in some cases, magnitude of known vectors. Among the examples are the Sun sensor, the star tracker and the magnetometer. Inertial sensors such as the gyroscope measure angular rates. Often these 2 classes of sensors are used to complement one another in a measurement system.

3.1.1 Sun Sensor

Sun sensors measure the azimuth and elevation of the Sun vector, providing a measurement of the Sun direction, hence considered a line-of-sight (LOS) sensor. As the orientation about the Sun direction cannot be measured, only two axes of attitude knowledge are obtained. As expected, these measurements are not available during periods of eclipse.

Sun sensors are one of the most widely used sensor type. Its usage and versatility is owed mainly to Sun's small angular radius in almost any orbit, which enables a point-source approximation and to its brightness relative to other astronomical objects, which enables simple equipment to sense it. According to [28] Sun sensors have fields of view (FOV) ranging from several square arc-minutes to approximately $\pi \ sr$, and resolutions from several degrees to less than an arc-second.

Sun Sensors vary from the most simple and reliable, but least accurate Coarse Sun Sensor (CSS), to the fine and digital Sun sensor, used whenever a better accuracy is needed. The CSS combines the variations with the Sun angle of the outputted current generated from attached photodiodes or solar panels and is used for safe-hold control of a spacecraft or in situations where there is no demand in accurate pointing. The fine and digital Sun sensor lets the Sun light pass through a narrow slit and manages to provide a digital representation of the angle between Sun radiation and the plane passing through the slit. A pair of these sensors arranged orthogonally provides two axes of attitude measurements.

The Sun sensor provides corrupted vector measurements of the Sun direction, in body coordinates, according to the model introduced in section 3.4.

3.1.2 Magnetometer

Magnetometers are vector sensors, sensing the direction and magnitude of the magnetic field, only providing two axes of attitude measurement. They are generally lightweight, reliable, require low power and with no moving parts. To use a magnetometer for attitude determination, a geomagnetic field model such as the World Magnetic Model (WMM) [32] is required for comparison.

There are various types of magnetometers, ranging from quantum magnetometers, using fundamental atomic properties, to induction magnetometers (search coil and fluxgate for example) based on Faraday's Law of Magnetic Inductance, and to MEMS magnetometers. The relation between size and accuracy make the MEMS magnetometers attractive for nanosatellite missions.

Magnetometers are only used for low accuracy attitude determination. They measure the sum of the external magnetic field that is of interest and any local field produced by the spacecraft, making them liable for perturbations from local fields produced by parts of the spacecraft or the Sun, that many times cannot be accurately known and compensated for. As Earth's magnetic field strength decreases with distance to Earth as $\frac{1}{r^3}$, generally magnetometers usage is limited to spacecraft below $1000km$ of altitude, hence to Low-Earth Orbit (LEO) [28].

The magnetometer provides corrupted vector measurements of all three components of the Earth's magnetic field, in body coordinates, according to the model introduced in section 3.4.

3.1.3 Star Tracker

Stars are available anywhere in the sky, being independent of the mission and orbit chosen, hence providing an highly accurate and available reference to any satellite. State-of-the-art star trackers are cameras that detect, identify and track several stars in their FOV, and then identify them in an internal catalog, being able to determine the attitude of the satellite using methods such as the ones described in chapter 4. Throughout the years, star trackers have evolved from heavy, expensive and power consuming, to miniaturized sensors which are light, less expensive and consume less power, but still maintaining a high accuracy in determining attitude within a few arcseconds of the true value, making them able to be employed in nanosatellite missions and superior to any other attitude sensors.

A state-of-the-art star tracker may output full attitude description, in a rotation matrix, three Euler angles or quaternion format, or as an alternative, output the directions of the stars in its FOV and the corresponding unit vectors in a inertial reference coordinate system. On this work, the latter is primarily considered for the use in the attitude determination methods of chapter 4.

The star tracker is a LOS sensor that provides corrupted vector measurements of the direction to several stars, in body coordinates, according to the model introduced in section 3.4.

3.1.4 Gyroscope

A gyroscope, or gyro for short, is a device that measures angular rates. Gyroscopes can generally be divided into Rate Gyros (RGs), that measure and output spacecraft angular rates, which may be integrated by an onboard computer to provide an estimate of attitude displacements from a initial reference, and into Rate Integrating Gyros (RIGs), that measure and output angular displacements directly.

Different types of gyroscopes exist concerning the physical mechanisms used for the measurements: spinning mass gyros, optical gyros and Coriolis vibratory gyros. Spinning mass gyros rely on the conservation of angular momentum of a rapidly rotating mass in inertial space to perform the measurements. Optical gyroscopes rely on the Sagnac effect which applies to light traveling in opposite directions around a closed path that is rotating with respect to an inertial frame. Coriolis vibratory gyroscopes (CVGs) sense the motion of a vibrational mode in a structure caused by Coriolis forces. CVGs manufactured with MEMS technology have low cost, mass and power requirements, but also low performance and a short lifetime compared to other gyros. The main gyro types used in small spacecraft are fiber optic gyros (FOGs) and MEMS gyros, with FOGs offering better performance at a mass and cost penalty.

Errors in RGs outputs are generally caused by nonlinearity, drift, and hysteresis.

3.1.5 Other Attitude Sensors

Some other sensors may be integrated in the ADCS system of a nanosatellite. Other examples are the Earth horizon sensors and the Global Positioning System (GPS) Receivers. Earth horizon sensors detect points on the Earth's horizon and with multiple points are able to compute the nadir vector. GPS Receivers can estimate the attitude of the spacecraft by measuring the phase of a GPS carrier signal received by different antennas.

3.2 Actuators

Being the primary focus of this work to study the attitude determination of a spacecraft and not its control and maneuvers, only a brief review of the most used actuators on board of small satellites is done in this section. Magnetic torquers, reaction wheels and thrusters are highlighted due to its importance and dominance.

Magnetic torquers or magnetorquers are used to control attitude in LEO, with high reliability and small mass, having a significant importance in the detumbling and stabilization phases. Sending a current through a coil will create a magnetic field perpendicular to the current loop that will interface with the Earth Magnetic Field and produce useful torque acting on the satellite and making it rotate. As control torques can only be provided in the plane perpendicular to the local magnetic field, full 3-axes stabilization is not possible at any given time using only magnetorquers.

Reaction wheels are used as the primary attitude control on most spacecraft and are now also dominant in nanosatellites, providing high accuracy. An electric motor attached to a flywheel making the rotation speed change causes the spacecraft to rotate through conservation of angular momentum. One or two reaction wheels can be used by momentum-bias spacecrafts, but 3 or more reaction wheels are needed for full three-axes attitude control. A four wheel configuration is often used to provide fault tolerance [33]. Due to external torques, reaction wheels need to be periodically de-saturated using other actuators providing external torques.

Miniaturized propulsion systems are now available, usually cold-gas based with high specific energy and efficiency, low power consumption and small volume and mass. They can be used for off-loading and attitude control.

3.3 Sensor Selection

The sensor selection for a nanosatellite mission follows strict constraints in terms of size, volume, mass, power consumption, cost and performance. The final selection is strongly determined by the latter and by the pointing requirements and goals of the mission. The 2 missions under the scope of the NANOSTAR Project require a precise sensor suite and full three axis knowledge.

Remembering from chapter 1, two different missions are in preliminary design. Depending on the mission, we can either be dealing with a LEO, or with orbits that gradually get farther and farther away from Earth until an orbit around the Moon.

To determine attitude, at least two external vector measurements are needed. A magnetometer may be used in LEO, but it will soon be useless as the distance to Earth increases significantly. A Sun sensor should, in principle, provide useful measurements, at a high frequency in any of the missions, but long duration eclipses may perturb the high pointing accuracy needed. A star tracker, providing the directions of stars available in its FOV would offer better conditions in terms of overall availability of measurements and accuracy, but it is a high cost sensor, and, generally, with low frequency, giving the estimators difficulties to determine the attitude with acceptable error if the angular velocity is high.

Inertial platforms are useful to hold and propagate external measurements or states, and give precise and important angular rate knowledge, at high rate.

Therefore, a sensor suite composed by an inertial unit containing a rate gyroscope and a magnetometer, with the addition of a Sun sensor and a star tracker is proposed. The rate gyroscope will provide the important angular rate knowledge, and the star tracker the accurate, but low rate, external measurements. Those accurate external measurements will be complemented by high rate measurements coming from the Sun sensor, and also the magnetometer in LEO.

3.4 Sensors Measurement Models

This section does not intend to expand on sensor mathematical models, hence simple but realistic measurement and noise models are used.

All attitude sensors are assumed to provide vector measurements that depend directly on the attitude alone. Two types of reference sensors are considered: complete vector sensors, which measure all three components of a quantity, and LOS sensors, that measure only a direction. In both cases, the measurements are assumed to be corrupted by noise and to be inherently discrete-time, following the model:

$$\mathbf{b}_k = \mathbf{A}_{\mathcal{BI}}\mathbf{r}_k + \boldsymbol{\eta}_k \tag{3.1}$$

where the subscript k represents the quantity at time k, $\mathbf{r}$ is a known reference vector in inertial coordinates, $\boldsymbol{\eta}_k$ is a vector of discrete noise, and $\mathbf{b}_k$ is the corrupted vector measurement in body coordinates.

Line-of-Sight (LOS) sensors measure the elevations of the line of sight as projected onto mutually perpendicular planes which contain the sensor boresight. Denoting these two angles as $\bar{\alpha}$ and $\bar{\zeta}$, the true measurement model for the sensed line of sight is given by:

$$\hat{\mathbf{b}}^{true} = \frac{1}{\sqrt{1+\tan^2(\bar{\alpha})+\tan^2(\bar{\zeta})}}\begin{bmatrix}\tan(\bar{\alpha})\\ \tan(\bar{\zeta})\\ 1\end{bmatrix} \tag{3.2}$$

where the caret denotes a unit vector measurement.

Reference [12] indicates that nearly all probability of the errors of a LOS measurement is concentrated on a very small area about the direction of $\hat{\mathbf{b}}^{true}$, so the sphere containing that point can be approximated by a tangent plane characterized by:

$$\hat{\mathbf{b}}_k = \mathbf{A}_{\mathcal{BI}}\hat{\mathbf{r}}_k + \boldsymbol{\eta}_k \tag{3.3}$$

where the sensor error $\boldsymbol{\eta}_k$ is approximately Gaussian, satisfying:

$$E\{\boldsymbol{\eta}_k\} = \mathbf{0} \tag{3.4a}$$

$$\mathbf{R}^{QUEST} \equiv E\{\boldsymbol{\eta}_k\boldsymbol{\eta}_k^T\} = \sigma^2\left(\mathbf{I}_3 - \hat{\mathbf{b}}^{true}\hat{\mathbf{b}}^{true^T}\right) \tag{3.4b}$$

where $E\{\}$ denotes expectation, $\mathbf{I}_3$ denotes a 3×3 identity matrix and σ is the standard deviation of the error assumed for the LOS sensor. Equation 3.4b is known as the QUEST Measurement model (QMM) [12, 34] and is used instead of the true model.

On the other hand, complete vector measurements following equation 3.1 do not provide measurement vectors with unit norm and the noise is usually assumed to be Gaussian, white, and with covariance $\mathbf{R}_{\eta_k}$, proportional to the identity matrix. As the magnitude of the observed vector does not contain relevant information for attitude determination, and the use of only unit vectors can be an advantage in some filters implementations [35], equation 3.1 can be normalized, giving the measurement model for the complete vector sensors as:

$$\hat{\mathbf{b}}_k = \mathbf{A}_{\mathcal{BI}}\hat{\mathbf{r}}_k + \hat{\boldsymbol{\eta}}_k \tag{3.5}$$

where $\hat{\boldsymbol{\eta}}_k$ is to good approximation Gaussian, white and related to $\boldsymbol{\eta}_k$ according to:

$$\hat{\boldsymbol{\eta}}_k = -\frac{1}{||\mathbf{r}_k||}\left[\mathbf{A}_{\mathcal{BI}}\hat{\mathbf{r}}_k\times\right]^2\boldsymbol{\eta}_k \tag{3.6}$$

The measurement covariance matrix of the unitized measurement is then given by:

$$\mathbf{R}_{\hat{\eta}_k} \equiv \frac{1}{||\mathbf{r}_k||^2}\left[\mathbf{A}_{\mathcal{BI}}\hat{\mathbf{r}}_k\times\right]^2\mathbf{R}_{\eta_k}\left[\mathbf{A}_{\mathcal{BI}}\hat{\mathbf{r}}_k\times\right]^2 \tag{3.7}$$

and as $\mathbf{R}_{\eta_k}$ is assumed proportional to the identity matrix, the covariance matrix of the QMM is reached:

$$\mathbf{R}_{\hat{\boldsymbol{\eta}}_k}^{QUEST} = \sigma_{\hat{\boldsymbol{\eta}}_k}^2(\mathbf{I}_3 - (\mathbf{A}_{\mathcal{BI}}\hat{\mathbf{r}}_k)(\mathbf{A}_{\mathcal{BI}}\hat{\mathbf{r}}_k)^T) \tag{3.8}$$

which was also the assumed covariance matrix for the LOS sensors.

Therefore, all the reference sensors provide unitized measurements following the QMM. The QMM has been widely used in spacecraft attitude estimation. The noise is approximated to lie in the tangent plane of the respective unit vector measurement, being axially symmetrically distributed about it. The QMM covariance is shown to be a reasonable approximation of the true covariance if the FOV is not too large but is necessarily singular. However, that singularity arising from the normalization of the measurement, can be set aside, as the attitude is insensitive to the length of the vector, and the singular measurement covariance matrix can be replaced with an equivalent one which is a multiple of the three dimensional identity matrix [34], having an efficient and simple implementation in the Kalman filter.

The measurement model that describes the gyroscope is the following continuous model:

$$\boldsymbol{\omega}_m(t) = \boldsymbol{\omega}(t) + \boldsymbol{\beta}(t) + \boldsymbol{\eta}_v(t) \tag{3.9a}$$

$$\dot{\boldsymbol{\beta}}(t) = \boldsymbol{\eta}_u(t) \tag{3.9b}$$

where $\boldsymbol{\omega}(t)$ is the true rate, $\boldsymbol{\omega}_m(t)$ is the measured rate, $\boldsymbol{\beta}(t)$ is the true bias or drift, and $\boldsymbol{\eta}_v(t)$ and $\boldsymbol{\eta}_u(t)$ are independent zero-mean Gaussian white-noise processes.

Chapter 4

Attitude Determination Methods

Chapter 3 described both the hardware and the principles behind gathering attitude related relevant data. This chapter describes the procedures by which that data is processed to determine the spacecraft attitude. The complete determination of the orientation of the three axes of the spacecraft body frame with respect to the inertial frame is desired, therefore only solutions capable of providing the three-axis attitude knowledge are considered.

Attitude determination methods come in various forms and an overview is given in section 1.2. The first appeared methods to solve the problem of determining the attitude of a spacecraft are referred in this work as deterministic methods or point-to-point approaches. These compute the attitude based on two or more vector measurements taken at the same time, in a purely algebraic way, without the use of motion models or information from past attitude.

Years after, the tendency soon started to be using estimation algorithms to solve the problem of attitude instead of deterministic ones. Solutions based on the Kalman filter started to be designed. These filtering approaches used dynamic models of the spacecraft's motion, and took into account past estimates as well as measurements coming from the sensors.

Limitations of the solutions based on Kalman Filtering led the research of new methods to determine the attitude. Among these, nonlinear observers have emerged as an attractive alternative given the possibility to establish convergence bounds and provide stability guarantees, with an associated lower computational cost, and a more straightforward tuning of the design parameters.

This chapter documents the theory underlying the attitude determination methods that are the focus of this work. Section 4.1 describes deterministic methods for combining measurements to produce a determination of the attitude, with special emphasis on the QUEST algorithm. Section 4.2 describes filtering methods to provide estimates of the attitude, with special emphasis on the development of the MEKF for attitude estimation. Section 4.3 presents a nonlinear observer with a sensor based framework at its core, reviewing its design.

4.1 Deterministic Methods

Deterministic methods compute the attitude solely based on vector measurements coming from the sensors, in a purely algebraic way, without the use of motion models or information from past attitude. These methods are very advantageous due to their low computational efforts, independence of attitude motion models and easily application but have serious limitations, such as the requirement of a minimum of two measurements.

As described in chapter 3, attitude related measurements are naturally represented as unit vectors, being typical examples the unit vectors giving the direction to the Sun, a star, or of the Earth's magnetic field.

The first static attitude determination method developed traces back to 1964 when Harold D. Black published what is usually called the TRIAD algorithm [8], which enabled the determination of the attitude of a spacecraft using two body and reference observations. To fight some of the flaws of TRIAD, one year later, Grace Wahba formulated a general criterion for attitude determination using two or more vector measurements [9]. The TRIAD algorithm and some solutions to the Wahba's Problem are discussed in this section, as well as some statistical factors such as the attitude error covariance that can be obtained for these solutions. The algorithms formulation in this section follows from references [10–12]

4.1.1 The TRIAD Algorithm

As mentioned before, the TRIaxial Attitude Determination (TRIAD) algorithm, first published by Harold D. Black, uses exactly two vector measurements to compute the attitude. It is a deterministic, non-optimal solution incorporating just enough observations to uniquely determine the attitude.

The method is based on the fact that, if two unit vectors are available in the spacecraft body frame, that are measurements of some quantity, and are also available the respective unit vectors of the same quantity, but this time represented in some reference frame, a relation can be used to obtain its attitude. As referred, 3 independent quantities are necessary to fully determine the attitude. By using only one measurement, owing to the norm constraint, only the orientation of the satellite axis relative to the vector line could be obtained, but not the degree of freedom involving the rotation about this line. To determine all 3 degrees of freedom, it is sufficient to use any other linearly independent vector in the same conditions as the first one. In fact, two vectors over-determine the attitude.

Denoting the representation of the two unit vectors in the body frame as $\mathbf{b}_1$ and $\mathbf{b}_2$, and the corresponding vectors in the reference frame by $\mathbf{r}_1$ and $\mathbf{r}_2$, the attitude matrix $\mathbf{A}$ that rotates vectors from the reference frame to the spacecraft body frame is such that:

$$\mathbf{A}\mathbf{r}_1 = \mathbf{b}_1 \tag{4.1a}$$

$$\mathbf{A}\mathbf{r}_2 = \mathbf{b}_2 \tag{4.1b}$$

Only for error free measurements the same solution could be found to the equations 4.1a and 4.1b, so the classical TRIAD algorithm is based on the assumption that $\mathbf{b}_1$ unit vector is more accurate than

$\mathbf{b}_2$, so the estimate satisfies exactly the equation 4.1a, and only approximately the equation 4.1b. In practical, the TRIAD algorithm determines the attitude by first discarding part of the measurements so that a solution exists.

The implementation of TRIAD passes by defining two orthonormal right-handed triad of vectors, one in the reference frame $(\mathfrak{v}_1, \mathfrak{v}_2, \mathfrak{v}_3)$, and one in the body frame $(\mathfrak{w}_1, \mathfrak{w}_2, \mathfrak{w}_3)$. The unique orthogonal matrix $\mathbf{A}$ that satisfies $\mathbf{A}\mathfrak{v}_i = \mathfrak{w}_i$ is given by:

$$\mathbf{A} \equiv \sum_{i=1}^{3} \mathfrak{w}_i \mathfrak{v}_i^T \tag{4.2}$$

with the triad $(\mathfrak{v}_1, \mathfrak{v}_2, \mathfrak{v}_3)$ formed from $\mathbf{r}_1$ and $\mathbf{r}_2$ and $(\mathfrak{w}_1, \mathfrak{w}_2, \mathfrak{w}_3)$ formed from $\mathbf{b}_1$ and $\mathbf{b}_2$ by means of:

$$\mathfrak{v}_1 = \mathbf{r}_1, \qquad \mathfrak{v}_2 = \mathbf{r}_x \equiv \frac{\mathbf{r}_1 \times \mathbf{r}_2}{||\mathbf{r}_1 \times \mathbf{r}_2||}, \qquad \mathfrak{v}_3 = \mathbf{r}_1 \times \mathbf{r}_x \tag{4.3a}$$

$$\mathfrak{w}_1 = \mathbf{b}_1, \qquad \mathfrak{w}_2 = \mathbf{b}_x \equiv \frac{\mathbf{b}_1 \times \mathbf{b}_2}{||\mathbf{b}_1 \times \mathbf{b}_2||}, \qquad \mathfrak{w}_3 = \mathbf{b}_1 \times \mathbf{b}_x \tag{4.3b}$$

4.1.2 Wahba's Problem

TRIAD, although powerful and simple, presented major limitations such as only being able to process two measurements, and its suboptimality in any statistical sense. Improvements were made by a problem posed in 1965 by Grace Wahba [9]. Wahba's problem is to find the orthogonal matrix $\mathbf{A}$ with determinant $+1$ that minimizes the following loss function:

$$L(\mathbf{A}) = \frac{1}{2} \sum_{i=1}^{N} a_i ||\mathbf{b}_i - \mathbf{A}\mathbf{r}_i||^2 \tag{4.4}$$

where $\mathbf{b}_i$ and $\mathbf{r}_i$ follow from last section and a_i are non-negative weights.

The loss function of equation 4.4 can be written in the form:

$$L(\mathbf{A}) = \lambda_0 - tr(\mathbf{A}\mathbb{B}^T) \tag{4.5}$$

with:

$$\lambda_0 \equiv \sum_{i=1}^{N} a_i, \qquad \mathbb{B} \equiv \sum_{i=1}^{N} a_i \mathbf{b}_i \mathbf{r}_i^T \tag{4.6}$$

The minimum loss function is achieved when $tr(\mathbf{A}\mathbb{B}^T)$ is maximized.

Several algorithms for solving Wahba's problem exist differing on computational load and the parameterization they solve to, either the attitude matrix directly or its quaternion representation.

4.1.3 Davenport's q Method

Paul Davenport presented the first useful solution to Wahba's problem for spacecraft attitude determination. His solution begins by applying some substitutions and using several quaternion relations, to the loss function of equation 4.5, finally expressing it as:

$$L(\mathbf{A}(\mathbf{q})) = \lambda_0 - \mathbf{q}^T \mathbb{K}(\mathbb{B})\mathbf{q} \tag{4.7}$$

where $\mathbb{K}(\mathbb{B})$ is the symmetric traceless matrix:

$$\mathbb{K}(\mathbb{B}) = \sum_{i=1}^{N} a_i [\mathbf{b}_i \otimes]^T [\mathbf{r}_i \odot] \tag{4.8}$$

The optimal attitude is represented by the unit quaternion that minimizes the right side of equation 4.7, and that is true when the quaternion is equal to the normalized eigenvector of $\mathbb{K}(\mathbb{B})$ with the largest eigenvalue.

The optimized loss function is then equal to:

$$L(A(\widehat{\mathbf{q}})) = \lambda_0 - \lambda_{max} \tag{4.9}$$

Very robust algorithms exist to solve the symmetric eigenvalue problem, which makes the Davenport's q method very robust, but considerably computationally heavy.

4.1.4 Quaternion Estimator (QUEST)

The QUEST algorithm appearance was justified by the difficulty of the computers at the time to perform the calculation of exact eigenvalues in Davenport's solution. QUEST maintained all the computational advantages of a fast deterministic algorithm like TRIAD while still yielding an optimal result like Davenport, and became the most widely used algorithm to solve Wahba's problem.

Davenport's eigenvalue condition can be expressed as:

$$\mathbf{0}_4 = \mathbb{H}(\lambda_{max})\widehat{\mathbf{q}} \tag{4.10}$$

where:

$$\mathbb{H}(\lambda) \equiv \lambda \mathbf{I}_4 - \mathbb{K}(\mathbb{B}) = \begin{bmatrix} (\lambda + tr(\mathbb{B}))\mathbf{I}_3 - \mathbb{S} & -\mathbb{z} \\ -\mathbb{z}^T & \lambda - tr(\mathbb{B}) \end{bmatrix} \tag{4.11}$$

with:

$$\mathbb{z} \equiv \begin{bmatrix} B_{23} - B_{32} \\ B_{31} - B_{13} \\ B_{12} - B_{21} \end{bmatrix} = \sum_{i=1}^{N} a_i (\mathbf{b}_i \times \mathbf{r}_i)\,, \qquad \mathbb{S} \equiv \mathbb{B} + \mathbb{B}^T \tag{4.12}$$

Equation 4.10 is equivalent to the following two equations:

$$(\zeta \mathbf{I}_3 - \mathbb{S})\,\widehat{\mathbf{q}}_{1:3} = \widehat{q}_4 \mathbb{z} \tag{4.13a}$$

$$(\lambda_{max} - tr(\mathbb{B}))\,\widehat{q}_4 - \mathbb{z}^T \widehat{\mathbf{q}}_{1:3} = 0 \tag{4.13b}$$

where:

$$\zeta \equiv \lambda_{max} + tr(\mathbb{B}) \tag{4.14}$$

If λ_{max} was known *a priori*, the optimal quaternion would be given as:

$$\widehat{\mathbf{q}} = \bar{\alpha} \begin{bmatrix} adj(\zeta \mathbf{I}_3 - \mathbb{S})\mathbb{z} \\ det(\zeta \mathbf{I}_3 - \mathbb{S}) \end{bmatrix} \tag{4.15}$$

where $\bar{\alpha}$ normalizes $\widehat{\mathbf{q}}$. If equation 4.15 is substituted into 4.13b, an implicit equation for the maximum eigenvalue is obtained, which is the characteristic equation of $\mathbb{K}(\mathbb{B})$:

$$(\lambda_{max} - tr(\mathbb{B}))det(\zeta \mathbf{I}_3 - \mathbb{S}) - \mathbb{z}^T adj(\zeta \mathbf{I}_3 - \mathbb{S})\mathbb{z} = 0 \tag{4.16}$$

Equation 4.16 can be written as a quartic equation for λ:

$$0 = [\lambda^2 - (tr(\mathbb{B}))^2 + \kappa][\lambda^2 - (tr(\mathbb{B}))^2 - ||\mathbb{z}||^2] - (\lambda - tr(\mathbb{B}))(\mathbb{z}^T\mathbb{S}\mathbb{z} + det\mathbb{S}) - \mathbb{z}^T\mathbb{S}^2\mathbb{z} \tag{4.17}$$

or under the following characteristic polynomial form:

$$f(\lambda) = \lambda^4 - (a+b)\lambda^2 - c\lambda + (ab + c\varrho - d) = 0 \tag{4.18}$$

with:

$$a = \varrho^2 - \kappa \tag{4.19a}$$

$$b = \varrho^2 + \mathbb{z}^T\mathbb{z} \tag{4.19b}$$

$$c = \varDelta + \mathbb{z}^T\mathbb{S}\mathbb{z} \tag{4.19c}$$

$$d = (\mathbb{z}^T\mathbb{S})(\mathbb{S}\mathbb{z}) \tag{4.19d}$$

$$\varrho = 0.5tr(\mathbb{S}) \tag{4.19e}$$

$$\kappa \equiv tr(adj(\mathbb{S})) \tag{4.19f}$$

$$\varDelta \equiv det(\mathbb{S}) \tag{4.19g}$$

Only the largest root, λ_{max}, of equation 4.17 is of interest. λ_{max} can be obtained by Newton-Raphson iteration of the same equation, starting from λ_0 as the initial estimate, since these two values will be nearly the same if the optimized loss function is small, with a single iteration being generally sufficient.

4.1.5 Error Analysis of Wahba's Problem

As important as an estimate, is knowing its accuracy. This subsection provides an estimate of the accuracy of a solution to Wahba's problem.

Remembering that the body frame vectors $\mathbf{b}_i$ and reference frame vectors $\mathbf{r}_i$ are given by:

$$\mathbf{r}_i = \mathbf{r}_i^{true} + \boldsymbol{\eta}_{r_i} \tag{4.20a}$$

$$\mathbf{b}_i = \mathbf{b}_i^{true} + \boldsymbol{\eta}_{b_i} \tag{4.20b}$$

where $\boldsymbol{\eta}_{r_i}$ and $\boldsymbol{\eta}_{b_i}$ are respectively the errors in the reference and body frame vector measurements. These errors are unknown, but some assumptions about them have been made in chapter 3, as the QUEST Measurement Model is the model of choice. The measurement covariance matrices of the noise for each measurement are then given by:

$$\mathbf{R}_{r_i} = \sigma_{r_i}^2 [I_3 - \mathbf{r}_i^{true} (\mathbf{r}_i^{true})^T] = -\sigma_{r_i}^2 [\mathbf{r}_i^{true} \times]^2 \tag{4.21a}$$

$$\mathbf{R}_{b_i} = \sigma_{b_i}^2 [I_3 - \mathbf{b}_i^{true} (\mathbf{b}_i^{true})^T] = -\sigma_{b_i}^2 [\mathbf{b}_i^{true} \times]^2 \tag{4.21b}$$

Additionally, it can be shown that choosing the weights a_i of Wahba's problem to be proportional to the inverses of the measurement variances of each sensor results in the best estimate:

$$a_i = c/\sigma_i^2 \tag{4.22}$$

for some constant c.

Then, using the rotation vector $\boldsymbol{\delta\vartheta}$ introduced in section 2.1.5, representing a rotation between the estimated body frame and the true body frame, represented in the spacecraft's body frame, as the parameterization for the attitude errors, the error covariance of the solution to Wahba's problem with the QMM, is given by:

$$\mathbf{P}_{\boldsymbol{\vartheta\vartheta}} = \left(\sum_{i=1}^{N} \sigma_i^{-2} \left[\mathbf{I}_3 - \mathbf{b}_i^{true} (\mathbf{b}_i^{true})^T \right] \right)^{-1} \tag{4.23}$$

In practice, it would be desired to compute an estimate of the covariance without knowing exactly the true body frame vectors. For that, the predicted body frame vectors $\widehat{\mathbf{A}}\mathbf{r}_i$ could be used instead of the true ones.

4.2 State Estimation Methods

State estimation refers to the process of obtaining the best state estimate, optimum by some measure, of the true system, using a dynamic model and measurements that are both corrupted with random noise of known statistics. The variables to be estimated are usually collected into a state vector. These type of approaches can be used to simultaneously estimate quantities and filter noisy measurement observations. Many times, a filter is also used to infer the state of a system that cannot be measured directly.

In state estimation processes, there are two basic ways to update the state vector. When a new estimate of the state vector is obtained after each observation, the process is called a sequential estimator or recursive. If, otherwise, all the observations are processed and then combined to produce a single update to the state vector, the process is referred to as a batch estimator [28].

This section introduces Kalman filter theory, discussing the concepts of state and observation models, and the formulation of the linear Kalman filter and the non-linear Extended Kalman filter (EKF). Afterwards, the design of an EKF that is typically used for actual onboard applications is presented.

4.2.1 Kalman Filtering

State and Observation Models

Modeling is a fundamental aspect when analyzing the behavior of a system and designing estimation algorithms for it. A system can either be described by a static model or a dynamic one.

Dynamic models are described by differential or difference equations. State vectors, composed by state variables, describe the condition of the system. The number of state variables in the state vector is known as the order of the system. Equation 4.24 describes the general structure of an n^{th}-order dynamic model.

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)), \mathbf{u}(t), t), \qquad \mathbf{x}(t_0) = \mathbf{x}_0 \tag{4.24}$$

where **f** is sufficiently differentiable, **x** is the state vector, and **u** denotes any input to the system that does not depend on the state elements.

One subset of these dynamic systems are the linear ones, which follow the superposition principle and can be represented as:

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t), \qquad \mathbf{x}(t_0) = \mathbf{x}_0 \tag{4.25}$$

where the matrices **F** and **B** are known as the state matrix and input matrix, respectively. The solution to equation 4.25 is given by [11]:

$$\mathbf{x}(t) = \mathbf{\Phi}(t, t_0)\mathbf{x}_0 + \int_{t_0}^{t} \mathbf{\Phi}(t, \tau)\mathbf{B}(\tau)\mathbf{u}(\tau) d\tau \tag{4.26}$$

where $\mathbf{\Phi}(t, t_0)$ is known as the state transition matrix, which has the following properties:

$$\mathbf{\Phi}(t_0, t_0) = \mathbf{I} \tag{4.27a}$$

$$\mathbf{\Phi}(t_0, t) = \mathbf{\Phi}(t, t_0)^{-1} \tag{4.27b}$$

$$\mathbf{\Phi}(t_2, t_0) = \mathbf{\Phi}(t_2, t_1)\mathbf{\Phi}(t_1, t_0) \tag{4.27c}$$

$$\dot{\mathbf{\Phi}}(t, t_0) = \mathbf{F}(t)\mathbf{\Phi}(t, t_0) \tag{4.27d}$$

Observations are gathered as outputs from the sensors. The general form of an observation vector, used to show how a sensor relates to various state quantities and inputs is given by:

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t), t) \tag{4.28}$$

where **h** is the observation vector. Linear observation models follow the same structure as linear state models:

$$\mathbf{y}(t) = \mathbf{H}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t) \tag{4.29}$$

where **H** is the observation matrix, or sensitivity matrix, and **D** is called direct transmission matrix [11].

Equations 4.25 and 4.29 form the state space description of a linear model.

Discrete-time systems have become a standard in most dynamic applications with the use of digital computers. Thus, a discrete time representation for the dynamic and observations models is derived. For that, let's consider a zero-order hold for sampling the system, which holds the sampled point to a constant value throughout the interval. With **F** and **B** constants, or slowly varying, and the time set to the first sample interval, then, equation 4.26, simplifies to:

$$\mathbf{x}(\varDelta t) = \mathbf{\Phi}\mathbf{x}(0) + \mathbf{\Gamma}\mathbf{u}(0) \tag{4.30}$$

where:

$$\mathbf{\Phi} \equiv e^{\mathbf{F}\varDelta t} \tag{4.31a}$$

$$\mathbf{\Gamma} \equiv \left[\int_0^{\varDelta t} e^{\mathbf{F}(t)} dt\right] \mathbf{B} \tag{4.31b}$$

Expanding equation 4.30 for $k+1$ samples and changing the notation in order that the entire discrete state-space is represented, gives:

$$\mathbf{x}_{k+1} = \mathbf{\Phi}\mathbf{x}_k + \mathbf{\Gamma}\mathbf{u}_k \tag{4.32}$$

being $\mathbf{x}_k$ the state at time t_k.

Equation 4.29 may also be represented in discrete time by:

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k \tag{4.33}$$

Kalman Filter

A Kalman Filter is an optimal recursive data processing algorithm [36]. With knowledge of the system and measurement errors, uncertainties in the dynamic models, any available information about the initial condition of the variables of interest, and all the available measurements (regardless of their precision), the Kalman Filter produces an optimum estimate of the desired variables. This estimate is statistically optimal with respect to any quadratic function of the estimation error [37]. The filter is recursive in the sense that it does not require all previous data to be kept in storage and reprocessed every time a new measurement is taken, just the previous one.

The operation of the Kalman Filter starts from an initial condition $\mathbf{x}(t_0)$. However, since this value may not be known precisely *a priori*, it will me modeled as a random vector that is normally distributed. Thus, the initialization stage is defined by the description of $\mathbf{x}(t_0)$, completely specified by the mean $\widehat{\mathbf{x}}_0$ and associated covariance $\mathbf{P}_0$:

$$E\{\mathbf{x}(t_0)\} = \widehat{\mathbf{x}}_0 \tag{4.34a}$$

$$E\{[\mathbf{x}(t_0) - \widehat{\mathbf{x}}_0][(\mathbf{x}(t_0) - \widehat{\mathbf{x}}_0]^T = \mathbf{P}_0 \tag{4.34b}$$

where $\mathbf{P}_0$ is an n-by-n matrix that is symmetric and positive semi-definite.

Afterwards, the filter performs recursively switching between two stages. The first one, called the prediction or propagation stage, propagates the estimate and its covariance from time t_k to t_{k+1}, yielding $\widehat{\mathbf{x}}_{k+1}^-$ and $\mathbf{P}_{k+1}^-$, based on a process describing the system dynamics. Then, comes the correction or update stage, where the *a priori* estimate $\widehat{\mathbf{x}}_{k+1}^-$ and covariance $\mathbf{P}_{k+1}^-$ suffer a correction based on the measurement $\mathbf{y}_{k+1}$, producing the *a posteriori* state estimate $\widehat{\mathbf{x}}_{k+1}^+$ and $\mathbf{P}_{k+1}^+$ for time t_{k+1}. The superscript $-$ is then used to refer to a value at a time prior to a update, whereas the superscript $+$ corresponds to a value already updated.

The dynamic process model is assumed to be in the form of a linear stochastic differential equation, obtained appending a noise term to the equation 4.25:

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{4.35}$$

where $\mathbf{w}$ is a zero-mean Gaussian white-noise process vector with spectral density given by:

$$E\{\mathbf{w}(t)\mathbf{w}^T(\tau)\} = \mathbf{Q}(t)\delta(t-\tau) \tag{4.36}$$

and $\mathbf{G}$ is called process noise distribution matrix. The process noise $\mathbf{w}(t)$ can also be described using the notation $\mathbf{w}(t) \sim N(0, \mathbf{Q}(t))$, where N stands for "normal distribution", the first argument is the mean and the second the spectral density.

Discrete-time noise-corrupted linear measurements $\mathbf{y}_k$ are assumed to be available, modeling the observations by:

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{4.37}$$

where $\mathbf{v}_k$ is a vector of discrete-time white Gaussian noise with statistics:

$$E\{\mathbf{v}_k\} = \mathbf{0} \tag{4.38a}$$

$$E\{\mathbf{v}_k\mathbf{v}_k^T\} = \mathbf{R}_k \tag{4.38b}$$

where $\mathbf{R}_k$ is the noise covariance matrix.

It is also assumed that $\mathbf{x}(t_0)$, $\mathbf{w}$ and $\mathbf{v}$ are independent of each other, and since all are assumed Gaussian, this is equivalent to assuming that they are uncorrelated.

According to the Separation Theorem, the overall estimator system can be designed by separating the estimator from the controller, hence, the control input $\mathbf{u}(t)$ is made null for the following design of the Kalman Filter.

The propagated estimated state, $\widehat{\mathbf{x}}$, obtained by performing the expectation of equation 4.35, follows:

$$\dot{\widehat{\mathbf{x}}}(t) = \mathbf{F}(t)\widehat{\mathbf{x}}(t) \tag{4.39}$$

Defining the residual error, $\mathit{\Delta}\mathbf{x}$, as the difference between the true state in equation 4.35 and the estimated state in equation 4.39:

$$\mathit{\Delta}\mathbf{x} \equiv \mathbf{x} - \widehat{\mathbf{x}} \tag{4.40}$$

The covariance of $\mathit{\Delta}\mathbf{x}$ is given by:

$$\mathbf{P} \equiv E\{\mathit{\Delta}\mathbf{x}\mathit{\Delta}\mathbf{x}^T\} \tag{4.41}$$

Performing the time derivative of equation 4.40, it can be derived that $\mathbf{P}$ obeys the equation:

$$\dot{\mathbf{P}} \equiv \mathbf{FP} + \mathbf{PF}^T + \mathbf{GQG}^T \tag{4.42}$$

which is solved for $\mathbf{P}(t)$ with initial condition $\mathbf{P}(t_0) = \mathbf{P}_0$. The equations 4.39 and 4.42 describe the propagation stage of the Kalman Filter.

At measurement time t_k, the measurement $\mathbf{y}_k$ becomes available. A discrete ”update” equation is required to process the measurements. The linear discrete-time update equation is given by:

$$\widehat{\mathbf{x}}_k^+ = \widehat{\mathbf{x}}_k^- + \mathbf{K}_k[\mathbf{y}_k - \mathbf{H}_k\widehat{\mathbf{x}}_k^-] \tag{4.43}$$

where $\widehat{\mathbf{x}}_k^-$ comes from the propagated estimate using equation 4.39, and $\mathbf{K}_k$ is the Kalman gain matrix.

The updated covariance expression is given by:

$$\mathbf{P}_k^+ = [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\mathbf{P}_k^-[\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T \tag{4.44}$$

where $\mathbf{P}_k^-$ comes from the propagated estimate using equation 4.42.

The optimal Kalman Filter gain $\mathbf{K}_k$ is determined by minimizing the trace of the updated covariance, leading to:

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T[\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k]^{-1} \tag{4.45}$$

Substituting equation 4.45 into 4.44 gives:

$$\mathbf{P}_k^+ = [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\mathbf{P}_k^- \tag{4.46}$$

For an extensive amount of problems, the discrete-time version of the dynamic model of equation 4.35 is usually adequate:

$$\mathbf{x}_{k+1} = \mathbf{\Phi}_k\mathbf{x}_k + \mathbf{\Gamma}_k\mathbf{u}_k + \mathbf{\Upsilon}_k\mathbf{w}_k \tag{4.47}$$

where $\mathbf{w}_k$ is a zero-mean Gaussian white-noise process vector with covariance $\mathbf{Q}_k$.

With this formulation, and employing again the Separation theorem, the discrete time estimate propagation is given by:

$$\widehat{\mathbf{x}}_{k+1}^{-} = \boldsymbol{\Phi}_k \widehat{\mathbf{x}}_k^{+} \tag{4.48}$$

and the discrete-time covariance propagation is given by [11]:

$$\mathbf{P}_{k+1}^{-} = \boldsymbol{\Phi}_k \mathbf{P}_k^{+} \boldsymbol{\Phi}_k^T + \boldsymbol{\Upsilon}_k \mathbf{Q}_k \boldsymbol{\Upsilon}_k^T \tag{4.49}$$

Table 4.1 summarizes the discrete-time Kalman Filter formulation.

Discrete-time Kalman Filter		
Model	$\mathbf{x}_{k+1} = \boldsymbol{\Phi}_k \mathbf{x}_k + \boldsymbol{\Gamma}_k \mathbf{u}_k + \boldsymbol{\Upsilon}_k \mathbf{w}_k,$ $\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k,$	with $\mathbf{w}_k \sim N(0, \mathbf{Q}_k)$ with $\mathbf{v}_k \sim N(0, \mathbf{R}_k)$
Initial condition	$\widehat{\mathbf{x}}(t_0)$=$\widehat{\mathbf{x}}_0$ $\mathbf{P}(t_0)$=$\mathbf{P}_0$	
Propagation	$\widehat{\mathbf{x}}_{k+1}^{-} = \boldsymbol{\Phi}_k \widehat{\mathbf{x}}_k^{+}$ $\mathbf{P}_{k+1}^{-} = \boldsymbol{\Phi}_k \mathbf{P}_k^{+} \boldsymbol{\Phi}_k^T + \boldsymbol{\Upsilon}_k \mathbf{Q}_k \boldsymbol{\Upsilon}_k^T$	
Gain	$\mathbf{K}_k = \mathbf{P}_k^{-} \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k^{-} \mathbf{H}_k^T + \mathbf{R}_k]^{-1}$	
Update	$\widehat{\mathbf{x}}_k^{+} = \widehat{\mathbf{x}}_k^{-} + \mathbf{K}_k [\mathbf{y}_k - \mathbf{H}_k \widehat{\mathbf{x}}_k^{-}]$ $\mathbf{P}_k^{+} = [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \mathbf{P}_k^{-}$	

Table 4.1: Discrete-time Kalman Filter formulation.

Extended Kalman Filter

Consider the general continuous nonlinear model:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{G}(t)\mathbf{w}(t)\,, \qquad \mathbf{w}(t) \sim N(0, \mathbf{Q}(t)) \tag{4.50a}$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t), t) + \mathbf{v}(t), \qquad \mathbf{v}(t) \sim N(0, \mathbf{R}(t)) \tag{4.50b}$$

or its discrete time equivalent:

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k) + \boldsymbol{\Upsilon}_k \mathbf{w}_k, \qquad \mathbf{w}_k \sim N(0, \mathbf{Q}_k) \tag{4.51a}$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k, \qquad \mathbf{v}_k \sim N(0, \mathbf{R}_k) \tag{4.51b}$$

where $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$ and $\mathbf{h}(\mathbf{x}(t), t)$ are continuously differentiable nonlinear functions, and $\mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k)$ and $\mathbf{h}_k(\mathbf{x}_k)$ are discrete-time differentiable nonlinear functions.

One of the assumptions made for the development of the Kalman filter in section 4.2.1, was that the system could be described by a linear model. However, a large class of estimation problems involve nonlinear models such as the ones in equations 4.50 and 4.51. This bears the issue that the probability density function of the noise is altered as it is transmitted through the nonlinear element. Meaning, a Gaussian output is not guaranteed by a Gaussian input, as in the linear case.

However, many of this nonlinear systems can be approximated locally by a linear one. That is, the functional dependence of the measurement or state dynamics on the system are nonlinear, but approximately linear for small perturbations in the values of the state variables. This means that methods of linear estimation theory, such as the Kalman Filter, may be applied to such nonlinear problems, by linear approximation of the effects of small perturbations in the state of the nonlinear system from a ”nominal” value [37].

For the cases which operational performance depends on staying close to an optimal trajectory, the nominal values of the state variables are well known beforehand. For these applications, the estimation problem can be linearized about this nominal trajectory, being able to compute the Kalman gains beforehand, but the deviation of the actual trajectory from the nominal one tends to increase with time. The nominal trajectory may also be defined ”on the fly” as the current best estimate of the actual trajectory. In this way, the perturbations include only the state estimation errors, generally smaller than the perturbations from any predefined nominal trajectory, and therefore better conditioned for linear approximation. However, for these cases, real-time computational cost increases. The latter is called the Extended Kalman Filter, and though not precisely ”optimum” has been successfully applied to many non-linear systems over the past many years [38].

The fundamental concept of the Extended Kalman Filter is assuming the true state is sufficiently close to the estimate state. With that assumption, the error dynamics can be represented by a linearized first-order Taylor series expansion [38].

The first order expansion of $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$ and $\mathbf{h}(\mathbf{x}(t), t)$, about a nominal state $\bar{\mathbf{x}}(t)$ is given by:

$$\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \cong \mathbf{f}(\bar{\mathbf{x}}(t), \mathbf{u}(t), t) + \left.\frac{\delta \mathbf{f}}{\delta \mathbf{x}}\right|_{\bar{\mathbf{x}}(t)} [\mathbf{x}(t) - \bar{\mathbf{x}}(t)] \tag{4.52a}$$

$$\mathbf{h}(\mathbf{x}(t), t) \cong \mathbf{h}(\bar{\mathbf{x}}(t), t) + \left.\frac{\delta \mathbf{h}}{\delta \mathbf{x}}\right|_{\bar{\mathbf{x}}(t)} [\mathbf{x}(t) - \bar{\mathbf{x}}(t)] \tag{4.52b}$$

where $\bar{\mathbf{x}}(t) = \widehat{\mathbf{x}}(t)$ in the case of the Extended Kalman Filter.

The linearized error dynamics can then be derived to be:

$$\mathit{\Delta}\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathit{\Delta}\mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{4.53}$$

where $\varDelta\mathbf{x}(t) = \mathbf{x}(t) - \widehat{\mathbf{x}}(t)$ and:

$$\mathbf{F}(t) \equiv \left.\frac{\delta \mathbf{f}}{\delta \mathbf{x}}\right|_{\widehat{\mathbf{x}}(t)} \tag{4.54}$$

As the error equation 4.53 is linear, the Extended Kalman Filter is formulated in terms of the error estimation.

A linear measurement model may also be defined by using the following measurement residual $\boldsymbol{\upsilon}_k$ at time t_k:

$$\boldsymbol{\upsilon}_k = \mathbf{y}_k - \widehat{\mathbf{y}}_k = \mathbf{y}_k - \mathbf{h}_k(\widehat{\mathbf{x}}_k^-) \tag{4.55}$$

where $\widehat{\mathbf{x}}_k^-$ is the *a priori* state estimate propagated from the previous step. A linear first order approximation for $\boldsymbol{\upsilon}_k$ is:

$$\boldsymbol{\upsilon}_k \cong \mathbf{H}_k \varDelta\mathbf{x}_k^- + \mathbf{v}_k, \qquad \mathbf{H}_k \equiv \left.\frac{\delta \mathbf{h}_k}{\delta \mathbf{x}}\right|_{\widehat{\mathbf{x}}_k^-} \tag{4.56}$$

where $\varDelta\mathbf{x}_k^- = \mathbf{x}_k - \widehat{\mathbf{x}}_k^-$.

Using the linearized measurement model of equation 4.56, the measurement update stage of the linear Kalman Filter may be applied to $\varDelta\mathbf{x}_k^-$, with some small changes:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k]^{-1} \tag{4.57a}$$

$$\varDelta\widehat{\mathbf{x}}_k^+ = \mathbf{K}_k [\mathbf{y}_k - \mathbf{h}_k(\widehat{\mathbf{x}}_k^-)] \tag{4.57b}$$

$$\widehat{\mathbf{x}}_k^+ = \widehat{\mathbf{x}}_k^- + \varDelta\widehat{\mathbf{x}}_k^+ \tag{4.57c}$$

$$\mathbf{P}_k^+ = [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \mathbf{P}_k^- \tag{4.57d}$$

where $\mathbf{K}_k$ and $\mathbf{P}_k^+$ come from equations 4.45 and 4.46 of the linear Kalman Filter, respectively.

The measurement update of the EKF can be summarized as follows. At time t_k, the uncertainty between the true and estimated state is $\varDelta\mathbf{x}_k^-$. With a measurement, $\varDelta\mathbf{x}_k^-$ is updated, reducing the uncertainty to $\varDelta\mathbf{x}_k^+$ using equation 4.57b. This estimate is used to yield the *a posteriori* state estimate $\widehat{\mathbf{x}}_k^+$, using equation 4.57c. The term $\varDelta\mathbf{x}_k^-$ does not appear in equation 4.57b because the estimate of uncertainty is not explicitly propagated, but propagated as a part of $\widehat{\mathbf{x}}_k^+$. Equation 4.57c may be seen as a ”full reset”.

The covariance and state propagation equations are yet to be defined. First of all, the propagation of the state estimate itself is made with resort to the nonlinear dynamics matrix. Taking the expectation of equation 4.51a:

$$\widehat{\mathbf{x}}_{k+1}^- = \mathbf{f}_k(\widehat{\mathbf{x}}_k^+) \tag{4.58}$$

In order to obtain the covariance propagation equation, the first-order Taylor series expansion of equation 4.51a is used, to get the propagation of the uncertainty from time t_k to t_{k+1}, obtaining:

$$\mathit{\Delta}\mathbf{x}^-_{k+1} = \boldsymbol{\Phi}_k \mathit{\Delta}\mathbf{x}^+_k + \boldsymbol{\Upsilon}_k \mathbf{w}_k, \qquad \boldsymbol{\Phi}_k \equiv \left.\frac{\delta \mathbf{f}_k}{\delta \mathbf{x}}\right|_{\widehat{\mathbf{x}}^+_k} \tag{4.59}$$

which results in the following expression for the covariance propagation:

$$\mathbf{P}^-_{k+1} = \boldsymbol{\Phi}_k \mathbf{P}^+_k \boldsymbol{\Phi}^T_k + \boldsymbol{\Upsilon}_k \mathbf{Q}_k \boldsymbol{\Upsilon}^T_k \tag{4.60}$$

Since the approach used in the EKF assumes that the true state is ”close” to the estimated state, this filter performance may be seriously affected for highly nonlinear applications with large initial condition errors [38].

Table 4.2 summarizes the discrete-time Extended Kalman Filter formulation.

Discrete-time Extended Kalman Filter		
Model	$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k) + \boldsymbol{\Upsilon}_k \mathbf{w}_k,$ $\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k,$	$\mathbf{w}_k \sim N(0, \mathbf{Q}_k)$ $\mathbf{v}_k \sim N(0, \mathbf{R}_k)$
Initial condition	$\widehat{\mathbf{x}}(t_0)$=$\widehat{\mathbf{x}}_0$ $\mathbf{P}(t_0)$=$\mathbf{P}_0$	
Propagation	$\widehat{\mathbf{x}}^-_{k+1} = \mathbf{f}_k(\widehat{\mathbf{x}}^+_k)$ $\mathbf{P}^-_{k+1} = \boldsymbol{\Phi}_k \mathbf{P}^+_k \boldsymbol{\Phi}^T_k + \boldsymbol{\Upsilon}_k \mathbf{Q}_k \boldsymbol{\Upsilon}^T_k,$	$\boldsymbol{\Phi}_k \equiv \left.\frac{\delta \mathbf{f}_k}{\delta \mathbf{x}}\right\|_{\widehat{\mathbf{x}}^+_k}$
Gain	$\mathbf{K}_k = \mathbf{P}^-_k \mathbf{H}^T_k [\mathbf{H}_k \mathbf{P}^-_k \mathbf{H}^T_k + \mathbf{R}_k]^{-1},$	$\mathbf{H}_k \equiv \left.\frac{\delta \mathbf{h}_k}{\delta \mathbf{x}}\right\|_{\widehat{\mathbf{x}}^-_k}$
Update	$\mathit{\Delta}\widehat{\mathbf{x}}^+_k = \mathbf{K}_k[\mathbf{y}_k - \mathbf{h}_k(\widehat{\mathbf{x}}^-_k)]$ $\widehat{\mathbf{x}}^+_k = \widehat{\mathbf{x}}^-_k + \mathit{\Delta}\widehat{\mathbf{x}}^+_k$ $\mathbf{P}^+_k = [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k]\mathbf{P}^-_k$	

Table 4.2: Discrete-time Extended Kalman Filter formulation.

4.2.2 Kalman Filtering for Spacecraft Attitude Estimation

A large class of orbit and attitude estimation problems are, in fact, nonlinear. Many times referred in literature, real-time spacecraft generally employ an EKF for attitude estimation [17]. Though the possibility of problems in filter performance due to divergence, many robust and reliable EKF have been employed.

Many of the representations presented in section 2.1 can be used in the EKF formulation. Although the rotation matrix is the fundamental representation of attitude, it’s composed by nine parameters, 6 of them redundant and orthogonality problems related with computation errors are frequent. As stated in section 2.1, special orthogonal group $SO(3)$ of rotation matrices has dimension three. Therefore, most EKFs use lower-dimensional parameterizations of SO(3) such as a minimal three-dimensional parameterization. However, while all three-parameter representations are singular or discontinuous, higher-dimensional parameterizations, such as the quaternion, can avoid these singularities or discontinuities. A dilemma is then faced, choosing between an attitude representation that is either singular or redundant [17]. The quaternion is the lowest non-singular parameterization of attitude, being easily employed in an EKF, hence being the parametrization chosen to formulate the following filters.

There are two ways to deal with the quaternion in the filter formulation. The first is to consider the four components of the quaternion as independent parameters, although it is not true, and is called Additive EKF. The second one is to consider a nonsingular representation for a reference attitude, and a three component representation for the deviations from this reference, called Multiplicative EKF.

Additive EKF

The Additive EKF is obtained by relaxing the quaternion normalization condition and treating the four components of the quaternion as independent parameters. The quaternion estimate and error are given by:

$$\widehat{\mathbf{q}} = E\{\mathbf{q}\} \tag{4.61a}$$

$$\varDelta\mathbf{q} \equiv \mathbf{q} - \widehat{\mathbf{q}} \tag{4.61b}$$

This leads to the relation:

$$E\{||\mathbf{q}||^2\} = E\{||\widehat{\mathbf{q}} + \varDelta\mathbf{q}||^2\} = ||\widehat{\mathbf{q}}||^2 + E\{||\varDelta\mathbf{q}||^2\} > ||\widehat{\mathbf{q}}||^2 \tag{4.62}$$

Equation 4.62 shows that if the random variable **q** has unit norm, the norm of its expectation must be less than unity. The unity norm constraint violation would be on the order of the variance of the attitude errors, but this problem is regarded as a conceptual problem of AEKF [17]. Some methods have been proposed to deal with the problem referred but none is considered completely satisfactory [11].

Multiplicative EKF

Evasion to the dilemma of a singular or redundant parameterization of attitude can be obtained using a nonsingular representation for a reference attitude, and a three component representation for the deviations from this reference. This method may be referred to as the Multiplicative EKF (MEKF) [17].

In MEKF, the true quaternion is written as the product of an error quaternion and the estimate:

$$\mathbf{q} = \delta\mathbf{q}(\bar{\mathbf{a}}) \otimes \widehat{\mathbf{q}} \tag{4.63}$$

It is notable that **q**, $\delta\mathbf{q}$ and $\widehat{\mathbf{q}}$ are unit quaternions and the rotation $\delta\mathbf{q}(\bar{\mathbf{a}})$ is parameterized by $\bar{\mathbf{a}}$, a three-component representation of the attitude error, defined in the body reference frame. $\boldsymbol{\delta\vartheta}$ error representation introduced in section 2.1.5 can be used as $\bar{\mathbf{a}}$, so the error quaternion can be expressed to first order by:

$$\delta\mathbf{q}(\delta\boldsymbol{\vartheta}) \approx \begin{bmatrix} \delta\boldsymbol{\vartheta}/2 \\ 1 \end{bmatrix} = \mathbf{I}_q + \frac{1}{2}\begin{bmatrix} \delta\boldsymbol{\vartheta} \\ 0 \end{bmatrix} \tag{4.64}$$

Using $\delta\boldsymbol{\vartheta}$ as the three component representation of the attitude error, the MEKF updates the error state:

$$\boldsymbol{\Delta}\mathbf{x} \equiv \begin{bmatrix} \delta\boldsymbol{\vartheta} \\ \boldsymbol{\Delta\xi} \end{bmatrix} \tag{4.65}$$

where $\boldsymbol{\xi} = \widehat{\boldsymbol{\xi}} + \boldsymbol{\Delta\xi}$ is a vector of other variables to be estimated.

Then the MEKF computes an unconstrained and unbiased estimate of $\Delta\mathbf{x}$. The estimate $\widehat{\mathbf{q}}$ is not actually part of the EKF, not being defined as an expectation, but a reset operation moves the updates into this global variable to keep the error quaternion $\boldsymbol{\delta}\mathbf{q}$ small and thus far away from any singularities.

As the MEKF requires less computational effort than the AEKF due to the lower dimensionality of its covariance matrix, and is also the most satisfying conceptually, respecting the dimensionality of the rotation group and the quaternion unit norm by definition [17], the proposed EKF for implementation in this work uses the Multiplicative formulation.

MEKF formulation

A normal MEKF implementation proceeds by iteration of three steps: the measurement update, the state vector reset and propagation to the next measurement time.

The measurement update step updates the error state vector with information obtained from measurements.

The observation model is given by equation 4.50b, specified for attitude estimation by:

$$\mathbf{y} = \mathbf{h}(\mathbf{q}, \boldsymbol{\xi}) + \mathbf{v} \tag{4.66}$$

As the MEKF updates the error state of equation 4.65, the measurement sensitivity matrix is:

$$\mathbf{H}(\mathbf{q}, \boldsymbol{\xi}) = \frac{\delta\mathbf{h}}{\delta(\boldsymbol{\Delta}\mathbf{x})} = \begin{bmatrix} \frac{\delta\mathbf{h}}{\delta(\delta\boldsymbol{\vartheta})} & \frac{\delta\mathbf{h}}{\delta(\boldsymbol{\Delta\xi})} \end{bmatrix} \tag{4.67}$$

After the reset and propagation steps, the error state vector is zero. However, it may be desired to process several measurements at one time without doing a reset between each other, leading to a error state vector with finite values. Therefore, and in order to avoid recalculating the non-linear function $\mathbf{h}(\widehat{\mathbf{q}}, \widehat{\boldsymbol{\xi}})$ several times, its first-order Taylor series is used to compute the expectation:

$$E\{\mathbf{h}(\mathbf{q}, \boldsymbol{\xi})\} \approx \mathbf{h}(\widehat{\mathbf{q}}, \widehat{\boldsymbol{\xi}}) + \mathbf{H}(\widehat{\mathbf{q}}, \widehat{\boldsymbol{\xi}}) \begin{bmatrix} \delta\widehat{\boldsymbol{\vartheta}} \\ \boldsymbol{\Delta}\widehat{\boldsymbol{\xi}} \end{bmatrix} \tag{4.68}$$

Then, the state update for the k^{th} measurement is then:

$$\begin{bmatrix} \boldsymbol{\delta}\widehat{\boldsymbol{\vartheta}}_k^+ \\ \boldsymbol{\Delta}\widehat{\boldsymbol{\xi}}_k^+ \end{bmatrix} = \begin{bmatrix} \boldsymbol{\delta}\widehat{\boldsymbol{\vartheta}}_k^- \\ \boldsymbol{\Delta}\widehat{\boldsymbol{\xi}}_k^- \end{bmatrix} + \mathbf{K}_k \left\{ \mathbf{y}_k - \mathbf{h}_k(\widehat{\mathbf{q}}_k^-, \widehat{\boldsymbol{\xi}}_k^-) - \mathbf{H}_k(\widehat{\mathbf{q}}_k^-, \widehat{\boldsymbol{\xi}}_k^-) \begin{bmatrix} \boldsymbol{\delta}\widehat{\boldsymbol{\vartheta}}_k^- \\ \boldsymbol{\Delta}\widehat{\boldsymbol{\xi}}_k^- \end{bmatrix} \right\} \tag{4.69}$$

where the Kalman gain $\mathbf{K}_k$ is computed according to equation 4.57a. The updated covariance matrix $\mathbf{P}_k^+$ may be computed according to equation 4.57d.

The reset step moves the updated information from the error state $\boldsymbol{\Delta}\mathbf{x}^+$ to the global state $\mathbf{x}^-$ and resets the components of the error state to zero. In this way, is obtained a post-updated estimate of the global state vector, i.e. $\widehat{\mathbf{q}}^+$ and $\widehat{\boldsymbol{\xi}}^+$, through:

$$\widehat{\mathbf{q}}^+ = \boldsymbol{\delta}\mathbf{q}(\boldsymbol{\delta}\widehat{\boldsymbol{\vartheta}}^+) \otimes \widehat{\mathbf{q}}^- = \frac{1}{\sqrt{1 + ||\boldsymbol{\delta}\widehat{\boldsymbol{\vartheta}}^+/2||^2}} \begin{bmatrix} \boldsymbol{\delta}\widehat{\boldsymbol{\vartheta}}^+/2 \\ 1 \end{bmatrix} \otimes \widehat{\mathbf{q}}^- \tag{4.70a}$$

$$\widehat{\boldsymbol{\xi}}^+ = \widehat{\boldsymbol{\xi}}^- + \boldsymbol{\Delta}\widehat{\boldsymbol{\xi}}^+ \tag{4.70b}$$

and after, $\boldsymbol{\delta}\widehat{\boldsymbol{\vartheta}}_k^-$ and $\boldsymbol{\Delta}\widehat{\boldsymbol{\xi}}_k^-$ are reset to zero.

The covariance is not affected by the reset because the mean rotation caused by the reset is zero.

The reset operation is fundamental before the beginning of the propagation step, to avoid the need of propagating the error state vector between measurements.

The propagation step propagates the global variables state and covariance to the time of the next measurement.

Equation 2.24 from section 2.3.2 defines the quaternion kinematics model. This equation and its expectation, repeated here, are used in the EKF for the propagation of the dynamics:

$$\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega})\mathbf{q} \tag{4.71}$$

and its expectation:

$$\dot{\widehat{\mathbf{q}}} = \frac{1}{2}\boldsymbol{\Omega}(\widehat{\boldsymbol{\omega}})\widehat{\mathbf{q}} \tag{4.72}$$

Combining these two equations, equation 4.63, and its derivative, an exact kinematic relationship may be derived [11]:

$$\boldsymbol{\delta}\dot{\mathbf{q}} = \frac{1}{2}\left(\begin{bmatrix} \widehat{\boldsymbol{\omega}} \\ 0 \end{bmatrix} \otimes \boldsymbol{\delta}\mathbf{q} - \boldsymbol{\delta}\mathbf{q} \otimes \begin{bmatrix} \widehat{\boldsymbol{\omega}} \\ 0 \end{bmatrix} \right) + \frac{1}{2} \begin{bmatrix} \boldsymbol{\delta\omega} \\ 0 \end{bmatrix} \otimes \boldsymbol{\delta}\mathbf{q} \tag{4.73}$$

that can be simplified (by ignoring some terms), to:

$$\boldsymbol{\delta}\dot{\mathbf{q}} = - \begin{bmatrix} \widehat{\boldsymbol{\omega}} \times \boldsymbol{\delta}\mathbf{q}_{1:3} \\ 0 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \boldsymbol{\delta\omega} \\ 0 \end{bmatrix} \tag{4.74}$$

The first three components of this, considering the error quaternion of equation 4.64 are:

$$\boldsymbol{\delta}\dot{\boldsymbol{\vartheta}} = -\widehat{\boldsymbol{\omega}} \times \boldsymbol{\delta\vartheta} + \boldsymbol{\delta\omega} \tag{4.75}$$

and its expectation is:

$$\boldsymbol{\delta}\dot{\widehat{\boldsymbol{\vartheta}}} = -\widehat{\boldsymbol{\omega}} \times \boldsymbol{\delta}\widehat{\boldsymbol{\vartheta}} \tag{4.76}$$

MEKF implementation

One traditional implementation of an MEKF onboard of satellites is the 6-state EKF presented in [11], that estimates the current attitude and gyroscope biases simultaneously, enabling an on-orbit calibration approach. That is the approach used in this work.

Dynamic models based on equations of section 2.3 are many times used for the formulation of these kind of filters, providing the precise angular rate information needed to filter noisy attitude sensor data and propagate the system. Another approach is many other times used, as most of the spacecraft are provided with accurate gyroscopes. These gyroscopes can be incorporated in an EKF as part of the dynamic model, which may referred as using gyros in the dynamic model replacement mode. The latter may be preferred in some situations: firstly, gyro information may be more accurate than the available models of rotational dynamics and torques, and, sometimes, developing these models is complex; secondly, it is far less computationally expensive to do it for onboard filtering. The filter formulated in this section uses this approach.

For the development of this filter, it is assumed that the sensors are perfectly aligned. It is an approximation that introduces error in the overall system because sensor misalignment is inevitable and would contribute to unreliable attitude estimates, but for the sake of simplicity, such is assumed here.

The 7-component global state vector is made up of the true quaternion **q** and the true gyro drift biases β:

$$\mathbf{x} \equiv \begin{bmatrix} \mathbf{q} \\ \beta \end{bmatrix} \tag{4.77}$$

and the 6-component state error vector is:

$$\boldsymbol{\Delta}\mathbf{x} \equiv \begin{bmatrix} \boldsymbol{\delta\vartheta} \\ \boldsymbol{\Delta\beta} \end{bmatrix} \tag{4.78}$$

where $\boldsymbol{\delta\vartheta}$ is the local vector of small attitude errors and $\boldsymbol{\Delta\beta}$ the error in the gyroscope biases defined as $\boldsymbol{\Delta\beta} \equiv \beta - \widehat{\boldsymbol{\beta}}$.

The gyroscope model is given by equations 3.9, where the spectral densities of $\boldsymbol{\eta}_v(t)$ and $\boldsymbol{\eta}_u(t)$ are respectively $\boldsymbol{\sigma}_v\mathbf{I}_3$ and $\boldsymbol{\sigma}_u\mathbf{I}_3$. From the gyro model, it can be deduced:

$$\dot{\widehat{\beta}} = \mathbf{0} \tag{4.79}$$

and:

$$\boldsymbol{\delta\omega} = -(\boldsymbol{\Delta\beta} + \boldsymbol{\eta_v}) \tag{4.80}$$

The error state vector obeys the linearized dynamic equation 4.53:

$$\Delta\dot{\mathbf{x}}(t) = \mathbf{F}(t)\Delta\mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{4.81}$$

where $\mathbf{F}(t)$ is the Jacobian of $\mathbf{f}(\mathbf{x}, t)$ and:

$$\mathbf{w}(t) \equiv \begin{bmatrix} \boldsymbol{\eta_v}(t) \\ \boldsymbol{\eta_u}(t) \end{bmatrix} \tag{4.82}$$

Resorting to equations 4.75 and 4.80, the matrices $\mathbf{F}(t)$, $\mathbf{G}(t)$ and $\mathbf{Q}(t)$ can be formed, according to [11], as:

$$\mathbf{F}(t) = \begin{bmatrix} -\left[\widehat{\boldsymbol{\omega}}(t)\times\right] & -\mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \tag{4.83}$$

$$\mathbf{G}(t) = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 \end{bmatrix} \tag{4.84}$$

$$\mathbf{Q}(t) = \begin{bmatrix} \boldsymbol{\sigma_v}^2\mathbf{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \boldsymbol{\sigma_u}^2\mathbf{I}_3 \end{bmatrix} \tag{4.85}$$

According to [11] and [38], closed-form expressions for the discrete-time matrices are possible, using a power series approach. The discrete error-state transition matrix is given by:

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\Phi}_{11} & \boldsymbol{\Phi}_{12} \\ \boldsymbol{\Phi}_{21} & \boldsymbol{\Phi}_{22} \end{bmatrix} \tag{4.86}$$

with:

$$\boldsymbol{\Phi}_{11} = \mathbf{I}_3 - \left[\widehat{\boldsymbol{\omega}}\times\right] \frac{\sin(||\widehat{\boldsymbol{\omega}}||\Delta t)}{||\widehat{\boldsymbol{\omega}}||} + \left[\widehat{\boldsymbol{\omega}}\times\right]^2 \frac{1-\cos(||\widehat{\boldsymbol{\omega}}||\Delta t)}{||\widehat{\boldsymbol{\omega}}||^2} \tag{4.87a}$$

$$\boldsymbol{\Phi}_{12} = \left[\widehat{\boldsymbol{\omega}}\times\right] \frac{1-\cos(||\widehat{\boldsymbol{\omega}}||\Delta t)}{||\widehat{\boldsymbol{\omega}}||^2} - \mathbf{I}_3\Delta t - \left[\widehat{\boldsymbol{\omega}}\times\right]^2 \frac{||\widehat{\boldsymbol{\omega}}||\Delta t - \sin(||\widehat{\boldsymbol{\omega}}||\Delta t)}{||\widehat{\boldsymbol{\omega}}||^3} \tag{4.87b}$$

$$\boldsymbol{\Phi}_{21} = \mathbf{0}_{3\times3} \tag{4.87c}$$

$$\boldsymbol{\Phi}_{22} = \mathbf{I}_3 \tag{4.87d}$$

The conversion from the spectral density $\mathbf{Q}(t)$ to discrete-time covariance $\mathbf{Q}_k$ is done resorting to some assumptions, namely that $\widehat{\omega}$ is constant throughout the sampling interval and that the sampling rate is below Nyquist's limit, which is often true, approximating the discrete process noise covariance by:

$$\mathbf{Q}_k \approx \begin{bmatrix} (\sigma_v^2\Delta t + \frac{1}{3}\sigma_u^2\Delta t^3)\mathbf{I}_3 & -(\frac{1}{2}\sigma_u^2\Delta t^2)\mathbf{I}_3 \\ -(\frac{1}{2}\sigma_u^2\Delta t^2)\mathbf{I}_3 & (\sigma_u^2\Delta t)\mathbf{I}_3 \end{bmatrix} \tag{4.88}$$

Assuming that discrete observations such as in equation 4.51b are available, the following $3N$ measurement vector can be formed:

$$\mathbf{y}_k = \begin{bmatrix} A(\mathbf{q})\mathbf{r}_1 \\ A(\mathbf{q})\mathbf{r}_2 \\ \vdots \\ A(\mathbf{q})\mathbf{r}_N \end{bmatrix} + \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_N \end{bmatrix} \equiv \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k \tag{4.89a}$$

$$\mathbf{R}_k = blkdiag\,(\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_N) \tag{4.89b}$$

where $\mathbf{R}_i$ is the covariance of $\mathbf{v}_i$. The Quest Measurement Model of equation 4.21 produces a singular matrix to be inverted in the EKF gain equation. Malcolm Shuster analyzed this singularity issue concluding that the QMM covariance matrix could be effectively replaced by $\boldsymbol{\sigma_i^2}\mathbf{I}_3$, leading to the same filter [34].

Using the relation between the true attitude matrix and the *a priori* attitude and the first order approximation of the error-attitude matrix $\mathbf{A}(\boldsymbol{\delta}\mathbf{q})$:

$$\mathbf{A}(\mathbf{q}) = \mathbf{A}(\boldsymbol{\delta}\mathbf{q})A(\widehat{\mathbf{q}}^-) \tag{4.90a}$$

$$\mathbf{A}(\boldsymbol{\delta}\mathbf{q}) \approx \mathbf{I}_3 - \left[\boldsymbol{\delta\vartheta}\times\right] \tag{4.90b}$$

it can be deduced that:

$$\mathit{\Delta}\mathbf{b} \equiv \mathbf{b} - \widehat{\mathbf{b}}^- \equiv \mathbf{A}(\mathbf{q})\mathbf{r} - \mathbf{A}(\widehat{\mathbf{q}}^-)\mathbf{r} = -\left[\boldsymbol{\delta\vartheta}\times\right]\mathbf{A}(\widehat{\mathbf{q}}^-)\mathbf{r} = \left[\widehat{\mathbf{b}}^-\times\right]\boldsymbol{\delta\vartheta} \tag{4.91}$$

and the sensitivity matrix for all measurement sets is therefore given by:

$$\mathbf{H}_k(\widehat{\mathbf{x}}_k^-) = \begin{bmatrix} \left[\widehat{\mathbf{b}}_1^-\times\right] & \mathbf{0}_{3\times 3} \\ \left[\widehat{\mathbf{b}}_2^-\times\right] & \mathbf{0}_{3\times 3} \\ \vdots & \vdots \\ \left[\widehat{\mathbf{b}}_N^-\times\right] & \mathbf{0}_{3\times 3} \end{bmatrix}_{t_k} \tag{4.92}$$

It is important to note that the measurements depend explicitly on the attitude but not on the gyro biases.

This gyro calibration MEKF implementation is summarized in Table 4.3. The filter has to be initialized with a known initial global state and error covariance matrix. Then the Kalman gain is computed using the measurement covariance matrix $\mathbf{R}_k$ and sensitivity matrix $\mathbf{H}_k$. Then, the state and error covariance updates follow the standard EKF update equations 4.57b and 4.57d, respectively. The global state is updated according to equations 4.70, followed by reset. The estimated angular velocity provided by the gyroscope is used to propagate the quaternion kinematic model and error covariance.

A discrete propagation of the quaternion model of equation 4.71 can be derived using a power series

Gyro Calibration MEKF formulation	
Initial conditions	$\widehat{\mathbf{q}}(t_0) = \widehat{\mathbf{q}}_0$ $\widehat{\boldsymbol{\beta}}(t_0) = \widehat{\boldsymbol{\beta}}_0$ $\mathbf{P}(t_0) = \mathbf{P}_0$
Gain	$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T(\widehat{\mathbf{x}}_k^-)[\mathbf{H}_k(\widehat{\mathbf{x}}_k^-)\mathbf{P}_k^- \mathbf{H}_k^T(\widehat{\mathbf{x}}_k^-) + \mathbf{R}_k]^{-1}$ $\mathbf{H}_k(\widehat{\mathbf{x}}_k^-) = \begin{bmatrix} \left[\widehat{\mathbf{b}}_1^- \times\right] & \mathbf{0}_{3\times 3} \\ \left[\widehat{\mathbf{b}}_2^- \times\right] & \mathbf{0}_{3\times 3} \\ \vdots & \vdots \\ \left[\widehat{\mathbf{b}}_N^- \times\right] & \mathbf{0}_{3\times 3} \end{bmatrix}_{t_k}$
Update	$\Delta\widehat{\mathbf{x}}_k^+ = \mathbf{K}_k[\mathbf{y}_k - \mathbf{h}_k(\widehat{\mathbf{x}}_k^-)]$ $\mathbf{P}_k^+ = [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k(\widehat{\mathbf{x}}_k^-)]\mathbf{P}_k^-$ $\mathbf{h}_k(\widehat{\mathbf{x}}_k^-) = \begin{bmatrix} A(\widehat{\mathbf{q}}^-)\mathbf{r}_1 \\ A(\widehat{\mathbf{q}}^-)\mathbf{r}_2 \\ \vdots \\ A(\widehat{\mathbf{q}}^-)\mathbf{r}_N \end{bmatrix}_{t_k}$ $\widehat{\mathbf{q}}_k^+ = \frac{1}{\sqrt{1+\Vert\boldsymbol{\delta}\widehat{\boldsymbol{\vartheta}}_k^+/2\Vert^2}} \begin{bmatrix} \boldsymbol{\delta}\widehat{\boldsymbol{\vartheta}}_k^+/2 \\ 1 \end{bmatrix} \otimes \widehat{\mathbf{q}}_k^-$ $\widehat{\boldsymbol{\beta}}_k^+ = \widehat{\boldsymbol{\beta}}_k^- + \boldsymbol{\Delta}\widehat{\boldsymbol{\beta}}_k^+$
Propagation	$\widehat{\boldsymbol{\omega}}(t) = \boldsymbol{\omega}_m(t) - \widehat{\boldsymbol{\beta}}(t)$ $\dot{\widehat{\mathbf{q}}} = \frac{1}{2}\boldsymbol{\Omega}(\widehat{\boldsymbol{\omega}})\widehat{\mathbf{q}}$ $\dot{\mathbf{P}}(t) = \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^T(t) + \mathbf{G}(t)\mathbf{Q}(t)\mathbf{G}^T(t)$

Table 4.3: Gyro Calibration MEKF formulation.

approach [38]. The propagated quaternion can be found using:

$$\widehat{\mathbf{q}}_{k+1}^- = \bar{\boldsymbol{\Omega}}(\widehat{\boldsymbol{\omega}}_k^+)\widehat{\mathbf{q}}_k^+ \tag{4.93}$$

with:

$$\bar{\boldsymbol{\Omega}}(\widehat{\boldsymbol{\omega}}_k^+) \equiv \begin{bmatrix} \cos(\frac{1}{2}||\widehat{\boldsymbol{\omega}}_k^+||\Delta t)\mathbf{I}_3 - \left[\widehat{\boldsymbol{\psi}}_k^+ \times\right] & \widehat{\boldsymbol{\psi}}_k^+ \\ -\widehat{\boldsymbol{\psi}}_k^{+T} & \cos(\frac{1}{2}||\widehat{\boldsymbol{\omega}}_k^+||\Delta t) \end{bmatrix} \tag{4.94}$$

and:

$$\widehat{\boldsymbol{\psi}}_k^+ \equiv \frac{\sin(\frac{1}{2}||\widehat{\boldsymbol{\omega}}_k^+||\Delta t)\widehat{\boldsymbol{\omega}}_k^+}{||\widehat{\boldsymbol{\omega}}_k^+||} \tag{4.95}$$

and Δt is the sampling interval in the gyroscope.

The discrete propagation of the covariance equation follows equation 4.60 with:

$$\boldsymbol{\Upsilon}_k = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{I}_3 \end{bmatrix} \tag{4.96}$$

So, the continuous propagation of equations 4.72 and 4.79 and covariance equation can be replaced by discrete equivalents 4.93, 4.60 and:

$$\widehat{\boldsymbol{\beta}}_{k+1}^{-} = \widehat{\boldsymbol{\beta}}_{k}^{+} \tag{4.97}$$

making the EKF suitable for on-board implementation.

Murrell's Version

The gain calculation of the filter shown before in Table 4.3 requires an inverse of a $3N \times 3N$ matrix in the gain calculation. A variation of this filter can be used based on an algorithm developed by Murrell [39]. As a linear update is performed, linear tools such as the principle of superposition can be used, and Murrell's filter uses this principle to process each 3×1 vector observation at a time. The schematic illustrated in figure 4.1 illustrates the steps of the algorithm. For each time t_k, the error covariance and state quantities are updated using a single vector observation. The update is continued and repeated until all vector observations are processed. Then, a reset moves the updated values into the global state representation, and then then the global state and error covariance are propagated to the next observation time.

4.3 GES Cascade Observers for Attitude Estimation

Section 4.1 focused on attitude estimation from vector measurements by means of different deterministic methods. Section 4.2 introduced the concept of state attitude estimation, describing the theory in Kalman Filtering and presenting a specific MEKF implementation. The EKF, although being a relatively simple and flexible tool with extensive heritage that can incorporate a great variety of measurements, having performed admirably in a vast number of applications, can show bad performance and divergence in the case of highly nonlinear dynamics or measurement models and bad initial estimates.

Common drawbacks [40] of attitude estimation solutions such as lack of convergence guarantees, singularities, unwinding phenomena, topological limitations for achieving global asymptotic stability, and slow convergence near unstable equilibrium points have been trying to be solved, and many nonlinear observers aiming for stability and convergence properties have emerged in that trend.

On the work of Batista, Silvestre, and Oliveira [41], the design, analysis, and performance evaluation of a sensor-based new class of globally asymptotically stable (GAS) filters for attitude estimation was presented. That solution did not have any of the previous stated limitations but it was computationally expensive. Following that work, the same authors created and presented in [40], the design, analysis, and performance evaluation of a novel cascade observer for attitude and bias estimation. The observer presented in that reference has globally exponentially stable (GES) error dynamics, is computationally efficient, is based on the angular motion kinematics, which are exact, builds on well-established Lyapunov results, explicitly estimates rate gyro bias and copes well with slowly time-varying bias, and has a complementary structure, fusing low bandwidth vector observations with high bandwidth rate gyro measurements.

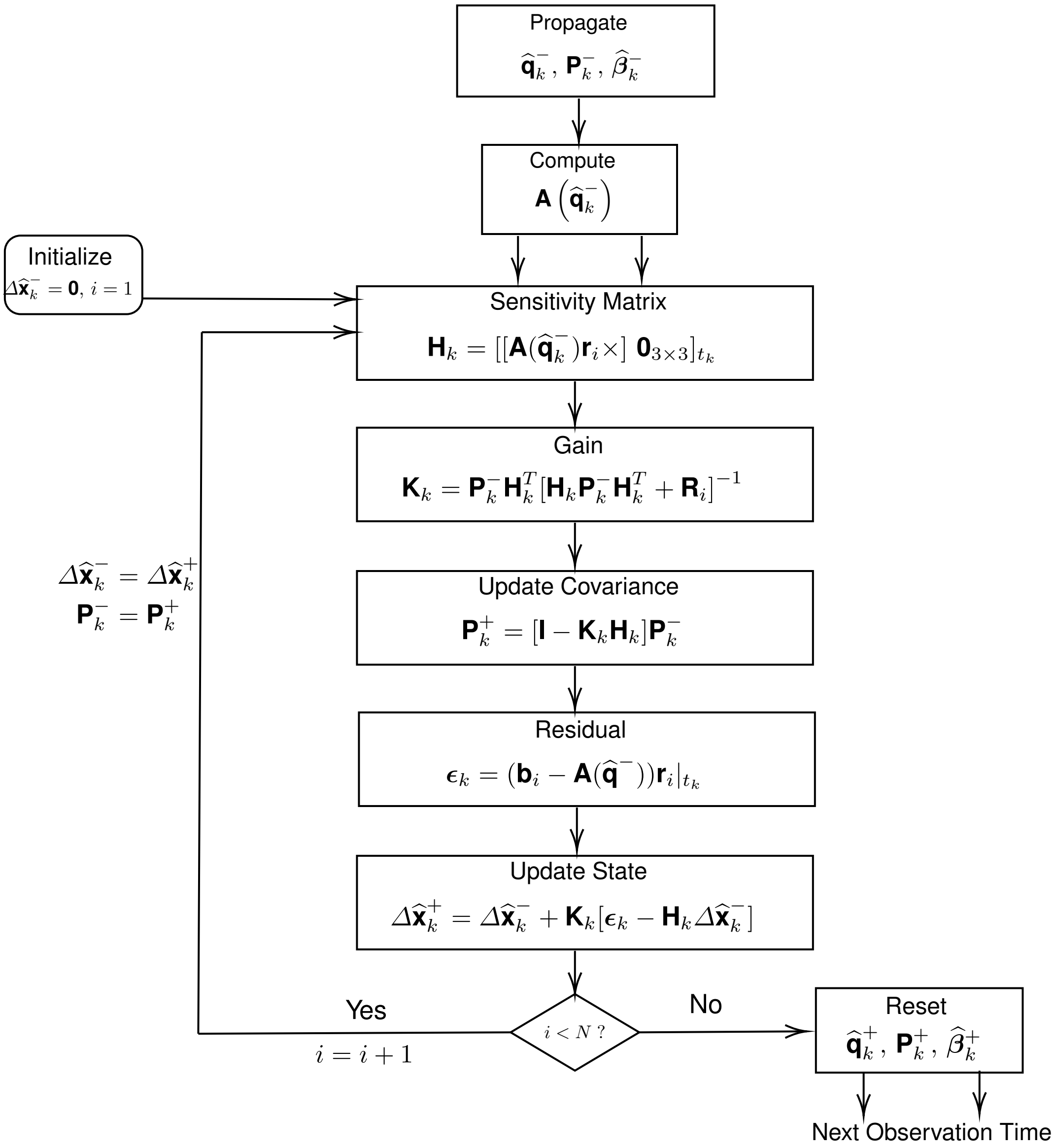

Propagate
$\widehat{\mathbf{q}}_k^-, \mathbf{P}_k^-, \widehat{\beta}_k^-$
Compute
$\mathbf{A}\left(\widehat{\mathbf{q}}_k^-\right)$
Initialize
$\Delta\widehat{\mathbf{x}}_k^- = \mathbf{0}, i = 1$
Sensitivity Matrix
$\mathbf{H}_k = [[\mathbf{A}(\widehat{\mathbf{q}}_k^-)\mathbf{r}_i\times]\ \mathbf{0}_{3\times3}]_{t_k}$
Gain
$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T[\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_i]^{-1}$
$\Delta\widehat{\mathbf{x}}_k^- = \Delta\widehat{\mathbf{x}}_k^+$
$\mathbf{P}_k^- = \mathbf{P}_k^+$
Update Covariance
$\mathbf{P}_k^+ = [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\mathbf{P}_k^-$
Residual
$\epsilon_k = (\mathbf{b}_i - \mathbf{A}(\widehat{\mathbf{q}}^-))\mathbf{r}_i|_{t_k}$
Update State
$\Delta\widehat{\mathbf{x}}_k^+ = \Delta\widehat{\mathbf{x}}_k^- + \mathbf{K}_k[\epsilon_k - \mathbf{H}_k\Delta\widehat{\mathbf{x}}_k^-]$
Yes
$i = i + 1$
$i < N\,?$
No
Reset
$\widehat{\mathbf{q}}_k^+, \mathbf{P}_k^+, \widehat{\beta}_k^+$
Next Observation Time

Figure 4.1: Flow diagram of Murrell's approach, adapted from [11].

The solution described in [40] is studied and its design is presented in this section. The analysis starts by reviewing nonlinear observer theory in section 4.3.1, then introducing the sensor based framework that is at the core of the proposed solution in section 4.3.2 and finally describing the observer design in section 4.3.3.

4.3.1 Nonlinear Observer Theory

Nonlinear observers design generally follows proposing an arbitrary filter, with parameters that can be tuned according to some criteria. The most important criteria is convergence, that can be analyzed and proved through Lyapunov theory, not discussed in this work. The motivation for the use of these nonlinear observers for attitude estimation is the usual proof of convergence and stability of the solutions.

Considering the nonlinear time varying system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \tag{4.98a}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) \tag{4.98b}$$

where **x** are the states, **u** are the inputs, and **y** are the measurements. A full state observer, under the assumption that **u** and **y** are available, is defined as:

$$\dot{\widehat{\mathbf{x}}} = \mathbf{f}(\widehat{\mathbf{x}}, \mathbf{u}, t) + \mathbf{L}(\mathbf{y} - \mathbf{h}(\widehat{\mathbf{x}})) \tag{4.99}$$

where $\widehat{\mathbf{x}}$ are the estimated states and **L** are the observer gains. Stability of the observer is done by defining the estimation error dynamics as:

$$\dot{\epsilon}(t) = \dot{\mathbf{x}}(t) - \dot{\widehat{\mathbf{x}}}(t) \tag{4.100}$$

and analyzing it for stability.

4.3.2 Sensor-based Framework

Traditional attitude estimation methods consist, generally, in using sensor data to obtain instantaneous measurements of the attitude, that are then fed to an observer or filter. With that approach, the transformations necessary to obtain an attitude representation distort the noise characteristics of the sensors. Additionally, many of these representations also exhibit problems such as singularities or unwinding phenomena, as referred in section 2.1 and [25]. In the design of the observer presented in this chapter, there's a different approach, using a sensor-based framework [40–42].

The core concept of the sensor-based framework is to design the filter directly in the space of the sensors. Then, using filtered estimates of the observations, the attitude can be determined. As the filtering occurs prior to the determination of attitude, the rotation matrix may be used as the attitude parameterization, which is unique, without singularities, and where topological restrictions on $SO(3)$ for achieving global asymptotic stability no longer apply and unwinding phenomena does not occur [42].

In this framework, the sensor measurements are included directly in the system dynamics, and the kinematics are propagated using the angular velocity provided by a three-axis rate gyro, whose biases are also considered.

Considering an inertial reference frame $\{\mathcal{I}\}$, a body-fixed reference frame $\{\mathcal{B}\}$ and the rotation matrix $\mathbf{A}(t) \in SO(3)$, from $\{\mathcal{B}\}$ to $\{\mathcal{I}\}$, the attitude kinematics are given by:

$$\dot{\mathbf{A}}(t) = \mathbf{A}(t)\mathbf{S}(\boldsymbol{\omega}(t)) \tag{4.101}$$

where $\boldsymbol{\omega}(t) \in \mathbb{R}^3$ is the angular velocity of $\{\mathcal{B}\}$, expressed in $\{\mathcal{B}\}$, and $\mathbf{S}(\mathbf{x})$ is the skew-symmetric matrix such that $\mathbf{S}(\mathbf{x})\mathbf{y} = \mathbf{x} \times \mathbf{y}$, formulated as explained in appendix A.2.

Assuming that a vector observation measurement $\mathbf{b}_1 \in \mathbb{R}^3$ is available, in body frame coordinates, of a known constant vector quantity $\mathbf{r}_1 \in \mathbb{R}^3$ in inertial coordinates:

$$\mathbf{r}_1 = \mathbf{A}(t)\mathbf{b}_1(t) \tag{4.102}$$

then, its dynamics are:

$$\dot{\mathbf{b}}_1(t) = -\mathbf{S}(\boldsymbol{\omega}(t))\mathbf{b}_1(t) \tag{4.103}$$

Many vector measurements can be considered as long as the quantities that originate them can be considered locally constant in inertial coordinates. As examples, the Sun direction and other star directions are possibilities for these observations.

Considering rate gyro measurements $\boldsymbol{\omega}_m(t) \in \mathbb{R}^3$ corrupted with a constant bias $\boldsymbol{\beta} \in \mathbb{R}^3$:

$$\boldsymbol{\omega}_m(t) = \boldsymbol{\omega}(t) + \boldsymbol{\beta}(t) \tag{4.104}$$

then, the system dynamics, extended for N vector observation measurements, may be written as:

$$\begin{cases} \dot{\mathbf{b}}_1(t) = -\mathbf{S}(\boldsymbol{\omega}_m(t))\mathbf{b}_1(t) + \mathbf{S}(\boldsymbol{\beta}(t))\mathbf{b}_1(t) = -\mathbf{S}(\boldsymbol{\omega}_m(t))\mathbf{b}_1(t) - \mathbf{S}(\mathbf{b}_1(t))\boldsymbol{\beta}(t) \\ \vdots \\ \dot{\mathbf{b}}_N(t) = -\mathbf{S}(\boldsymbol{\omega}_m(t))\mathbf{b}_N(t) + \mathbf{S}(\boldsymbol{\beta}(t))\mathbf{b}_N(t) = -\mathbf{S}(\boldsymbol{\omega}_m(t))\mathbf{b}_N(t) - \mathbf{S}(\mathbf{b}_N(t))\boldsymbol{\beta}(t) \\ \dot{\boldsymbol{\beta}}(t) = \mathbf{0} \end{cases} \tag{4.105}$$

Based on the nominal nonlinear system 4.105, filter solutions may be designed. Once filtered estimates of $\mathbf{b} = \begin{bmatrix}\mathbf{b}_1 \dots \mathbf{b}_N\end{bmatrix}^T \in \mathbb{R}^{3N}$ are obtained, the attitude can be computed using various methods.

4.3.3 Observers Design

In this section, the design of a cascade attitude observer for the rotation matrix $\mathbf{A}(t)$ and the rate gyro bias $\boldsymbol{\beta}(t)$, with GES error dynamics is presented. First a bias observer with GES error dynamics is presented, resorting directly to the vector measurements. Afterwards, an attitude observer with GES error dynamics is presented, assuming that the rate gyro bias is known. Lastly, the overall cascade attitude observer with resulting GES error dynamics is presented.

Bias Observer

The following design is based on the assumption that there exist at least two non-collinear reference vectors, i.e., there exist i and j such that $\mathbf{r}_i \times \mathbf{r}_j \neq 0$ (from now on referred as Assumption 1). Assumption 1 is necessary for attitude estimation, as stated in previous chapters.

The bias observer is composed by a set of states that correspond to the set of vector observations, with the addition of the rate gyro bias. Using the system dynamics 4.105 the bias observer is given by:

$$\begin{cases} \dot{\widehat{\mathbf{b}}}_1(t) = -\mathbf{S}(\boldsymbol{\omega}_m(t))\widehat{\mathbf{b}}_1(t) - \mathbf{S}(\mathbf{b}_1(t))\widehat{\boldsymbol{\beta}}(t) + \alpha_1\tilde{\mathbf{b}}_1(t) \\ \vdots \\ \dot{\widehat{\mathbf{b}}}_N(t) = -\mathbf{S}(\boldsymbol{\omega}_m(t))\widehat{\mathbf{b}}_N(t) - \mathbf{S}(\mathbf{b}_N(t))\widehat{\boldsymbol{\beta}}(t) + \alpha_N\tilde{\mathbf{b}}_N(t) \\ \dot{\widehat{\boldsymbol{\beta}}}(t) = \sum_{i=1}^{N} \gamma_i \mathbf{S}(\mathbf{b}_i(t))\tilde{\mathbf{b}}_i(t) \end{cases} \tag{4.106}$$

where $i = 1, ..., N$, $\tilde{\mathbf{b}}_i(t) \equiv \mathbf{b}_i(t) - \widehat{\mathbf{b}}_i(t)$, are the errors of the vector observation estimates, and α_i, γ_i, are positive scalar constants. Defining the bias estimation error as $\tilde{\boldsymbol{\beta}}(t) \equiv \boldsymbol{\beta}(t) - \widehat{\boldsymbol{\beta}}(t)$, the bias observer error dynamics are given by:

$$\begin{cases} \dot{\tilde{\mathbf{b}}}_1(t) = -\mathbf{S}(\boldsymbol{\omega}_m(t))\tilde{\mathbf{b}}_1(t) - \mathbf{S}(\mathbf{b}_1(t))\tilde{\boldsymbol{\beta}}(t) + \alpha_1\tilde{\mathbf{b}}_1(t) \\ \vdots \\ \dot{\tilde{\mathbf{b}}}_N(t) = -\mathbf{S}(\boldsymbol{\omega}_m(t))\tilde{\mathbf{b}}_N(t) - \mathbf{S}(\mathbf{b}_N(t))\tilde{\boldsymbol{\beta}}(t) + \alpha_N\tilde{\mathbf{b}}_N(t) \\ \dot{\tilde{\boldsymbol{\beta}}}(t) = -\sum_{i=1}^{N} \gamma_i \mathbf{S}(\mathbf{b}_i(t))\tilde{\mathbf{b}}_i(t) \end{cases} \tag{4.107}$$

or, in compact form:

$$\dot{\tilde{\boldsymbol{\chi}}}_1(t) = A_1(t)\tilde{\boldsymbol{\chi}}_1(t) \tag{4.108}$$

with:

$$\tilde{\boldsymbol{\chi}}_1(t) = \begin{bmatrix} \tilde{\mathbf{b}}_1^T(t) & \dots & \tilde{\mathbf{b}}_N^T(t) & \tilde{\boldsymbol{\beta}}^T(t) \end{bmatrix}^T \in \mathbb{R}^{3(N+1)} \tag{4.109a}$$

$$A_1(t) = \quad -diag(\alpha_1\mathbf{I} + \mathbf{S}(\boldsymbol{\omega}_m(t)), \dots, \alpha_N\mathbf{I} + \mathbf{S}(\boldsymbol{\omega}_m(t)), \mathbf{0}) \tag{4.109b}$$

$$+ \begin{bmatrix} \mathbf{0} & \dots & \mathbf{0} & -\mathbf{S}(\mathbf{b}_1(t)) \\ \vdots & & \vdots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & -\mathbf{S}(\mathbf{b}_N(t)) \\ -\gamma_1\mathbf{S}(\mathbf{b}_1(t)) & \dots & -\gamma_N\mathbf{S}(\mathbf{b}_N(t)) & \mathbf{0} \end{bmatrix} \tag{4.109c}$$

Under Assumption 1, the origin of the observer error dynamics 4.107 is a GES equilibrium point. Proof for this statement may be found in Section 3.1 of [40].

Attitude Observer

The following design is made on the assumption that the matrix $\begin{bmatrix}\mathbf{r}_1 \dots \mathbf{r}_N\end{bmatrix} \in \mathbb{R}^{3\times3N}$ has full rank (from now on referred as Assumption 2), which, given a set of reference vectors that satisfy Assumption 1, it is always possible: If $\mathbf{r}_i \in \mathbb{R}^3$ and $\mathbf{r}_j \in \mathbb{R}^3$ denote two non-collinear reference vectors, then, the set $\{\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_i \times \mathbf{r}_j\}$ satisfies Assumption 2.

Let's consider an alternative column representation of the rotation matrix $\mathbf{A}(t)$ given by:

$$\boldsymbol{\chi}_2(t) = \begin{bmatrix} \mathbf{z}_1(t) \\ \mathbf{z}_2(t) \\ \mathbf{z}_3(t) \end{bmatrix} \in \mathbb{R}^9 \tag{4.110}$$

where:

$$\mathbf{A}(t) = \begin{bmatrix} \mathbf{z}_1^T(t) \\ \mathbf{z}_2^T(t) \\ \mathbf{z}_3^T(t) \end{bmatrix}, \mathbf{z}_i(t) \in \mathbb{R}^3, i = 1, .., 3. \tag{4.111}$$

Under Assumption 2 and knowing the rate gyro bias, the following dynamics take place:

$$\dot{\boldsymbol{\chi}}_2(t) = -\mathbf{S}_3(\boldsymbol{\omega}_m(t) - \boldsymbol{\beta}(t))\boldsymbol{\chi}_2(t) \tag{4.112}$$

where:

$$\mathbf{S}_3(\mathbf{x}) \equiv diag(\mathbf{S}(\mathbf{x}), \mathbf{S}(\mathbf{x}), \mathbf{S}(\mathbf{x})) \in \mathbb{R}^{9\times 9} \tag{4.113}$$

Using 4.102, the vector observations can be written as a function of $\boldsymbol{\chi}_2(t)$ by:

$$\mathbf{b}(t) = \mathbf{C}_2\boldsymbol{\chi}_2(t) \tag{4.114}$$

with:

$$\mathbf{C}_2 = \begin{bmatrix} r_{11} & 0 & 0 & r_{12} & 0 & 0 & r_{13} & 0 & 0 \\ 0 & r_{11} & 0 & 0 & r_{12} & 0 & 0 & r_{13} & 0 \\ 0 & 0 & r_{11} & 0 & 0 & r_{12} & 0 & 0 & r_{13} \\ & & & & \vdots & & & & \\ r_{N1} & 0 & 0 & r_{N2} & 0 & 0 & r_{N3} & 0 & 0 \\ 0 & r_{N1} & 0 & 0 & r_{N2} & 0 & 0 & r_{N3} & 0 \\ 0 & 0 & r_{N1} & 0 & 0 & r_{N2} & 0 & 0 & r_{N3} \end{bmatrix} \in \mathbb{R}^{3N\times 9} \qquad \mathbf{r}_i = \begin{bmatrix} r_{i1} \\ r_{i2} \\ r_{i3} \end{bmatrix} \in \mathbb{R}^3 \tag{4.115}$$

The attitude observer can then be given by:

$$\dot{\widehat{\boldsymbol{\chi}}}_2(t) = -\mathbf{S}_3(\boldsymbol{\omega}_m(t) - \boldsymbol{\beta}(t))\widehat{\boldsymbol{\chi}}_2(t) + \mathbf{C}_2^T \mathtt{Q}^{-1}[\mathbf{b}(t) - \mathbf{C}_2\widehat{\boldsymbol{\chi}}_2(t)] \tag{4.116}$$

where $\mathtt{Q} = \mathtt{Q}^T \in \mathbb{R}^{3N\times 3N}$ is a positive definite matrix.

Defining the error variable $\tilde{\boldsymbol{\chi}}_2(t) = \boldsymbol{\chi}_2(t) - \widehat{\boldsymbol{\chi}}_2(t)$, the observer error dynamics are given by:

$$\dot{\tilde{\boldsymbol{\chi}}}_2(t) = \mathcal{A}_2(t)\tilde{\boldsymbol{\chi}}_2(t) \tag{4.117}$$

where:

$$\mathcal{A}_2(t) \equiv -[\mathbf{S}_3(\boldsymbol{\omega}_m(t) - \boldsymbol{\beta}(t)) + \mathbf{C}_2^T \mathtt{Q}^{-1}\mathbf{C}_2] \tag{4.118}$$

With knowledge of the rate gyro bias and under Assumption 2, the origin of the observer error dynamics 4.117 is a globally exponentially stable equilibrium point. Proof for this statement may be found in section 3.2 of [40].

Cascade Observer

Two observers were presented in the last sections. Firstly, an observer that provides an estimate of the bias, based directly on the vector observations and with GES error dynamics. Then, an attitude observer, also with GES error dynamics, but assuming the rate gyro bias is known. The idea of the cascade observer presented in this section is to feed the attitude observer with the bias estimate provided by the bias observer. The final nonlinear cascade observer becomes:

$$\begin{cases} \dot{\widehat{\mathbf{b}}}_1(t) = -\mathbf{S}(\boldsymbol{\omega}_m(t))\widehat{\mathbf{b}}_1(t) - \mathbf{S}(\mathbf{b}_1(t))\widehat{\boldsymbol{\beta}}(t) + \alpha_1\tilde{\mathbf{b}}_1(t) \\ \vdots \\ \dot{\widehat{\mathbf{b}}}_N(t) = -\mathbf{S}(\boldsymbol{\omega}_m(t))\widehat{\mathbf{b}}_N(t) - \mathbf{S}(\mathbf{b}_N(t))\widehat{\boldsymbol{\beta}}(t) + \alpha_N\tilde{\mathbf{b}}_N(t) \\ \dot{\widehat{\boldsymbol{\beta}}}(t) = \sum_{i=1}^{N}\gamma_i\mathbf{S}(\mathbf{b}_i(t))\tilde{\mathbf{b}}_i(t) \\ \dot{\widehat{\boldsymbol{\chi}}}_2(t) = -\mathbf{S}_3(\boldsymbol{\omega}_m(t) - \widehat{\boldsymbol{\beta}}(t))\widehat{\boldsymbol{\chi}}_2(t) + \mathbf{C}_2^T\mathtt{Q}^{-1}[\mathbf{b}(t) - \mathbf{C}_2\widehat{\boldsymbol{\chi}}_2(t)] \end{cases} \tag{4.119}$$

The error dynamics of the cascade observer can be written as:

$$\begin{cases} \dot{\tilde{\boldsymbol{\chi}}}_1(t) = \mathsf{A}_1(t)\tilde{\boldsymbol{\chi}}_1(t) \\ \dot{\tilde{\boldsymbol{\chi}}}_2(t) = [\mathsf{A}_2(t) - \mathbf{S}_3(\tilde{\boldsymbol{\beta}}(t))]\tilde{\boldsymbol{\chi}}_2(t) + \mathbf{u}_2(t) \end{cases} \tag{4.120}$$

where:

$$\mathbf{u}_2(t) \equiv \mathbf{S}_3(\tilde{\boldsymbol{\beta}}(t))\boldsymbol{\chi}_2(t) \tag{4.121}$$

The origin of the observer error dynamics 4.120 is a GES equilibrium point. Proof for this statement may be found in [40].

The attitude observer considers only the bias estimate, disregarding the vector estimates. For performance purposes, particularly in the presence of sensor noise, it may be better to employ the vector estimate $\widehat{\mathbf{b}}(t)$ provided by the bias observer for the feedback. The full cascade observer becomes:

$$\begin{cases} \dot{\widehat{\mathbf{b}}}_1(t) = -\mathbf{S}(\boldsymbol{\omega}_m(t))\widehat{\mathbf{b}}_1(t) - \mathbf{S}(\mathbf{b}_1(t))\widehat{\boldsymbol{\beta}}(t) + \alpha_1\tilde{\mathbf{b}}_1(t) \\ \vdots \\ \dot{\widehat{\mathbf{b}}}_N(t) = -\mathbf{S}(\boldsymbol{\omega}_m(t))\widehat{\mathbf{b}}_N(t) - \mathbf{S}(\mathbf{b}_N(t))\widehat{\boldsymbol{\beta}}(t) + \alpha_N\tilde{\mathbf{b}}_N(t) \\ \dot{\widehat{\boldsymbol{\beta}}}(t) = \sum_{i=1}^{N}\gamma_i\mathbf{S}(\mathbf{b}_i(t))\tilde{\mathbf{b}}_i(t) \\ \dot{\widehat{\boldsymbol{\chi}}}_2(t) = -\mathbf{S}_3(\boldsymbol{\omega}_m(t) - \widehat{\boldsymbol{\beta}}(t))\widehat{\boldsymbol{\chi}}_2(t) + \mathbf{C}_2^T\mathtt{Q}^{-1}[\widehat{\mathbf{b}}(t) - \mathbf{C}_2\widehat{\boldsymbol{\chi}}_2(t)] \end{cases} \tag{4.122}$$

The error dynamics of 4.122 are similar to 4.120, but with $\mathbf{u}_2(t) \equiv \mathbf{S}_3(\tilde{\boldsymbol{\beta}}(t))\boldsymbol{\chi}_2(t) + \mathbf{C}_2^T\mathtt{Q}^{-1}\tilde{\mathbf{b}}(t)$. It is shown in [40] that this refinement does not affect the nominal asymptotic stability analysis of the cascade attitude observer 4.119.

Additionally, there is nothing in the observer structure imposing the restriction that the estimate of the rotation matrix $\widehat{\mathbf{A}}(t)$ belongs to $SO(3)$. As both the vector observations and the rate gyro readings are subject to noise, that induces errors in the rotation matrix estimate. However, as the error converges to a tight neighborhood of zero, estimates arbitrarily close to $SO(3)$ can be obtained by employing computationally efficient orthogonalization cycles given by:

$$\widehat{\mathbf{A}}_f(t) = \frac{1}{2}(\widehat{\mathbf{A}}(t) + [\widehat{\mathbf{A}}^T(t)]^{-1}) \tag{4.123}$$

The overall system presented is computationally efficient and appropriate for application in platforms where computational resources are scarce [40].

Chapter 5

Implementation

This chapter discusses, in section 5.1, the development of a simulation environment that realistically describes the environment and motion conditions of a spacecraft, enabling the generation of realistically data to feed the sensors and properly test the estimation algorithms implemented. It also describes the integration and implementation of the attitude estimation algorithms of chapter 4, in section 5.2. Finally, it analysis the computational resources efficiency of the three algorithms, in section 5.3.

5.1 Simulation Environment

The simulation environment developed aims at realistically describe the conditions felt by the spacecraft. The simulation tools used were Matlab and Simulink. It is composed of an environment model, that simulates the position of other celestial objects and disturbance effects on the acceleration and torque of the spacecraft, a dynamic model that simulates the effects of forces and torques on the nanosatellite, propagating its orbit and attitude, a suite of sensor models that calculate the sensors outputs, and, finally, the implementation of the estimation algorithms proposed.

5.1.1 Environment Model

The environment model aims at recreating the space environment around the satellite. It simulates the position of celestial objects such as the Sun, the Moon, and stars, recreating their effects in terms of forces and torques upon the spacecraft. It also recreates Earth's magnetic field and atmosphere, and the effects caused by them.

Sun and Moon positions were obtained from accurate orbital ephemerides of the Sun and Moon computed by the Jet Propulsion Laboratory (JPL) of the California Institute of Technology, namely the DE405 ephemerid. These ephemerides are obtained by precise numerical integration of the equations of motion of the celestial bodies, which are formulated taking into account a considerable amount of perturbations and celestial influences and are generally created to support spacecraft mission planning. Sun shadow is simulated assuming that the shadow created by the Earth is a cylindrical projection of the Earth's diameter along the direction of the Sun to the Earth [11].

The Earth's magnetic field model implemented in this work is The World Magnetic Model 2015 (WMM), which represents only the long wavelength portion generated by the main geomagnetic field, but is accurate enough for the purposes of this simulation. The WMM consists of a degree and order 12 spherical-harmonic main field model comprised of 168 spherical-harmonica Gauss coefficients and degree and order 12 spherical-harmonica Secular-Variation field model. It is the standard model used by the United States Department of Defense or the North Atlantic Treaty Organization (NATO), for navigation, attitude and heading referencing systems using the geomagnetic field [32].

Earth's atmosphere emulation follows a Exponentially Decaying Model Atmosphere, which is a fully static model that assumes the atmospheric density decays exponentially with increasing height, and assumes an axially symmetric atmosphere about the polar axis. It is a simple model but yields decent results, being suitable for the implementation in this work. Atmospheric density values were retrieved from [11].

Perturbation forces and perturbation torques were computed based on equations presented in section 2.3.3.

5.1.2 Dynamic Model

The dynamic model propagates the orbit and attitude of the satellite, taking into account the forces and torques acting on it. For that, the differential equations 2.22, 2.32, and 2.31d, representing, respectively the orbital dynamics, attitude dynamics and attitude kinematics, are numerically integrated to obtain the inertial position, velocity, angular rate and attitude of the satellite. The perturbations computed by the environment model are taken into account.

A 6U CubeSat was considered, with magnetic dipole **m** per axis of $0.0280Am^2$ and a moment of inertia $\boldsymbol{J}_{ii} = [0.0541, 0.0914, 0.1054] kg\,m^2$.

For the orbit propagation, Cowell's method of special perturbations is used, being the orbit first described as a Keplerian orbit, and then perturbations are added. A class of orbits can be simulated by altering the position and velocity initial conditions.

5.1.3 Sensors Model

As detailed in section 3.3, the sensor suite assumed available onboard for purposes of attitude determination is composed by a rate gyroscope, a Sun sensor, a magnetometer and a star tracker.

Reference quantities such as the true angular rate, the position of the Sun and other stars, or the magnetic field, were provided by the environment and dynamic models. Sensor outputs were created using the models of section 3.4.

The star tracker is assumed to provide the directions of three stars in its FOV and the respective unit vectors in an inertial frame, at all times.

The rate gyroscope measurements were created following equations 3.9, assuming perfect alignment, where $\boldsymbol{\omega}$ comes from the dynamic model. A slowly time-varying bias was considered, which is a very common and undesirable characteristic of rate gyros.

Reference [33] provides an overview of the current state of the art of small spacecraft technology, with particular emphasis on the state of the art of CubeSat-related technology. From that reference, state-of-the-art hardware was selected, in order to retrieve their characteristics and provide realistic sensor information. The sensor characteristics selected are those of the ST-16 star tracker from Sinclair Interplanetary, the RM3100 magnetometer from PNI Corp, the Fine Digital Sun Sensor from New Space systems and the LN-200S gyroscope from the Northrop Grumman. Sensors characteristics are presented in Table 5.1.

Sensor	Sample Frequency	Accuracy	Noise Std. Dev.
Gyroscope	100 Hz		$\sigma_v = 1.18 \times 10^{-2}\ {}^\circ/s$ $\sigma_u = 2.78 \times 10^{-4}\ {}^\circ/s$
Magnetometer	100 Hz	30 nT	$\sigma_{mag} = 30nT$
Sun sensor	100 Hz	0.1°	$\sigma_{ss} \approx 1.70 \times 10^{-3} rad$
Star tracker	10 Hz	74 arcsec	$\sigma_{star} = 3.59 \times 10^{-4} rad$

Table 5.1: Sensors characteristics.

5.2 Attitude Determination Methods Implementation

Three types of attitude determination methods were studied in chapter 4, namely deterministic methods, state estimation methods and nonlinear observers. For simulation purposes, the QUEST algorithm of section 4.1.4, the MEKF of section 4.2.2 adapted to Murrell's version and the nonlinear cascaded observers of section 4.3, were implemented. That implementation is described in this section.

5.2.1 Deterministic Methods Implementation

The implementation of QUEST followed directly from section 4.1.4, with non-negative weights a_i defined as the inverse of the respective sensors variance. The error covariance **P** $_{\vartheta\vartheta}$ was also computed, providing important knowledge about the estimate accuracy.

The QUEST algorithm ran at the rate of $10\,Hz$.

5.2.2 MEKF Implementation

The gyro calibration MEKF was implemented as formulated in Table 4.3 with the respective adaptations to Murrell's version for more computational efficiency, as illustrated in flowchart 4.1.

The angular rate information provided by the gyroscope was used in the filter construction and **Q** and **R** matrices were computed, according to section 4.2.2, using the noise characteristics of the sensors provided in Table 5.1.

The MEKF ran at the rate of $100\,Hz$.

5.2.3 Nonlinear Observer Implementation

The nonlinear observer presented in section 4.3, based directly on the continuous-time system dynamics, was not immediately implemented, as the sensors considered output discrete-time data, at different rates. For that, the observer must be discretized. As stated before, for the NL observer operation, only vector measurements originated from quantities that can be considered locally constant in inertial coordinates, can be used. The latter implicated that the magnetometer could not be used, as its representation in inertial coordinates has serious variations.

The dynamical system 4.122 was discretized using the first-order Euler method, with the right side of the system subject to sample-and-hold. The Euler discretization of the system dynamics gives:

$$\begin{cases} \widehat{\mathbf{b}}_{1_{k+1}} = \widehat{\mathbf{b}}_{1_k} + T_g(-\mathbf{S}(\boldsymbol{\omega}_{m_k})\widehat{\mathbf{b}}_{1_k} - \mathbf{S}(\mathbf{b}_{1_k})\widehat{\boldsymbol{\beta}}_k + \alpha_1\tilde{\mathbf{b}}_{1_k}) \\ \vdots \\ \widehat{\mathbf{b}}_{N_{k+1}} = \widehat{\mathbf{b}}_{N_k} + T_g(-\mathbf{S}(\boldsymbol{\omega}_{m_k})\widehat{\mathbf{b}}_{N_k} - \mathbf{S}(\mathbf{b}_{N_k})\widehat{\boldsymbol{\beta}}_k + \alpha_N\tilde{\mathbf{b}}_{N_k}) \\ \widehat{\boldsymbol{\beta}}_{k+1} = \widehat{\boldsymbol{\beta}}_k + T_g(\sum_{i=1}^{N}\gamma_i\mathbf{S}(\mathbf{b}_{i_k})\tilde{\mathbf{b}}_{i_k}) \\ \widehat{\boldsymbol{\chi}}_{2_{k+1}} = \widehat{\boldsymbol{\chi}}_{2_k} + T_g(-\mathbf{S}_3(\boldsymbol{\omega}_{m_k} - \widehat{\boldsymbol{\beta}}_k)\widehat{\boldsymbol{\chi}}_{2_k} + \mathbf{C}_2^T\mathtt{Q}^{-1}[\widehat{\mathbf{b}}_k - \mathbf{C}_2\widehat{\boldsymbol{\chi}}_{2_k}]) \end{cases} \quad (5.1)$$

where T_g denotes the sampling rate at which the observer runs, corresponding to the sampling rate of the gyro. Therefore, the NL observer has a rate of $100\,Hz$.

The rate gyro, Sun sensor, and star tracker provide measurements at different rates. The full vector of measurements **b** is then not available at every time step k, hence the system 5.1, as it is presented, cannot be propagated. To overcome this problem, an estimate of the measurements $\mathbf{b}_k^e \approx \mathbf{b}_k$ is used instead. At every time k, Sun sensor readings are available, but star tracker measurements do not. Between the instants at which the star tracker provides measurements, open-loop integration of the last star tracker measurements using the angular information coming from the rate gyro are used. In this way, sensors with low and high sampling rates are fused to perform the estimation.

Parameter Tuning

The implemented cascade observers can essentially be divided into a bias observer and an attitude observer. The bias observer has as tuning parameters the positive scalar constants $\alpha_i\,, \gamma_i,\ i = 1, ..., N$, while the attitude observer only has the positive definite matrix $\mathtt{Q}$.

In tuning nonlinear observers, the characteristics of the sensor noise are taken into account by proper tuning the available parameters. An intuitive approach can be taken: the larger the parameters are (with the exception of $\mathtt{Q}$ that is inverted in the system), the faster the error converges to zero. However, as the parameters increase, the sensitivity of the estimates to sensor noise increases, possibly even undesirably amplifying the measurement noise, affecting performance. It will always be a trade-off between convergence speed and performance.

The knowledge about the measurement noise characteristics of the sensors can be exploited in the design of the observer gains. This can be done through running the observer in simulation, replicating the sensors characteristics, and searching for the minimum estimate error over a discrete array of values for the parameters, analyzing convergence times at the same time. After doing that analysis, a balance between estimation error and convergence time was achieved for the parameters, $\alpha_i = 1\,, i = 1, ..., N$, $\gamma_i = 0.016\,, i = 1, ..., N$ and $\mathtt{Q} = 0.03$. These were the parameters used in the simulations.

5.3 Computational Resources Efficiency Analysis

A real-world implementation of any of the methods from chapter 4 on a spacecraft will always be under serious constraints in terms of computational resources. Therefore, the amount of computational complexity or volume of work of an algorithm should be known in advance before selecting it.

One way to quantify the computational complexity of an algorithm is to count FLOPs (Floating Point Operations per second). A FLOP (Floating Point Operation) is either an addition, subtraction, multiplication or division of floating point numbers [43]. The convention used in this work assumes square-root and trigonometric operations to also require 1 FLOP. The real implementation complexity and FLOP count for an algorithm operation will always depend on the hardware chosen for the implementation.

In this section, the number of FLOPs needed to run each of the three algorithms is analyzed and compared. Filter operation with the rates specified in section 5.2 and with four available vector measurements, namely the Sun sensor measurement and the three star tracker measurements, was considered.

5.3.1 QUEST FLOPs Analysis

QUEST can be summarized as follows from section 4.1.4:

1. Compute the initial estimate λ_0.

2. Construct the quartic equation 4.18 for λ.

3. Iterate through Newton-Raphson method obtaining λ_{max}.

4. Compute the quaternion estimate from λ_{max} with equation 4.15.

Item 1 requires the computation of four a_i weights, and their sum, reaching the total of 11 FLOP.

Item 2 requires the construction of matrix $\mathbb{B}$, requiring 99 FLOP, the construction of $\mathbb{z}$ requiring 3 FLOP and the construction of matrix $\mathbb{S}$, requiring 9 FLOP. Calculating ϱ needs 3 FLOP, κ requires 11 FLOP and $\mathit{\Delta}$ requires 14 FLOP. Calculation of *a* needs 2 FLOP, *b* needs 7 FLOP, *c* requires 21 FLOP, and *d* needs 20 FLOP. Therefore, constructing the quartic polynomial needs the total of 189 FLOP.

Item 3 requires the calculation of $\lambda_{k+1} = \lambda_k - \frac{f(\lambda_0)}{f\prime(\lambda_0)}$. It implicates the evaluation of $f(\lambda_0)$ requiring only 13 FLOP and then the evaluation of $f\prime(\lambda_0) = 4\lambda_0^3 - 2(a + b)\lambda_0 - c$, requiring 5 FLOP. Each iteration requires then the total of 18 FLOP. Assuming a single iteration Item 3 requires the total of 18 FLOP.

Item 4 requires the computation of the optimal quaternion and its normalization, in a total of 83 FLOP.

Finally, a complete iteration of the algorithm requires $11+189+18+83 = 301$ FLOP. QUEST algorithm runs at 10Hz, making the total number of $301 \times 10 = 3010$ FLOPs.

5.3.2 MEKF FLOPs Analysis

The MEKF implementation, follows Murrell's version, and it can be summarized as follows from flowchart figure 4.1.

1. Compute the estimated rotation matrix from the estimated quaternion.
2. Compute the sensitivity matrix.
3. Compute the Kalman gain.
4. Update the covariance.
5. Compute the residual and update state.
6. Repeat steps 2 to 5 until all measurements are processed.
7. Reset.
8. Propagate until the next observation time.

These items can be divided into update step, contemplating items 1 to 6, reset step, and propagation step.

Item 1 takes 56 FLOP. Item 2 requires 15 FLOP. To compute the Kalman gain, 648 FLOP are needed. Item 4 takes 612 FLOP. Item 5 requires 75 FLOP. Item 1 just runs one time for each update step, but items 2 to 5, are repeated for each measurement available. The sensors available have different rates. The Sun sensor runs at 100 Hz and the star tracker at 10 Hz. Then, for an update with 4 measurements, $56 + (15 + 648 + 612 + 75) \times 4 = 5456$ FLOP are needed, as for an update with 1 measurement, only 1406 FLOP. Noting again that the EKF runs at the rate of 100 Hz, it means that for the total of 1 second, there are 90 measurement updates with only 1 measurement, and 10 using 4 measurements. Hence for the total of 1 second, the measurement update step needs $90 \times 1406 + 10 \times 5456 = 181100$ FLOP.

The reset step requires 43 FLOP and happens, once for each iteration of the total filter, hence for a total of 1 second, $43 \times 100 = 4300$ FLOP are required.

The propagation from one measurement time to the next, requires 1111 FLOP and it happens once for each iteration of the total filter, hence for a total of 1 second, $1111 \times 100 = 111100$ FLOP are required.

Then, the MEKF implemented requires $181100 + 4300 + 111100 = 296500$ FLOPs.

5.3.3 Nonlinear Observer FLOPs Analysis

The nonlinear observer implementation can be summarized as follows:

1. Combine high frequency measurements, with low ones, propagating the low ones, to create an estimate, $\mathbf{b}_k^e$, of the measurements.
2. Compute the necessary parameters.
3. Obtain estimates with bias observer.

4. Obtain estimate with attitude observer.

5. Perform orthogonalization cycles.

Item 1 requires 64 FLOP. Item 2 needs 39 FLOP. Item 3 can be divided into 4 measurement estimates, each one requiring 45 FLOP, and a bias estimate requiring 87 FLOP, hence the total of $45 \times 4 + 87 = 267$. The attitude observer requires the total of 723 FLOP. Finally each orthogonalization cycle require 68 FLOP, and 2 are performed. The total number of FLOP for a single iteration of the nonlinear observer is $64 + 39 + 267 + 723 + (68 \times 2) = 1229$ FLOP. Running at 100Hz, the nonlinear observer requires 122900 FLOPs.

An alternative metric can be defined, where it is assumed that the four available sensors provide vector measurements at all discrete times. Based on that assumption, and following the steps stated above, the evolution of the FLOPs required for each algorithm, depending on the operation rate, can be analyzed and compared. The stated evolution is illustrated in figure 5.1.

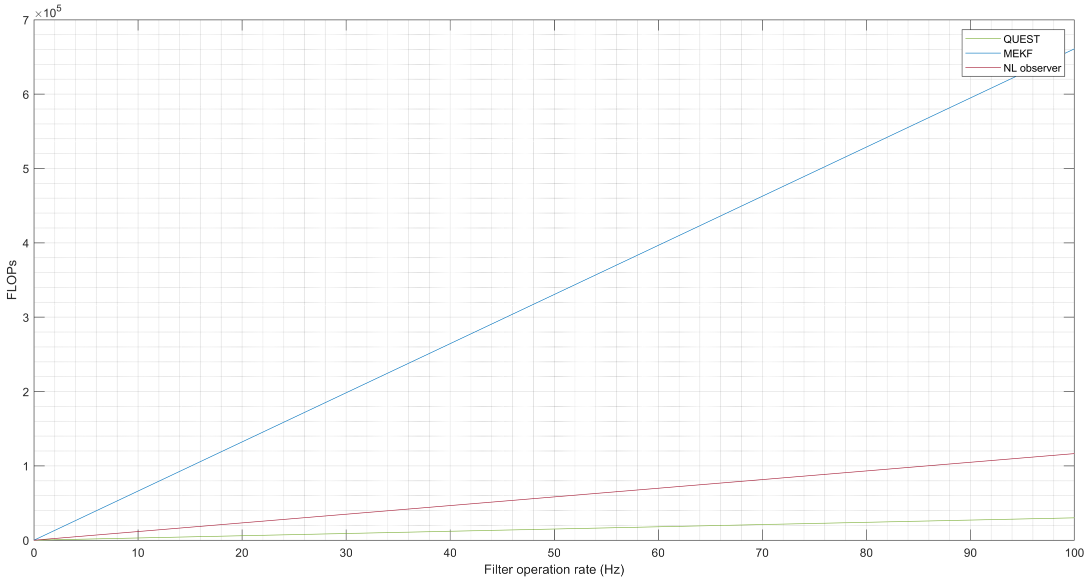


Figure 5.1: Required FLOPs for the three algorithms, depending on the operation rate.

Chapter 6

Results

The final goal of this work was to use the implemented algorithms in the realistic simulation environment created, in order to compare their performances. For that purpose, different simulation cases were created. This chapter describes those case scenarios. Then, the results obtained are presented and section 6.1 provides a discussion of those results.

First of all, the satellite dynamics and kinematics are simulated in continuous-time, while the attitude determination methods are run in discrete-time. The solver used was chosen based on the analysis done in [44], and is the fixed-step eighth order Dormand-Prince method. This method computes the state at the next time step as an explicit function of the current value of the state and the state derivatives approximated at intermediate points. The solver runs with a fixed time-step of $0.01\,s$.

In the simulations performed, it is assumed that the satellite has been detumbled before the attitude estimation begins and that all sensors work independently of each other and without problems at the angular rates experienced by the satellite. Additionally, no control torque is applied during the simulations, i.e. no particular attitude is actively maintained, and only the error between the true value and the estimated value is used to perform the accuracy studies proposed. The true attitude is assumed to be the one generated by the simulator, which is considered reality in the simulation.

Two different orbits were used to perform the simulations. Orbit 1 was selected as a Geostationary Transfer Orbit (GTO), the predefined initial orbit for Mission 1. Orbit 2 is a typical Low-Earth Orbit that can realistically describe the orbit conditions for Mission 2. The characteristics of the two orbits are presented in Table 6.1. Implications of these two orbits essentially differ on the acceleration and torque perturbations present, and on the availability of the magnetometer sensor, which is not available in Orbit 1. All other sensors are available in both orbits.

Different simulation cases were created to emulate different realistic scenarios and draw different conclusions.

Case 1

Case 1 aimed at comparing the three attitude determination algorithms implemented, with all the sensors available. As stated before, the magnetometer is not available in Orbit 1 and cannot be used by

Parameter	Unit	Orbit 1	Orbit 2
Epoch	-	16 Dec 2023 10h:45min:40s	01 Jan 2019 00h:00min:00s
Orbit semi-major axis	km	24364.64	6781.20
Orbit eccentricity	-	0.73	8.46×10^{-4}
Orbit inclination	$\deg$	27.50	51.73
Orbit RAAN	$\deg$	360	112.70
Orbit argument of perigee	$\deg$	267	80
Orbit initial true anomaly	$\deg$	122.50	162.20
Orbital period	h	10.50	1.54

Table 6.1: Orbits characteristics.

the NL observer.

The true initial attitude was defined as the identity quaternion $\mathbf{q}_0 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$ and the initial bias was $\boldsymbol{\beta}_0 = \frac{\pi}{180} \begin{bmatrix} -0.02 & 0.03 & -0.01 \end{bmatrix}^T rad\, s^{-1}$.

The angular rate followed $\boldsymbol{\omega} = \frac{\pi}{180} \begin{bmatrix} 0.1\, \cos(\frac{2\pi}{200}t) & 0.15\, \cos(\frac{2\pi}{180}t) & 0.05\, \cos(\frac{2\pi}{200}t) \end{bmatrix}^T rad/s$.

The MEKF and NL observer were initialized with a considerable attitude and bias estimation error, defining these errors as the difference between the true attitude and bias, and the estimated ones coming from the algorithms, respectively.

The initial attitude estimate was $\widehat{\mathbf{q}}_0 = \begin{bmatrix} 6.0692 \times 10^{-02} & 6.9371 \times 10^{-01} & 6.0692 \times 10^{-02} & 7.1512 \times 10^{-01} \end{bmatrix}^T$, corresponding to an initial attitude estimation error in terms of Euler angles of $\delta\phi = 80^\circ$, $\delta\theta = 80^\circ$ and $\delta\psi = 80^\circ$. The initial bias estimate was $\widehat{\boldsymbol{\beta}}_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T rad\, s^{-1}$, corresponding to a initial bias estimation error of $||\Delta\widehat{\boldsymbol{\beta}}_0|| = 134.70^\circ\ h^{-1}$, in magnitude.

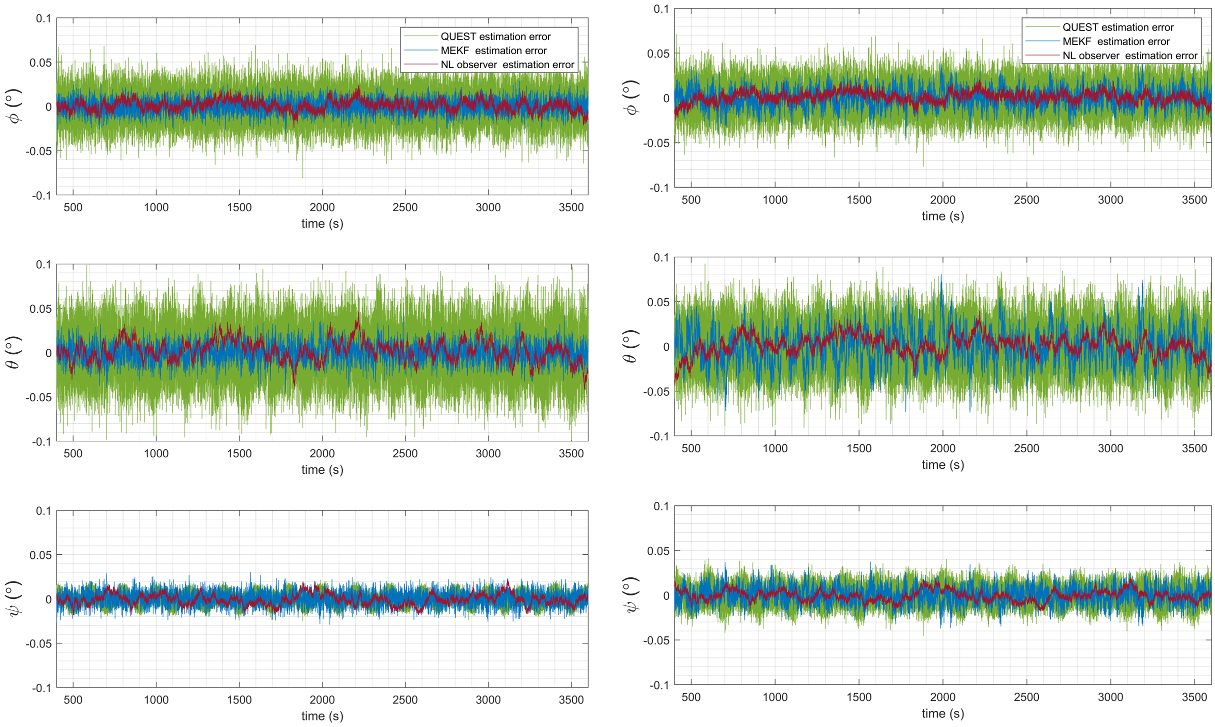
The initial covariance matrix for the MEKF was defined as $\mathbf{P}_0 = diag(100, 100, 100, 10, 10, 10)$, reflecting the uncertainty of the initial estimates. The NL observer estimates for the measurements were initialized as zero, and the observer parameters were defined as described in section 5.2.3.

The three algorithms ran in parallel, in a simulation of $3600s$. The detailed evolution of the angular estimation error, represented as Euler angles, after the initial transient faded out, for the two orbits, is presented in figure 6.1, and the respective root-mean-square errors are available in Table 6.2.

	Orbit 1			Orbit 2		
	QUEST	MEKF	NL observer	QUEST	MEKF	NL observer
ϕ_{RMS} (°)	1.7214×10^{-2}	7.0025×10^{-3}	6.2568×10^{-3}	1.6992×10^{-2}	1.0059×10^{-2}	5.9629×10^{-3}
θ_{RMS} (°)	2.5538×10^{-2}	1.0120×10^{-2}	1.1909×10^{-2}	2.3371×10^{-2}	1.9754×10^{-2}	1.1167×10^{-2}
ψ_{RMS} (°)	5.4279×10^{-3}	6.8745×10^{-3}	6.2647×10^{-3}	9.6787×10^{-3}	9.0957×10^{-3}	5.9824×10^{-3}

Table 6.2: Root Mean Square of the angular estimation error for the QUEST, the MEKF and the NL observer for simulation case 1 in the interval $(400s$; $3600s)$.

The initial convergence of the angular estimation error, for the simulation in Orbit 1 is represented in figure 6.2, and the initial convergence and detailed evolution of the bias estimation error can be seen in figure 6.3.



(a) Orbit 1

(b) Orbit 2

Figure 6.1: Detailed evolution of the angular estimation error for simulation case 1.

Case 2

Case 2 aimed at studying the estimation performance when the star tracker has a fault, stopping to provide its 3 star measurements successively.

The same initial simulation conditions of Case 1 were used. The star tracker was assumed to enter into fault at time $500s$, when the first star sensor measurement is lost. The second is lost at time $800s$, and the third at $1100s$, entering into complete failure.

After the complete failure of the sensor, in the case of simulations in orbit 1, just the Sun sensor was left available, while in the case of simulations in orbit 2, both the Sun sensor and the Magnetometer were available. When just one measurement is available, the QUEST and NL Observer cannot run, hence, an open loop propagation takes place starting from the last estimate of each algorithm. While the propagation for QUEST was done directly using the angular rate provided by the gyro, for the NL observer, the same angular rate was corrected using the last bias estimate of the filter.

The three algorithms ran in parallel. The detailed evolution of the angular estimation error, represented as Euler angles, with the star tracker fault happening after the initial transient faded out, for the two orbits, is presented in figure 6.4, and the respective root-mean-square errors are available in Table 6.3.

Case 3

Case 3 aimed at studying the estimation performance when only the star tracker was available.

The same initial conditions of Case 1 were used. A simulation in Orbit 2 was performed where at

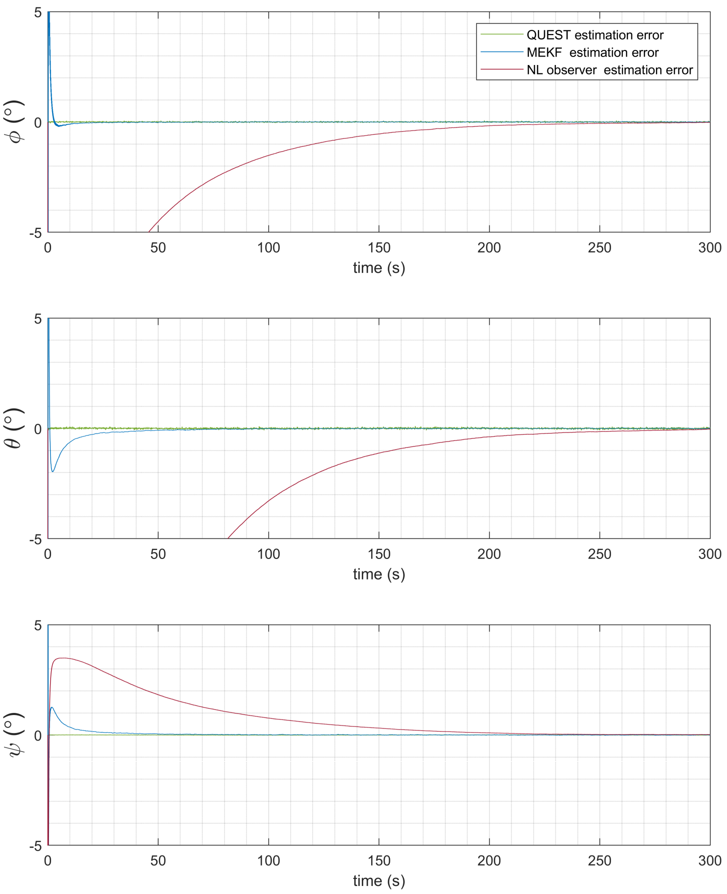


Figure 6.2: Initial convergence of the angular estimation error for simulation case 1.

	Orbit 1			Orbit 2		
	QUEST	MEKF	NL observer	QUEST	MEKF	NL observer
ϕ_{RMS} (°)	3.4094×10^{1}	2.3245×10^{-1}	6.9455×10^{-1}	1.2279×10^{-1}	1.0339×10^{-2}	6.0389×10^{-1}
θ_{RMS} (°)	3.1325×10^{1}	5.3866×10^{-1}	5.0615×10^{-1}	1.8045×10^{-1}	1.6615×10^{-2}	4.6671×10^{-1}
ψ_{RMS} (°)	4.2578×10^{1}	5.1819×10^{-2}	9.2886×10^{-1}	8.7629×10^{-2}	8.4730×10^{-3}	9.4055×10^{-1}

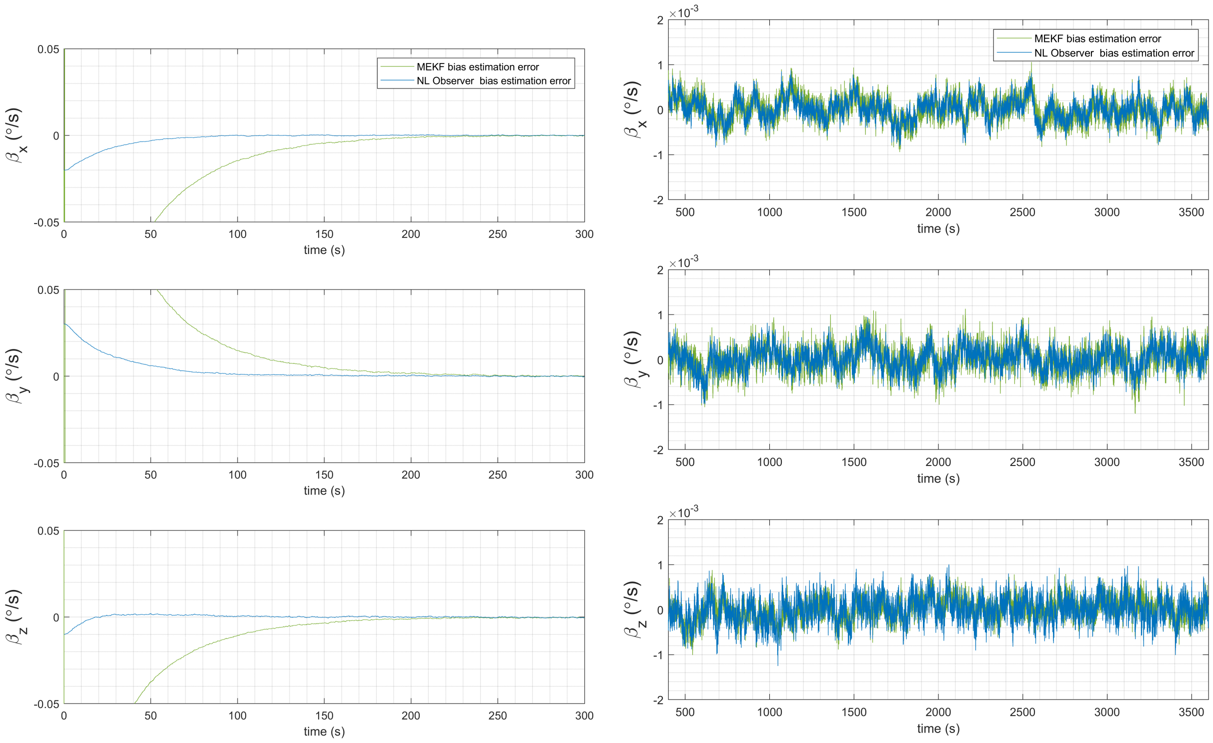
Table 6.3: Root Mean Square of the angular estimation error for the QUEST, the MEKF and the NL observer for simulation case 2 in the interval $(1100s\ ;\ 3600s)$.

$750s$ the Sun sensor and the magnetometer suffered a fault. From $750s$ to $3600s$, the three estimation filters ran only with the measurements provided by the star tracker.

The detailed evolution of the angular estimation error, represented as Euler angles, after the initial transient faded out, is presented in figure 6.5, and the respective root-mean-square errors are available in Table 6.4.

	QUEST	MEKF	NL observer
ϕ_{RMS} (°)	1.7251×10^{-2}	9.5974×10^{-3}	4.9036×10^{-3}
θ_{RMS} (°)	2.3712×10^{-2}	9.1414×10^{-3}	8.4132×10^{-3}
ψ_{RMS} (°)	9.6778×10^{-3}	5.7051×10^{-3}	7.4242×10^{-3}

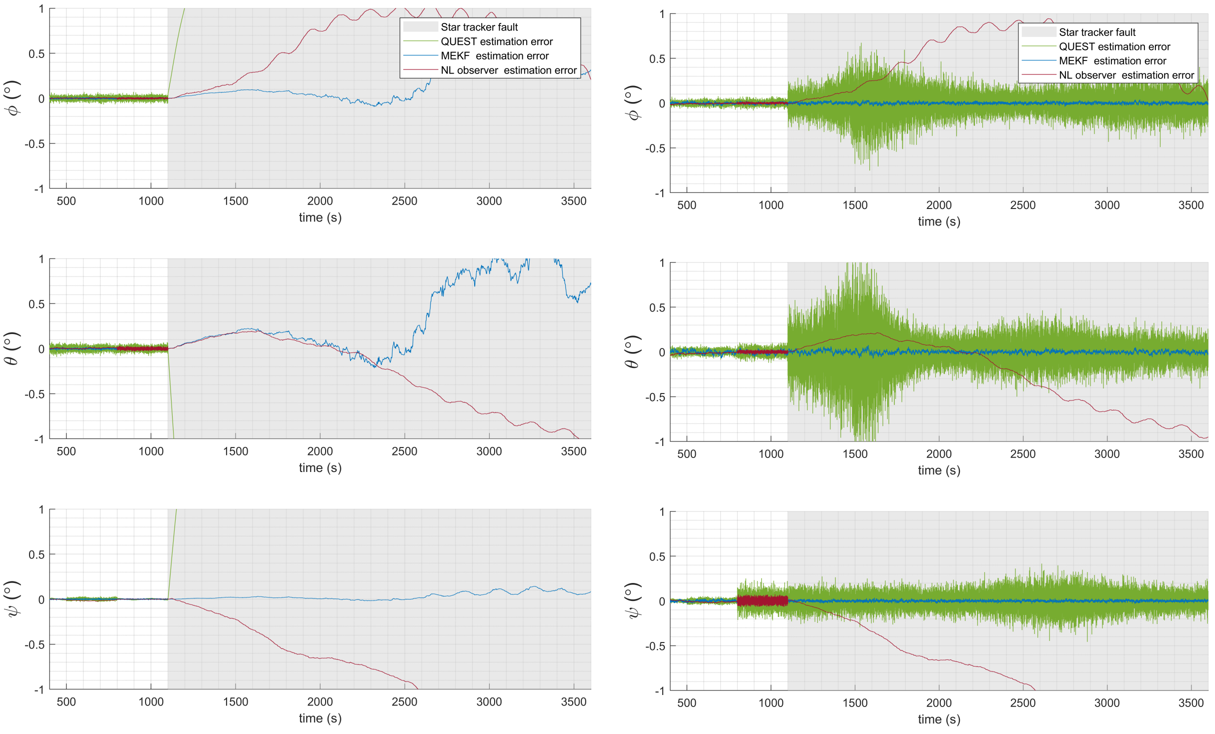
Table 6.4: Root Mean Square of the angular estimation error for the QUEST, the MEKF and the NL observer for simulation case 3 in the interval $(750s\ ;\ 3600s)$.



(a) Bias estimation error initial convergence.

(b) Bias estimation error detailed evolution.

Figure 6.3: Initial convergence and detailed evolution of the bias estimation error for simulation case 1.



(a) Orbit 1

(b) Orbit 2

Figure 6.4: Detailed evolution of the angular estimation error for simulation case 2.

Case 4

Case 4 aimed at studying how the 3 filters responded in a situation of re-acquisition of sensors after a drift due to faults.

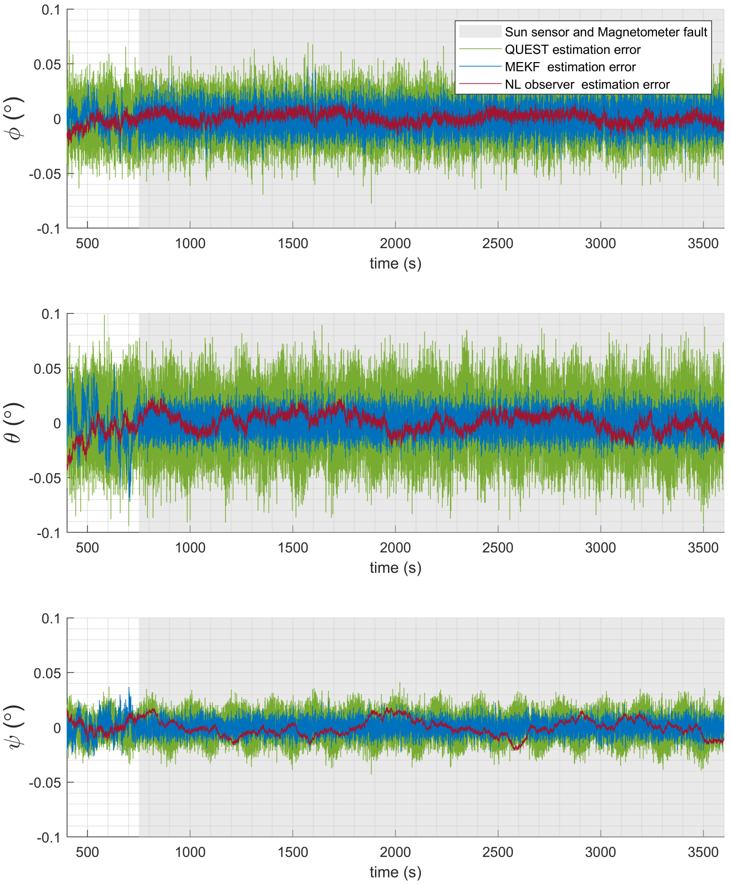


Figure 6.5: Detailed evolution of the angular estimation error for simulation case 3.

The end of the simulation case 2, orbit 1, is a good example of a situation where a drift leads to considerable angular estimation errors, and was used as a starting point for simulation in case 4. After $3600s$, all the attitude sensors became available again. The three algorithms ran in parallel and the evolution of the angular estimation error, represented as Euler angles is presented in figure 6.6.

6.1 Discussion

Case 1

Case 1 enabled the comparison of the QUEST, MEKF and NL Observer in a situation of initial attitude and bias acquisition and in nominal situation, its ideal working mode, with all sensors fully available depending on the orbit.

The analysis of results for Case 1 may start with the analysis of the initial transients for attitude and bias estimation. Figure 6.2 shows the initial transient of the estimation error for each algorithm and it can be seen that the 3 algorithms behave very differently. The QUEST algorithm, owing to its characteristics, does not have a transient period, achieving steady state performance from time $0s$. The MEKF achieves steady state for the three Euler angles before time $50s$. The NL observer has shown to be the algorithm with the biggest transient, needing $300s$ to achieve its steady state performance.

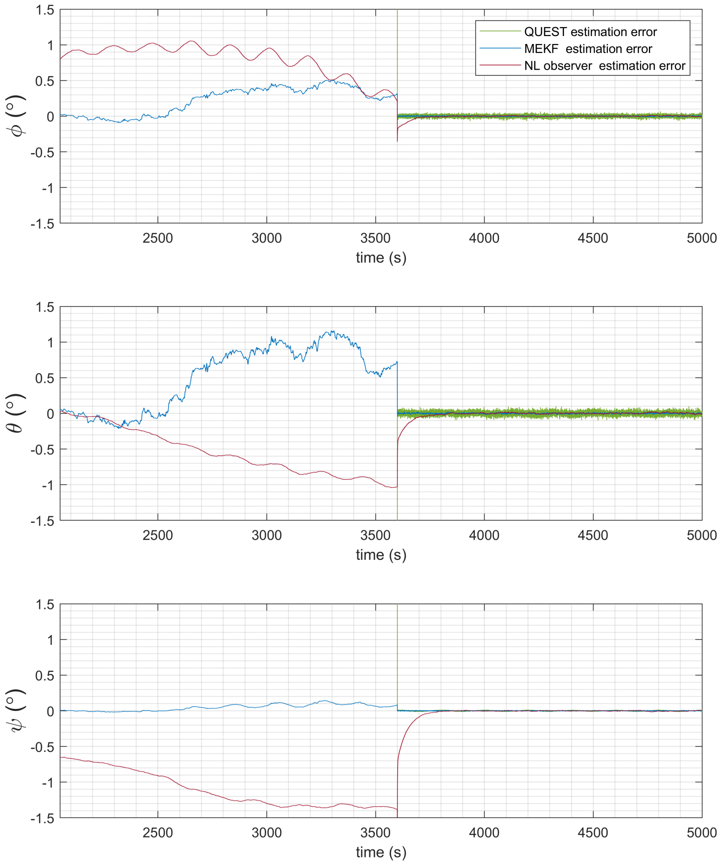


Figure 6.6: Evolution of the angular estimation error for simulation case 4 with sensors reacquisition at $3600s$.

The bias estimation error initial convergence is represented in figure 6.3(a). Contrary to what happens in terms of attitude convergence, the NL observer bias estimation shows to converge to zero, in less than $100s$, which is faster than the MEKF bias estimation, which requires at least $200s$ to reach the same values.

It is important to note that Case 1 simulates extreme conditions in terms of initial angular and bias estimation errors, which do not affect the QUEST estimates but have impact in the transient period for the MEKF and NL observer. The uncertainty of the initial estimates led to the definition of a initial covariance matrix for the MEKF with relatively high values, and to observer parameters reflecting a trade-off between convergence time and steady-state performance. The high values in the initial covariance matrix have impact in the highly rapid variations of the attitude and bias estimation for the MEKF in the initial seconds. A different tuning for the initial covariance, and for the observer parameters may translate in faster transients for the two algorithms. However, even with extreme conditions, the convergence times presented should not implicate any disadvantages in terms of a real application, and less initial error should translate into smaller transient periods.

The nominal mode or steady-state performance for the 3 algorithms can be analyzed by inspecting the detailed evolution of angular and bias estimation errors in figures 6.1 and 6.3(b), respectively. The behavior of the detailed evolution of the angular estimation error is similar for the simulation in Orbit 1 or Orbit 2, for the three algorithms, which indicates that the presence of the magnetometer, when all the other sensors are available, does not reflect in a better attitude estimation accuracy for the QUEST and MEKF. Overall, the QUEST algorithm has the biggest estimation error, followed by the MEKF and NL Observer that have approximately the same performance. Root-mean-square errors did not exceed $2.5538 \times 10^{-2\circ}$ for the QUEST, $1.9754 \times 10^{-2\circ}$ for the MEKF and $1.1909 \times 10^{-2\circ}$ for the observer.

In terms of the analysis of the bias estimation error evolution, the performance of the MEKF and the NL Observer is similar, with maximum root mean square error for the bias estimation of $5.045 \times 10^{-6\circ}/s$ for the two cases.

This case analyzes probably the most important aspects when it comes to the three algorithms comparison, which are their convergence characteristics and steady state performance in terms of bias and angular error estimation. It's important to note that while the results achieved for the NL Observer and MEKF are a reflection of the tuning parameters and therefore can yet be adjusted, in the QUEST case, not much can be done.

To finalize, the rates at which these three algorithms run also have an impact on their performance. The filter operation rates chosen were not the same for the three filters. While QUEST ran at 10 Hz, the MEKF and NL Observer ran at 100 Hz. This choice was justified by the availability of star tracker measurements. QUEST strong dependence on the accuracy of the measurements justified its only use when star tracker measurements became available, hence the rate of 10 Hz. On the other hand, the MEKF and NL observer strongly benefit from the use of gyroscopic information, hence the rate chosen was the same as the one of the gyro. Choosing lower rates diminishes the computational resources complexity, but also means that with higher angular rates, a considerable movement can occur from one sample to the next, leading to greater following errors.

Case 2

Case 2 enabled the comparison of the performance of the 3 estimation algorithms when an essential sensor like the star tracker suffers a fault. The analysis of figure 6.4 and Table 6.3 leads to very different analysis for the two orbits, given the sensors left available in the two cases, which were the Sun sensor in the case of orbit 1 and the magnetometer and Sun sensor in the case of Orbit 2.

In the case of Orbit 1, analyzing figure 6.4(a) after the star tracker failure, the QUEST estimate is propagated with non-corrected angular rate measurements from the gyro, leading to a very fast drift that translates into a maximum root-mean-square error of $4.2578 \times 10^{1\circ}$. The NL observer estimate is propagated with angular rate measurements corrected by the last bias estimate, which minimizes the drift effect for a longer period, but eventually reaches fairly high values of error when in comparison of those obtained in nominal mode. The MEKF shows to be robust in this situation, showing to be the one that looses estimation accuracy more slowly, and being able to keep the lower estimation error with only one measurement. The maximum root-mean-square errors achieved in this period are for the MEKF of $5.1819 \times 10^{-2\circ}$ and $9.2886 \times 10^{-1\circ}$ for the NL Observer.

In the case of Orbit 2, figure 6.4(b), both the QUEST and MEKF are able to continue running with the Sun sensor and magnetometer measurements, but the NL Observer is in the same situation as in

Orbit 1. After the star tracker failure, the angular error estimate for the QUEST shows to increase, with a root-mean-square error one order of magnitude higher. In the case of the MEKF, the angular estimation error increases in comparison with Case 1, but a good following is still achieved.

This case made evident some of the flaws or weaknesses of the three algorithms compared. It showed that QUEST is highly dependent on the availability of two vector measurements. In the case of their unavailability a propagation aiming at minimizing the errors from lack of new estimates can still be done, as most spacecrafts incorporate a gyroscope, but this has shown to not be robust in practice, due to the lack of bias estimates. Additionally, without the star tracker, the increased angular estimation error evidenced reflects the QUEST highly dependence on the accuracy of the measurements used.

On the other hand, the NL observer will always have the possibility of correcting the rate gyro measurements with its last bias estimate, and mitigate the propagation errors in the case of lack of measurements. However, if this propagation is done for a long period, it will eventually drift to higher errors.

Finally,the MEKF has shown to be the more robust of the 3 algorithms. Even with just one measurement, the filter showed to be the one minimizing the errors, and with the sun sensor and magnetometer available, showed to be able to raise the importance of the rate gyro measurements when compared with the noisy sun sensor and magnetometer. In this last situation, those two sensors are mainly used for correction.

Case 3

Case 3 enabled the study of the 3 estimation algorithms when only the star tracker was left available. From figure 6.5 and Table 6.4, it is clear that the QUEST performance remained the same, with approximately the same root-mean-square errors as the ones obtained in Case 1. It is an expected result as the star tracker measurements are much more weighted in the QUEST algorithm than those coming from the Sun sensor and magnetometer.

When it comes to the MEKF and NL Observer, a slightly better performance can be noticed, with less angular error variations, as can be deduced by comparing the root-mean-square errors in Table 6.4, with those of table 6.2. The results obtained indicate that combining a magnetometer and Sun sensor with a star tracker does not necessarily give improved estimate accuracy, with the greater part of the accuracy coming from the star tracker. In fact, the bigger value of noise in the Sun sensor and magnetometer measurements can introduce variations in the estimates of the MEKF and NL Observer that disappear when only the more precise but less frequent star tracker measurements are used. That is something that can be taken into account in future tuning of parameters, lowering the importance given to these two sensors in those processes.

Case 4

Case 4 has the importance of showing how these algorithms respond to a re-acquisition of sensors that were not working properly due to eclipses or faults. That is illustrated by figure 6.6.

The simulation conditions of Case 2, orbit 1 are replicated for the first $3600s$. At time $3600s$, figure

6.6 points a considerable angular error estimation of more than 1° for some Euler angles. At that time, all the sensors measurements become available again. After the sensors re-acquisition, the QUEST convergence is immediate, as already was illustrated in the transient analysis in Case 1. The MEKF also shows an almost immediate convergence to values close to 0. On the other hand, the slower transients seen for the NL Observer in Case 1, are also visible in this analysis, where $200s$ are needed for the NL Observer angular estimate to converge again to values close to 0.

This simulation appeals once more to the convergence characteristics of the three methods, where it is illustrated that even after a long drift due to unavailability of measurements, the nominal working mode can be retaken if the sensors start working again.

Chapter 7

Conclusions

The main aim of this dissertation was to use the three estimation algorithms proposed to solve the problem of attitude estimation, in conditions representative of the two specified missions under the scope of the NANOSTAR project, giving meaningful insight on the advantages, disadvantages, computational resources efficiency and performance of the three algorithms. This would provide the project with a grounded analysis that could be used for decision making in terms of the ADCS system design.

In order to accomplish that aim, the first step was to select a sensor suite that could provide the necessary measurements, taking into account the constraints imposed for this kind of satellite. The sensor suite selected to feature the ADCS system determined the quantity and quality of the measurements available for the estimation, impacting its accuracy. A suite composed by a state-of-the-art Sun sensor, magnetometer, rate gyro and star tracker was selected. The decision for this set of sensors was made based on the type of mission at hands, the availability of measurements, and the accuracy required. The conclusion that combining the star tracker with a magnetometer and a Sun sensor did not give improved estimate accuracy could later be drawn. However, the added redundancy made the system more robust as could be seen in the results for simulation cases 2 and 3.

Afterwards, a simulation environment was created that replicated the translational and rotational motion of a satellite orbiting the Earth, feeding realistic space environment information to realistic sensor models, hence realistically simulating conditions similar to the ones expected for a satellite in orbit.

Finally, using the platform developed, three different algorithms for attitude determination were studied, implemented and compared in terms of computational resources efficiency, steady state performance, and performance in the case of failures, through realistic simulation cases, and conclusions were drawn.

The deterministic method studied, QUEST, was the least computationally complex algorithm employed, requiring approximately 40 times less FLOPs than required by the NL Observer, the second least complex. The chosen rate at which the QUEST algorithm ran, ten times lower than the one of the NL Observer, contributed to the less computational resources usage, but evidence was shown that with the same rate, QUEST would still be the least computationally complex algorithm. A steady state performance in terms of angular estimation similar to the other two algorithms was achieved by QUEST, when all sensors were available. However, the accuracy obtained has shown to be highly dependent on the quality of the sensors measurements, and their availability, showing serious limitations in terms of robustness when the sensors suffer failures. Additionally, the lack of estimation of other quantities such as gyro biases is a negative point for this kind of methods, that ultimately translates into a serious practical disadvantage.

The MEKF, on the other hand, showed to be a very robust solution, dealing favorably with sensor faults. Its steady state performance in terms of angular estimation has revealed the best, in pair with the results for the NL Observer. However, its computational complexity has shown to be the greatest of the three algorithms requiring more than the double of FLOPs than the NL Observer. The MEKF is also the algorithm with the most complex formulation, without proofs of convergence, and with the need to define initial estimates that can negatively impact its performance. Its ability to also estimate gyro biases proved extremely important as the gyro measurements could be calibrated.

The Nonlinear observer offered a steady state performance approximately the same as the one provided by the MEKF. It also required less than half of the FLOPs when compared to the MEKF, was more easily formulated and implemented, and had a set of parameters that could be more intuitively tuned depending on the constraints. It has shown, however, to be less robust than the MEKF, due to its inability to use the magnetometer, but still with a moderate response to faults, and the ability to calibrate the gyroscope measurements. Overall, it has shown to be an attractive solution in comparison with the MEKF due to its proof of globally exponentially stability.

To finalize, the selection of one of these three algorithms for the implementation into the ADCS of a NANOSTAR mission will always be a trade-off between computer and implementation complexity, the importance of robustness, and the steady state accuracy needed.

7.1 Future Work

Future work under the scope of the project includes the possibility of studying and implementing, under the simulation platform created, other interesting attitude determination solutions and sensors, as an extension to this work. Additionally, the development of a similar study as this one but regarding attitude control methods and actuators should be addressed. The next step would be to perform the final selection of one determination and one control methods, and a package of sensors and actuators, combining everything into the development of a full ADCS model.

The simulations and results provided in this work were products of a single set of scenarios, with well defined characteristics. However, it would be of interest to perform some sensitivity analysis through random variations in some key parameters of the final ADCS model. This could be done with Monte Carlo simulations, and would be an important step towards the validation of the solutions proposed.

After the validation step suggested, hardware implementation and testing should be addressed.

Bibliography

[1] NANOSTAR. What is a nanosatellite?, 2018. URL http://nanostarproject.eu/the-project/what-is-a-nanosatellite/.

[2] P. CubeSat. CubeSat Design Specification. 2015.

[3] J. Bouwmeester and J. Guo. Survey of worldwide pico- and nanosatellite missions, distributions and subsystem technology. *Acta Astronautica*, 67(7-8):854–862, oct 2010.

[4] J. Guo and C. Han. Where is the Limit: The Analysis of CubeSat ADCS Performance. *The 4S Symposium*, (1):1–15, 2016.

[5] X. Xia, G. Sun, K. Zhang, S. Wu, T. Wang, L. Xia, and S. Liu. NanoSats/CubeSats ADCS survey. In *2017 29th Chinese Control And Decision Conference (CCDC)*, pages 5151–5158. IEEE, may 2017.

[6] NANOSTAR. The Project, 2018. URL http://nanostarproject.eu/the-project/.

[7] NANOSTAR. Preliminary Design Challenge Mission Requirements. Technical report, 2019. URL https://nanostarproject.eu/wp-content/uploads/2019/02/NANOST-REQ-042-Space-mission-requirements{_}V2.pdf.

[8] H. D. Black. A passive system for determining the attitude of a satellite. *AIAA Journal*, 2(7):1350–1351, jul 1964.

[9] G. Wahba. A Least Squares Estimate of Satellite Attitude (Grace Wahba). *SIAM Review*, 8(3): 384–386, jul 1966.

[10] F. L. Markley and D. Mortari. Quaternion Attitude Estimation Using Vector Observations. *Journal of the Astronautical Sciences*, 48, 2000.

[11] F. L. Markley and J. L. Crassidis. *Fundamentals of Spacecraft Attitude Determination and Control*. 2014. ISBN 9781493908011.

[12] M. D. Shuster and S. D. OH. Three-axis attitude determination from vector observations. *Journal of Guidance and Control*, 4(1):70–77, jan 1981.

[13] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35, 1960.

[14] R. E. Kalman and R. S. Bucy. New Results in Linear Filtering and Prediction Theory. *Journal of Basic Engineering*, 83(1):95, 1961.

[15] J. L. Farrell. Attitude determination by kalman filtering. *Automatica*, 6(3):419–430, may 1970.

[16] E. J. Lefferts, F. L. Markley, and M. D. Shuster. Kalman filtering for Spacecraft Attitude Estimation. In *20th Aerospace Sciences Meeting*, Reston, Virigina, jan 1982. American Institute of Aeronautics and Astronautics.

[17] F. L. Markley. Multiplicative vs. Additive Filtering for Spacecraft Attitude Determination. page 8, 2003.

[18] J. L. Crassidis, F. L. Markley, and Y. Cheng. Survey of Nonlinear Attitude Estimation Methods. *Journal of Guidance, Control, and Dynamics*, 30(1):12–28, jan 2007.

[19] M. D. Shuster. A simple Kalman Filter and Smoother for Spacecraft Attitude. *Journal of the Astronautical Sciences*, 37:89–106, 1989.

[20] I. Y. Bar-Itzhack. REQUEST - A recursive QUEST algorithm for sequential attitude determination. *Journal of Guidance, Control, and Dynamics*, 19(5):1034–1038, sep 1996.

[21] S. Julier, J. Uhlmann, and H. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45(3): 477–482, mar 2000.

[22] J. L. Crassidis and F. L. Markley. Unscented Filtering for Spacecraft Attitude Estimation. *Journal of Guidance, Control, and Dynamics*, 26(4):536–542, jul 2003.

[23] G. T. Haupt, N. J. Kasdin, G. M. Keiser, and B. W. Parkinson. Optimal recursive iterative algorithm for discrete nonlinear least-squares estimation. *Journal of Guidance, Control, and Dynamics*, 19 (3):643–649, may 1996.

[24] N. J. Kasdin and T. Weaver. Recursive Satellite Attitude Estimation with the Two-Step Optimal Estimator. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Reston, Virigina, aug 2002. American Institute of Aeronautics and Astronautics.

[25] M. D. Shuster. A survey of attitude representations. *Journal of the Astronautical Sciences*, 41: 439–517, 1993.

[26] N. A. Chaturvedi, A. K. Sanyal, and N. H. McClamroch. Rigid-Body Attitude Control. *IEEE Control Systems*, 31(3):30–51, jun 2011.

[27] Federal Agency for Cartography and Geodesy. The 3rd realization of the International Celestial Reference Frame (ICRF3), 2013. URL `https://www.iers.org/IERS/EN/DataProducts/ICRF/ICRF3/icrf3.html`.

[28] J. R. Wertz. *Spacecraft Attitude Determination and Control*, volume 73 of *Astrophysics and Space Science Library*. Springer Netherlands, Dordrecht, 1978. ISBN 978-90-277-1204-2.

[29] J. R. Wertz and W. J. Larson. *Space Mission Analysis and Design*. Microcosm Press, 3 edition, 2011. ISBN 9781881883159.

[30] P. C. Hughes. *Spacecraft Attitude Dynamics*. Dover Publications, 2004. ISBN 0486439259.

[31] D. A. Vallado. *Fundamentals of Astrodynamics and Applications*. Microcosm Press, 4 edition, 2013.

[32] N. C. f. E. I. NOAA. The World Magnetic Model, 2019. URL `https://www.ngdc.noaa.gov/geomag/WMM/back.shtml`.

[33] D. Mission Design. Small Spacecraft Technology State of the Art. Technical report, NASA, Moffett Field, California, 2015.

[34] M. D. Shuster. Kalman filtering of spacecraft attitude and the QUEST model. *Journal of the Astronautical Sciences*, 38:377—-393, 1990.

[35] Y. Cheng, J. L. Crassidis, and F. L. Markley. Attitude estimation for large field-of-view sensors. *The Journal of the Astronautical Sciences*, 54(3-4):433–448, dec 2006.

[36] P. S. Maybeck. *Stochastic Models, Estimation and Control*. Academic Press, London, 1979.

[37] M. S. Grewal and A. P. Andrews. *Kalman Filtering: Theory and Practice*, volume 5. John Wiley & Sons, Inc., New York, USA, jan 2002. ISBN 0471392545.

[38] J. L. Crassidis and J. L. Junkins. *Optimal Estimation of Dynamic Systems*, volume 20040427 of *Chapman & Hall/CRC Applied Mathematics & Nonlinear Science*. Chapman and Hall/CRC, apr 2004. ISBN 978-1-58488-391-3.

[39] J. W. Murrell. Precision Attitude Determination for Multimission Spacecraft. In *Guidance and Control Conference*, Reston, Virigina, aug 1978. American Institute of Aeronautics and Astronautics.

[40] P. Batista, C. Silvestre, and P. Oliveira. Globally exponentially stable cascade observers for attitude estimation. *Control Engineering Practice*, 20(2):148–155, feb 2012.

[41] P. Batista, C. Silvestre, and P. Oliveira. Sensor-based complementary globally asymptotically stable filters for attitude estimation. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 7563–7568. IEEE, dec 2009.

[42] P. T. Martins Batista. *Sensor-based Navigation and Control of Autonomous Vehicles*. PhD thesis, Instituto Superior Técnico, Universidade de Lisboa, 2010.

[43] G. H. Golub and C. F. V. Loan. *Matrix Computations*. The Johns Hopkins University Press, 4 edition, 2013.

[44] J. H. F. Freitas. *Model-in-the-Loop Simulation of the ECOSat-III Attitude Determination and Control System*. PhD thesis, Instituto Superior Técnico, Universidade de Lisboa, 2016.

[45] F. L. Markley, J. Crassidis, and Y. Cheng. Nonlinear Attitude Filtering Methods. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Reston, Virigina, aug 2005. American Institute of Aeronautics and Astronautics.

Appendix A

Review of Notation

This appendix is intended to familiarize the reader with the mathematical notation and conventions adopted in this work. A review of some mathematical concepts are presented, as adequate.

A.1 Orthonormal Bases, Change of Basis

Let us consider two orthonormal sets of basis vectors, defined by $\{e_1, e_2, ..., e_n\}$ and $\{e'_1, e'_2, ..., e'_n\}$. The abstract generic n-dimensional vector **p** can be expressed in either basis:

$$\mathbf{p} = \sum_{j=1}^{n} p_j \mathbf{e}_j = \sum_{k=1}^{n} p'_k \mathbf{e}'_j \tag{A.1}$$

Taking the dot product of this equation with $\mathbf{e}_j$ or $\mathbf{e}'_k$ gives:

$$p_j = \mathbf{e}_j \cdot \mathbf{p} = \sum_{k=1}^{n} (\mathbf{e}_j \cdot \mathbf{e}'_k) p'_k \tag{A.2}$$

$$p'_k = \mathbf{e}'_k \cdot \mathbf{p} = \sum_{j=1}^{n} (\mathbf{e}'_k \cdot \mathbf{e}_j) p_j \tag{A.3}$$

Equations A.2 and A.3 can be used to express the relations between $\mathbf{p}_e$ and $\mathbf{p}_{e'}$, respectively **p** representation in each of the coordinate frames e and e', as matrix products:

$$\mathbf{p}_e = \mathbf{D}_{ee'} \mathbf{p}'_e \tag{A.4}$$

$$\mathbf{p}_{e'} = \mathbf{D}_{e'e} \mathbf{p}_e \tag{A.5}$$

where:

$$\mathbf{D}_{ee'} = \begin{bmatrix} \mathbf{e}_1 \cdot \mathbf{e}'_1 & \mathbf{e}_1 \cdot \mathbf{e}'_2 & \dots & \mathbf{e}_1 \cdot \mathbf{e}'_n \\ \mathbf{e}_2 \cdot \mathbf{e}'_1 & \mathbf{e}_2 \cdot \mathbf{e}'_2 & \dots & \mathbf{e}_2 \cdot \mathbf{e}'_n \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{e}_n \cdot \mathbf{e}'_1 & \mathbf{e}_n \cdot \mathbf{e}'_2 & \dots & \mathbf{e}_n \cdot \mathbf{e}'_n \end{bmatrix} \tag{A.6}$$

$$\mathbf{D}_{e'e} = \begin{bmatrix} \mathbf{e}'_1 \cdot \mathbf{e}_1 & \mathbf{e}'_1 \cdot \mathbf{e}_2 & \dots & \mathbf{e}'_1 \cdot \mathbf{e}_n \\ \mathbf{e}'_2 \cdot \mathbf{e}_1 & \mathbf{e}'_2 \cdot \mathbf{e}_2 & \dots & \mathbf{e}'_2 \cdot \mathbf{e}_n \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{e}'_n \cdot \mathbf{e}_1 & \mathbf{e}'_n \cdot \mathbf{e}_2 & \dots & \mathbf{e}'_n \cdot \mathbf{e}_n \end{bmatrix} \tag{A.7}$$

The matrices $\mathbf{D}_{ee'}$ and $\mathbf{D}_{e'e}$ are known as direction cosine matrices (DCMs) because their elements are the cosines of the angles between the basis vectors in the two reference frames. It is also of note that:

$$\mathbf{D}_{ee'} = \mathbf{D}_{e'e}^T \tag{A.8}$$

The elements of the direction cosine matrix are not all independent. The relationships among elements can be summarized by the statement that the product of a DCM **A** and its transpose is the identity matrix, meaning that **A** is a real orthogonal matrix, and that its determinant equals $+1$:

$$\mathbf{A}\mathbf{A}^T = 1 \tag{A.9a}$$

$$det(\mathbf{A}) = 1 \tag{A.9b}$$

Thus **A** is a proper real orthogonal matrix. A proper real orthogonal matrix transformation preserves the length of vectors and the angles between them, and thus represents a rotation. The group of matrices that have these properties constitute the Special Orthogonal *group* SO(n).

A.2 Matrix Representation of Vectors

The cross product is written as:

$$\mathbf{u} \times \mathbf{v} = \left[\mathbf{u}\times\right] \mathbf{v} \tag{A.10}$$

with:

$$\left[\mathbf{u}\times\right] = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \tag{A.11}$$

$\left[\mathbf{u}\times\right]$ can also be represented in this work by the notation **S(u)**.

A.3 Quaternion Definition and Quaternion Operations

There are different conceptual versions for the quaternion. Here we adopt the convention from references [11, 45], which is conceptually different from that firstly introduced by Hamilton in 1844.

Let's consider a quaternion as a four-dimensional vector with some additional operations defined on it. A quaternion $\mathbf{q}$ has a three-vector part $\mathbf{q}_{1:3}$ and a scalar part q_4:

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_{1:3} \\ q_4 \end{bmatrix} \tag{A.12}$$

Two important different products of a pair of quaternions $\bar{\mathbf{q}}$ and $\mathbf{q}$ are defined as:

$$\bar{\mathbf{q}} \otimes \mathbf{q} = \begin{bmatrix} q_4\bar{\mathbf{q}}_{1:3} + \bar{q}_4\mathbf{q}_{1:3} - \bar{\mathbf{q}}_{1:3} \times \mathbf{q}_{1:3} \\ \bar{q}_4 q_4 - \bar{\mathbf{q}}_{1:3} \cdot \mathbf{q}_{1:3} \end{bmatrix} \tag{A.13a}$$

$$\bar{\mathbf{q}} \odot \mathbf{q} = \begin{bmatrix} q_4\bar{\mathbf{q}}_{1:3} + \bar{q}_4\mathbf{q}_{1:3} + \bar{\mathbf{q}}_{1:3} \times \mathbf{q}_{1:3} \\ \bar{q}_4 q_4 - \bar{\mathbf{q}}_{1:3} \cdot \mathbf{q}_{1:3} \end{bmatrix} \tag{A.13b}$$

which can be represented by matrix multiplication by:

$$\mathbf{q} \otimes \bar{\mathbf{q}} = \left[\mathbf{q}\otimes\right] \bar{\mathbf{q}} \tag{A.14a}$$

$$\mathbf{q} \odot \bar{\mathbf{q}} = \left[\mathbf{q}\odot\right] \bar{\mathbf{q}} \tag{A.14b}$$

where:

$$\left[\mathbf{q}\otimes\right] \equiv \begin{bmatrix} q_4 I_3 - \left[\mathbf{q}_{1:3}\times\right] & \mathbf{q}_{1:3} \\ -\mathbf{q}_{1:3}^T & q_4 \end{bmatrix} = \begin{bmatrix} \Psi(\mathbf{q}) & \mathbf{q} \end{bmatrix} \tag{A.15a}$$

$$\left[\mathbf{q}\odot\right] \equiv \begin{bmatrix} q_4 I_3 + \left[\mathbf{q}_{1:3}\times\right] & \mathbf{q}_{1:3} \\ -\mathbf{q}_{1:3}^T & q_4 \end{bmatrix} = \begin{bmatrix} \Xi(\mathbf{q}) & \mathbf{q} \end{bmatrix} \tag{A.15b}$$

Quaternion multiplication is associative, distributive, but not commutative in general.

The identity quaternion is defined as:

$$\mathbf{I}_q \equiv \begin{bmatrix} \mathbf{0}_3 \\ 1 \end{bmatrix} \tag{A.16}$$

obeying all the properties commonly know of the identity.

The conjugate $\mathbf{q}^*$ of a quaternion is:

$$\mathbf{q}^* = \begin{bmatrix} \mathbf{q}_{1:3} \\ q_4 \end{bmatrix}^* \equiv \begin{bmatrix} -\mathbf{q}_{1:3} \\ q_4 \end{bmatrix} \tag{A.17}$$

and the inverse of a quaternion having non-zero norm is:

$$\mathbf{q}^{-1} \equiv \mathbf{q}^*/||\mathbf{q}||^2 \tag{A.18}$$

obeying all the properties required by the definition of an inverse.

A rotation of a three-component vector **p** is implemented by the quaternion product:

$$\mathbf{q} \otimes \mathbf{p} \otimes \mathbf{q}^* = \left[\mathbf{q}\odot\right]^T \left[\mathbf{q}\otimes\right] \begin{bmatrix} \mathbf{p} \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Xi}^T(\mathbf{q})\boldsymbol{\Psi}(\mathbf{q})\mathbf{p} \\ 0 \end{bmatrix} \tag{A.19}$$

The rule for performing successive transformations using quaternions may be derived from equation A.19. Applying a transformation by a second quaternion $\bar{\mathbf{q}}$ may be written as:

$$(\bar{\mathbf{q}} \otimes \mathbf{q}) \otimes \mathbf{p} \otimes (\bar{\mathbf{q}} \otimes \mathbf{q})^* = \begin{bmatrix} A(\bar{\mathbf{q}} \otimes \mathbf{q})\mathbf{p} \\ 0 \end{bmatrix} \tag{A.20}$$

with:

$$A(\bar{\mathbf{q}} \otimes \mathbf{q}) = A(\bar{\mathbf{q}})A(\mathbf{q}) \tag{A.21}$$