TÉCNICO LISBOA

# Attitude Determination and Control System of the ORCASat

**Bernardo Lúcio de Melo Sabino**

Thesis to obtain the Master of Science Degree in

## Aerospace Engineering

Supervisor(s): Prof. Afzal Suleman

### Examination Committee

Chairperson: Prof. Fernando José Parracho Lau
Supervisor: Prof. Afzal Suleman
Member of the Committee: Prof. Alexandra Bento Moutinho

**September 2020**

To you reader, I hope you can find the answer to your problem in this thesis.

# Acknowledgments

I would like to start to thanking my thesis supervisor, Professor Afzal Suleman, for the amazing opportunity to work on my thesis abroad in the Centre for Aerospace Research of University of Victoria. Thank you so much;

To all my colleagues from the Optical and Radio Calibration Satellite (ORCASat) project, specially Bernardo Lobo-Fernandes;

I would like to thank all of my friends that went with me to Canada and with whom I experienced many adventures;

To all the people from the office with whom I had a great time;

A special thanks to my friend and one of the most intelligent and interesting people I've ever met, Mario who taught me a lot and with whom I spent long nights in the office, a great time in Fels and in downtown Victoria;

To my very close friends from Portugal who help me through difficult times and whose contact was incredibly valuable; Hopefully we will have dinner together one of these days!

To Kim, one of the most amazing people I've ever met and with whom I had a great number of adventures and hopefully more will come in the near future.
Danke schön for all of your support and love!

A special thanks to my parents, brother and Nagum who had the difficult task to deal with me during all this time I've been at home writing my thesis.

Finally, I would also like to leave a message to my future self to keep pushing and never give up and if I eventually read this acknowledgments page again to take a minute of silence to remember the past and to think about the future.

Não posso deixar de agradecer, em especial, e em Português, novamente à minha família e à Kim pelo apoio que me deram, e aos meus amigos, Marco, Joaquim e Ramalho, pela ajuda que me deram na reta final ao reverem a tese comigo e pelo seu contributo na minha apresentação em PowerPoint.

Deixo aqui também os parabéns ao meu primo João que será pai muito brevemente e à Diva, e que a Aurora venha a ser engenheira!

# Resumo

O objetivo desta tese é projetar um Sistema de Determinação e Controle de Atitude (ADCS) capaz de atender aos requisitos de missão do ORCASat, um CubeSat 2U: Capacidade de desacelerar o satélite de velocidades angulares iniciais elevadas; Garantir um erro entre a atitude real e a desejada inferior a $10^\circ$ ; E garantir um erro de estimação de atitude inferior a $2^\circ$ .

O controlo ativo do ADCS é feito através de atuadores eletromagnéticos, também denominados *magnetorquers*, enquanto um volante de inércia, também conhecido como *momentum wheel*, assegura a estabilização passiva da atitude. Quatro algoritmos diferentes de controlo de atitude - *Constant Gain Controller* (CGC), *Finite Horizon Controller* (FHC), *Infinite Horizon Controller* (IHC) e *Sliding Mode Controller* (SMC) - foram testados e comparados em termos de eficiência e precisão. O algoritmo com melhor desempenho foi depois selecionado e analisado em diferentes cenários, incluindo incertezas de modelo como a incerteza da matriz de inércia e a degradação do desempenho do volante de inércia. Por forma a desacelerar o satélite, uma versão melhorada do controlador B-dot foi implementada e o seu desempenho foi testado em diferentes condições.

O satélite tem como sensores de atitude quatro sensores solares, um magnetómetro e um giroscópio. O *Quaternion Estimator* (QUEST) foi implementado por forma a inicializar o principal estimador de atitude, o *Multiplicative Extended Kalman Filter* (MEKF). Um filtro para a calibração em tempo real do magnetómetro, o *Magnetometer Calibration Extended Kalman Filter* (MCEKF), foi estudado e implementado como uma tentativa de melhorar o desempenho do MEKF. Diferentes casos foram criados de forma a poder analisar o desempenho destes algoritmos.

Todas as simulações foram realizadas num ambiente realista desenvolvido em Matlab/Simulink. Os resultados das simulações realizadas mostraram que o ADCS proposto é capaz de satisfazer com êxito os requisitos de missão estabelecidos, mesmo com a existência de incertezas de modelo.

**Palavras-chave:** ADCS, CubeSat, estimação de atitude, controlo de atitude, calibração de magnetómetro, atuador eletromagnético

# Abstract

The goal of this thesis is to design an Attitude Determination and Control System (ADCS) capable of achieving the mission requirements of the ORCASat, a 2U CubeSat: Ability to detumble the satellite from high initial angular velocities; Guarantee a pointing error smaller than $10^\circ$ ; And guarantee an attitude estimation error smaller than $2^\circ$ .

The active ADCS control is provided by magnetorquers while a momentum wheel is used for passive attitude stabilization. Four different attitude control algorithms – Constant Gain Controller (CGC), Finite Horizon Controller (FHC), Infinite Horizon Controller (IHC), and Sliding Mode Controller (SMC) - were tested and compared in terms of efficiency and pointing accuracy. The best performing algorithm was selected, and its performance was analyzed under different scenarios including model uncertainties such as the inertia matrix uncertainty and the momentum wheel performance degradation. In order to detumble the satellite, a modified B-dot controller was implemented, and its performance was tested under different conditions.

The ORCASat's attitude sensor suite includes four sun sensors, a magnetometer, and a gyroscope. The Quaternion Estimator (QUEST) algorithm was implemented to initialize the main attitude estimator, the Multiplicative Extended Kalman Filter (MEKF). A real-time magnetometer calibration filter, the Magnetometer Calibration Extended Kalman Filter (MCEKF), was implemented and studied as an attempt to improve the performance of the MEKF. Different cases were devised to analyze the performance of these algorithms.

All the simulations were performed under a realistic Matlab/Simulink environment. The different simulations showed that the proposed ADCS could fulfill the mission requirements, even with the existence of model uncertainties.



**Keywords:** ADCS, CubeSat, attitude determination, attitude control, magnetometer calibration, magnetorquer

# Contents

# List of Tables

# List of Figures

# List of Symbols

The next list describes several symbols that will be later used within the body of the document. Vectors and matrices are written in a boldface type. Matrices, however, are displayed in uppercase, while vectors are usually displayed in lowercase. Vectors with dimensions $3 \times 1$ have an arrow sign on top. Matrices and vectors related to the control algorithms are written in italic type, while matrices and vectors related to attitude estimation algorithms are written in upright type. Scalars are shown in a lightface type and can be either lowercase or uppercase. Roman and Greek symbols bear no special distinction.

**Reference frames**

$B = \{\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \hat{\mathbf{b}}_3\}$ Spacecraft body frame.

$E = \{\hat{\boldsymbol{\epsilon}}_1, \hat{\boldsymbol{\epsilon}}_2, \hat{\boldsymbol{\epsilon}}_3\}$ Earth-Centered/Earth-Fixed reference frame.

$I = \{\hat{\mathbf{i}}_1, \hat{\mathbf{i}}_2, \hat{\mathbf{i}}_3\}$ Earth-Centered Inertial reference frame.

$O = \{\hat{\mathbf{o}}_1, \hat{\mathbf{o}}_2, \hat{\mathbf{o}}_3\}$ Local-Vertical/Local-Horizontal reference frame.

**Functions and Operators**

$\mathrm{adj}(\square)$ Adjoint matrix operator.

blkdiag$(\square)$ Block diagonal matrix.

$[\square\times]$ Cross product matrix operator.

$J(\square)$ Cost function.

$\square^*$ Conjugate.

$\delta(t-\tau)$ Dirac delta function.

$\delta_{kj}$ Kronecker delta function.

$\det(\square)$ Determinant.

$\mathrm{diag}(\square)$ Diagonal matrix.

$\dot{\square}$ Derivative with respect to time.

$\ddot{\square}$ Second derivative with respect to time.

$\hat{\square}$ Estimated value.

$E\{\square\}$ Expected value.

$\boldsymbol{\nabla}\square$ Gradient operator.

$H(\square)$ Heaviside function.

$\mathrm{INT}(\square)$ Integer operator.

$\gg$ Much greater than.

$\mathrm{mod}\{\square,\square\}$ Modulo operator.

$\mathcal{N}(\mu,\mathbf{Q})$ Normal distribution with mean $\mu$ and covariance $\mathbf{Q}$.

$\|\square\|$ Norm operator.

$\odot$ Quaternion product.

$\otimes$ Quaternion product.

$\mathrm{sign}(\square)$ Sign function.

$\mathrm{tr}(\square)$ Trace.

**Greek symbols**

$\Delta\vec{\boldsymbol{\beta}}$ Gyroscope bias error vector.

$\vec{\boldsymbol{\beta}}$ Gyroscope bias vector.

$\Delta\vec{\boldsymbol{\omega}}$ Error angular velocity vector.

$\boldsymbol{\epsilon}$ State error vector.

$\lambda$ Sliding condition gain of the Sliding Mode Controller.

$\mu_{\oplus}$ Earth's standard gravitational parameter.

$\bar{\omega}$ Measured angular velocity.

$\delta\vec{\boldsymbol{\omega}}$ Small perturbation of the angular velocity $\vec{\boldsymbol{\omega}}_B^{B/O}$ from the reference angular velocity $\vec{\boldsymbol{\omega}}_{B_{ref}}^{B/O}$.

$\omega_0$ Orbital rate.

$\vec{\boldsymbol{\omega}}$ Spacecraft inertial angular velocity - shorthand for $\vec{\boldsymbol{\omega}}^{B/I}$.

$\vec{\boldsymbol{\omega}}_B^s$ Angular velocity of the momentum wheel about its spin axis with respect to the spacecraft's body.

$\vec{\boldsymbol{\omega}}^{B/I}$ Angular velocity of frame $B$ with respect to frame $I$.

$\vec{\boldsymbol{\omega}}^{B/O}$ Angular velocity of frame $B$ with respect to frame $O$.

$\vec{\boldsymbol{\omega}}^{i/B}$ Angular velocity of the $i$'th rigid body relatively to the body frame.

$\vec{\boldsymbol{\omega}}^{i/I}$ Inertial angular velocity of the $i$'th rigid body.

$\vec{\boldsymbol{\omega}}^{O/I}$ Angular velocity of frame $O$ with respect to frame $I$.

$\vec{\boldsymbol{\omega}}_0$ Initial angular velocity.

$\vec{\boldsymbol{\omega}}_e$ Error angular velocity - short for $\vec{\boldsymbol{\omega}}_O^{B/O}$.

$\vec{\boldsymbol{\omega}}_\oplus$ Earth's angular velocity vector in frame $I$ coordinates.

$\boldsymbol{\Phi}$ Error-state transition matrix.

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

$\rho$ Atmospheric density.

$\Sigma$ Covariance matrix of the magnetometer measurement noise.

$\sigma$ Standard Deviation.

$\sigma_{abs}^i$ Absorption coefficient of the $i$'th panel.

$\sigma_{diff}^i$ Diffuse reflection coefficient of the $i$'th panel.

$\sigma_{spec}^i$ Specular reflection coefficient of the $i$'th panel.

$\tau$ Shorthand for $\vec{\boldsymbol{\tau}}^c$.

$\vec{\boldsymbol{\tau}}_{eq}^{\parallel}$ Component of $\vec{\boldsymbol{\tau}}_{eq}$ parallel to $\vec{\mathbf{s}}$.

$\vec{\boldsymbol{\tau}}_{eq}^{\perp}$ Component of $\vec{\boldsymbol{\tau}}_{eq}$ perpendicular to $\vec{\mathbf{s}}$.

$\vec{\boldsymbol{\tau}}^c$ Torque applied to the system about its center of mass $c$.

$\vec{\boldsymbol{\tau}}^{perturb}$ Net torque due to environmental perturbations about the spacecraft's center of mass.

$\vec{\boldsymbol{\tau}}_{des}$ Desired control torque.

$\vec{\boldsymbol{\tau}}_{eq}$ Equivalent control torque.

$\delta\vec{\boldsymbol{\theta}}$ Attitude quaternion small angle parametrization vector.

$\theta$ Euler rotation angle.

$\theta_{GMST}$ Greenwich Mean Sidereal Time angle.

$\xi$ Inclination of the spacecraft orbit relative to the geomagnetic equatorial plane.

**Roman symbols**

$\vec{\mathbf{0}}$ $3 \times 1$ null vector.

$\mathbf{0}_n$ $n \times 1$ null vector.

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

$\delta\mathbf{A}$ Small rotation of the attitude matrix $\mathbf{A}_O^B$ from the reference attitude matrix $\mathbf{A}_{O_{ref}}^B$.

$\vec{\mathbf{a}}$ Total acceleration applied to the spacecraft's center of mass.

$\vec{\mathbf{a}}^{perturb}$ Total acceleration due to environmental perturbations.

$\boldsymbol{A}$ State matrix.

$\mathbf{A}_I^B$ DCM or attitude matrix representing the rotation of frame $B$ with respect to frame $I$.

$\mathbf{A}_I^E$ DCM or attitude matrix representing the rotation of frame $E$ with respect to frame $I$.

$\mathbf{A}_R^M$ DCM or attitude matrix representing the rotation of the magnetometer frame $M$ with respect to a reference frame $R$.

$a$ Magnetic spherical reference radius.

$A_i$ Area of the $i$'th panel of the spacecraft.

$\bar{\boldsymbol{B}}$ Control matrix $\hat{\boldsymbol{B}}$ averaged within one orbit.

$\vec{\mathbf{B}}$ Geomagnetic field vector.

$\vec{\mathbf{b}}$ Magnetometer bias vector.

$\vec{\mathbf{B}}_M$ Magnetic field measured by the magnetometer.

$\boldsymbol{B}$ Control matrix.

$\hat{\boldsymbol{B}}$ Periodic control matrix.

$C_D$ Drag coefficient

$\mathbf{D}$ Fully populated symmetric scale factors and non-orthogonality corrections matrix.

$\vec{\mathbf{e}}$ Euler rotation axis.

$\vec{\mathbf{F}}^i$ Force acting on the $i$'th plate of the spacecraft's body.

$\vec{\mathbf{F}}_{grav}$ Gravity force.

$\mathbf{F}$ Dynamics matrix.

$\mathbf{G}$ Process noise distribution matrix.

$G$ Universal gravitational constant.

$\vec{\mathbf{h}}$ Angular momentum vector.

$\vec{\mathbf{h}}^c$ Angular momentum of the system about its center of mass.

$\vec{\mathbf{h}}^s$ Angular momentum vector contribution of the momentum wheel about its spin axis due to its angular velocity relatively to the body frame.

$\vec{\mathbf{h}}^{i/c_i}$ Angular momentum of the $i$'th rigid body about its own center of mass $c_i$.

$\vec{\mathbf{h}}^{i/c}$ Angular momentum of the $i$'th rigid body about the system's center of mass $c$.

$\mathbf{H}$ Measurement matrix.

$\mathbf{I}_n$ $n \times n$ Identity Matrix.

$\mathbf{I}_q$ Identity quaternion.

$\mathbf{J}$ Shorthand for $\mathbf{J}^c$.

$\mathbf{J}^c$ Inertia tensor of the complete system about its center of mass $c$.

$\mathbf{J}^w$ Inertia tensor of the momentum wheel about its center of mass.

$\mathbf{J}^{i/c_i}$ Inertia matrix of the $i$'th rigid body about its own center of mass.

$J^s$ Principal moment of inertia of the momentum wheel about its spin axis.

$J^\perp$ Principal moment of inertia of the momentum wheel about any axis perpendicular to the wheel's spin axis.

$\boldsymbol{K}$ Controller gain matrix.

$\mathbf{K}$ Kalman gain matrix.

$K$ Scalar gain.

$\vec{\mathbf{m}}$ Dipole moment vector.

$\vec{\mathbf{m}}^{\parallel}$ Component of $\vec{\mathbf{m}}$ parallel to the local geomagnetic field $\vec{\mathbf{B}}$.

$\vec{\mathbf{m}}^{\perp}$ Component of $\vec{\mathbf{m}}$ perpendicular to the local geomagnetic field $\vec{\mathbf{B}}$.

$\vec{\mathbf{m}}_{cmd}$ Controller commanded dipole moment.

$\vec{\mathbf{m}}_{ctrl}$ Magnetorquers generated magnetic dipole moment.

$m$ Mass of the spacecraft.

$M_\oplus$ Earth's mass.

$\hat{\mathbf{n}}_i$ Outward normal unit vector of the $i$'th panel of the spacecraft.

$\mathbf{P}$ Covariance matrix of the state error vector.

$p_\odot$ Solar radiation pressure.

$\boldsymbol{P}$ Riccati matrix.

$\boldsymbol{P}_\infty$ Periodic Riccati matrix.

$\boldsymbol{P}_f$ Riccati final condition matrix.

$\hat{\boldsymbol{P}}$ Periodic extension of the Riccati matrix.

$\delta q_4$ Scalar component of $\delta\mathbf{q}$

$\delta\vec{\boldsymbol{q}}$ Vector part of $\delta\boldsymbol{q}$.

$\delta\boldsymbol{q}$ Small rotation of the attitude quaternion $\mathbf{q}_O^B$ from the reference attitude quaternion $\mathbf{q}_{O_{ref}}^B$.

$\delta\vec{\mathbf{q}}$ Vector component of $\delta\mathbf{q}$.

$\delta\mathbf{q}$ Multiplicative error quaternion.

$\vec{\mathbf{q}}$ Vector part of $\mathbf{q}$.

$\boldsymbol{Q}$ Error weighted matrix.

$\mathbf{Q}$ Process noise covariance matrix.

$\mathbf{q}$ Attitude quaternion.

$\mathbf{q}_I^B$ Attitude quaternion defining the rotation of frame $B$ with respect to frame $I$.

$\mathbf{q}_I^O$ Attitude quaternion defining the rotation of frame $O$ with respect to frame $I$.

$\mathbf{q}_O^B$ Attitude quaternion defining the rotation of frame $B$ with respect to frame $O$.

$\mathbf{q}_e$ Attitude quaternion error - shorthand for $\mathrm{q}_O^B$.

$q_4$ Scalar component of $\mathbf{q}$.

$\vec{\mathbf{r}}$ Position of the spacecraft with respect to the Earth's center of mass.

$\vec{\mathbf{r}}^i$ Position vector of the center of pressure of the $i$'th panel of the spacecraft with respect to the spacecraft's center of mass.

$\vec{\mathbf{r}}^{c_i/c}$ Position of the $i$'th rigid body center of mass relative to the system's center of mass.

$\mathbf{R}$ Measurement noise covariance matrix.

$R_\oplus$ Earth's mean equatorial radius.

$\mathcal{S}$ Sliding manifold.

$\vec{\mathbf{s}}$ Sliding variable.

$\hat{\mathbf{s}}_\odot$ Unit vector pointing from the spacecraft to the Sun.

$S_i$ Projected area of the $i$'th panel of the spacecraft in the direction of $\vec{\mathbf{v}}_{rel}$.

$\Delta t$ Sampling time interval.

$t$ Time.

$T_{orb}$ Orbital period.

$\vec{u}$ Control vector.

$\vec{\mathbf{u}}$ Shorthand for $\vec{\mathbf{u}}_{ctrl}$.

$\vec{\mathbf{u}}_{ctrl}$ Control torque written in frame $B$ coordinates.

$U$ Gravitational potential function.

$\mathcal{V}$ Lyapunov candidate function.

$\vec{\mathbf{v}}$ Shorthand for $\vec{\mathbf{v}}_{spacecraft}$.

$\vec{\mathbf{v}}^{c_i/c}$ Inertial velocity of the $i$'th rigid body center of mass relative to the system's center of mass.

$\vec{\mathbf{v}}_{atmosphere}$ Velocity of the Earth's atmosphere.

$\vec{\mathbf{v}}_{rel}$ Velocity of the spacecraft with respect to the Earth's atmosphere.

$\vec{\mathbf{v}}_{spacecraft}$ Velocity of the spacecraft with respect to the Earth's center of mass.

$\mathbf{v}$ Measurement noise column vector.

$V$ Magnetic field potential function.

$v$ Measurement noise.

$\mathbf{w}$ Process noise column vector.

$\hat{\mathbf{w}}^s$ Spin axis of the momentum wheel.

$\hat{\mathbf{w}}^{\perp_1}$ Momentum wheel axis perpendicular to the spin axis and to $\hat{\mathbf{w}}^{\perp_2}$.

$\hat{\mathbf{w}}^{\perp_2}$ Momentum wheel axis perpendicular to the spin axis and to $\hat{\mathbf{w}}^{\perp_1}$.

$\Delta\mathbf{x}$ State vector correction.

$\mathbf{x}$ State vector.

$\mathbf{x}_e$ Equilibrium point.

$\mathbf{y}$ Measurement vector.

$\vec{\mathbf{z}}_I$ Measurement unit vector written in frame $I$ coordinates given by a measurement model.

**Subscripts**

$aero$ Aerodynamic.

$i$ Computational index.

$k$ Denotes the $k$'th time-step.

| | |
|---|---|
| $gg$ | Gravity gradient. |
| $0$ | Initial. |
| $mag$ | Magnetic. |
| ref | Reference condition. |
| $srp$ | Solar radiation pressure. |
| $B$ | Variable written in frame $B$ coordinates. |
| $I$ | Variable written in frame $I$ coordinates. |
| $O$ | Variable written in frame $O$ coordinates. |

**Superscripts**

| | |
|---|---|
| $+$ | Updated, post-update. |
| $-$ | Propagated, pre-update. |
| $\top$ | Transpose. |

# Acronyms

**ADC** Analog-to-Digital Converter.

**ADCS** Attitude Determination and Control System.

**ARE** Algebraic Riccati Equation.

**ARW** Angular Random Walk.

**CAD** Computer-Aided Design.

**Cal Poly** California Polytechnic State University.

**CCP** Canadian CubeSat Project.

**CfAR** Centre for Aerospace Research.

**CGC** Constant Gain Controller.

**CHIME** Canadian Hydrogen Intensity Mapping Experiment.

**CMOS** Complementary Metal–Oxide–Semiconductor.

**COTS** Commercial Off-The-Shelf.

**CSA** Canadian Space Agency.

**DCM** Direction Cosine Matrix.

**DRE** Differential Riccati Equation.

**ECEF** Earth-Centered/Earth-Fixed.

**ECI** Earth-Centered Inertial.

**EGM2008** Earth Gravitational Model 2008.

**EKF** Extended Kalman Filter.

**FHC** Finite Horizon Controller.

**GMST** Greenwich Mean Sidereal Time.

**GNSS** Global Navigation Satellite System.

**HIL** Hardware-In-the-Loop.

**IHC** Infinite Horizon Controller.

**IMU** Inertial Measurement Unit.

**ISS** International Space Station.

**JPL** Jet Propulsion Laboratory.

**LEO** Low Earth Orbit.

**LQR** Linear Quadratic Regulator.

**LVLH** Local-Vertical/Local-Horizontal.

**MCEKF** Magnetometer Calibration Extended Kalman Filter.

**MEKF** Multiplicative Extended Kalman Filter.

**MIL** Model-In-the-loop.

**MSR** Method of Sequential Rotations.

**NRLMSISE** US Naval Research Laboratory Mass Spectrometer and Incoherent Scatter Radar Exosphere.

**OBC** On-Board Computer.

**ORCASat** Optical and Radio Calibration Satellite.

**PD** Proportional-Derivative.

**QUEST** Quaternion Estimator.

**RMS** Root Mean Square.

**RRW** Rate Random Walk.

**SD** Standard Deviation.

**SIL** Software-In-the-Loop.

**SMC** Sliding Mode Controller.

**SMS** Spacecraft Mechanics Simulator.

**SRP** Solar Radiation Pressure.

**SSDL** Stanford University's Space Systems Development Laboratory.

**SVD** Singular Value Decomposition.

**TAM** Three-Axis Magnetometer.

**TRIAD** Triaxial Attitude Determination.

**UTC** Coordinated Universal Time.

**UT1** Universal Time 1.

**UVic** University of Victoria.

**WMM** World Magnetic Model.

# Chapter 1

# Introduction

Satellites are usually divided into classes by their weight. Figure 1.1 shows the typical classification of the different satellite classes in an intuitive way.


Figure 1.1: Satellite classification according to their weight [1].

A particular class of satellites, the CubeSat has been growing continuously over the years. The global CubeSat market is projected to grow from USD 152 million in 2018 to USD 375 million by 2023 [2] and more than 1200 CubeSats have been launched as of January 2020 [3].

A CubeSat is a square-shaped miniature satellite built to standard dimensions (Units or "U") of $10\text{cm} \times 10\text{cm} \times 10\text{cm}$, with a mass of up to $1.33\text{kg}$ per U [1]. A CubeSat can be used alone (1U) or in groups of multiple units, originating 2U, 3U, 6U, up to a maximum 24U. The CubeSat Project[1] began in 1999 as a collaborative effort between Jordi Puig-Suari, a professor at California Polytechnic State

[1] http://www.cubesat.org/

University (Cal Poly), and Bob Twiggs, a professor at Stanford University's Space Systems Development Laboratory (SSDL), to provide a standard for designing satellites in order to reduce cost and development time, increase accessibility to space, and to provide the ability to sustain frequent launches [4].

Although CubeSats were initially developed as an educational tool, they are increasingly being put to active use in orbit for technology demonstration, scientific studies, and even commercial purposes. Figure 1.2 provides a synthesis of what is a CubeSat, its advantages and missions.

The small size of these satellites presents some challenges for the design of an Attitude Determination and Control System (ADCS) since these satellites are very limited not only in terms of mass and volume but they must also obey strict power and cost limitations. The first CubeSats had a relatively simple ADCS making use of magnetometers, photodiodes, and magnetorquers or gravity gradient booms as basic attitude sensors and actuators. With the development of CMOS sun sensors and miniaturized star trackers and reaction wheels, more complex ADCS started to emerge enabling a more variety of CubeSats missions which require for instance accurate attitude knowledge and full 3-axis control [5].

This thesis addresses some of the challenges faced while designing the ADCS of a CubeSat and provides and evaluates the efficiency of a fully conceptualized ADCS under the framework of the Optical and Radio Calibration Satellite (ORCASat) project developed at the Centre for Aerospace Research (CfAR) at University of Victoria (UVic). The ADCS of the ORCASat features a sensor suite composed of a three-axis digital magnetometer and gyroscope Inertial Measurement Unit (IMU), four digital sun sensors, and a Global Navigation Satellite System (GNSS) receiver. The active ADCS control is provided by three orthogonal magnetorquers while a momentum wheel is used for passive attitude stabilization.

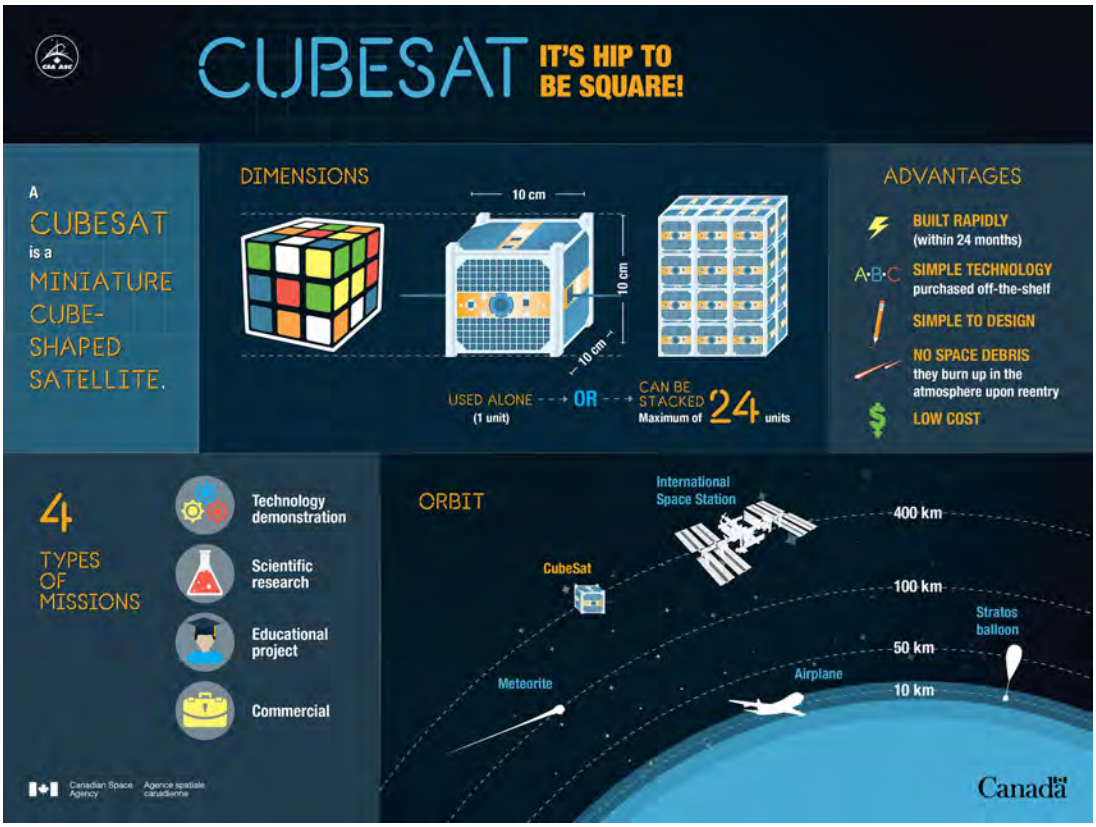



Figure 1.2: CubeSat info-graphic [1].

## 1.1 Motivation

The ORCASat project[2] is a multi-payload, collaborative project between teams at the University of British Columbia, Simon Fraser University, the University of Victoria (British Columbia, Canada), and Instituto Superior Técnico (Lisbon, Portugal) [6]. This project is part of the Canadian CubeSat Project (CCP)[3] which offers post-secondary institutions (colleges and universities) from each province and territory in Canada the opportunity for their students to take part in a real space mission by designing, building, launching, and operating their own CubeSat. This is thought to help to increase students' interest in science, technology, engineering and math as well as develop students' expertise in space domains and advance Canadian space science and technology. This project will be guided by Canadian Space Agency (CSA) experts and representatives from the Canadian space industry which will help the teams to optimize the success of each mission.

The ORCASat is scheduled for launch from the International Space Station (ISS) in the final quarter of 2021. The purpose of the ORCASat will be to provide a reference light source from orbit to calibrate ground-based optical observatories that are used to study the expansion rate of the universe and the effects of dark energy. Once in orbit, ORCASat will flash a laser light source while traveling across the sky above the observatory. On-board sensors will measure and record the actual output of the light source while at the same time an observatory on the ground will take its own readings. The values recorded on-board the satellite will then be transmitted to the observatory where astronomers will compare the actual and perceived emission helping to calibrate the observatories [7]. Additionally, its radio source will be used to calibrate the Canadian Hydrogen Intensity Mapping Experiment (CHIME) radio observatory conceived to form a 3-dimensional map of hydrogen density, which will be used to measure the expansion history of the universe [8]. The CubeSat also provides a store and forward repeater, as service to the amateur radio community.

## 1.2 Topic Overview

The ADCS of a spacecraft is the subsystem responsible for determining the orientation of the spacecraft and then controlling it so that the spacecraft points in some desired direction. Authors such as F. Landis Markley, John L. Crassidis, James R. Wertz, and Bong Wie have published many articles exploring this subject which can be found compiled in references [9–12].

The first spacecrafts starting with the Sputnik in 1957 had no ADCS, others made use of passive stabilization methods, while infrequent control torques to be applied by active mechanisms were computed on the ground and telemetered to the spacecraft. Most modern-day missions and spacecrafts are however different, having specific pointing requirements [9].

The first 3-axis attitude determination system was invented by Harold Black for the Transit satellite system and was published in 1964 [13]. This algorithm made use of two and only two vector observations and came to be known as the Triaxial Attitude Determination (TRIAD) algorithm [14]. One year

[2] https://www.orcasat.ca/

[3] https://www.asc-csa.gc.ca/eng/satellites/cubesat/default.asp

later Grace Wahba published her famous attitude determination problem involving any number of vector observations [15]. The first successful application of Wahba's problem was the q-method algorithm devised by Paul Davenport as reported in 1977 in reference [16] by using the quaternion parametrization for the attitude. This algorithm still presented some challenges for the computing power of the computers at that time due to its necessity to solve an eigenvalue/eigenvector problem. In 1978 the Quaternion Estimator (QUEST) algorithm developed by Shuster [17] was presented as an alternative to the q-method which avoided the eigenvalue/eigenvector decomposition being less computationally expensive. Since then many other solutions for Wahba's problem have been developed. Reference [18] provides a summary of all those methods as well as comparing the performance between them.

The attitude determination methods mention above belong to the class of static attitude determination methods which combines measurements taken at the same time. A different class of attitude determination methods is the recursive attitude determination methods which make use of a model of the dynamics of the spacecraft. The latter methods have the advantage of providing more accurate solutions and also providing attitude estimates when less than two independent measurements are available. One of the most popular recursive algorithms for attitude estimation is the Extended Kalman Filter (EKF), being considered the workhorse of real-time spacecraft attitude estimation [19]. The first publication of the use of the EKF for attitude estimation was done in 1970 by James Farrell [20] where the Euler angles were used for the attitude parametrization. These filters, however, started only to be implemented in the late 1970s, due to their very high computational burden [21]. Reference [19] provides a broad view of current attitude estimation methods.

According to [19], most applications in spacecrafts use the EKF, specially in the form known as the Multiplicative Extended Kalman Filter (MEKF), as the method of choice for solving the attitude determination problem [22–29]. In some cases, this filter is coupled with a static attitude determination methods, such as the Singular Value Decomposition (SVD) algorithm or the QUEST algorithm, for sanity check and for providing an initial state estimation for the recursive estimation method [26–28].

Although the history of attitude determination has been relatively well documented, the history of attitude control is less known mainly because it was classified secret in the early days of spacecraft mission design [9]. Similarly to the estimation problem, the parametrization of choice for the control problem is the quaternion parametrization. One of the first publications using the quaternion as attitude parametrization was produced by Bong Wie and Peter Barba in [30] which essentially uses a classic Proportional-Derivative (PD) controller which is proved stable by using a Lyapunov analysis. Since the first publications, control theory has continued to evolve, producing a vast stream of theoretically-based publications that continues to this day. Reference [9] provides a good synopsis of the history of attitude control as well as many different control algorithms.

Currently, spacecraft attitude control methods can be divided into two classes, passive and active control methods. Passive control methods use the natural spacecraft dynamics to satisfy their pointing requirements, these include momentum bias techniques using spin stabilization or momentum wheels, gravity-gradient stabilization, and permanent magnets stabilization. On the other hand, active control schemes rely on actuators such as thrusters, magnetorquers, and reaction wheels, among others, to

accomplish the spacecraft's pointing requirements. Active control schemes allow for missions that have much more demanding pointing requirements including three-axis stabilization. Most applications in CubeSats use magnetorquers as main actuators due to its simplicity, low cost, and the fact that they can be simultaneously used for detumbling and 3-axis stabilization [31]. These actuators are often used together with passive stabilization methods such as the use of a momentum wheel or gravity gradient boom [27, 32–37]. The magnetic spacecraft control problem is significantly different from the conventional attitude regulation one since due to the magnetic field properties, it is not possible by means of magnetic actuators to provide three independent control torques at each time instant [38]. A large amount of literature has been developed solely to address this particular problem such as in references [39] and [40]. References [38, 41, 42] study the 3-axis attitude control using only magnetorquers. Examples of 3-axis attitude control using only magnetorquers can be seen in references [28, 35, 43–45]. Advances in miniaturization allowed CubeSats to be equipped with reaction wheels as primary actuators, while using magnetorquers for momentum damping, for better pointing accuracy [22, 25, 29, 46]. The miniaturization advances also allowed for more sophisticated sensors to be present in CubeSats, such as miniaturized star trackers [47] which reduce the estimation error and consequently improve the pointing performance.

## 1.3 Objectives

The objective of this thesis comes from the need to have a robust, simple, and cost-effective ADCS for the ORCASat. The three mission requirements for the ADCS of the ORCASat are:

- Ability to detumble the satellite;
- Guarantee an attitude estimation error smaller than $2^\circ$ ;
- Guarantee a pointing error smaller than $10^\circ$ .

These requirements must be guaranteed for the full orbit independently if the CubeSat is in eclipse or in sunlit. This thesis had contributions from previous works done at CfAR such as [48] and [49] but includes an improved attitude estimation algorithm and the study and testing of new and improved control algorithms. The goals for this master's thesis are summarized as follows:

- Present a functional ADCS prototype clearly separated from the Model-In-the-loop (MIL) framework from [48];
- Investigate and study the performance of different control algorithms;
- Investigate and study the performance of the estimation algorithm implemented in [48] and implement the necessary changes to meet the requirements;
- Define the initializing sequence of the ADCS and its operating modes;
- Improve the environment simulator using the current Computer-Aided Design (CAD) model;
- Demonstrate that the designed ADCS can meet the mission requirements.

## 1.4 Thesis Outline

This thesis is organized as follows:

- **Chapter 2** includes a summary of the required background concepts for the research presented in this document, such as different attitude parametrizations, the frames of reference that are used throughout the work as well as some concepts about the stability of nonlinear systems.
- **Chapter 3** describes the spacecraft mechanics, which includes the orbital motion of the spacecraft, the attitude dynamics and kinematics as well as the different perturbation forces and torques applied to the spacecraft. This chapter also describes the linearized model for the dynamics and kinematics of the spacecraft, which will be used for the synthesis of the Linear Quadratic Regulator (LQR) controllers.
- **Chapter 4** presents the theory behind the different attitude determination/estimation algorithms implemented in the ADCS as well as a real-time magnetometer calibration filter based on the Extended Kalman Filter.
- **Chapter 5** provides the theory behind the different magnetic control algorithms studied. This includes both the detumbling controller and the nadir-pointing controllers.
- **Chapter 6** provides an overview of the implementation and a description of the functioning of the simulator developed in Matlab/Simulink to test the different algorithms and to simulate the satellite, including the different environmental models used, the hardware models and the selected hardware for the ORCASat and the ADCS implementation.
- **Chapter 7** provides the simulation results. A comparison between the performance of the different nadir-pointing control algorithms studied is done as well as the selection of the nominal controller for the ORCASat. The validation of the performance of the selected nadir-pointing controller, of the detumbling controller and the different estimation filters under different scenarios is also presented.
- **Chapter 8** summarizes the work performed in this thesis with a conclusion and presents some recommendations for future work.

# Chapter 2

# Theoretical Background

This chapter will start, in Section 2.1, with the definition of the reference frames used throughout this work which are useful in the context of the subject of attitude determination and control, as well as the transformations used to switch between these reference frames. It then follows, in Section 2.2, presenting the different attitude parameterizations used, i.e., the different ways to mathematically represent the orientation of a body, and lastly, in Section 2.3, the concepts associated with the stability of nonlinear systems are described.

## 2.1 Reference Frames

A reference frame consists of a coordinate system that is specified by the location of its origin and the orientation of its coordinate axes [9]. In order to apply the different equations that describe the motion of the spacecraft in space, and to be able to determine its position and attitude, one needs to define a reference frame.

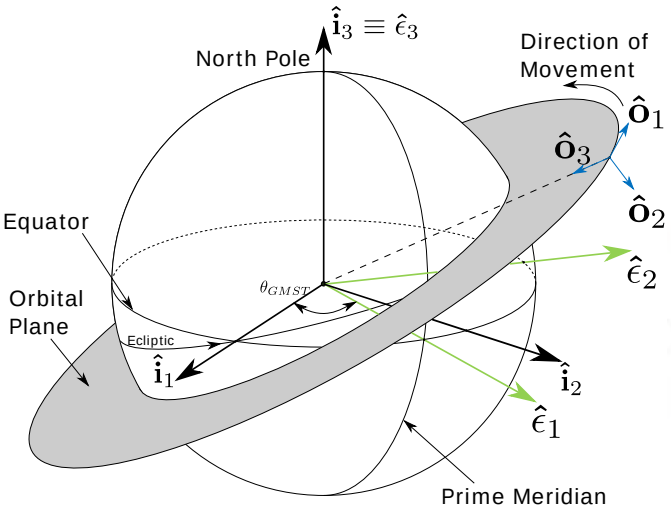



Figure 2.1: Representation of the different reference frames.

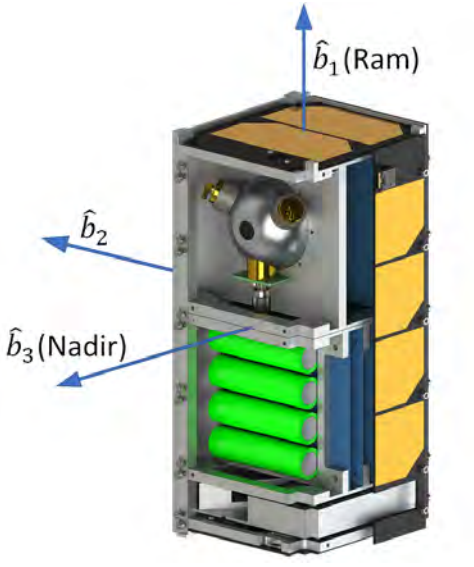



Figure 2.2: Spacecraft body frame.

### 2.1.1 Earth-Centered Inertial (ECI) Frame

The ECI frame, here designated by $I$, is an approximate inertial frame whose origin is in the center of mass of the Earth. It is an approximate inertial frame because it has a linear acceleration due to the movement of the Earth around the Sun, but it does not rotate. This frame is denoted by the triad $\{\hat{\mathbf{i}}_1, \hat{\mathbf{i}}_2, \hat{\mathbf{i}}_3\}$ and is depicted in Figure 2.1. The $\hat{\mathbf{i}}_3$ axis of this frame is aligned with the Earth's North pole, and the $\hat{\mathbf{i}}_1$ axis is aligned with the *vernal equinox* which is the intersection of the Earth's equatorial plane with the plane of the apparent path of the Sun in the Earth's sky (the ecliptic plane) in the direction of the Sun's position relative to the Earth in the first day of spring. The $\hat{\mathbf{i}}_2$ axis completes the right-handed triad.

Due to the Earth's precession and nutation, neither the intersection between the ecliptic plane and the Earth's equatorial plane nor the polar axis are initially fixed, and thus this system is defined by the mean orientation of those axes at a fixed epoch. The epoch used in this work is the current standard epoch (at the time of writing), the J2000.

### 2.1.2 Earth-Centered/Earth-Fixed (ECEF) Frame

Similarly to the ECI frame, the ECEF frame, designated in this work by $E$, has its origin fixed in the center of the Earth, but unlike the previous frame, the ECEF frame rotates with the Earth. This frame is denoted by $\{\hat{\boldsymbol{\epsilon}}_1, \hat{\boldsymbol{\epsilon}}_2, \hat{\boldsymbol{\epsilon}}_3\}$. The $\hat{\boldsymbol{\epsilon}}_3$ axis is aligned with the north pole, being $\hat{\boldsymbol{\epsilon}}_3 \equiv \hat{\mathbf{i}}_3$, and the $\hat{\boldsymbol{\epsilon}}_1$ axis points in the direction of Earth's prime meridian. The $\hat{\boldsymbol{\epsilon}}_2$ axis completes the right-handed triad.

The rotation angle $\theta_{GMST}$ in Figure 2.1 is the Greenwich Mean Sidereal Time (GMST). With this angle, it is possible to write the transformation of a vector $\vec{\mathbf{v}}$ from its representation in the ECI frame to the ECEF frame. This transformation is given by

$$\vec{\mathbf{v}}_E = \mathbf{A}_I^E \vec{\mathbf{v}}_I = \begin{bmatrix} \cos\theta_{GMST} & \sin\theta_{GMST} & 0 \\ -\sin\theta_{GMST} & \cos\theta_{GMST} & 0 \\ 0 & 0 & 1 \end{bmatrix} \vec{\mathbf{v}}_I \tag{2.1}$$

where $\vec{\mathbf{v}}_E$ and $\vec{\mathbf{v}}_I$ are the representations of the vector $\vec{\mathbf{v}}$ in the ECEF frame $E$ and in the ECI frame $I$ respectively. The $\theta_{GMST}$ angle is given in degrees by [50]

$$\theta_{GMST} = \frac{1}{240} \operatorname{mod}\{24110.54841 + 8640184.812866 T_0 + 0.093104 T_0^2 - \\ - 6.2 \times 10^{-6} T_0^3 + 1.002737909350795(3600h + 60m + s), 86400\} \tag{2.2}$$

where $h$, $m$, $s$ are the hours, minutes and seconds given in Universal Time 1 (UT1) and $T_0$ is the number of Julian centuries elapsed from the epoch J2000 to zero hours of the date in question, and is given by

$$T_0 = \frac{JD(Y, M, D, 0, 0, 0) - 2451545}{36525} \tag{2.3}$$

where $Y$ is a given year between 1901 and 2099 and $M$ and $D$ are the month and the day of the desired

date. The Julian date $JD$ is computed by

$$JD(Y,M,D,h,m,s) = 1721013.5 + 367Y - \text{INT}\left\{\frac{7}{4}Y + \left[\text{INT}\left(\frac{M+9}{12}\right)\right]\right\} + \text{INT}\left(\frac{275M}{9}\right) + D + \\ + \frac{60h + m + s/60^*}{1440} \quad (2.4)$$

where $\text{INT}$ means the integer (floor) part and $60^*$ denotes using $61\text{s}$ for days with a leap second.

### 2.1.3 Local-Vertical/Local-Horizontal (LVLH) Frame

The LVLH frame is very useful for Earth pointing spacecrafts. This frame, also known as the orbit frame, is going to be designated in this work by $O = \{\hat{\mathbf{o}}_1, \hat{\mathbf{o}}_2, \hat{\mathbf{o}}_3\}$, and is depicted in Figure 2.1. The origin of this frame is attached to the spacecraft, the $\hat{\mathbf{o}}_3$ axis points along the nadir (geocentric) direction, the $\hat{\mathbf{o}}_2$ axis points along the negative orbit normal, and the $\hat{\mathbf{o}}_1$ axis completes the right-handed triad.

In order to transform a vector $\vec{\mathbf{v}}$ written in the orbital frame coordinates, $\vec{\mathbf{v}}_O$, to its representation in the ECI frame, $\vec{\mathbf{v}}_I$, one can use the following equation,

$$\vec{\mathbf{v}}_I = \mathbf{A}_O^I \vec{\mathbf{v}}_O = \begin{bmatrix} \hat{\mathbf{o}}_{1_I} & \hat{\mathbf{o}}_{2_I} & \hat{\mathbf{o}}_{3_I} \end{bmatrix} \vec{\mathbf{v}}_O \quad (2.5)$$

with

$$\hat{\mathbf{o}}_{3_I} = -\frac{\vec{\mathbf{r}}_I}{\|\vec{\mathbf{r}}_I\|} \quad (2.6a)$$

$$\hat{\mathbf{o}}_{2_I} = -\frac{(\vec{\mathbf{r}}_I \times \vec{\mathbf{v}}_I)}{\|\vec{\mathbf{r}}_I \times \vec{\mathbf{v}}_I\|} \quad (2.6b)$$

$$\hat{\mathbf{o}}_{1_I} = \hat{\mathbf{o}}_{2_I} \times \hat{\mathbf{o}}_{3_I} \quad (2.6c)$$

where $\vec{\mathbf{r}}_I$ and $\vec{\mathbf{v}}_I$ are the position and velocity of the spacecraft in frame $I$ and $\hat{\mathbf{o}}_{1_I}$ $\hat{\mathbf{o}}_{2_I}$ and $\hat{\mathbf{o}}_{3_I}$ are the representations of the $O$ frame axes in frame $I$.

### 2.1.4 Body Frame

The spacecraft body frame $B = \{\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \hat{\mathbf{b}}_3\}$ has its origin fixed on the center of mass of the spacecraft, and its axes rotate with the spacecraft. This frame is depicted in Figure 2.2. The body frame of the ORCASat is oriented so that in nominal mode, the payload is nadir pointing, and under a null pointing error, frame $B$ is coincident with frame $O$ (Figure 2.1). The $\hat{\mathbf{b}}_3$ axis is normal to the large face where the payload is, pointing in the payload's direction while the $\hat{\mathbf{b}}_2$ axis is normal to the other set of large faces and, in the absence of pointing error, has the opposite direction of the orbit normal. The $\hat{\mathbf{b}}_1$ axis forms the right-handed system being normal to the set of smaller faces of the CubeSat.

## 2.2 Attitude Representations

The attitude of a spacecraft consists of the orientation of the spacecraft relative to a certain reference frame. This attitude can be represented in different ways. The most commonly used are the quaternion representation, which makes use of only four parameters, being lowest-dimensional parameterization that is free from singularities [51], the Direction Cosine Matrix (DCM) representation which uses 9 different parameters for representing the attitude, the Euler Angles representation and the Gibbs vector or Rodrigues Parameter representation which are the lowest-parameter representations, using only 3 parameters, which makes them have singularities. Due to the non-singularity free nature of the Euler angles and Gibbs vector attitude representations, they are not going to be used for the attitude parameterization of the ORCASat and so they will not be presented in the following development. The advantages and disadvantages of each attitude parameterization are presented in Table 2.1 [11, 52].

Table 2.1: Attitude parameterizations.

| **Parameterization** | **Advantages** | **Disadvantages** |
|---|---|---|
| DCM | Uniquely defines the attitude<br>No singularities<br>No trigonometric functions<br>Convenient rule for successive rotations | Inefficient - Six redundant parameters<br>Difficult to enforce the orthogonality constraint |
| Quaternions | No singularities<br>No trigonometric functions<br>Convenient rule for successive rotations<br>Only one redundant parameter<br>Easy to enforce the normalization constraint<br>Easy to convert into a rotation matrix | Not intuitive<br>Non-uniqueness |
| Euler angles | Physical interpretation<br>No redundant parameters | Involve trigonometric functions<br>They have singularities for some orientations<br>Not convenient rule for successive rotations |
| Gibbs Vector | No redundant parameter<br>No trigonometric functions<br>Convenient rule for successive rotations | Singularity for 180 degrees rotation |

### 2.2.1 Direction Cosine Matrix (DCM)

The same vector $\vec{\mathbf{v}}$ can be represented in different reference frames. By saying for instance that $\vec{\mathbf{v}}_I$ is the representation of vector $\vec{\mathbf{v}}$ in the inertial frame $I$ and $\vec{\mathbf{v}}_B$ is the representation of vector $\vec{\mathbf{v}}$ in the body frame $B$, without loss of generality, these two representations are related by [52]

$$\vec{\mathbf{v}}_B = \mathbf{A}_I^B \vec{\mathbf{v}}_I \tag{2.7}$$

where the matrix $\mathbf{A}_I^B$, known as DCM or rotation matrix, is a $3 \times 3$ orthogonal matrix with determinant $+1$, meaning that

$$\mathbf{A}_I^B \left(\mathbf{A}_I^B\right)^\top = \left(\mathbf{A}_I^B\right)^\top \mathbf{A}_I^B = \mathbf{I}_3 = \mathbf{A}_I^B \left(\mathbf{A}_I^B\right)^{-1} = \left(\mathbf{A}_I^B\right)^{-1} \mathbf{A}_I^B \tag{2.8}$$

where $\mathbf{I}_3$ is the $3 \times 3$ identity matrix. The property expressed by Equation (2.8) is very important because it preserves both lengths of vectors and angles between them, independently of the reference frame in which they are represented, allowing to perform the inner product between the two different vectors in any frame. If $\vec{\mathbf{v}}_B$ and $\vec{\mathbf{v}}_I$ are written as

$$\vec{\mathbf{v}}_B = \begin{bmatrix} v_{1_B} \\ v_{2_B} \\ v_{3_B} \end{bmatrix} = \begin{bmatrix} \vec{\mathbf{v}} \cdot \hat{\mathbf{b}}_1 \\ \vec{\mathbf{v}} \cdot \hat{\mathbf{b}}_2 \\ \vec{\mathbf{v}} \cdot \hat{\mathbf{b}}_3 \end{bmatrix} \qquad \vec{\mathbf{v}}_I = \begin{bmatrix} v_{1_I} \\ v_{2_I} \\ v_{3_I} \end{bmatrix} = \begin{bmatrix} \vec{\mathbf{v}} \cdot \hat{\mathbf{i}}_1 \\ \vec{\mathbf{v}} \cdot \hat{\mathbf{i}}_2 \\ \vec{\mathbf{v}} \cdot \hat{\mathbf{i}}_3 \end{bmatrix} \tag{2.9}$$

where $\hat{\mathbf{b}}_i$ and $\hat{\mathbf{i}}_i$ with $i = 1, 2, 3$ are the basis unit vectors of frames $B$ and $I$ respectively, and $v_{i_B}$ and $v_{i_I}$ are the components of the vector $\vec{\mathbf{v}}$ in frames $B$ and $I$ respectively, then the DCM $\mathbf{A}_I^B$ is given by

$$\mathbf{A}_I^B = \begin{bmatrix} \hat{\mathbf{b}}_1 \cdot \hat{\mathbf{i}}_1 & \hat{\mathbf{b}}_1 \cdot \hat{\mathbf{i}}_2 & \hat{\mathbf{b}}_1 \cdot \hat{\mathbf{i}}_3 \\ \hat{\mathbf{b}}_2 \cdot \hat{\mathbf{i}}_1 & \hat{\mathbf{b}}_2 \cdot \hat{\mathbf{i}}_2 & \hat{\mathbf{b}}_2 \cdot \hat{\mathbf{i}}_3 \\ \hat{\mathbf{b}}_3 \cdot \hat{\mathbf{i}}_1 & \hat{\mathbf{b}}_3 \cdot \hat{\mathbf{i}}_2 & \hat{\mathbf{b}}_3 \cdot \hat{\mathbf{i}}_3 \end{bmatrix} = \begin{bmatrix} (\hat{\mathbf{b}}_{1_I})^\top \\ (\hat{\mathbf{b}}_{2_I})^\top \\ (\hat{\mathbf{b}}_{3_I})^\top \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{i}}_{1_B} & \hat{\mathbf{i}}_{2_B} & \hat{\mathbf{i}}_{3_B} \end{bmatrix} \tag{2.10}$$

where $\hat{\mathbf{b}}_{k_I}$ with $k = 1, 2, 3$ is the representation of the basis vectors of $B$ in frame $I$ and $\hat{\mathbf{i}}_{k_B}$ is the representation of the basis vectors of $I$ in frame $B$. The inverse rotation can be written as

$$\vec{\mathbf{v}}_I = \mathbf{A}_B^I \vec{\mathbf{v}}_B = \left(\mathbf{A}_I^B\right)^\top \vec{\mathbf{v}}_B \tag{2.11}$$

If there are one or more intermediate frames, like for instance frames $J$ and $K$, then

$$\vec{\mathbf{v}}_B = \mathbf{A}_I^B \vec{\mathbf{v}}_I = \mathbf{A}_K^B \mathbf{A}_J^K \mathbf{A}_I^J \vec{\mathbf{v}}_I \tag{2.12}$$

where $\mathbf{A}_I^J$ corresponds to the first rotation performed, $\mathbf{A}_J^K$ to the second and $\mathbf{A}_K^B$ corresponds to the third and final rotation performed. In general, the rotations do not commute, which means that the order of the rotation is important,

$$\mathbf{A}_K^B \mathbf{A}_I^K \neq \mathbf{A}_I^K \mathbf{A}_K^B \tag{2.13}$$

### 2.2.2 Quaternion

The quaternions were introduced by Hamilton as a result of the searching for hypercomplex numbers that could be represented by points in three-dimensional space [12]. The quaternions can be defined as a four-component vector given by,

$$\mathbf{q} = \begin{bmatrix} \vec{\mathbf{q}} \\ q_4 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{2.14}$$

where $\vec{\mathbf{q}}$ is the vector part of the quaternion and $q_4$ is the scalar component of the quaternion. The quaternions used to parameterize rotations are unit quaternions [9] , i.e.:

$$\mathbf{q}^\top \mathbf{q} = 1 \tag{2.15}$$

The vector part of the attitude quaternions and the scalar part can be represented as

$$\vec{\mathbf{q}} = \vec{\mathbf{e}} \sin\left(\frac{\theta}{2}\right) = \begin{bmatrix} e_1 \sin(\theta/2) \\ e_2 \sin(\theta/2) \\ e_3 \sin(\theta/2) \end{bmatrix} \qquad q_4 = \cos\left(\frac{\theta}{2}\right) \tag{2.16}$$

where $\vec{\mathbf{e}} = \begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}^\top$ is the Euler rotation axis associated to the rotation between two different frames and $\theta$ is the rotation angle about that axis. From here on, whenever the quaternion is mentioned, it is implied that it is a unit quaternion without any loss of generality.

**Quaternion Algebra**

In order to apply the quaternions for attitude representation, it is important to understand how the quaternion algebra works. The most important operations used in this work are represented in this section and provide a better understanding of the following sections, namely for the attitude kinematics of the spacecraft. The relations presented here are valid for all quaternions and not only to the attitude quaternions [9, 12].

The product of two quaternions can be given by two different rules. This product is associative, distributive but generally not commutative. Having two quaternions, $\mathbf{p} = \begin{bmatrix} \vec{\mathbf{p}}^\top & p_4 \end{bmatrix}^\top$ and $\mathbf{q} = \begin{bmatrix} \vec{\mathbf{q}}^\top & q_4 \end{bmatrix}^\top$, the product between the two is given by

$$\mathbf{q} \otimes \mathbf{p} = \begin{bmatrix} p_4 \vec{\mathbf{q}} + q_4 \vec{\mathbf{p}} - \vec{\mathbf{q}} \times \vec{\mathbf{p}} \\ q_4 p_4 - \vec{\mathbf{q}} \cdot \vec{\mathbf{p}} \end{bmatrix} \tag{2.17}$$

$$\mathbf{q} \odot \mathbf{p} = \begin{bmatrix} p_4 \vec{\mathbf{q}} + q_4 \vec{\mathbf{p}} + \vec{\mathbf{q}} \times \vec{\mathbf{p}} \\ q_4 p_4 - \vec{\mathbf{q}} \cdot \vec{\mathbf{p}} \end{bmatrix} \tag{2.18}$$

As one can see, the difference between the two formulations is the sign of the cross product between the vector part of each of the quaternions $\mathbf{p}$ and $\mathbf{q}$. The relation between the two formulations is given by

$$\mathbf{p} \otimes \mathbf{q} = \mathbf{q} \odot \mathbf{p} \tag{2.19}$$

The products of Equations (2.17) and (2.18) can be represented as a matrix multiplication very much like the cross product operation

$$\mathbf{q} \otimes \mathbf{p} = [\mathbf{q}\otimes]\mathbf{p} = \mathbf{p} \odot \mathbf{q} \tag{2.20a}$$

$$\mathbf{q} \odot \mathbf{p} = [\mathbf{q}\odot]\mathbf{p} = \mathbf{p} \otimes \mathbf{q} \tag{2.20b}$$

where

$$[\mathbf{q}\otimes] = \begin{bmatrix} \boldsymbol{\Psi}(\mathbf{q}) & \mathbf{q} \end{bmatrix} = \begin{bmatrix} q_4\mathbf{I}_3 - [\vec{\mathbf{q}}\times] & \vec{\mathbf{q}} \\ -\vec{\mathbf{q}}^\top & q_4 \end{bmatrix} \tag{2.21}$$

and

$$[\mathbf{q}\odot] = \begin{bmatrix} \boldsymbol{\Xi}(\mathbf{q}) & \mathbf{q} \end{bmatrix} = \begin{bmatrix} q_4\mathbf{I}_3 + [\vec{\mathbf{q}}\times] & \vec{\mathbf{q}} \\ -\vec{\mathbf{q}}^\top & q_4 \end{bmatrix} \tag{2.22}$$

with $\boldsymbol{\Psi}(\mathbf{q})$ and $\boldsymbol{\Xi}(\mathbf{q})$ $4 \times 3$ matrices given by

$$\boldsymbol{\Psi}(\mathbf{q}) = \begin{bmatrix} q_4\mathbf{I}_3 - [\vec{\mathbf{q}}\times] \\ -\vec{\mathbf{q}}^\top \end{bmatrix} = \begin{bmatrix} q_4 & q_3 & -q_2 \\ -q_3 & q_4 & q_1 \\ q_2 & -q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \tag{2.23}$$

$$\boldsymbol{\Xi}(\mathbf{q}) = \begin{bmatrix} q_4\mathbf{I}_3 + [\vec{\mathbf{q}}\times] \\ -\vec{\mathbf{q}}^\top \end{bmatrix} = \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \tag{2.24}$$

Using Equations (2.23) and (2.24) it is possible to prove that

$$\boldsymbol{\Psi}^\top(\mathbf{q})\mathbf{q} = \boldsymbol{\Xi}^\top(\mathbf{q})\mathbf{q} = \vec{\mathbf{0}} \tag{2.25}$$

where $\vec{\mathbf{0}}$ is a $3 \times 1$ vector of zeros. The identity quaternion must obey:

$$\mathbf{I}_q \otimes \mathbf{q} = \mathbf{q} \otimes \mathbf{I}_q = \mathbf{I}_q \odot \mathbf{q} = \mathbf{q} \odot \mathbf{I}_q = \mathbf{q} \tag{2.26}$$

being thus defined as

$$\mathbf{I}_q = \begin{bmatrix} \vec{\mathbf{0}} \\ 1 \end{bmatrix} \tag{2.27}$$

The conjugate of a quaternion is defined by changing the sign of the vector part of the quaternion:

$$\mathbf{q}^* = \begin{bmatrix} \vec{\mathbf{q}} \\ q_4 \end{bmatrix}^* = \begin{bmatrix} -\vec{\mathbf{q}} \\ q_4 \end{bmatrix} \tag{2.28}$$

The product of the quaternion with its conjugate is then the square of its norm times the identity quaternion:

$$\mathbf{q} \otimes \mathbf{q}^* = \mathbf{q}^* \otimes \mathbf{q} = \mathbf{q} \odot \mathbf{q}^* = \mathbf{q}^* \odot \mathbf{q} = \|\mathbf{q}\|^2 \mathbf{I}_q \tag{2.29}$$

By replacing the conjugate quaternion (Eq. (2.28)) in Equations (2.21) and (2.22), it is easy to see that

$$[\mathbf{q}^*\otimes] = [\mathbf{q}\otimes]^\top \qquad \text{and} \qquad [\mathbf{q}^*\odot] = [\mathbf{q}\odot]^\top \tag{2.30}$$

The inverse of a quaternion having nonzero norm is defined as

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{\|\mathbf{q}\|^2} \tag{2.31}$$

so that

$$\mathbf{q} \otimes \mathbf{q}^{-1} = \mathbf{q}^{-1} \otimes \mathbf{q} = \mathbf{q} \odot \mathbf{q}^{-1} = \mathbf{q}^{-1} \odot \mathbf{q} = \mathbf{I}_q \tag{2.32}$$

By applying Equation (2.31) to an attitude quaternion, whose norm is equal to one, one can see that the inverse of an attitude quaternion is equal to its conjugate quaternion, which is analogous to the fact that the inverse of a rotation matrix is equal to its transpose (Equation (2.8)). Analogous to the attitude matrix case, one can think on the conjugate/inverse of the attitude quaternion as the representation of the inverse rotation.

The multiplication between a $3 \times 1$ vector $\vec{\mathbf{v}}$ and a quaternion is done by defining the auxiliary vector $\mathbf{v} = \begin{bmatrix} \vec{\mathbf{v}} \\ 0 \end{bmatrix}$. The multiplication is then performed as if it were a multiplication between any two quaternions:

$$\mathbf{v} \otimes \mathbf{q} = \begin{bmatrix} \vec{\mathbf{v}} \\ 0 \end{bmatrix} \otimes \mathbf{q} = [\mathbf{v}\otimes]\mathbf{q} \qquad \text{and} \qquad \mathbf{q} \otimes \mathbf{v} = \mathbf{q} \otimes \begin{bmatrix} \vec{\mathbf{v}} \\ 0 \end{bmatrix} = [\mathbf{q}\otimes]\mathbf{v} \tag{2.33}$$

$$\mathbf{v} \odot \mathbf{q} = \begin{bmatrix} \vec{\mathbf{v}} \\ 0 \end{bmatrix} \odot \mathbf{q} = [\mathbf{v}\odot]\mathbf{q} \qquad \text{and} \qquad \mathbf{q} \odot \mathbf{v} = \mathbf{q} \odot \begin{bmatrix} \vec{\mathbf{v}} \\ 0 \end{bmatrix} = [\mathbf{q}\odot]\mathbf{v} \tag{2.34}$$

where the products $[\mathbf{q}\otimes]\mathbf{v}$ and $[\mathbf{q}\odot]\mathbf{v}$ can be simplified as

$$[\mathbf{q}\otimes]\mathbf{v} = \mathbf{\Psi}(\mathbf{q})\vec{\mathbf{v}} \qquad \text{and} \qquad [\mathbf{q}\odot]\mathbf{v} = \mathbf{\Xi}(\mathbf{q})\vec{\mathbf{v}} \tag{2.35}$$

**Quaternion to DCM**

The conversion from the attitude quaternion representation to DCM is given by [9]

$$\begin{aligned} \mathbf{A}(\mathbf{q}) &= \left(q_4^2 - \|\vec{\mathbf{q}}\|^2\right)\mathbf{I}_3 - 2q_4[\vec{\mathbf{q}}\times] + 2\vec{\mathbf{q}}\vec{\mathbf{q}}^\top = \mathbf{\Xi}^\top(\mathbf{q})\mathbf{\Psi}(\mathbf{q}) = \\ &= \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_2q_1 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_3q_1 + q_2q_4) & 2(q_3q_2 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \end{aligned} \tag{2.36}$$

which is a quadratic function of the elements of the quaternion, requiring no trigonometric or other transcendental functions.

**DCM to Quaternion**

According to [53], the extraction of an attitude quaternion from a DCM is done by choosing one of the four different vectors $\mathbf{x}_i = 4q_i\mathbf{q}$ from Equation (2.38) and by normalizing it, obtaining then the desired attitude quaternion $\mathbf{q}$. Let $\mathbf{A}$ denote the DCM to transform into a quaternion, with $A_{ij}$, $i, j = 1, 2, 3$ the

element of $\mathbf{A}$ in the $i$'th row and $j$'th column, and $\mathrm{tr}(\mathbf{A})$ the trace of $\mathbf{A}$. The solution $\mathbf{x}_i$ to choose from the set $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$ that minimizes the numerical errors is the one corresponding to the same element position as the maximum (most positive) element of the set $\{A_{11}, A_{22}, A_{33}, \mathrm{tr}\,\mathbf{A}\}$. The desired attitude quaternion $\mathbf{q}$ is then given by

$$\mathbf{q} = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|} \tag{2.37}$$

$$\mathbf{x}_1(\mathbf{A}) = \begin{bmatrix} 1 - \mathrm{tr}(\mathbf{A}) + 2A_{11} \\ A_{12} + A_{21} \\ A_{13} + A_{31} \\ A_{23} - A_{32} \end{bmatrix} \tag{2.38a}$$

$$\mathbf{x}_2(\mathbf{A}) = \begin{bmatrix} A_{21} + A_{12} \\ 1 - \mathrm{tr}(\mathbf{A}) + 2A_{22} \\ A_{23} + A_{32} \\ A_{31} - A_{13} \end{bmatrix} \tag{2.38b}$$

$$\mathbf{x}_3(\mathbf{A}) = \begin{bmatrix} A_{31} + A_{13} \\ A_{32} + A_{23} \\ 1 - \mathrm{tr}(\mathbf{A}) + 2A_{33} \\ A_{12} - A_{21} \end{bmatrix} \tag{2.38c}$$

$$\mathbf{x}_4(\mathbf{A}) = \begin{bmatrix} A_{23} - A_{32} \\ A_{31} - A_{13} \\ A_{12} - A_{21} \\ 1 + \mathrm{tr}(\mathbf{A}) \end{bmatrix} \tag{2.38d}$$

**Quaternion Transformation Between Reference Frames**

The transformation of a vector $\vec{\mathbf{v}}$ written in an arbitrary frame $I$ to an arbitrary frame $B$ can be given by the double quaternion product $\mathbf{q}_I^B \otimes \mathbf{v}_I \otimes \left(\mathbf{q}_I^B\right)^*$ using the results from Equations (2.35), (2.20), (2.30), (2.22) and (2.36):

$$\vec{\mathbf{v}}_B = \mathbf{q}_I^B \otimes \mathbf{v}_I \otimes \left(\mathbf{q}_I^B\right)^* = \left[\left(\mathbf{q}_I^B\right)^* \odot\right] \mathbf{\Psi}(\mathbf{q}_I^B)\vec{\mathbf{v}} = \begin{bmatrix} \mathbf{\Xi}^\top(\mathbf{q}_I^B)\mathbf{\Psi}(\mathbf{q}_I^B)\vec{\mathbf{v}}_I \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{A}(\mathbf{q}_\mathbf{I}^\mathbf{B})\vec{\mathbf{v}}_I \\ 0 \end{bmatrix} \tag{2.39}$$

where $\mathbf{A}(\mathbf{q}_I^B) \equiv \mathbf{A}_I^B$. Applying now a second transformation to $\vec{\mathbf{v}}$ using $\mathbf{q}_B^K$ gives

$$\vec{\mathbf{v}}_K = \mathbf{q}_B^K \otimes \mathbf{v}_B \otimes \left(\mathbf{q}_B^K\right)^* = \mathbf{q}_B^K \otimes \left[\mathbf{q}_I^B \otimes \mathbf{v}_I \otimes \left(\mathbf{q}_I^B\right)^*\right] \otimes \left(\mathbf{q}_B^K\right)^* = \begin{bmatrix} \mathbf{A}(\mathbf{q}_\mathbf{B}^\mathbf{K})\mathbf{A}(\mathbf{q}_\mathbf{I}^\mathbf{B})\vec{\mathbf{v}}_I \\ 0 \end{bmatrix} \tag{2.40}$$

where by using the associative property of the quaternion multiplication,

$$\mathbf{q}_B^K \otimes \left[\mathbf{q}_I^B \otimes \mathbf{v}_I \otimes \left(\mathbf{q}_I^B\right)^*\right] \otimes \left(\mathbf{q}_B^K\right)^* = (\mathbf{q}_B^K \otimes \mathbf{q}_I^B) \otimes \mathbf{v}_I \otimes \left(\mathbf{q}_B^K \otimes \mathbf{q}_I^B\right)^* = \begin{bmatrix} \mathbf{A}\left(\mathbf{q}_\mathbf{B}^\mathbf{K} \otimes \mathbf{q}_\mathbf{I}^\mathbf{B}\right)\vec{\mathbf{v}}_I \\ 0 \end{bmatrix} \tag{2.41}$$

proving that

$$\mathbf{A}\left(\mathbf{q}_\mathbf{B}^\mathbf{K} \otimes \mathbf{q}_\mathbf{I}^\mathbf{B}\right) = \mathbf{A}\left(\mathbf{q}_\mathbf{B}^\mathbf{K}\right)\mathbf{A}\left(\mathbf{q}_\mathbf{I}^\mathbf{B}\right) \tag{2.42}$$

and thus the quaternion representation of successive transformations is the product of the quaternions that constitute that transformation, in the same way that the attitude matrix of the combined transformation is the product of the individual attitude matrices, being the order of the quaternion product $\otimes$ identical to the order of matrix multiplication [9].

## 2.3 Stability of Nonlinear Systems

In this section, the concepts of stability and asymptotic stability as well as the stability analysis of a nonlinear dynamic system are introduced. These concepts will allow us to evaluate and predict what the evolution of the nonlinear system will be when driven by the different control laws. The stability theory presented in this section follows from [12]. Consider the following nonlinear system described by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) \qquad\qquad \mathbf{f}(\mathbf{x}_e, t) = \mathbf{0}, \quad \forall t \tag{2.43}$$

where $\mathbf{x}$ is the system's state vector and $\mathbf{x}_e$ is an equilibrium point of the system.

**Definition 1** (Lyapunov stability)**.** An isolated equilibrium point $\mathbf{x}_e$ is Lyapunov stable if for any $\epsilon > 0$ there exists a real positive number $\delta(\epsilon, t_0)$ such that $\|\mathbf{x}(t_0) - \mathbf{x}_e\| \leqslant \delta \Rightarrow \|\mathbf{x}(t) - \mathbf{x}_e\| \leqslant \epsilon, \quad \forall t \geqslant t_0$.

**Definition 2** (Local asymptotic stability)**.** An isolated equilibrium point $\mathbf{x}_e$ is said to be locally asymptotically stable if it is Lyapunov stable and $\|\mathbf{x}(t_0) - \mathbf{x}_e\| \leqslant \delta \Rightarrow \mathbf{x}(t) \to \mathbf{x}_e \quad \text{as} \quad t \to \infty$.

**Definition 3** (Global asymptotic stability)**.** An equilibrium point $\mathbf{x}_e$ is said to be global asymptotic stable if it is Lyapunov stable and $\mathbf{x}(t) \to \mathbf{x}_e \quad \text{as} \quad t \to \infty$ for any initial conditions $\mathbf{x}(t_0)$.

**Definition 4** (Instability)**.** An equilibrium point is said to be unstable if it is not Lyapunov stable nor asymptotically stable.

In simple terms, Definition 1 states that if the system starts near an equilibrium point $\mathbf{x}_e$, it will stay near the equilibrium forever, while Definition $2$ more strongly states that if the system starts near an equilibrium point $\mathbf{x}_e$, then it will converge to $\mathbf{x}_e$ and not only stay near it. Definition 3 implies that a necessary condition for the equilibrium point to be globally asymptotically stable is that it is the only equilibrium point. Consider now the following scalar function $\mathcal{V}(\mathbf{x})$, with continuous first partial derivatives with respect to $\mathbf{x}$, satisfying the following properties in a neighborhood $D$ of the equilibrium point $\mathbf{x}_e$

$$\mathcal{V}(\mathbf{x}) > 0, \quad \forall \mathbf{x} \neq \mathbf{x}_e \qquad\qquad \mathcal{V}(\mathbf{x}_e) = 0 \tag{2.44}$$

**Theorem 1** (Lyapunov stability)**.** The solution $\mathbf{x} = \mathbf{x}_e$ of the system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ with $\mathbf{f}(\mathbf{x}_e) = 0$ is Lyapunov stable (Definition 1) if $\dot{\mathcal{V}}(\mathbf{x}) \leqslant 0, \quad \forall \mathbf{x} \neq \mathbf{x}_e$ in $D$ and $t$.

**Theorem 2** (Local asymptotic stability)**.** The solution $\mathbf{x} = \mathbf{x}_e$ of the system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ with $\mathbf{f}(\mathbf{x}_e) = 0$ is locally asymptotically stable (Definition 2) if $\dot{\mathcal{V}}(\mathbf{x}) < 0 \quad \forall \mathbf{x} \neq \mathbf{x}_e$ in $D$ and $t$.

**Theorem 3** (Global asymptotic stability)**.** The solution $\mathbf{x} = \mathbf{x}_e$ of the system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ with $\mathbf{f}(\mathbf{x}_e) = 0$ is globally asymptotically stable (Definition 3) if $\dot{\mathcal{V}}(\mathbf{x}) < 0, \quad \forall \mathbf{x} \neq \mathbf{x}_e$ and if in addition there is an entire state space where $\mathcal{V}(\mathbf{x}) > 0, \quad \forall \mathbf{x} \neq \mathbf{x}_e$, is radially unbounded, i.e., $\mathcal{V}(\mathbf{x}) \to \infty \quad \text{as} \quad \|\mathbf{x}\| \to \infty$.

**Theorem 4** (Instability)**.** The solution $\mathbf{x} = \mathbf{x}_e$ of the system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ with $\mathbf{f}(\mathbf{x}_e) = 0$ is unstable (Definition 4) if $\dot{\mathcal{V}}(\mathbf{x}) > 0, \quad \forall \mathbf{x} \neq \mathbf{x}_e$, and $\dot{\mathcal{V}}(\mathbf{x}_e) = 0, \quad \forall t$.

The function $\mathcal{V}(\mathbf{x})$ is called the Lyapunov function. The proper selection of this function will allow us to prove the nonlinear system stability using theorems 1 through 4.

# Chapter 3

# Spacecraft Mechanics

This chapter will be concerned with the laws that describe the motion of the spacecraft under the influence of a system of forces and torques. First, in Section 3.1, the orbital mechanics as well as the mathematical model for the gravity force are described. In Section 3.2, the attitude kinematics of the spacecraft are presented. Then, in Section 3.3, the effect of the forces and torques applied to the spacecraft are going to be presented for an analysis in terms of attitude dynamics. In Section 3.4, the main external forces and torques which influence the motion of the spacecraft in space are characterized. Finally, in Section 3.5, a linearized model for the spacecraft's attitude dynamics and kinematics is described.

## 3.1 Orbital Mechanics

The orbit of a spacecraft is defined as its path through space [54]. With the help of the observational data from the Danish astronomer Tycho Brahe, Johannes Kepler discovered the three laws of planetary motion. Later these laws were formalized by Isaac Newton which described the motion of two celestial bodies with masses $M$ and $m$ orbiting their common center of mass by [9]

$$\ddot{\vec{\mathbf{r}}} = -\frac{G(M+m)}{\|\vec{\mathbf{r}}\|^3}\vec{\mathbf{r}} \tag{3.1}$$

where $\vec{\mathbf{r}}$ is the relative position of both masses $M$ and $m$ measured in an inertial frame and $G$ is the universal gravitational constant. Equation (3.1) ignores the non-spherical symmetry of the two bodies, the perturbations due to other bodies and non-gravitational forces like the air drag and solar radiation pressure. This special type of orbits is termed as Keplerian orbit. Equation (3.1) can be further simplified if one considers that $M \gg m$ which for the case of a spacecraft orbiting the Earth is a valid assumption. In this way, the center of mass of the system (spacecraft + Earth) is coincident with the center of mass of the Earth. The motion of the spacecraft is now described about an inertial frame whose origin is coincident with the Earth's center of mass, such as the ECI frame from Section 2.1.1. Equation (3.1)

can thus be written as

$$\ddot{\vec{\mathbf{r}}}_I = -\frac{\mu_\oplus}{\|\vec{\mathbf{r}}_I\|^3}\vec{\mathbf{r}}_I \tag{3.2}$$

where $\mu_\oplus = GM_\oplus$ is the Earth's standard gravitational parameter, $M_\oplus$ is the Earth's mass, and $\vec{\mathbf{r}}_I$ is the position of the spacecraft in the ECI frame $I$.

Since the orbit being considered for the ORCASat is a Low Earth Orbit (LEO), the effects of the Earth's non-spherical mass distribution are one of the major sources of perturbations [11]. A better approximation for the Earth's gravity field can be given by the use of spherical harmonics. It can be proved that the gravity force, and therefore by Newton's second law, the gravity acceleration $\ddot{\vec{\mathbf{r}}}$, can be expressed as a gradient of a scalar function $U$:

$$\ddot{\vec{\mathbf{r}}} = -\boldsymbol{\nabla} U \tag{3.3}$$

In the ECEF frame (Section 2.1.2), this scalar function, the gravitational potential, can be expressed in a spherical harmonics expansion by [9]

$$U = \frac{\mu_\oplus}{\|\vec{\mathbf{r}}\|}\left\{1 + \sum_{n=1}^{\infty}\left(\frac{R_\oplus}{\|\vec{\mathbf{r}}\|}\right)^n \sum_{m=0}^{n} P_n^m(\sin\lambda)\left[C_n^m\cos(m\phi) + S_n^m\sin(m\phi)\right]\right\} \tag{3.4}$$

where $R_\oplus$ is the Earth's mean equatorial radius, $\lambda$ and $\phi$ are respectively the geocentric latitude and the longitude of the point mass, $P_n^m$ is the associate Legendre function of degree $n$ and order $m$, and $C_n^m$ and $S_n^m$ are geopotential coefficients which are found by fitting gravity measurement data. The gravity acceleration from Equation (3.3), which as seen was expressed in the ECEF frame, must now be transformed to the ECI frame:

$$\ddot{\vec{\mathbf{r}}}_I = -\mathbf{A}_E^I \boldsymbol{\nabla} U \tag{3.5}$$

where $\mathbf{A}_E^I = \left(\mathbf{A}_I^E\right)^\top$, with $\mathbf{A}_I^E$ given by Equation (2.1). Equation (3.5) can be modified to incorporate the effects due to other perturbative forces acting on the spacecraft as

$$\ddot{\vec{\mathbf{r}}}_I = -\mathbf{A}_E^I \boldsymbol{\nabla} U + \vec{\mathbf{a}}_I^{perturb} \tag{3.6}$$

These perturbative accelerations/forces will be outlined in Section 3.4.

## 3.2 Attitude Kinematics

In Section 2.2 two different attitude parameterizations were considered. This section will now present the kinematic equations for those parameterizations allowing to describe the time-dependent relationship between the different reference frames.

By considering two reference frames, for instance, the body frame $B$ and the ECI frame $I$ presented in Section 2.1, without loss of generality, the angular rate of frame $B$ with respect to frame $I$ is going to be denoted by $\vec{\boldsymbol{\omega}}^{B/I}$ and is in general time-dependent, $\vec{\boldsymbol{\omega}}^{B/I} = \vec{\boldsymbol{\omega}}^{B/I}(t)$. The kinematic differential equation which translates the relative orientation of frame $B$ with respect to frame $I$ can thus be written

as [12]

$$\dot{\mathbf{A}}_I^B = -\left[\vec{\boldsymbol{\omega}}_B^{B/I}\times\right]\mathbf{A}_I^B = -\begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}\mathbf{A}_I^B \tag{3.7}$$

where $\vec{\boldsymbol{\omega}}_B^{B/I} = \begin{bmatrix}\omega_1 & \omega_2 & \omega_3\end{bmatrix}^\top$ is the angular rate $\vec{\boldsymbol{\omega}}^{B/I}$ written in $B$ frame coordinates. The fundamental equation of attitude kinematics, Eq. (3.7), can also be written in the quaternion form being given by [9]

$$\dot{\mathbf{q}}_I^B = \frac{1}{2}\mathbf{q}_I^B \odot \boldsymbol{\omega}_B^{B/I} = \frac{1}{2}\mathbf{q}_I^B \odot \begin{bmatrix}\vec{\boldsymbol{\omega}}_B^{B/I} \\ 0\end{bmatrix} = \frac{1}{2}\begin{bmatrix}\vec{\boldsymbol{\omega}}_B^{B/I} \\ 0\end{bmatrix} \otimes \mathbf{q}_I^B \tag{3.8}$$

By making use of the results from Equations (2.20) and (2.35), Equation (3.8) can be given in matrix form by

$$\dot{\mathbf{q}}_I^B = \frac{1}{2}\Xi(\mathbf{q}_I^B)\vec{\boldsymbol{\omega}}_B^{B/I} = \frac{1}{2}\Omega(\vec{\boldsymbol{\omega}}_B^{B/I})\mathbf{q}_I^B = \frac{1}{2}\begin{bmatrix} q_{I_4}^B\vec{\boldsymbol{\omega}}_B^{B/I} - \vec{\boldsymbol{\omega}}_B^{B/I}\times\vec{\mathbf{q}}_I^B \\ -\vec{\boldsymbol{\omega}}_B^{B/I}\cdot\vec{\mathbf{q}}_I^B \end{bmatrix} \tag{3.9}$$

where $\Xi(\mathbf{q})$ is given by Equation (2.24) and $\Omega(\vec{\boldsymbol{\omega}})$ is obtained by replacing $\begin{bmatrix}\vec{\boldsymbol{\omega}}_B^{B/I} \\ 0\end{bmatrix}$ in Equation (2.21):

$$\Omega(\vec{\boldsymbol{\omega}}) = \begin{bmatrix} -[\vec{\boldsymbol{\omega}}\times] & \vec{\boldsymbol{\omega}} \\ -\vec{\boldsymbol{\omega}}^\top & 0 \end{bmatrix} = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \tag{3.10}$$

with $\vec{\boldsymbol{\omega}} = \begin{bmatrix}\omega_1 & \omega_2 & \omega_3\end{bmatrix}^\top$. By knowing $\vec{\boldsymbol{\omega}}_B^{B/I}$, which is possible with the use of gyroscopes for instance, Equations (3.7) and (3.9) can be numerically integrated to propagate the attitude of the spacecraft.

## 3.3 Attitude Dynamics

The mathematical model of a spacecraft is not complete without the description of the dynamic equations of motion. Section 3.1 was concerned with the motion of the center of mass of objects in space whereas this section will be concerned with the motion of the body about its center of mass. For this analysis, the spacecraft will be considered as a rigid body.

Euler's equation of motion states that the derivative of the angular momentum $\dot{\vec{\mathbf{h}}}_I^c$ of a body about its center of mass $c$ is equal to the net applied torque $\vec{\boldsymbol{\tau}}_I^c$ about its center of mass [9]:

$$\dot{\vec{\mathbf{h}}}_I^c = \vec{\boldsymbol{\tau}}_I^c \tag{3.11}$$

Note that Equation (3.11) is only valid in an inertial frame like the ECI frame described in Section 2.1, hence the use of the subscript $I$. The rigid body assumption allows to write the angular momentum $\vec{\mathbf{h}}_I^c$

as

$$\vec{\mathbf{h}}_I^c = \mathbf{J}_I^c \vec{\boldsymbol{\omega}}_I^{B/I} \tag{3.12}$$

where $\vec{\boldsymbol{\omega}}_I^{B/I}$ is the angular velocity of the body with respect to frame $I$ expressed in the $I$ frame and $\mathbf{J}_I^c$ is the inertia tensor of the body about its center of mass measured in the $I$ frame. The inertia tensor is given in a general frame by [9, 55]

$$\mathbf{J}^c = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{yx} & J_{yy} & J_{yz} \\ J_{zx} & J_{zy} & J_{zz} \end{bmatrix} = \int \|\vec{\mathbf{r}}\|^2 \mathbf{I}_3 - \vec{\mathbf{r}}\,\vec{\mathbf{r}}^\top dm \tag{3.13}$$

with

$$\begin{aligned} J_{xx} &= \int (y^2 + z^2) dm & J_{xy} = J_{yx} &= -\int xy\, dm \\ J_{yy} &= \int (x^2 + z^2) dm & J_{xz} = J_{zx} &= -\int xz\, dm \\ J_{zz} &= \int (x^2 + y^2) dm & J_{yz} = J_{zy} &= -\int yz\, dm \end{aligned}$$

where the terms on the left-hand side are the principal moments of inertia and the terms on the right-hand side are the products of inertia. Vector $\vec{\mathbf{r}} = \begin{bmatrix} x & y & z \end{bmatrix}^\top$ is the position vector of a point particle of mass $dm$ with respect to the body's center of mass. The inertia tensor $\mathbf{J}_I^c$ is in general not constant, and so Equation (3.12) is often specified in the body frame where it is constant since the position of each point particle with respect to the body's center of mass remains constant in this frame (under rigid body assumption). The angular momentum of the spacecraft written in body frame coordinates is then given by

$$\vec{\mathbf{h}}_B^c = \mathbf{A}_I^B \vec{\mathbf{h}}_I^c = \mathbf{A}_I^B \mathbf{J}_I^c \left(\mathbf{A}_I^B\right)^\top \mathbf{A}_I^B \vec{\boldsymbol{\omega}}_I^{B/I} = \mathbf{J}_B^c \vec{\boldsymbol{\omega}}_B^{B/I} \tag{3.14}$$

The fact that $\mathbf{J}^c$ is constant in the body frame together with the fact that the external torques applied to the center of mass of the body are more easily computed in the body frame means that Equation (3.11) is usually given in body frame coordinates [9]. Equation (3.11) is thus replaced by

$$\dot{\vec{\mathbf{h}}}_B^c + \vec{\boldsymbol{\omega}}_B^{B/I} \times \vec{\mathbf{h}}_B^c = \vec{\boldsymbol{\tau}}_B^c \tag{3.15}$$

Substituting Equation (3.14) in Equation (3.15) and remembering that the inertia matrix $\mathbf{J}_B^c$ is constant in body frame coordinates, allows to obtain the Euler's rotational equation:

$$\dot{\vec{\boldsymbol{\omega}}}_B^{B/I} = (\mathbf{J}_B^c)^{-1} \left[ \left( \mathbf{J}_B^c \vec{\boldsymbol{\omega}}_B^{B/I} \right) \times \vec{\boldsymbol{\omega}}_B^{B/I} + \vec{\boldsymbol{\tau}}_B^c \right] \tag{3.16}$$

This equation together with the kinematics equation, Eq. (3.7) or Eq. (3.9), provide a complete description of the motion of the rigid body [9].

In some cases, however, it is not possible to model the spacecraft as a single rigid body. A more complex model consists on analyzing the spacecraft as a system of rigid bodies. In a system of $N$ rigid

bodies, the angular momentum of the system about its center of mass $c$ corresponds to the sum of the angular momentum of each rigid body about the system's center of mass [56]:

$$\vec{\mathbf{h}}^{c} = \sum_{i=1}^{N} \vec{\mathbf{h}}^{i/c} \tag{3.17}$$

where the angular momentum of the $i$'th rigid body about the system's center of mass $\vec{\mathbf{h}}^{i/c}$ is given by

$$\vec{\mathbf{h}}^{i/c} = \vec{\mathbf{h}}^{i/c_i} + m_i(\vec{\mathbf{r}}^{\,c_i/c} \times \vec{\mathbf{v}}^{c_i/c}) \tag{3.18}$$

where $m_i$ is the mass of the $i$'th rigid body, $\vec{\mathbf{h}}^{i/c_i}$ is the angular momentum of the $i$'th rigid body about its own center of mass $c_i$, and $\vec{\mathbf{r}}^{\,c_i/c}$ and $\vec{\mathbf{v}}^{c_i/c}$ are the position and inertial velocity of the $i$'th rigid body center of mass relative to the system's center of mass. The angular momentum $\vec{\mathbf{h}}^{i/c}$ is then given in the body frame by

$$\vec{\mathbf{h}}_B^{i/c} = \mathbf{J}_B^{i/c_i}\omega_B^{i/I} + m_i\left[\vec{\mathbf{r}}_B^{\,c_i/c} \times \left(\dot{\vec{\mathbf{r}}}_B^{\,c_i/c} + \vec{\boldsymbol{\omega}}_B^{B/I} \times \vec{\mathbf{r}}_B^{\,c_i/c}\right)\right] \tag{3.19}$$

where $\mathbf{J}_B^{i/c_i}$ is the inertia matrix of the $i$'th rigid body about its own center of mass $c_i$ written in body frame coordinates, and $\omega_B^{i/I}$ is the inertial angular velocity of the $i$'th rigid body written in body frame coordinates. Under rigid body assumption, the position of $c_i$ relative to $c$ is constant in the body frame, meaning that $\dot{\vec{\mathbf{r}}}_B^{\,c_i/c} = \vec{\mathbf{0}}$. By noting that $\vec{\boldsymbol{\omega}}_B^{i/I} = \vec{\boldsymbol{\omega}}_B^{i/B} + \vec{\boldsymbol{\omega}}_B^{B/I}$, Equation (3.19) can by rewritten as

$$\vec{\mathbf{h}}_B^{i/c} = \left\{\mathbf{J}_B^{i/c_i} + m_i\left[\left\|\vec{\mathbf{r}}_B^{\,c_i/c}\right\|^2\mathbf{I}_3 - \vec{\mathbf{r}}_B^{\,c_i/c}\left(\vec{\mathbf{r}}_B^{\,c_i/c}\right)^\top\right]\right\}\vec{\boldsymbol{\omega}}_B^{B/I} + \mathbf{J}_B^{i/c_i}\vec{\boldsymbol{\omega}}_B^{i/B} \tag{3.20}$$

which explicitly shows in the term $\mathbf{J}_B^{i/c_i}\vec{\boldsymbol{\omega}}_B^{i/B}$ the contribution of the the $i$'th rigid body to the total system's angular momentum due to its angular velocity relatively to the body frame.

The ORCASat requires the inclusion of a momentum wheel. Using Equation (3.17) together with Equation (3.20), the total angular momentum $\vec{\mathbf{h}}$ of the system about its center of mass is given in the body frame by

$$\vec{\mathbf{h}}_B^{c} = \mathbf{J}_B\vec{\boldsymbol{\omega}}_B^{B/I} + \mathbf{J}_B^{w}\vec{\boldsymbol{\omega}}_B^{s} = \mathbf{J}_B\vec{\boldsymbol{\omega}}_B^{B/I} + \vec{\mathbf{h}}_B^{s} \tag{3.21}$$

where $\mathbf{J}_B$ is the inertia matrix of the CubeSat (including the momentum wheel) about its center of mass, $\mathbf{J}_B^w$ is the inertia matrix of the wheel about its own center of mass given in body frame coordinates, and $\vec{\boldsymbol{\omega}}_B^s$ is the angular velocity of the wheel relative to the body frame, i.e., its spin angular velocity. Due to the wheel symmetry about its spinning axis, $\mathbf{J}^w$ is given in a frame $W = \{\hat{\mathbf{w}}^s, \hat{\mathbf{w}}^{\perp_1}, \hat{\mathbf{w}}^{\perp_2}\}$ aligned with the wheel's spin axis by

$$\mathbf{J}_W^w = \operatorname{diag}\begin{pmatrix} J^s & J^\perp & J^\perp \end{pmatrix} \tag{3.22}$$

where $J^s$ is the principal moment of inertia of the momentum wheel about its spin axis $\hat{\mathbf{w}}^s$, which is a principal axis, and $J^\perp$ is the principal moment of inertia of the momentum wheel about any other two axes $\hat{\mathbf{w}}^{\perp_1}$ and $\hat{\mathbf{w}}^{\perp_2}$ perpendicular to the spin axis and to themselves. The inertia tensor $\mathbf{J}_B^w$ is then given by

$$\mathbf{J}_B^w = \mathbf{A}_W^B\mathbf{J}_W^w\left(\mathbf{A}_W^B\right)^\top = J^s\left[\hat{\mathbf{w}}_B^s\left(\hat{\mathbf{w}}_B^s\right)^\top\right] + J^\perp\left[\mathbf{I}_3 - \hat{\mathbf{w}}_B^s\left(\hat{\mathbf{w}}_B^s\right)^\top\right] \tag{3.23}$$

with

$$\mathbf{A}_W^B = \begin{bmatrix} \hat{\mathbf{w}}_B^s & \hat{\mathbf{w}}_B^{\perp_1} & \hat{\mathbf{w}}_B^{\perp_2} \end{bmatrix} \tag{3.24}$$

defined according to Equation (2.10). By noting that $\vec{\boldsymbol{\omega}}_B^s = \omega \hat{\mathbf{w}}_B^s$, Equations (3.21) and (3.23) allow to express the wheel spin angular momentum contribution $\vec{\mathbf{h}}_B^s$ as

$$\vec{\mathbf{h}}_B^s = J^s \omega^s \hat{\mathbf{w}}_B^s \tag{3.25}$$

By using the results from Equations (3.21) and (3.25) in Equation (3.15), the dynamic equation of the ORCASat is obtained:

$$\mathbf{J}_B \dot{\vec{\boldsymbol{\omega}}}_B^{B/I} = (\mathbf{J}_B \vec{\boldsymbol{\omega}}_B^{B/I} + \vec{\mathbf{h}}_B^s) \times \vec{\boldsymbol{\omega}}_B^{B/I} - \dot{\vec{\mathbf{h}}}_B^s + \vec{\boldsymbol{\tau}}_B^c \tag{3.26}$$

with

$$\vec{\boldsymbol{\tau}}_B^c = \vec{\boldsymbol{\tau}}_B^{perturb} + \vec{\mathbf{u}} \tag{3.27}$$

and

$$\dot{\vec{\mathbf{h}}}_B^s = J^s \dot{\omega}^s \hat{\mathbf{w}}_B^s \tag{3.28}$$

where $\vec{\boldsymbol{\tau}}_B^{perturb}$ represents the perturbative torques acting on the spacecraft's center of mass and $\vec{\mathbf{u}}$ is the control torque about the system's center of mass. In the ORCASat, the wheel spin axis $\hat{\mathbf{w}}_B^s \equiv -\hat{\mathbf{b}}_2$ and so the wheel angular momentum and torque contributions, Equations (3.25) and (3.28) can also be written as

$$\vec{\mathbf{h}}_B^s = -J^s \omega^s \hat{\mathbf{b}}_2 = -h^s \hat{\mathbf{b}}_2 \tag{3.29}$$

$$\dot{\vec{\mathbf{h}}}_B^s = -J^s \dot{\omega}^s \hat{\mathbf{b}}_2 = -\dot{h}^s \hat{\mathbf{b}}_2 \tag{3.30}$$

## 3.4 Environmental Perturbations

The environmental perturbations are responsible for deviating the true orbit of the spacecraft from the Keplerian orbit through the perturbative forces and for changing the overall momentum of the spacecraft through the perturbative moments. According to reference [11], the perturbative forces can be divided into four classes: nongravitational forces, third-body interactions, nonspherical mass distributions, and relativistic mechanics. The main nongravitational forces are due to aerodynamic drag and solar radiation pressure. Relativistic mechanics may be completely neglected in most applications, and for spacecrafts in LEO, the third-body interactions are negligible. The nonspherical mass distribution has already been accounted for in Section 3.1. The main perturbative torques affecting a spacecraft in LEO are the gravity gradient torque, the aerodynamic and Solar Radiation Pressure (SRP) torque, and the magnetic torque [9, 57].

### 3.4.1 Aerodynamic Drag

The aerodynamic drag results from the interaction between the satellite's surface and the upper atmosphere. This force acts in the opposite direction of the spacecraft's velocity relative to the atmosphere and is responsible for generating torques on the satellite and for its orbital decay, playing an important roll on the spacecraft's mission lifetime in LEO. The drag force is proportional to the atmospheric density and so it decreases approximately exponentially with the increase in altitude. In order to model this phenomenon, several assumptions were made [57]: the momentum of molecules arriving at the body's surface is totally lost to the surface; the mean thermal motion of the atmosphere is much smaller than the speed of the spacecraft through the atmosphere; momentum transfer from molecules leaving the surface is negligible; for spinning vehicles, the relative motion between surface elements is much smaller than the speed of the mass center.

For a spacecraft modeled as $N$ flat plates (which is a very good representation of the ORCASat), the aerodynamic drag force acting on plate $i$ can be described as follows:

$$\vec{\mathbf{F}}^i_{aero} = -\frac{1}{2}\rho C_D \|\vec{\mathbf{v}}_{rel}\| S_i \vec{\mathbf{v}}_{rel} \tag{3.31}$$

with

$$S_i = A_i \hat{\mathbf{n}}_i \cdot \frac{\vec{\mathbf{v}}_{rel}}{\|\vec{\mathbf{v}}_{rel}\|} H\left(\hat{\mathbf{n}}^i \cdot \frac{\vec{\mathbf{v}}_{rel}}{\|\vec{\mathbf{v}}_{rel}\|}\right) \tag{3.32}$$

where $\rho$ is the local atmospheric density, $C_D$ is the drag coefficient, $A_i$ is the area of the $i$'th panel of the satellite and $S_i$ corresponds to the projected area of the $i$'th panel in the direction of $\vec{\mathbf{v}}_{rel}$. The Heaviside function $H(\square)$ was used so that only the panels which are facing the flow are accounted for the drag force. The velocity $\vec{\mathbf{v}}_{rel}$ is the velocity of the spacecraft with respect to the atmosphere and is given by

$$\vec{\mathbf{v}}_{rel} = \vec{\mathbf{v}}_{spacecraft} - \vec{\mathbf{v}}_{atmosphere} = \vec{\mathbf{v}}_I - \vec{\boldsymbol{\omega}}_\oplus \times \vec{\mathbf{r}}_I \tag{3.33}$$

where the assumption that the atmosphere co-rotates with the Earth was made. Vectors $\vec{\mathbf{v}}_I$ and $\vec{\mathbf{r}}_I$ are the velocity and position of the spacecraft in the ECI frame $I$ and $\vec{\boldsymbol{\omega}}_\oplus = \omega_\oplus \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top$ is the Earth's angular velocity vector in frame $I$. The total acceleration of the spacecraft's center of mass due to the aerodynamic force is then given by

$$\vec{\mathbf{a}}_{aero} = \frac{1}{m}\sum_{i=1}^{N} \vec{\mathbf{F}}^i_{aero} \tag{3.34}$$

where $m$ is the total mass of the spacecraft. The torque disturbance due to the aerodynamic drag is given by

$$\vec{\boldsymbol{\tau}}_{aero} = \sum_{i=1}^{N} \vec{\mathbf{r}}^i \times \vec{\mathbf{F}}^i_{aero} \tag{3.35}$$

where $\vec{\mathbf{r}}^i$ is the position of the center of pressure of the $i$'th plate with respect to the spacecraft's center of mass.

### 3.4.2 Solar Radiation Pressure (SRP)

The SRP affects the spacecraft through the exchange of momentum between the spacecraft and the photons incident on its surface. Unlike other perturbations, the SRP is not always present, having a contribution only when the spacecraft is out of eclipse. The model used here for the SRP considers the following assumptions [9, 57]: the Sun is the only source of radiation interacting with the spacecraft (no reflected radiation from the Earth or the Moon is considered); for spinning vehicles the relative surface velocities are negligible; the force due to thermal radiation emitted from the spacecraft is ignored and the spacecraft's self-shadowing is ignored. For the current analysis, this last hypothesis is not a major source of uncertainty since the spacecraft in study is a CubeSat without any concave surfaces which could produce self-shadowing.

Similarly to what was done for the aerodynamic drag, the spacecraft is going to be modeled as a collection of $N$ plates. The SRP force exerted on the $i$'th panel can thus be given by

$$\vec{\mathbf{F}}^i_{srp} = -p_\odot S_i \left\{ 2 \left[ \frac{\sigma^i_{diff}}{3} + \sigma^i_{spec} (\hat{\mathbf{s}}_\odot \cdot \hat{\mathbf{n}}^i) \right] \hat{\mathbf{n}}_i + (1 - \sigma^i_{spec}) \hat{\mathbf{s}}_\odot \right\} \tag{3.36}$$

with

$$S_i = A_i (\hat{\mathbf{n}}^i \cdot \hat{\mathbf{s}}_\odot) H \left( \hat{\mathbf{n}}^i \cdot \hat{\mathbf{s}}_\odot \right) \tag{3.37}$$

where $p_\odot$ is the solar radiation pressure, $\hat{\mathbf{s}}_\odot$ is the unit vector pointing from the spacecraft to the Sun, $A_i$ is the area of the $i$'th panel, and $\hat{\mathbf{n}}^i$ is its respective outward normal unit vector. Similarly to the aerodynamic drag, the Heaviside function $H(\square)$ was used so that only the panels that are visible to the Sun, $\hat{\mathbf{s}}_\odot \cdot \hat{\mathbf{n}}^i > 0$, are accounted for the SRP force. The coefficients $\sigma^i_{diff}$, $\sigma^i_{spec}$ are respectively the diffuse and specular reflection coefficients of the $i$'th panel. The absorption coefficient $\sigma^i_{abs}$ was implicitly used since all the coefficients sum to unit, $\sigma^i_{diff} + \sigma^i_{spec} + \sigma^i_{abs} = 1$. The total acceleration due to the SRP is then given by

$$\vec{\mathbf{a}}_{srp} = \frac{1}{m} \sum_{i=1}^{N} \vec{\mathbf{F}}^i_{srp} \tag{3.38}$$

where $m$ is the total mass of the spacecraft. The torque perturbation due to SRP is given by

$$\vec{\boldsymbol{\tau}}_{srp} = \sum_{i=1}^{N} \vec{\mathbf{r}}^i \times \vec{\mathbf{F}}^i_{srp} \tag{3.39}$$

where $\vec{\mathbf{r}}^i$ is the vector from the spacecraft's center of mass to the center of pressure of the SRP on the $i$'th panel.

### 3.4.3 Magnetic Torque

The Earth's magnetic field can have a significant effect on a spacecraft, specially in LEO where the field is stronger. The magnetic torque results from the interaction between the spacecraft's magnetic field and the geomagnetic field, being the spacecraft's magnetic moment usually the dominant source of this torque [11]. The magnetic disturbance torque about the spacecraft's center of mass can be described

by

$$\vec{\boldsymbol{\tau}}_{mag} = \vec{\mathbf{m}} \times \vec{\mathbf{B}} \tag{3.40}$$

where $\vec{\mathbf{m}}$ (in A m$^2$) is the net magnetic dipole moment generated by the spacecraft and $\vec{\mathbf{B}}$ is the Earth's magnetic field. The magnetic torque can be used to control the attitude of the spacecraft, by using permanent magnets or magnetorquers, but there are also magnetic dipole moments that can lead to magnetic disturbance torques. These disturbance dipole moments are usually generated by current loops present in the spacecraft or by other electronics or scientific instruments [57].

Similarly to the gravity force from Section 3.1, the Earth's magnetic field can be written as a gradient of a scalar potential function $V$ [9, 11]:

$$\vec{\mathbf{B}} = -\boldsymbol{\nabla} V \tag{3.41}$$

where the magnetic potential $V$ is given in spherical harmonics expansion in the ECEF frame by [58]

$$V = a \sum_{n=1}^{\infty} \left( \frac{a}{\|\vec{\mathbf{r}}\|} \right)^{n+1} \sum_{m=0}^{n} \bar{P}_n^m(\sin\lambda) \left[ g_n^m(t) \cos(m\phi) + h_n^m(t) \sin(m\phi) \right] \tag{3.42}$$

where $a$ is the magnetic spherical reference radius, $\lambda$ and $\phi$ are the geocentric latitude and longitude respectively and $\vec{\mathbf{r}}$ is the position of a point in space with respect to the Earth's center of mass. The coefficients $g_n^m(t)$ and $h_n^m(t)$ are the time-dependent Gauss coefficients of degree $n$ and order $m$ describing the Earth's main magnetic field and are conventionally given in nanotesla (nT). The parameters $\bar{P}_n^m$ are the Schmidt semi-normalized associated Legendre functions.

### 3.4.4 Gravity Gradient Torque

The non-symmetrical distribution of mass in a rigid body leads to a gravitational torque about the body's center of mass. This torque is called gravity gradient torque. The expression derived for the gravity gradient disturbance torque will have in consideration the following assumptions [57]: only the influence of the Earth is going to be considered; the Earth's gravity field possesses a spherically symmetrical mass distribution; the distance from the spacecraft to the center of mass of the Earth is much bigger than the size of the spacecraft itself; the spacecraft consists on a single rigid body. The first two assumptions allow us to write the gravity force as

$$\vec{\mathbf{F}}_{grav} = -\mu_{\oplus} \int \frac{\vec{\mathbf{r}}}{\|\vec{\mathbf{r}}\|^3} \, dm \tag{3.43}$$

where $\mu_{\oplus}$ is the gravitational parameter of the Earth and $\vec{\mathbf{r}}$ is the position of the mass element $dm$ with respect to the center of mass of the Earth. By considering

$$\vec{\mathbf{r}} = \vec{\mathbf{R}}^c + \vec{\mathbf{r}}\,' \tag{3.44}$$

where $\vec{\mathbf{R}}^c$ is the position of the center of mass of the spacecraft with respect to the center of mass of the Earth and $\vec{\mathbf{r}}\,'$ is the position of the element $dm$ relatively to the spacecraft's center of mass, assumption

three allows $\frac{1}{\|\vec{\mathbf{r}}\|^3}$ to be approximated as

$$\frac{1}{\|\vec{\mathbf{r}}\|^3} \simeq \frac{1}{\left\|\vec{\mathbf{R}}^c\right\|^3}\left(1 - 3\frac{\vec{\mathbf{r}}' \cdot \vec{\mathbf{R}}^c}{\left\|\vec{\mathbf{R}}^c\right\|^2}\right) \tag{3.45}$$

The torque about the spacecraft's center of mass due to gravity is given by

$$\vec{\boldsymbol{\tau}}_{gg} = \int \vec{\mathbf{r}}' \times d\vec{\mathbf{F}}_{grav} = -\mu_{\oplus} \int \frac{\vec{\mathbf{r}}' \times \vec{\mathbf{r}}}{\|\vec{\mathbf{r}}\|^3}\, dm \tag{3.46}$$

By using the results from Equations (3.45) and (3.44) in Equation (3.46)

$$\vec{\boldsymbol{\tau}}_{gg} = -\frac{\mu_{\oplus}}{\left\|\vec{\mathbf{R}}^c\right\|^3}\left(\int \vec{\mathbf{r}}'\, dm\right) \times \vec{\mathbf{R}}^c + \frac{3\mu_{\oplus}}{\left\|\vec{\mathbf{R}}^c\right\|^5}\int (\vec{\mathbf{r}}' \times \vec{\mathbf{R}}^c)(\vec{\mathbf{r}}' \cdot \vec{\mathbf{R}}^c)\, dm \tag{3.47}$$

The first term on the right-hand side is zero by definition of center of mass $\int \vec{\mathbf{r}}'\, dm = \vec{\mathbf{0}}$. The second term can be rearranged as follows:

$$\vec{\boldsymbol{\tau}}_{gg} = -\frac{3\mu_{\oplus}}{\left\|\vec{\mathbf{R}}^c\right\|^5}\vec{\mathbf{R}}^c \times \left(\int \vec{\mathbf{r}}'(\vec{\mathbf{r}}')^\top\, dm\right)\vec{\mathbf{R}}^c = \tag{3.48a}$$

$$= -\frac{3\mu_{\oplus}}{\left\|\vec{\mathbf{R}}^c\right\|^5}\vec{\mathbf{R}}^c \times \left(\int \|\vec{\mathbf{r}}'\|^2 \mathbf{I}_3\, dm - \mathbf{J}^c\right)\vec{\mathbf{R}}^c = \tag{3.48b}$$

$$= -\frac{3\mu_{\oplus}}{\left\|\vec{\mathbf{R}}^c\right\|^5}\vec{\mathbf{R}}^c \times \left(\int \|\vec{\mathbf{r}}'\|^2 \mathbf{I}_3\, dm\right)\vec{\mathbf{R}}^c + \frac{3\mu_{\oplus}}{\left\|\vec{\mathbf{R}}^c\right\|^5}\vec{\mathbf{R}}^c \times \mathbf{J}^c\vec{\mathbf{R}}^c \tag{3.48c}$$

where by recalling Equation (3.13), $\mathbf{J}^c = \int \|\vec{\mathbf{r}}'\|^2 \mathbf{I}_3 - \vec{\mathbf{r}}'(\vec{\mathbf{r}}')^\top\, dm$ is the inertia tensor of the spacecraft about its center of mass. The first term of Equation (3.48c) is zero since $\vec{\mathbf{R}}^c \times k\mathbf{I}_3\vec{\mathbf{R}}^c = k\vec{\mathbf{R}}^c \times \vec{\mathbf{R}}^c = \vec{\mathbf{0}}$. The gravity gradient torque about the spacecrafts center of mass is then given by

$$\vec{\boldsymbol{\tau}}_{gg} = \frac{3\mu_{\oplus}}{\left\|\vec{\mathbf{R}}^c\right\|^3}\hat{\mathbf{o}}_3 \times \mathbf{J}^c\hat{\mathbf{o}}_3 \tag{3.49}$$

where $\hat{\mathbf{o}}_3$ is the $z$ axis of the orbital frame $O$ defined in Section 2.1.3.

## 3.5 Linearized Model

A linearized model for the equations of motion of the ORCASat is going to be used in Chapter 5 in the development of the LQR controllers. In order to linearize the equations of motion of the satellite, an equilibrium state must be chosen about which the linearization is going to be performed. Since the nominal state is nadir pointing, the natural choice is to let the angular velocity of the satellite relative to the LVLH frame $O$, $\vec{\boldsymbol{\omega}}^{B/O}$, go to zero and by making the body frame coincide with the orbital frame so that the attitude quaternion which defines the rotation of the body frame with respect to the orbital frame,

$\mathbf{q}_O^B$, is the identity quaternion:

$$\vec{\boldsymbol{\omega}}_{B_{ref}}^{B/O} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top \qquad \mathbf{q}_{O_{ref}}^B = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^\top \tag{3.50}$$

The angular velocity of the satellite with respect to frame $O$ will then consist of a perturbation $\delta\vec{\boldsymbol{\omega}}$ from the reference $\vec{\boldsymbol{\omega}}_{B_{ref}}^{B/O}$:

$$\vec{\boldsymbol{\omega}}_B^{B/O} = \begin{bmatrix} \delta\omega_1 \\ \delta\omega_2 \\ \delta\omega_3 \end{bmatrix} + \vec{\boldsymbol{\omega}}_{B_{ref}}^{B/O} = \delta\vec{\boldsymbol{\omega}} \tag{3.51}$$

while the attitude quaternion will consist of a small rotation $\delta\boldsymbol{q}$ about the reference attitude quaternion $\mathbf{q}_{O_{ref}}^B$:

$$\mathbf{q}_O^B = \delta\boldsymbol{q} \otimes \mathbf{q}_{O_{ref}}^B = \delta\boldsymbol{q} \tag{3.52}$$

By using the definition of quaternion as stated in Equation (2.16) and using the small-angle approximation $\sin\theta \simeq \theta$ and $\cos\theta \simeq 1$, $\delta\boldsymbol{q}$ can be approximated by

$$\delta\boldsymbol{q} = \begin{bmatrix} e_1 \sin\frac{\delta\theta}{2} \\ e_2 \sin\frac{\delta\theta}{2} \\ e_3 \sin\frac{\delta\theta}{2} \\ \cos\frac{\delta\theta}{2} \end{bmatrix} \simeq \begin{bmatrix} \delta\theta_1/2 \\ \delta\theta_2/2 \\ \delta\theta_3/2 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\delta\vec{\boldsymbol{\theta}} \\ 1 \end{bmatrix} = \begin{bmatrix} \delta\vec{\boldsymbol{q}} \\ 1 \end{bmatrix} \tag{3.53}$$

where $\delta\theta_i/2 = e_i(\delta\theta/2)$. By using Equations (3.51), (3.52) and (3.53) together in Equation (3.9) and disregarding the second-order terms, the linearized kinematics are given by

$$\dot{\mathbf{q}}_O^B = \begin{bmatrix} \delta\dot{\vec{\boldsymbol{q}}} \\ 0 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} \delta\vec{\boldsymbol{\omega}} \\ 0 \end{bmatrix} \tag{3.54}$$

The first step to linearize the dynamics equation, Eq. (3.26), consists in relating the angular velocity of the spacecraft $\vec{\boldsymbol{\omega}}^{B/I}$ with the perturbed angular velocity $\delta\vec{\boldsymbol{\omega}}$ from Equation (3.51). The inertial angular velocity of the spacecraft in the body frame $\vec{\boldsymbol{\omega}}_B^{B/I}$ can be expressed as

$$\vec{\boldsymbol{\omega}}_B^{B/I} = \vec{\boldsymbol{\omega}}_B^{B/O} + \vec{\boldsymbol{\omega}}_B^{O/I} \tag{3.55}$$

For linearization purposes, the angular velocity of the orbit frame with respect to the inertial frame, $\vec{\boldsymbol{\omega}}_O^{O/I}$, is going to be considered as constant:

$$\vec{\boldsymbol{\omega}}_O^{O/I} \simeq \begin{bmatrix} 0 \\ -\omega_0 \\ 0 \end{bmatrix} \tag{3.56}$$

where $\omega_0 = 2\pi/T_{orb}$ is the average value of the orbital rate. This is a good approximation since the orbit considered (see Chapter 7) is almost circular, having an eccentricity value of $8.3 \times 10^{-4}$. The relationship between $\vec{\boldsymbol{\omega}}_B^{O/I}$ and $\vec{\boldsymbol{\omega}}_O^{O/I}$ is done using the DCM $\mathbf{A}_O^B$ which represents the rotation of the body frame with

respect to the orbit frame as follows

$$\vec{\boldsymbol{\omega}}_B^{O/I} = \mathbf{A}_O^B \vec{\boldsymbol{\omega}}_O^{O/I} \tag{3.57}$$

The rotation matrix $\mathbf{A}_O^B$ can also be written as a small rotation about a reference rotation matrix like done previously with the attitude quaternion in Equation (3.52):

$$\mathbf{A}_O^B = (\delta\mathbf{A})\mathbf{A}_{O_{ref}}^B = (\delta\mathbf{A})\mathbf{I}_3 = \delta\mathbf{A} \tag{3.58}$$

The matrix $\delta\mathbf{A}$ can be obtained by substituting the attitude quaternion from Equation (3.53) in Equation (2.36). Disregarding second-order terms it gives

$$\delta\mathbf{A} \simeq \begin{bmatrix} 1 & 2\delta q_3 & -2\delta q_2 \\ -2\delta q_3 & 1 & 2\delta q_1 \\ 2\delta q_2 & -2\delta q_1 & 1 \end{bmatrix} = \mathbf{I}_3 - 2\Big[\delta\vec{\boldsymbol{q}}\times\Big] \tag{3.59}$$

where $\delta q_i$, $i = 1, 2, 3$ are the components of $\delta\vec{\boldsymbol{q}}$. The angular velocity $\vec{\boldsymbol{\omega}}_B^{O/I}$ can thus be approximated by

$$\vec{\boldsymbol{\omega}}_B^{O/I} = \mathbf{A}_O^B \vec{\boldsymbol{\omega}}_O^{O/I} \simeq \delta\mathbf{A} \begin{bmatrix} 0 \\ -\omega_0 \\ 0 \end{bmatrix} \simeq \begin{bmatrix} -2\omega_0\delta q_3 \\ -\omega_0 \\ 2\omega_0\delta q_1 \end{bmatrix} \tag{3.60}$$

By replacing Equations (3.51) and (3.60) in Equation (3.55), the inertial angular velocity of the body in body-frame coordinates is given by

$$\vec{\boldsymbol{\omega}}_B^{B/I} = \delta\vec{\boldsymbol{\omega}} + \begin{bmatrix} -2\omega_0\delta q_3 \\ -\omega_0 \\ 2\omega_0\delta q_1 \end{bmatrix} \tag{3.61}$$

By replacing Equation (3.61) in the dynamics equation, Eq. (3.26), and by setting $\dot{\vec{\mathbf{h}}}^s = 0$, since in nominal mode the momentum wheel will be spinning at a fixed rate, $\dot{\omega} = 0$, the dynamics equation is thus given by

$$\mathbf{J}\left(\delta\dot{\vec{\boldsymbol{\omega}}} + \begin{bmatrix} -2\omega_0\delta\dot{q_3} \\ 0 \\ 2\omega_0\delta\dot{q_1} \end{bmatrix}\right) = \left[\mathbf{J}\left(\delta\vec{\boldsymbol{\omega}} + \begin{bmatrix} -2\omega_0\delta q_3 \\ -\omega_0 \\ 2\omega_0\delta q_1 \end{bmatrix}\right) + \vec{\mathbf{h}}^s\right] \times \left[\delta\vec{\boldsymbol{\omega}} + \begin{bmatrix} -2\omega_0\delta q_3 \\ -\omega_0 \\ 2\omega_0\delta q_1 \end{bmatrix}\right] + \vec{\mathbf{u}} \tag{3.62}$$

By expanding the different terms from Equation (3.62) and by using the linearized kinematics equation,

Eq. (3.54), in Equation (3.62), the dynamics equation can be written as

$$\mathbf{J}\delta\dot{\vec{\boldsymbol{\omega}}} + \mathbf{J}\begin{bmatrix} -\omega_0\delta\omega_3 \\ 0 \\ \omega_0\delta\omega_1 \end{bmatrix} = (\mathbf{J}\delta\vec{\boldsymbol{\omega}}) \times \delta\vec{\boldsymbol{\omega}} + (\mathbf{J}\delta\vec{\boldsymbol{\omega}}) \times \begin{bmatrix} -2\omega_0\delta q_3 \\ -\omega_0 \\ 2\omega_0\delta q_1 \end{bmatrix} + \left( \mathbf{J}\begin{bmatrix} -2\omega_0\delta q_3 \\ -\omega_0 \\ 2\omega_0\delta q_1 \end{bmatrix} \right) \times \delta\vec{\boldsymbol{\omega}} + \\ + \left( \mathbf{J}\begin{bmatrix} -2\omega_0\delta q_3 \\ -\omega_0 \\ 2\omega_0\delta q_1 \end{bmatrix} \right) \times \begin{bmatrix} -2\omega_0\delta q_3 \\ -\omega_0 \\ 2\omega_0\delta q_1 \end{bmatrix} + \vec{\mathbf{h}}^s \times \delta\vec{\boldsymbol{\omega}} + \vec{\mathbf{h}}^s \times \begin{bmatrix} -2\omega_0\delta q_3 \\ -\omega_0 \\ 2\omega_0\delta q_1 \end{bmatrix} + \vec{\mathbf{u}} \quad (3.63)$$

By using $\mathbf{J} = \operatorname{diag}\begin{pmatrix} J_1 & J_2 & J_3 \end{pmatrix}$ as the inertia tensor of the spacecraft whose principal moments of inertia are aligned with the body frame $B$, then, by developing all the cross products and disregarding second-order terms one gets

$$\mathbf{J}\delta\dot{\vec{\boldsymbol{\omega}}} = \omega_0\mathbf{J}\begin{bmatrix} \delta\omega_3 \\ 0 \\ -\delta\omega_1 \end{bmatrix} + \omega_0\begin{bmatrix} J_3\delta\omega_3 \\ 0 \\ -J_1\delta\omega_1 \end{bmatrix} + \omega_0\begin{bmatrix} -J_2\delta\omega_3 \\ 0 \\ J_2\delta\omega_1 \end{bmatrix} + 2\omega_0^2\begin{bmatrix} -K_xJ_1\delta q_1 \\ 0 \\ K_zJ_3\delta q_3 \end{bmatrix} + h^s\begin{bmatrix} -\delta\omega_3 \\ 0 \\ \delta\omega_1 \end{bmatrix} - 2\omega_0h^s\begin{bmatrix} \delta q_1 \\ 0 \\ \delta q_3 \end{bmatrix} + \vec{\mathbf{u}} \quad (3.64)$$

where $K_x = \dfrac{J_2 - J_3}{J_1}$ and $K_z = \dfrac{J_1 - J_2}{J_3}$ and $h^s$ is defined in Equation (3.29). Equation (3.64) can be further simplified by multiplying all terms by $\mathbf{J}^{-1}$, giving

$$\delta\dot{\vec{\boldsymbol{\omega}}} = \omega_0\begin{bmatrix} \delta\omega_3 \\ 0 \\ -\delta\omega_1 \end{bmatrix} - \omega_0\begin{bmatrix} K_x\delta\omega_3 \\ 0 \\ K_z\delta\omega_1 \end{bmatrix} + 2\omega_0^2\begin{bmatrix} -K_x\delta q_1 \\ 0 \\ K_z\delta q_3 \end{bmatrix} + h^s\begin{bmatrix} -\delta\omega_3/J_1 \\ 0 \\ \delta\omega_1/J_3 \end{bmatrix} - 2\omega_0h^s\begin{bmatrix} \delta q_1/J_1 \\ 0 \\ \delta q_3/J_3 \end{bmatrix} + \vec{\mathbf{u}}' \quad (3.65)$$

$$\Leftrightarrow \delta\dot{\vec{\boldsymbol{\omega}}} = \begin{bmatrix} [w_0(1 - K_x) - h^s/J_1]\delta\omega_3 - 2\omega_0(\omega_0K_x + h^s/J_1)\delta q_1 \\ 0 \\ [h^s/J_3 - \omega_0(1 + K_z)]\delta\omega_1 + 2\omega_0(\omega_0K_z - h^s/J_3)\delta q_3 \end{bmatrix} + \vec{\mathbf{u}}' \quad (3.66)$$

where $\vec{\mathbf{u}}' = \mathbf{J}^{-1}\vec{\mathbf{u}}$. By defining the state vector $\mathbf{x}$ as

$$\mathbf{x} = \begin{bmatrix} \delta\omega_1 & \delta\omega_2 & \delta\omega_3 & \delta q_1 & \delta q_2 & \delta q_3 \end{bmatrix}^\top \quad (3.67)$$

the linearized equations of motion of the satellite are written as

$$\dot{\mathbf{x}} = \boldsymbol{A}\mathbf{x} + \begin{bmatrix} \mathbf{J}^{-1}\vec{\mathbf{u}} \\ \mathbf{0}_{3\times3} \end{bmatrix} \quad (3.68)$$

where $\mathbf{0}_{3\times3}$ is a $3 \times 3$ matrix fully populated with zeros. The state matrix $\boldsymbol{A}$ is given by

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & w_0\left(1-K_x\right) - \dfrac{h^s}{J_1} & -2\omega_0\left(\omega_0 K_x + \dfrac{h^s}{J_1}\right) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \dfrac{h^s}{J_3} - \omega_0\left(1+K_z\right) & 0 & 0 & 0 & 0 & 2\omega_0\left(\omega_0 K_z - \dfrac{h^s}{J_3}\right) \\ 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 \end{bmatrix} \tag{3.69}$$

The control torque $\vec{\mathbf{u}}$ is given by

$$\vec{\mathbf{u}} = \vec{\mathbf{m}}_{ctrl} \times \vec{\mathbf{B}}_B = \vec{\mathbf{m}}_{ctrl} \times \left(\vec{\mathbf{A}}_O^B \vec{\mathbf{B}}_O\right) \tag{3.70}$$

where $\vec{\mathbf{m}}_{ctrl}$ is the dipole moment generated by the magnetorquers, $\vec{\mathbf{B}}_B$ is the local geomagnetic field in body frame coordinates and $\vec{\mathbf{B}}_O$ is the local geomagnetic field is the orbit frame. By using Equation (3.58) the previous equation writes:

$$\vec{\mathbf{u}} = \vec{\mathbf{m}}_{ctrl} \times (\vec{\mathbf{B}}_O - 2\delta\vec{\boldsymbol{q}} \times \vec{\mathbf{B}}_O) \simeq \vec{\mathbf{m}}_{ctrl} \times \vec{\mathbf{B}}_O \tag{3.71}$$

where the second equality is true because when the linear feedback of the form $\vec{\mathbf{m}}_{ctrl} = -\boldsymbol{K}'\mathbf{x}$ is implemented, with $\boldsymbol{K}'$ the $3 \times 6$ gain matrix, the term $-2\vec{\mathbf{m}}_{ctrl} \times (\delta\vec{\boldsymbol{q}} \times \vec{\mathbf{B}}_O)$ becomes a second-order term.

# Chapter 4

# Attitude Determination

In this chapter, the different algorithms related to the estimation of the attitude of the ORCASat are outlined. In Section 4.1, Wahba's problem is presented and solved using the quaternion parametrization by means of the Quaternion Estimator (QUEST) algorithm. The Method of Sequential Rotations (MSR) used to overcome the singularity problem of the QUEST algorithm is presented as well as the solution for the Wahba's problem only using two vector observations which are related to the set of sensors present in the ORCASat. In Section 4.2, the EKF which will be the base for the following two algorithms is reviewed. In Section 4.2.1, the Multiplicative Extended Kalman Filter (MEKF) is developed, and finally, in Section 4.2.2, the Magnetometer Calibration Extended Kalman Filter (MCEKF) is presented. This last estimation filter will be used to provide a real-time calibration of the magnetometer of the ORCASat in an attempt to improve the performance of the QUEST algorithm and the performance of the MEKF. The flowcharts corresponding to the implementation of the different algorithms are present in Appendix A.

## 4.1 Quaternion Estimator (QUEST)

Wahba's problem tries to find the orthogonal matrix $\mathbf{A}$ with determinant $+1$ that minimizes the cost function [9]

$$J(\mathbf{A}) = \frac{1}{2}\sum_{i=1}^{N} a_i \left\| \vec{\mathbf{b}}_i - \mathbf{A}\vec{\mathbf{r}}_i \right\|^2 \tag{4.1}$$

This least squares formula tries to find the best fit for the attitude matrix $\mathbf{A}$ given the set of $N$ independent unit vectors $\vec{\mathbf{b}}_i$ measured in a spacecraft's body frame and the corresponding unit vectors $\vec{\mathbf{r}}_i$ given in a reference frame. The scalar values $a_i$ are non-negative weights. Wahba's Problem has different ways to be solved comprising two different classes of problems, the ones that solve the attitude matrix directly and the ones that solve the quaternion representation of the attitude matrix. Since the main attitude representation for the ORCASat is the quaternion representation, the method chosen was the QUEST algorithm. This method was also chosen due to the fact that it is the most widely used algorithm for solving Wahba's problem [21].

Equation (4.1) can be expressed in a more convenient form by expanding the quadratic term $\left\| \vec{\mathbf{b}}_i - \mathbf{A}\vec{\mathbf{r}}_i \right\|^2$.

Taking in consideration that vectors $\vec{\mathbf{r}}_i$ and $\vec{\mathbf{b}}_i$ are unit vectors and that matrix $\mathbf{A}$ is orthogonal, the quadratic term can be simplified as

$$\left\|\vec{\mathbf{b}}_i - \mathbf{A}\vec{\mathbf{r}}_i\right\|^2 = \left(\vec{\mathbf{b}}_i - \mathbf{A}\vec{\mathbf{r}}_i\right)^\top \left(\vec{\mathbf{b}}_i - \mathbf{A}\vec{\mathbf{r}}_i\right) = (\vec{\mathbf{b}}_i)^\top\vec{\mathbf{b}}_i + (\mathbf{A}\vec{\mathbf{r}}_i)^\top\mathbf{A}\vec{\mathbf{r}}_i - (\vec{\mathbf{b}}_i)^\top\mathbf{A}\vec{\mathbf{r}}_i - (\mathbf{A}\vec{\mathbf{r}}_i)^\top\vec{\mathbf{b}}_i = 2 - 2(\vec{\mathbf{b}}_i)^\top\mathbf{A}\vec{\mathbf{r}}_i \tag{4.2}$$

where the last equality is true due to the fact that the product $(\mathbf{A}\vec{\mathbf{r}}_i)^\top\vec{\mathbf{b}}_i$ is a scalar and the transpose of a scalar is equal to that same scalar. By substituting Equation (2.36) in Equation (4.2) and using the results from Equations (2.35) and (2.20), the last term of Equation (4.2) can be written in an explicit quaternion form as follows:

$$(\vec{\mathbf{b}}_i)^\top\mathbf{\Xi}^\top(\mathbf{q})\mathbf{\Psi}(\mathbf{q})\vec{\mathbf{r}}_i = \left(\mathbf{\Xi}(\mathbf{q})\vec{\mathbf{b}}_i\right)^\top\mathbf{\Psi}(\mathbf{q})\vec{\mathbf{r}}_i = (\mathbf{q}\odot\mathbf{b}_i)^\top(\mathbf{q}\otimes\mathbf{r}_i) = (\mathbf{b}_i\otimes\mathbf{q})^\top(\mathbf{r}_i\odot\mathbf{q}) = \mathbf{q}^\top[\mathbf{b}_i\otimes]^\top[\mathbf{r}_i\odot]\mathbf{q} \tag{4.3}$$

where $\mathbf{b}_i = \begin{bmatrix}\vec{\mathbf{b}}_i \\ 0\end{bmatrix}$ and $\mathbf{r}_i = \begin{bmatrix}\vec{\mathbf{r}}_i \\ 0\end{bmatrix}$. Equation (4.1) can thus be rewritten in quaternion form as

$$J(\mathbf{q}) = \lambda_0 - \mathbf{q}^\top\mathbf{K}\mathbf{q} \tag{4.4}$$

where

$$\lambda_0 = \sum_{i=1}^{N} a_i \tag{4.5}$$

and $\mathbf{K}$ is a symmetric traceless matrix given by

$$\mathbf{K} = \begin{bmatrix}\mathbf{B} + \mathbf{B}^\top - (\operatorname{tr}\mathbf{B})\mathbf{I}_3 & \vec{\mathbf{z}} \\ \vec{\mathbf{z}}^\top & \operatorname{tr}\mathbf{B}\end{bmatrix} \tag{4.6}$$

with $\mathbf{B}$, the attitude profile matrix, and the vector $\vec{\mathbf{z}}$ given by

$$\mathbf{B} = \sum_{i=1}^{N} a_i\vec{\mathbf{b}}_i\vec{\mathbf{r}}_i^\top \qquad\qquad \vec{\mathbf{z}} = \begin{bmatrix}B_{23} - B_{32} \\ B_{31} - B_{13} \\ B_{12} - B_{21}\end{bmatrix} \tag{4.7}$$

where $B_{ij}$ is the element from the $i$'th row and $j$'th column of matrix $\mathbf{B}$. The cost function from Equation (4.4) is minimized if $\mathbf{q}^\top\mathbf{K}\mathbf{q}$ is maximized. By using the Lagrange multiplier to append the quaternion unit norm constrain to the cost function $J(\mathbf{q})$, the function to maximize is the following:

$$g(\mathbf{q}) = \mathbf{q}^\top\mathbf{K}\mathbf{q} + \lambda(1 - \mathbf{q}^\top\mathbf{q}) \tag{4.8}$$

In order to find an extreme of Equation (4.8) one can take its derivative with respect to $\mathbf{q}$

$$\frac{\mathrm{d}g(\mathbf{q})}{\mathrm{d}\mathbf{q}} = 0 \Leftrightarrow \mathbf{K}\mathbf{q} = \lambda\mathbf{q} \tag{4.9}$$

which is the know eigenvalue/eigenvector problem applied to the $\mathbf{K}$ matrix:

$$(\mathbf{K} - \lambda \mathbf{I}_4)\mathbf{q} = 0 \tag{4.10}$$

with $\mathbf{I}_4$ defining the $4 \times 4$ identity matrix. By replacing Equation (4.9) in Equation (4.8), function $g(\mathbf{q})$ can be rewritten as

$$g(\mathbf{q}) = \mathbf{q}^\top \lambda \mathbf{q} + \lambda(1 - \mathbf{q}^\top \mathbf{q}) = \lambda \tag{4.11}$$

Equations (4.9) and (4.11) show that the estimated quaternion is the normalized eigenvector of $\mathbf{K}$ that corresponds to the largest eigenvalue. This method of solving Wahba's problem, by computing the largest eigenvalue of $\mathbf{K}$ and by normalizing the respective eigenvector, is known as the q-method and was the first useful solution of Wahba's problem for spacecraft attitude determination [9].

The QUEST algorithm on the other hand tries to solve Wahba's problem without solving the eigenvalue/eigenvector problem. This method should be faster than the q-method although with some accuracy loss. Equation (4.9) is equivalent to the following equations

$$(\rho \mathbf{I}_3 - \mathbf{S})\vec{\mathbf{q}} = q_4 \vec{\mathbf{z}} \tag{4.12a}$$

$$(\lambda_{max} - \operatorname{tr}\mathbf{B})q_4 - (\vec{\mathbf{z}})^\top \vec{\mathbf{q}} = 0 \tag{4.12b}$$

where $\lambda_{max}$ is the maximum eigenvalue of $\mathbf{K}$, $\rho = \lambda_{max} + \operatorname{tr}\mathbf{B}$, $\mathbf{S} = \mathbf{B} + \mathbf{B}^\top$ and $\vec{\mathbf{q}}$ is the vector part of the attitude quaternion $\mathbf{q}$. Using the unit property of the attitude quaternion $\|\vec{\mathbf{q}}\|^2 + q_4^2 = 1$ together with Equation (4.12a), it is possible to obtain the optimal attitude quaternion as function of $\lambda_{max}$. Equation (4.12a) can be rewritten as

$$\vec{\mathbf{q}} = \frac{q_4}{\det(\rho \mathbf{I}_3 - \mathbf{S})}[\operatorname{adj}(\rho \mathbf{I}_3 - \mathbf{S})]\vec{\mathbf{z}} \tag{4.13}$$

By using the previous equation together with the unit norm constrain condition of the attitude quaternion, it is possible to write

$$\|\vec{\mathbf{q}}\|^2 + q_4^2 = 1 = \frac{q_4^2}{\det^2(\rho \mathbf{I}_3 - \mathbf{S})}\|[\operatorname{adj}(\rho \mathbf{I}_3 - \mathbf{S})]\vec{\mathbf{z}}\|^2 + q_4^2 \tag{4.14}$$

By solving Equation (4.14) in order to $q_4$:

$$q_4 = \frac{\det(\rho \mathbf{I}_3 - \mathbf{S})}{\sqrt{\det^2(\rho \mathbf{I}_3 - \mathbf{S}) + \|[\operatorname{adj}(\rho \mathbf{I}_3 - \mathbf{S})]\vec{\mathbf{z}}\|^2}} \tag{4.15}$$

The expression for the attitude quaternion $\mathbf{q}$ is then given by using Equation (4.15) in Equation (4.13)

$$\mathbf{q} = \frac{1}{\alpha}\begin{bmatrix} \operatorname{adj}(\rho \mathbf{I}_3 - \mathbf{S})\vec{\mathbf{z}} \\ \det(\rho \mathbf{I}_3 - \mathbf{S}) \end{bmatrix} \tag{4.16}$$

where $\alpha = \sqrt{\det^2(\rho \mathbf{I}_3 - \mathbf{S}) + \|[\operatorname{adj}(\rho \mathbf{I}_3 - \mathbf{S})]\vec{\mathbf{z}}\|^2}$ and is usually determined by normalization of $\mathbf{q}$. Equation (4.16) shows that if $\lambda_{max}$ is known the optimal attitude quaternion can be computed. Using Equation

(4.16) in Equation (4.12b) yields

$$(\lambda_{max} - \operatorname{tr}\mathbf{B})\det(\rho\mathbf{I}_3 - \mathbf{S}) - \vec{\mathbf{z}}^\top \operatorname{adj}(\rho\mathbf{I}_3 - \mathbf{S})\vec{\mathbf{z}} = 0 \tag{4.17}$$

Equation (4.17) can be converted into an explicit equation for $\lambda_{max}$ by applying some algebra and by using the definition of the adjoint and the determinant applied to the matrix $\rho\mathbf{I}_3 - \mathbf{S}$ as explained in [9]. Equation (4.17) can thus be rewritten as:

$$\begin{aligned} 0 = \psi_{QUEST}(\lambda) = [\lambda^2 - (\operatorname{tr}\mathbf{B})^2 + \operatorname{tr}(\operatorname{adj}\mathbf{S})][\lambda^2 - (\operatorname{tr}\mathbf{B})^2 - \|\vec{\mathbf{z}}\|^2] - \\ - (\lambda - \operatorname{tr}\mathbf{B})(\vec{\mathbf{z}}^\top\mathbf{S}\vec{\mathbf{z}} + \det\mathbf{S}) - \vec{\mathbf{z}}^\top\mathbf{S}^2\vec{\mathbf{z}} \end{aligned} \tag{4.18}$$

The value of $\lambda_{max}$ can be obtained by solving Equation (4.18) with the Newton-Raphson method using as first estimation $\lambda_{max} = \lambda_0$ where $\lambda_0$ follows from Equation (4.5). The QUEST algorithm presented has a singularity when $\alpha = 0$, or equivalently when $\rho\mathbf{I}_3 - \mathbf{S}$ is singular. This happens when the estimated attitude quaternion corresponds to a $180^\circ$ rotation, i.e. $q_4 = 0$ [9, 18].

**Method of Sequential Rotations (MSR)**

The method of sequential rotations allows us to overcome the singularity problem of QUEST by computing the attitude with respect to a rotated reference frame $I_k$, where $k$ means the rotation about the $k$'th coordinate axis of the original reference frame $I$. This rotation is implemented by writing the coordinates of the reference vectors $\vec{\mathbf{r}}_i$ in the new frame $I_k$ by simply multiplying them by the appropriate rotation matrix $\mathbf{A}_I^{I_k}$. After solving the attitude with respect to the rotated reference frame $I_k$, an inverse rotation is performed to obtain the attitude in the desired frame. For simplicity reasons, the frames are rotated $180^\circ$ about one of the coordinate axes. The relation between the desired attitude quaternion $\mathbf{q} \equiv \mathbf{q}_I^B$ and the solution obtained in the rotated reference frame $\mathbf{q}^k \equiv \mathbf{q}_{I_k}^B$ is given by

$$\mathbf{q} \equiv \mathbf{q}_I^B = \mathbf{q}_{I_k}^B \otimes \mathbf{q}_I^{I_k} = \mathbf{q}^k \otimes \mathbf{q}(\vec{\mathbf{e}}_k, \pi) = \begin{bmatrix} \vec{\mathbf{q}}^k \\ q_4^k \end{bmatrix} \otimes \begin{bmatrix} \vec{\mathbf{e}}_k \\ 0 \end{bmatrix} = \begin{bmatrix} q_4^k\vec{\mathbf{e}}_k + \vec{\mathbf{e}}_k \times \vec{\mathbf{q}}^k \\ -\vec{\mathbf{q}}^k \cdot \vec{\mathbf{e}}_k \end{bmatrix} = \mathbf{E}_k\mathbf{q}^k \tag{4.19}$$

where $\vec{\mathbf{e}}_k$ is the unit vector of the $k$'th coordinate axis of $I$ and $\mathbf{E}_k$ is a skew-symmetric orthogonal matrix given by

$$\mathbf{E}_k = \begin{bmatrix} [\vec{\mathbf{e}}_k\times] & \vec{\mathbf{e}}_k \\ -\vec{\mathbf{e}}_k^\top & 0 \end{bmatrix} = \begin{bmatrix} 0 & -e_3 & e_2 & e_1 \\ e_3 & 0 & -e_1 & e_2 \\ -e_2 & e_1 & 0 & e_3 \\ -e_1 & -e_2 & -e_3 & 0 \end{bmatrix} \tag{4.20}$$

The inverse relation is simply given by

$$\mathbf{q}^k = \mathbf{E}_k^\top \mathbf{q} = \begin{bmatrix} 0 & e_3 & -e_2 & -e_1 \\ -e_3 & 0 & e_1 & -e_2 \\ e_2 & -e_1 & 0 & -e_3 \\ e_1 & e_2 & e_3 & 0 \end{bmatrix} \mathbf{q} \tag{4.21}$$

It can be seen from Equation (4.21) that the $k$'th component of the attitude quaternion $\mathbf{q}$ ends up as the fourth component of $\mathbf{q}^k$. Since the QUEST algorithm works better if the forth component of the attitude quaternion being computed is the further away from zero, the method of sequential rotations allows to find the reference frame in which $q_4$ is at least $1/2$ due to the unit norm of the attitude quaternion. The optimal reference frame rotation is the one about the axis corresponding to the component of $\mathbf{q}$ having the largest magnitude. If the largest component of $\mathbf{q}$ is the fourth, i.e., the scalar one, thus no rotation is needed to be performed. This selection can be performed by using a previously computed quaternion. The implementation of the QUEST algorithm together with the MSR is summarized in Table 4.1.

Table 4.1: QUEST algorithm implementation including the MSR.

| |
|---|
| Normalization of input vectors $\vec{\mathbf{b}}_i$ and $\vec{\mathbf{r}}_i$ |
| Determination of the component of the previously computed quaternion with the largest magnitude - MSR |
| If necessary, apply the $180^\circ$ rotation about the $k$'th coordinate axis of $I$, using $\mathbf{A}_I^{I_k}$, to the reference frame vectors $\vec{\mathbf{r}}_i$ - MSR |
| Construction of the matrix $\mathbf{B}$ and of the vector $\vec{\mathbf{z}}$ using Equation (4.7) |
| Construction of the matrix $\mathbf{S}$: $\mathbf{S} = \mathbf{B} + \mathbf{B}^\top$ |
| Determination of $\lambda_0$ using Equation (4.5) |
| Newton-Raphson iteration of Equation (4.18) to find $\lambda_{max}$ using $\lambda_0$ as initial condition |
| Calculation of the attitude quaternion using Equation (4.16) (where $\rho = \lambda_{max} + \operatorname{tr}\mathbf{B}$) |
| Calculation of the attitude quaternion in the original frame by undoing the initial rotation using Equation (4.19) - MSR |

**Two vector observations**

In the case of the ORCASat, only two observations will be available at the same time, the sun vector and the magnetic field vector. This provides some simplifications to the previous equations and a simple closed-form solution for the characteristic equation of the $\mathbf{K}$ matrix. The solution for Wahba's problem using the two vector observations case also provides a generalization of the TRIAD algorithm using arbitrary measurement weights. It is possible to show that the maximum eigenvalue $\lambda_{max}$ can be given by [9]

$$\lambda_{max} = \sqrt{a_1^2 + a_2^2 + 2a_1a_2\left[(\vec{\mathbf{b}}_1 \cdot \vec{\mathbf{b}}_2)(\vec{\mathbf{r}}_1 \cdot \vec{\mathbf{r}}_2) + \left\|\vec{\mathbf{b}}_1 \times \vec{\mathbf{b}}_2\right\|\|\vec{\mathbf{r}}_1 \times \vec{\mathbf{r}}_2\|\right]} \tag{4.22}$$

This equation avoids the need of using the Newton-Raphson algorithm to obtained the value of $\lambda_{max}$. The closed form solution for the attitude quaternion is given by [9]

$$\mathbf{q} = \begin{cases} \dfrac{1}{2\sqrt{\gamma(\gamma+\alpha)\left(1+\vec{\mathbf{b}}_3\cdot\vec{\mathbf{r}}_3\right)}} \begin{bmatrix} (\gamma+\alpha)\left(\vec{\mathbf{b}}_3\times\vec{\mathbf{r}}_3\right)+\beta\left(\vec{\mathbf{b}}_3+\vec{\mathbf{r}}_3\right) \\ (\gamma+\alpha)\left(1+\vec{\mathbf{b}}_3\cdot\vec{\mathbf{r}}_3\right) \end{bmatrix} & \text{for} \quad \alpha \geqslant 0 \\ \dfrac{1}{2\sqrt{\gamma(\gamma-\alpha)\left(1+\vec{\mathbf{b}}_3\cdot\vec{\mathbf{r}}_3\right)}} \begin{bmatrix} \beta\left(\vec{\mathbf{b}}_3\times\vec{\mathbf{r}}_3\right)+(\gamma-\alpha)\left(\vec{\mathbf{b}}_3+\vec{\mathbf{r}}_3\right) \\ \beta\left(1+\vec{\mathbf{b}}_3\cdot\vec{\mathbf{r}}_3\right) \end{bmatrix} & \text{for} \quad \alpha < 0 \end{cases} \tag{4.23}$$

where $\vec{\mathbf{b}}_3$, $\vec{\mathbf{r}}_3$, $\alpha$, $\beta$ and $\gamma$ are given respectively by

$$\vec{\mathbf{b}}_3 = \frac{\vec{\mathbf{b}}_1\times\vec{\mathbf{b}}_2}{\left\|\vec{\mathbf{b}}_1\times\vec{\mathbf{b}}_2\right\|} \tag{4.24}$$

$$\vec{\mathbf{r}}_3 = \frac{\vec{\mathbf{r}}_1\times\vec{\mathbf{r}}_2}{\|\vec{\mathbf{r}}_1\times\vec{\mathbf{r}}_2\|} \tag{4.25}$$

$$\alpha = \left(1+\vec{\mathbf{b}}_3\cdot\vec{\mathbf{r}}_3\right)\left(a_1\vec{\mathbf{b}}_1\cdot\vec{\mathbf{r}}_1+a_2\vec{\mathbf{b}}_2\cdot\vec{\mathbf{r}}_2\right)+\left(\vec{\mathbf{b}}_3\times\vec{\mathbf{r}}_3\right)\cdot\left(a_1\vec{\mathbf{b}}_1\times\vec{\mathbf{r}}_1+a_2\vec{\mathbf{b}}_2\times\vec{\mathbf{r}}_2\right) \tag{4.26}$$

$$\beta = \left(\vec{\mathbf{b}}_3+\vec{\mathbf{r}}_3\right)\cdot\left(a_1\vec{\mathbf{b}}_1\times\vec{\mathbf{r}}_1+a_2\vec{\mathbf{b}}_2\times\vec{\mathbf{r}}_2\right) \tag{4.27}$$

$$\gamma = \sqrt{\alpha^2+\beta^2} \tag{4.28}$$

When $\vec{\mathbf{b}}_3\cdot\vec{\mathbf{r}}_3 = -1$, Equation (4.23) does not have a solution. This problem can be solved by using the MSR where for the case of the two vector observations, a 180° rotation about the $k$'th axis of the reference frame gives

$$\left(\vec{\mathbf{b}}_3\cdot\vec{\mathbf{r}}_3^{\,rotated}\right) = 2\left(\vec{\mathbf{b}}_3\right)_k\left(\vec{\mathbf{r}}_3^{\,unrotated}\right)_k - \vec{\mathbf{b}}_3\cdot\vec{\mathbf{r}}_3^{\,unrotated} \tag{4.29}$$

where $(\vec{\mathbf{b}}_3)_k$ and $(\vec{\mathbf{r}}_3^{\,unrotated})_k$ are the $k$'th component of vectors $\vec{\mathbf{b}}_3$ and $\vec{\mathbf{r}}_3^{\,unrotated}$ respectively. From Equation (4.29), it is possible to see that if $\left(\vec{\mathbf{b}}_3\cdot\vec{\mathbf{r}}_3^{\,unrotated}\right)$ is greater than any of the other products $\left(\vec{\mathbf{b}}_3\right)_k\left(\vec{\mathbf{r}}_3^{\,unrotated}\right)_k$ there is no need for a rotation, otherwise, in order to ensure the largest value for $\left(\vec{\mathbf{b}}_3\cdot\vec{\mathbf{r}}_3^{\,rotated}\right)$, a rotation is performed about the $k$'th axis of the reference frame where the value of $\left(\vec{\mathbf{b}}_3\right)_k\left(\vec{\mathbf{r}}_3^{\,unrotated}\right)_k$ is greater.

The implementation of the QUEST algorithm using two vector observations and the MSR is summarized in Appendix A under the form of a flowchart by Figure A.5.

## 4.2 Extended Kalman Filter (EKF)

In Kalman Filtering, the system in study is assumed to be described by a linear model. In practice, however, not every system can be described by using a linear model. There are different methods to tackle this problem, being a common approach to use the EKF, which consists of representing the error dynamics by a linearized first-order Taylor series expansion about the system's current best state estimate, which is considered to be sufficiently close to the true state. The main attitude estimator which will operate in the ORCASat will be the MEKF which as the name implies is a variant of the EKF.

Consider the following system truth model with discrete-time measurements given by

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{G}(t)\mathbf{w}(t), \quad \mathbf{w}(t) \sim \mathcal{N}(0, \mathbf{Q}(t)) \tag{4.30}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k, \quad \mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k) \tag{4.31}$$

where $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$ is a sufficiently differentiable $n \times 1$ vector function and $\mathbf{h}(\mathbf{x}_k)$ is the $m \times 1$ observation vector function, $\mathbf{x}(t)$ is the $n \times 1$ state vector and $\mathbf{y}(t)$ is the $m \times 1$ measurement vector. The $n \times 1$ $\mathbf{w}(t)$ vector and the $m \times 1$ $\mathbf{v}(t)$ vector are zero-mean Gaussian white-noise process vectors with covariances

$$E\{\mathbf{w}(t)\mathbf{w}^\top(\tau)\} = \mathbf{Q}(t)\delta(t - \tau) \tag{4.32a}$$

$$E\{\mathbf{v}_k\mathbf{v}_j^\top\} = \mathbf{R}_k\delta_{kj} \tag{4.32b}$$

where $E\{\square\}$ denotes the expected value operator, $\delta(t - \tau)$ and $\delta_{kj}$ are Dirac delta function and the Kronecker delta function respectively, $\mathbf{Q}(t)$ is an $n \times n$ covariance matrix and $\mathbf{R}_k$ and $m \times m$ covariance matrix. The estimated state is going to be denoted with a wide caret and is given by taking the expectation of the state $\mathbf{x}$:

$$\hat{\mathbf{x}}(t) = E\{\mathbf{x}(t)\} \tag{4.33}$$

The EKF involves the notion that the true state is sufficiently close to the estimated state and so the error dynamics can be represented fairly accurately by a linearized first-order Taylor series expansion:

$$\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \simeq \mathbf{f}(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) + \mathbf{F}(t)[\mathbf{x}(t) - \hat{\mathbf{x}}(t)] \tag{4.34}$$

$$\mathbf{h}(\mathbf{x}_k) \simeq \mathbf{h}(\hat{\mathbf{x}}_k) + \mathbf{H}_k[\mathbf{x}_k - \hat{\mathbf{x}}_k] \tag{4.35}$$

with $\mathbf{F}$ and $\mathbf{H}$ defined as

$$\mathbf{F}(t) = \left.\frac{\mathrm{d}\mathbf{f}}{\mathrm{d}\mathbf{x}}\right|_{\hat{\mathbf{x}}(t), \mathbf{u}(t)} \qquad \mathbf{H}_k = \left.\frac{\mathrm{d}\mathbf{h}}{\mathrm{d}\mathbf{x}}\right|_{\hat{\mathbf{x}}_k} \tag{4.36}$$

where $\hat{\mathbf{x}}_k$ is the estimated state $\hat{\mathbf{x}}$ evaluated at time $t = t_k$. By taking the expectation of Equation (4.30), the predicted estimated state differential equation is given by

$$\dot{\hat{x}}(t) = \mathbf{f}(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) \tag{4.37}$$

By defining the propagation error $\boldsymbol{\epsilon}^-(t)$ as

$$\boldsymbol{\epsilon}^-(t) = \mathbf{x}(t) - \hat{\mathbf{x}}^-(t) \tag{4.38}$$

with $\hat{\mathbf{x}}^-(t) \equiv \hat{\mathbf{x}}(t)$ and taking its derivative with respect to time while using the result from Equation (4.37) and Equation (4.30) together with Equation (4.34), the dynamics of the propagation error are given by

$$\dot{\boldsymbol{\epsilon}}^-(t) = \mathbf{F}(t)\boldsymbol{\epsilon}^-(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{4.39}$$

where the superscript $-$ was used to make evident that this error is prior to the update stage. The covariance of $\boldsymbol{\epsilon}^-(t)$ is given by

$$\mathbf{P}^-(t) = E\left\{\boldsymbol{\epsilon}^-(t)\left[\boldsymbol{\epsilon}^-(t)\right]^\top\right\} \tag{4.40}$$

and obeys the dynamic equation [9]

$$\dot{\mathbf{P}}^-(t) = E\left\{\boldsymbol{\epsilon}^-(t)(\dot{\boldsymbol{\epsilon}}^-)^\top\right\} + E\left\{\dot{\boldsymbol{\epsilon}}^-\left[\boldsymbol{\epsilon}^-(t)\right]^\top\right\} = \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}(t)^\top + \mathbf{G}(t)\mathbf{Q}(t)\mathbf{G}(t)^\top \tag{4.41}$$

The discrete-time update equation of the EKF is given by

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \Delta\mathbf{x} = \hat{\mathbf{x}}_k^- + \mathbf{K}_k[\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-)] \tag{4.42}$$

where the superscript $+$ in $\hat{\mathbf{x}}_k^+$ denotes the discrete-time right after the update stage, $\mathbf{x}_k^-$ is the propagated estimated state $\hat{\mathbf{x}}^-(t)$ evaluated at time $t_k$ and $\Delta\mathbf{x} = \mathbf{K}_k[\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-)]$ is a correction which is function of the difference between the measured and the predicted system output. The $\mathbf{K}$ matrix is the $n \times m$ Kalman gain matrix. By defining the update error $\boldsymbol{\epsilon}^+$ at time $t = t_k$ as

$$\boldsymbol{\epsilon}_k^+ = \mathbf{x}_k - \hat{\mathbf{x}}_k^+ \tag{4.43}$$

where $\mathbf{x}_k = \mathbf{x}(t = t_k)$ is the true state of the system at time $t_k$, and substituting Equation (4.31) with $\mathbf{h}(\mathbf{x}_k)$ linearized about $\hat{\mathbf{x}}_k^-$ according to Equation (4.35) in Equation (4.42), the update error $\boldsymbol{\epsilon}^+$ can be written as

$$\boldsymbol{\epsilon}_k^+ = [\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k]\boldsymbol{\epsilon}_k^- + \mathbf{K}_k\mathbf{v}_k \tag{4.44}$$

where $\mathbf{I}_n$ is the $n \times n$ identity matrix. The updated covariance matrix $\mathbf{P}_k^+$ is then given by

$$\mathbf{P}_k^+ = E\left\{\boldsymbol{\epsilon}_k^+\left(\boldsymbol{\epsilon}_k^+\right)^\top\right\} = \left[\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k\right]\mathbf{P}_k^-\left[\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k\right]^\top + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^\top \tag{4.45}$$

where $\mathbf{P}_k^- = \mathbf{P}^-(t = t_k)$ is given from the propagated system by Equation (4.41). The optimal gain matrix $\mathbf{K}_k$ is then obtained by minimizing $J_k = \operatorname{tr}\mathbf{P}_k^+$ which leads to [9]

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^\top\left(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^\top + \mathbf{R}_k\right)^{-1} \tag{4.46}$$

Substituting Equation (4.46) into Equation (4.45) gives the covariance matrix of the updated error:

$$\mathbf{P}_k^+ = (\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^- \tag{4.47}$$

The continuous-discrete EKF algorithm is then given by the set of Equations (4.48) and (4.49), where Equations (4.48) represent the propagation stage, and Equations (4.49) represent the update stage.

$$\dot{\hat{x}}(t) = \mathbf{f}(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) \tag{4.48a}$$

$$\dot{P}(t) = \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^\top(t) + \mathbf{G}(t)\mathbf{Q}(t)\mathbf{G}^\top(t) \tag{4.48b}$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^\top \left(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^\top + \mathbf{R}_k\right)^{-1} \tag{4.49a}$$

$$\mathbf{P}_k^+ = (\mathbf{I}_n - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k^- \tag{4.49b}$$

$$\Delta\mathbf{x} = \mathbf{K}_k \left[\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-)\right] \tag{4.49c}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \Delta\mathbf{x} \tag{4.49d}$$

### 4.2.1 Multiplicative Extended Kalman Filter (MEKF)

The great advantage of the MEKF is the use of the multiplicative quaternion error formulation given by

$$\delta\mathbf{q} = \mathbf{q} \otimes \hat{\mathbf{q}}^{-1} \tag{4.50}$$

with $\mathbf{q} \equiv \mathbf{q}_I^B$ and $\hat{\mathbf{q}} \equiv \mathbf{q}_I^{\hat{B}}$, which allows to enforce the unit norm constrain of the estimated quaternion with fewer problems than the addictive quaternion formulation counterpart where the error quaternion $\Delta\mathbf{q}$ is defined as $\Delta\mathbf{q} = \mathbf{q} - \hat{\mathbf{q}}$. The term $\delta\mathbf{q}$ in Equation (4.50) represents the rotation of true body frame $B$ with respect to the estimated body frame $\hat{B}$ and can be seen as the quaternion counterpart of the rotation matrix $\mathbf{A}_{\hat{B}}^B$ by following the notation of Section 2.2.1. The objective of the MEKF is to estimate the error $\delta\mathbf{q}$ and use it to update the propagated attitude quaternion $\hat{\mathbf{q}}$. The first step is the determination of the kinematics model for the quaternion error $\delta\mathbf{q}$. Taking the the time derivative of Equation (4.50) yields

$$\dot{\delta\mathbf{q}} = \dot{\mathbf{q}} \otimes \hat{\mathbf{q}}^{-1} + \mathbf{q} \otimes \dot{\hat{q}}^{-1} \tag{4.51}$$

Finding an expression for $\dot{\hat{q}}^{-1}$ can be done by recalling that by definition, the product of a quaternion with its inverse is the identity quaternion, as seen in Equation (2.32) and repeated here as

$$\hat{\mathbf{q}} \otimes \hat{\mathbf{q}}^{-1} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^\top \tag{4.52}$$

Taking the derivative with respect to time of the previous equation gives

$$\dot{\hat{q}} \otimes \hat{\mathbf{q}}^{-1} + \hat{\mathbf{q}} \otimes \dot{\hat{q}}^{-1} = \mathbf{0}_4 \tag{4.53}$$

The true and estimated quaternions satisfy the kinematic equations

$$\dot{\mathbf{q}} = \frac{1}{2} \begin{bmatrix} \vec{\boldsymbol{\omega}} \\ 0 \end{bmatrix} \otimes \mathbf{q} \tag{4.54a}$$

$$\dot{\hat{q}} = \frac{1}{2} \begin{bmatrix} \hat{\boldsymbol{\omega}} \\ 0 \end{bmatrix} \otimes \hat{\mathbf{q}} \tag{4.54b}$$

By substituting Equation (4.54b) in Eq. (4.53) and using the result from Equation (4.52), Equation (4.53) can be written as

$$\frac{1}{2}\begin{bmatrix}\hat{\boldsymbol{\omega}}\\0\end{bmatrix}\otimes\hat{\mathbf{q}}\otimes\hat{\mathbf{q}}^{-1}+\hat{\mathbf{q}}\otimes\dot{\hat{q}}^{-1}=\frac{1}{2}\begin{bmatrix}\hat{\boldsymbol{\omega}}\\0\end{bmatrix}+\hat{\mathbf{q}}\otimes\dot{\hat{q}}^{-1}=\mathbf{0}_4 \tag{4.55}$$

By multiplying both sides of Equation (4.55) by $[\hat{\mathbf{q}}^{-1}\otimes]$, the expression for $\dot{\hat{q}}^{-1}$ can be obtained:

$$\dot{\hat{q}}^{-1}=-\frac{1}{2}\hat{\mathbf{q}}^{-1}\otimes\begin{bmatrix}\hat{\boldsymbol{\omega}}\\0\end{bmatrix} \tag{4.56}$$

Substituting the quaternion kinematic model from Equation (4.54a) together with Equation (4.56) in Equation (4.51) gives

$$\dot{\delta\mathbf{q}}=\frac{1}{2}\begin{bmatrix}\vec{\boldsymbol{\omega}}\\0\end{bmatrix}\otimes\mathbf{q}\otimes\hat{\mathbf{q}}^{-1}-\frac{1}{2}\mathbf{q}\otimes\hat{\mathbf{q}}^{-1}\otimes\begin{bmatrix}\hat{\boldsymbol{\omega}}\\0\end{bmatrix}=\frac{1}{2}\left(\begin{bmatrix}\vec{\boldsymbol{\omega}}\\0\end{bmatrix}\otimes\delta\mathbf{q}-\delta\mathbf{q}\otimes\begin{bmatrix}\hat{\boldsymbol{\omega}}\\0\end{bmatrix}\right) \tag{4.57}$$

where the last equality was obtained by recovering the definition of the error quaternion given by Equation (4.50). By defining the error angular velocity vector $\Delta\vec{\boldsymbol{\omega}}$ as

$$\Delta\vec{\boldsymbol{\omega}}=\vec{\boldsymbol{\omega}}-\hat{\boldsymbol{\omega}} \tag{4.58}$$

and by substituting $\vec{\boldsymbol{\omega}}=\hat{\boldsymbol{\omega}}+\Delta\vec{\boldsymbol{\omega}}$ from Equation (4.58) in Equation (4.57) gives

$$\dot{\delta\mathbf{q}}=\frac{1}{2}\left(\begin{bmatrix}\hat{\boldsymbol{\omega}}\\0\end{bmatrix}\otimes\delta\mathbf{q}-\delta\mathbf{q}\otimes\begin{bmatrix}\hat{\boldsymbol{\omega}}\\0\end{bmatrix}\right)+\frac{1}{2}\begin{bmatrix}\Delta\vec{\boldsymbol{\omega}}\\0\end{bmatrix}\otimes\delta\mathbf{q} \tag{4.59}$$

This last equation can be further simplified by using the quaternion product properties from Equations (2.20) to (2.22) giving

$$\dot{\delta\mathbf{q}}=-\begin{bmatrix}[\hat{\boldsymbol{\omega}}\times]\delta\vec{\mathbf{q}}\\0\end{bmatrix}+\frac{1}{2}\begin{bmatrix}\Delta\vec{\boldsymbol{\omega}}\\0\end{bmatrix}\otimes\delta\mathbf{q} \tag{4.60}$$

where $\delta\vec{\mathbf{q}}$ is the vector part of $\delta\mathbf{q}$: $\delta\mathbf{q}=\begin{bmatrix}\delta\vec{\mathbf{q}}^\top & \delta q_4\end{bmatrix}^\top$. The only non-linear term corresponds to the last term on the right-hand side of the previous equation. By using the first-order approximation of $\delta\mathbf{q}$ (Equation (3.53)) and by neglecting the second-order terms, the term on the right-hand side results in

$$\frac{1}{2}\begin{bmatrix}\Delta\vec{\boldsymbol{\omega}}\\0\end{bmatrix}\otimes\delta\mathbf{q}\simeq\frac{1}{2}\begin{bmatrix}\Delta\vec{\boldsymbol{\omega}}\\0\end{bmatrix}\otimes\left(\mathbf{I}_q+\begin{bmatrix}\delta\vec{\mathbf{q}}\\0\end{bmatrix}\right)\simeq\frac{1}{2}\begin{bmatrix}\Delta\vec{\boldsymbol{\omega}}\\0\end{bmatrix} \tag{4.61}$$

Finally, the linearized model for $\delta\mathbf{q}$ is obtain by replacing the result from Equation (4.61) in Equation (4.60):

$$\dot{\delta\vec{\mathbf{q}}}=-[\hat{\boldsymbol{\omega}}\times]\delta\vec{\mathbf{q}}+\frac{1}{2}\Delta\vec{\boldsymbol{\omega}} \tag{4.62a}$$

$$\dot{\delta q_4}=0 \tag{4.62b}$$

The first-order approximation assumes that the error quaternion is very small with $\delta q_4 \simeq 1$ constant, being the estimated attitude quaternion close to the true quaternion and allowing to reduce the order of the system by one state [10]. Equation (4.62a) is going to be re-written using again the small-angle approximation $\delta\vec{\mathbf{q}} = \delta\vec{\boldsymbol{\theta}}/2$, yielding

$$\dot{\delta\vec{\theta}} = -[\hat{\boldsymbol{\omega}}\times]\delta\vec{\boldsymbol{\theta}} + \Delta\vec{\boldsymbol{\omega}} \tag{4.63}$$

The angular velocity $\vec{\boldsymbol{\omega}}$ will be measured using gyroscopes sensors. The sensor model, assuming a three-axis rate-integrating gyro, is given by the first-order Markov process [10]:

$$\vec{\boldsymbol{\omega}}(t) = \bar{\boldsymbol{\omega}}(t) - \vec{\boldsymbol{\beta}}(t) - \vec{\boldsymbol{\eta}}_v(t) \tag{4.64a}$$

$$\dot{\vec{\boldsymbol{\beta}}}(t) = \vec{\boldsymbol{\eta}}_u(t) \tag{4.64b}$$

where $\bar{\boldsymbol{\omega}}(t)$ is the measured angular velocity, $\vec{\boldsymbol{\beta}}(t)$ is the gyroscope bias vector, and $\vec{\boldsymbol{\eta}}_v(t)$ and $\vec{\boldsymbol{\eta}}_u(t)$ are zero-mean Gaussian white-noise processes with spectral densities given by $\sigma_v^2\mathbf{I}_3$ and $\sigma_u^2\mathbf{I}_3$ respectively. The parameters $\sigma_v$ and $\sigma_u$ are known as the Angular Random Walk (ARW) and Rate Random Walk (RRW) respectively. The error angular velocity $\Delta\vec{\boldsymbol{\omega}}$ from Equation (4.58) can now be written as

$$\Delta\vec{\boldsymbol{\omega}} = -\Delta\vec{\boldsymbol{\beta}} - \vec{\boldsymbol{\eta}}_v \tag{4.65}$$

with $\Delta\vec{\boldsymbol{\beta}} = \vec{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}$ by using $\hat{\boldsymbol{\omega}} = \bar{\omega} - \hat{\boldsymbol{\beta}}$ when taking the expectation of Equation (4.64a). The bias error $\Delta\vec{\boldsymbol{\beta}}$ together with the quaternion error $\delta\vec{\boldsymbol{\theta}}$ will make the new state $\mathbf{x} = \begin{bmatrix}\left(\delta\vec{\boldsymbol{\theta}}\right)^\top & \left(\Delta\vec{\boldsymbol{\beta}}\right)^\top\end{bmatrix}^\top$ to be estimated and whose dynamics are given by

$$\dot{\mathbf{x}} = \begin{bmatrix}\dot{\delta\vec{\boldsymbol{\theta}}} \\ \dot{\Delta\vec{\boldsymbol{\beta}}}\end{bmatrix} = \begin{bmatrix}-[\hat{\boldsymbol{\omega}}\times]\delta\vec{\boldsymbol{\theta}} - \Delta\vec{\boldsymbol{\beta}} - \vec{\boldsymbol{\eta}}_v \\ \vec{\boldsymbol{\eta}}_u\end{bmatrix} \tag{4.66}$$

The dynamics of the estimated state $\hat{\mathbf{x}}$ are then given by

$$\dot{\hat{x}} = E\{\dot{\mathbf{x}}\} = \begin{bmatrix}-[\hat{\boldsymbol{\omega}}\times]\delta\hat{\boldsymbol{\theta}} - (\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}) \\ \vec{\mathbf{0}}\end{bmatrix} = \mathbf{0}_6 \tag{4.67}$$

where the last equality was obtained by noting that by definition (Equation (4.50))

$$\delta\hat{\mathbf{q}} = E\{\delta\mathbf{q}\} = \hat{\mathbf{q}} \otimes \hat{\mathbf{q}}^{-1} = \begin{bmatrix}\vec{\mathbf{0}}_3 \\ 1\end{bmatrix} \Rightarrow \delta\hat{\boldsymbol{\theta}} = \vec{\mathbf{0}} \tag{4.68}$$

Equation (4.67) means that in the propagation stage, the estimated state will remain constant. Since by definition $\hat{\mathbf{x}} = \mathbf{0}_6$, by using the result from Equation (4.68) and by noting that $\Delta\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}} = \vec{\mathbf{0}}$, the estimated quaternion and bias error will remain constant and equal to zero in the prediction step. This result is the key feature of the MEKF. The MEKF propagates the global variables to the time of the next representation while the error variables (defined by state $\mathbf{x}$) do not need to be propagated because they are identically zero over the propagation step. The update stage updates the error variables which will

then be used to update the global variables $\mathbf{q}$ and $\vec{\boldsymbol{\beta}}$ [9]. The MEKF error model can thus be written as

$$\dot{\boldsymbol{\epsilon}}^{-}(t) = \mathbf{F}(t)\boldsymbol{\epsilon}^{-}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{4.69}$$

where $\boldsymbol{\epsilon}^{-}(t) = \mathbf{x} - \hat{\mathbf{x}}^{-} = \begin{bmatrix} \delta\vec{\boldsymbol{\theta}}^{\top} & \Delta\vec{\boldsymbol{\beta}}^{\top} \end{bmatrix}^{\top}$ is the $6 \times 1$ error state vector and $\mathbf{w}(t) = \begin{bmatrix} \vec{\boldsymbol{\eta}}_v^{\top} & \vec{\boldsymbol{\eta}}_u^{\top} \end{bmatrix}^{\top}$ is the $6 \times 1$ process noise vector. The dynamics and process noise distribution $6 \times 6$ matrices $\mathbf{F}(t)$ and $\mathbf{G}(t)$ and the $6 \times 6$ covariance matrix $\mathbf{Q}(t)$ of $\mathbf{w}(t)$ are given respectively by

$$\mathbf{F}(t) = \begin{bmatrix} -[\hat{\boldsymbol{\omega}}(t)\times] & -\mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \qquad \mathbf{G}(t) = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 \end{bmatrix} \qquad \mathbf{Q}(t) = \begin{bmatrix} \sigma_v^2\mathbf{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \sigma_u^2\mathbf{I}_3 \end{bmatrix} \tag{4.70}$$

Equation (4.69) is similar to the one used to express the error for the EKF, Equation (4.39). From here on the development of this filter follows from the EKF, namely the determination of the updated covariance matrix $\mathbf{P}_k^{+}$, Equation (4.47), and the determination of the Kalman gain matrix $\mathbf{K}_k$, Equation (4.46). The updated state variables $\delta\hat{\boldsymbol{\theta}}^{+}$ and $\Delta\vec{\boldsymbol{\beta}}^{+}$ are simply given by

$$\begin{bmatrix} \delta\hat{\boldsymbol{\theta}}_k^{+} \\ \Delta\hat{\boldsymbol{\beta}}_k^{+} \end{bmatrix} = \begin{bmatrix} \vec{\mathbf{0}} \\ \vec{\mathbf{0}} \end{bmatrix} + \mathbf{K}_k \left[ \mathbf{y}_k - \mathbf{h}\left(\hat{\mathbf{x}}_k^{-}\right) \right] \tag{4.71}$$

The update of the estimated gyro bias and angular velocity are given by

$$\hat{\boldsymbol{\beta}}_k^{+} = \hat{\boldsymbol{\beta}}_k^{-} + \Delta\vec{\boldsymbol{\beta}}_k^{+} \tag{4.72}$$

$$\hat{\boldsymbol{\omega}}_k^{+} = \bar{\boldsymbol{\omega}}_k - \hat{\boldsymbol{\beta}}_k^{+} \tag{4.73}$$

while the update of the global attitude representation, the attitude quaternion, is performed by two steps:

$$\hat{\mathbf{q}}' = \begin{bmatrix} \frac{1}{2}\delta\hat{\boldsymbol{\theta}}_k^{+} \\ 1 \end{bmatrix} \otimes \hat{\mathbf{q}}_k^{-} \tag{4.74}$$

$$\hat{\mathbf{q}}_k^{+} = \hat{\mathbf{q}}' / \left\| \hat{\mathbf{q}}' \right\| \tag{4.75}$$

The final step consists on finding $\mathbf{y}_k$ and the sensitivity matrix $\mathbf{H}_k$. The measurement vector $\vec{\mathbf{y}}_k$ follows the formulation of Equation (4.31) and is given by

$$\vec{\mathbf{y}}_k = \vec{\mathbf{h}}(\mathbf{x}_k) + \vec{\mathbf{v}}_k = \mathbf{A}(\mathbf{q})\vec{\mathbf{z}}_I \Big|_{t_k} + \vec{\mathbf{v}}_k, \quad \mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k) \qquad \mathbf{R}_k = \sigma_y^2\mathbf{I}_3 \tag{4.76}$$

where $\vec{\mathbf{z}}_I$ is a measurement vector given by the measurement model and $\mathbf{R}_k$ was defined assuming that the measurement errors are isotropic [9] with $\sigma_y$ the noise standard deviation of the sensor. By definition (Equation (4.50)):

$$\mathbf{A}(\mathbf{q}) = \mathbf{A}(\delta\mathbf{q})\mathbf{A}(\hat{\mathbf{q}}^{-}) = \left( \mathbf{I}_3 - \left[ \delta\vec{\boldsymbol{\theta}} \times \right] \right) \mathbf{A}(\hat{\mathbf{q}}^{-}) \tag{4.77}$$

where the first-order approximation for $\mathbf{A}(\delta\mathbf{q})$ is given by Equation (3.59) by using $\delta\vec{\mathbf{q}} = \delta\vec{\boldsymbol{\theta}}/2$. The linearized observation vector $\vec{\mathbf{h}}(\mathbf{x}_k)$ is then given by

$$\vec{\mathbf{h}}(\mathbf{x}_k) \simeq \mathbf{A}(\hat{\mathbf{q}}^-)\vec{\mathbf{z}}_I\Big|_{t_k} + \left\{\left[\mathbf{A}(\hat{\mathbf{q}}^-)\vec{\mathbf{z}}_I\right] \times \delta\vec{\boldsymbol{\theta}}\right\}\Big|_{t_k} \tag{4.78}$$

The sensitivity matrix is then obtained from Equation (4.36) and is given by

$$\mathbf{H}_k(\hat{\mathbf{x}}_k^-) = \Big[\left[(\mathbf{A}(\hat{\mathbf{q}}^-)\vec{\mathbf{z}}_I)\times\right] \quad \mathbf{0}_{3\times 3}\Big]\Big|_{t_k} \tag{4.79}$$

The observation vector $\vec{\mathbf{h}}(\hat{\mathbf{x}}_k^-)$ used in Equation (4.71) is given by

$$\vec{\mathbf{h}}(\hat{\mathbf{x}}_k^-) = \mathbf{A}(\hat{\mathbf{q}}^-)\vec{\mathbf{z}}_I\Big|_{t_k} \tag{4.80}$$

since $\delta\hat{\boldsymbol{\theta}}_k^- = \vec{\mathbf{0}}$. For the update stage described here, only a single measurement was considered. This is not a concern since the MEKF update is linear and so the principle of superposition can be used. The update stage can thus be run repeatedly for each measurement at a time, before the next propagation stage. This is known as Murrell's method [10]. In order to obtain the updated covariance matrix $\mathbf{P}_k^+$ and the updated attitude quaternion $\hat{\mathbf{q}}_k^+$, it is first necessary to propagate their values. This can be done by numerically integrating Equations (4.41) and (4.54b) respectively or by using a discrete propagation technique as done in this work. The discrete propagation of the quaternion from Equation (4.54b) is found using a power series approach [10]. The propagated attitude quaternion is then given by

$$\hat{\mathbf{q}}_{k+1}^- = \boldsymbol{\Omega}(\hat{\boldsymbol{\omega}}_k^+)\hat{\mathbf{q}}_k^+ \tag{4.81}$$

where

$$\boldsymbol{\Omega}(\hat{\boldsymbol{\omega}}_k^+) = \begin{bmatrix} \cos\left(\frac{1}{2}\|\hat{\boldsymbol{\omega}}_k^+\|\Delta t\right)\mathbf{I}_3 - \left[\vec{\boldsymbol{\psi}}_k^+\times\right] & \vec{\boldsymbol{\psi}}_k^+ \\ -\left(\vec{\boldsymbol{\psi}}_k^+\right)^\top & \cos\left(\frac{1}{2}\|\hat{\boldsymbol{\omega}}_k^+\|\Delta t\right) \end{bmatrix} \quad \text{and} \quad \vec{\boldsymbol{\psi}}_k^+ = \frac{\sin\left(\frac{1}{2}\|\hat{\boldsymbol{\omega}}_k^+\|\Delta t\right)\hat{\boldsymbol{\omega}}_k^+}{\|\hat{\boldsymbol{\omega}}_k^+\|} \tag{4.82}$$

where the $+$ and $-$ signs were used once again to indicate the post-update and pre-update estimation stage and $\Delta t$ is the sampling time of the gyro. Given the post-update estimate of the gyro bias $\hat{\boldsymbol{\beta}}_k^+$, and since $\dot{\hat{\beta}} = \vec{\mathbf{0}}$ (Equation (4.64b)), the propagated gyro bias $\hat{\boldsymbol{\beta}}_{k+1}$ is given in discrete-time by

$$\hat{\boldsymbol{\beta}}_{k+1}^- = \hat{\boldsymbol{\beta}}_k^+ \tag{4.83}$$

The discrete-time propagation of the covariance matrix $\mathbf{P}$ is given by

$$\mathbf{P}_{k+1}^- = \boldsymbol{\Phi}_k\mathbf{P}_k^+\boldsymbol{\Phi}_k^\top + \boldsymbol{\Gamma}_k\mathbf{Q}_k\boldsymbol{\Gamma}_k^\top \tag{4.84}$$

with $\boldsymbol{\Gamma}_k$ given by

$$\boldsymbol{\Gamma}_k = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{I}_3 \end{bmatrix} \tag{4.85}$$

The $6 \times 6$ discrete error-state transition matrix $\mathbf{\Phi}_k$ can also be derived using a power series approach [10]:

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

with

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

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

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

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

The discrete process noise covariance $\mathbf{Q}_k$ is given by

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

The implementation of the MEKF is given under the form of a flowchart by Figure A.6 in Appendix A.

### 4.2.2 Magnetometer Calibration Extended Kalman Filter (MCEKF)

The better the accuracy of the attitude determination sensors is, the better is going to be the attitude estimate. In order to improve the attitude accuracy obtained using the magnetometers of the ORCASat, the MCEKF was implemented. The magnetometer measurements are going to be modeled as [9]

$$\vec{\mathbf{B}}_{M_k} = (\mathbf{I}_3 + \mathbf{D})^{-1}(\mathcal{O}^\top \mathbf{A}_{R_k}^M \vec{\mathbf{B}}_{R_k} + \vec{\mathbf{b}} + \vec{\boldsymbol{\nu}}_k), \quad \vec{\boldsymbol{\nu}}_k \sim \mathcal{N}(0, \mathbf{\Sigma}_k) \tag{4.89}$$

where $\vec{\mathbf{B}}_{M_k}$ is the magnetic field measured by the magnetometer at time $t = t_k$, $\vec{\mathbf{B}}_{R_k}$ is the corresponding model of the geomagnetic field in a chosen reference frame such as frame $E$ or frame $I$ from Section 2.1, $\mathbf{A}_{R_k}^M$ is the unknown attitude matrix of the magnetometer frame $M$ with respect to the reference frame $R$, $\mathbf{D}$ is a fully-populated $3 \times 3$ matrix of scale factors and non-orthogonality corrections, $\mathcal{O}$ is an orthogonal matrix, $\vec{\mathbf{b}}$ is the magnetometer bias vector and $\vec{\boldsymbol{\nu}}_k$ is the measurement noise vector which is assumed to be a zero-mean Gaussian process with covariance $\mathbf{\Sigma}_k$. Since the attitude matrix $\mathbf{A}_{R_k}^M$ is unknown, the parameters of $\mathcal{O}$ cannot be determined. The algorithm presented here assumes that the matrix $\mathbf{D}$ is symmetric since any skew-symmetric contribution is equivalent to a rotation that can be absorbed in $\mathcal{O}$ [9]. An attitude independent approach is possible by computing

$$y_k = \left\|\vec{\mathbf{B}}_{M_k}\right\|^2 - \left\|\vec{\mathbf{B}}_{R_k}\right\|^2 \tag{4.90}$$

In order to eliminate the dependence on the attitude matrix and on matrix $\mathcal{O}$, Equation (4.89) can be written as

$$(\mathbf{I}_3 + \mathbf{D})\vec{\mathbf{B}}_{M_k} - \vec{\mathbf{b}} - \vec{\boldsymbol{\nu}}_k = \mathcal{O}^\top \mathbf{A}_{R_k}^M \vec{\mathbf{B}}_{R_k} \tag{4.91}$$

Taking the square norm of both sides of the previous equation gives

$$\left\|(\mathbf{I}_3 + \mathbf{D})\vec{\mathbf{B}}_{M_k} - \vec{\mathbf{b}} - \vec{\boldsymbol{\nu}}_k\right\|^2 = \left\|(\mathbf{I}_3 + \mathbf{D})\vec{\mathbf{B}}_{M_k}\right\|^2 - 2\left[(\mathbf{I}_3 + \mathbf{D})\vec{\mathbf{B}}_{M_k}\right] \cdot (\vec{\mathbf{b}} + \vec{\boldsymbol{\nu}_k}) + \left\|\vec{\mathbf{b}}\right\|^2 + \|\vec{\boldsymbol{\nu}_k}\|^2 + 2\vec{\mathbf{b}} \cdot \vec{\boldsymbol{\nu}_k} = \left\|\vec{\mathbf{B}}_{R_k}\right\|^2 \tag{4.92}$$

Substituting Equation (4.92) in Equation (4.90) gives

$$y_k = (\vec{\mathbf{B}}_{M_k})^\top \vec{\mathbf{B}}_{M_k} - \left[(\mathbf{I}_3 + \mathbf{D})\vec{\mathbf{B}}_{M_k}\right]^\top \left[(\mathbf{I}_3 + \mathbf{D})\vec{\mathbf{B}}_{M_k}\right] + 2\left[(\mathbf{I}_3 + \mathbf{D})\vec{\mathbf{B}}_{M_k}\right] \cdot \vec{\mathbf{b}} - \left\|\vec{\mathbf{b}}\right\|^2 + 2\left[(\mathbf{I}_3 + \mathbf{D})\vec{\mathbf{B}}_{M_k} - \vec{\mathbf{b}}\right] \cdot \vec{\boldsymbol{\nu}_k} - \|\vec{\boldsymbol{\nu}}_k\|^2 \tag{4.93}$$

The first 2 terms on the right-hand side can be grouped and simplified by using the symmetric property of $\mathbf{D}$ giving

$$y_k = -(\vec{\mathbf{B}}_{M_k})^\top (2\mathbf{D} + \mathbf{D}^2)\vec{\mathbf{B}}_{M_k} + 2(\vec{\mathbf{B}}_{M_k})^\top (\mathbf{I}_3 + \mathbf{D})\vec{\mathbf{b}} - \left\|\vec{\mathbf{b}}\right\|^2 + v_k = h_k(\mathbf{x}) + v_k \tag{4.94}$$

where the effective noise $v_k$ given by

$$v_k = 2\left[(\mathbf{I}_3 + \mathbf{D})\,\vec{\mathbf{B}}_{M_k} - \vec{\mathbf{b}}\right]^\top \vec{\boldsymbol{\nu}}_k - \|\vec{\boldsymbol{\nu}}_k\|^2 \tag{4.95}$$

is approximately Gaussian with variance $\sigma_k^2$ given by

$$\sigma_k^2 = E\{v_k^2\} - E^2\{v_k\} = 4\left[(\mathbf{I}_3 + \mathbf{D})\vec{\mathbf{B}}_{M_k} - \vec{\mathbf{b}}\right]^\top \boldsymbol{\Sigma}_k \left[(\mathbf{I}_3 + \mathbf{D})\vec{\mathbf{B}}_{M_k} - \vec{\mathbf{b}}\right] + 2\,\mathrm{tr}\,\boldsymbol{\Sigma}_k^2 \tag{4.96}$$

The system state vector $\mathbf{x}$ is defined as

$$\mathbf{x} = \begin{bmatrix}\vec{\mathbf{b}}^\top & \mathbf{D}_v^\top\end{bmatrix}^\top \tag{4.97}$$

with $\mathbf{D}_v$ defined as

$$\mathbf{D}_v = \begin{bmatrix}D_{11} & D_{22} & D_{33} & D_{12} & D_{13} & D_{23}\end{bmatrix}^\top \tag{4.98}$$

where $D_{ij}$ is the element from the $i$'th row and $j$'th column of matrix $\mathbf{D}$. Since the state vector is constant and no noise appears in the state model, $\dot{\mathbf{x}} = \mathbf{0}_9 \Rightarrow \dot{\hat{x}} = \mathbf{0}_9$, meaning that in the propagation stage the estimated state $\hat{\mathbf{x}}$ will remain constant and equal to the value from the previous update step: $\hat{\mathbf{x}}_k^- = \hat{\mathbf{x}}_{k-1}^+$. The dynamics of the propagation error (Equation (4.38)) is given by $\dot{\epsilon}^- = \dot{\mathbf{x}} - \dot{\hat{x}}^- = \mathbf{0}_9$, meaning that the covariance matrix $\mathbf{P}$ will also remain constant in the propagation stage (Equation (4.41)) and equal to the value from the previous update step: $\mathbf{P}_k^- = \mathbf{P}_{k-1}^+$. The EKF equations, Eqs. (4.48) and (4.49), are

thus reduced to

$$\mathbf{K}_k = \mathbf{P}_{k-1}^{+}\mathbf{H}_k^{\top}(\hat{\mathbf{x}}_{k-1}^{+})\left[\mathbf{H}_k(\hat{\mathbf{x}}_{k-1}^{+})\mathbf{P}_{k-1}^{+}\mathbf{H}_k^{\top}(\hat{\mathbf{x}}_{k-1}^{+}) + \sigma_k^2(\hat{\mathbf{x}}_{k-1}^{+})\right]^{-1} \tag{4.99}$$

$$\hat{\mathbf{x}}_k^{+} = \hat{\mathbf{x}}_{k-1}^{+} + \mathbf{K}_k\left[y_k - h_k(\hat{\mathbf{x}}_{k-1}^{+})\right] \tag{4.100}$$

$$\mathbf{P}_k^{+} = \left[\mathbf{I}_9 - \mathbf{K}_k\mathbf{H}_k(\hat{\mathbf{x}}_{k-1}^{+})\right]\mathbf{P}_{k-1}^{+} \tag{4.101}$$

The first term on the right-hand side of Equation (4.94), $-(\vec{\mathbf{B}}_{M_k})^{\top}(2\mathbf{D}+\mathbf{D}^2)\vec{\mathbf{B}}_{M_k}$, can be written as $-(\vec{\mathbf{B}}_{M_k})^{\top}(2\mathbf{D}+\mathbf{D}^2)\vec{\mathbf{B}}_{M_k} = -\mathbf{S}_k^{\top}\mathbf{E}_{\mathbf{v}}$ with $\mathbf{E}_v$ and $\mathbf{S}_k$ defined as

$$\mathbf{E}_v = \begin{bmatrix} E_{11} & E_{22} & E_{33} & E_{12} & E_{13} & E_{23} \end{bmatrix}^{\top} \tag{4.102}$$

$$\mathbf{S}_k = \begin{bmatrix} B_1^2 & B_2^2 & B_3^2 & 2B_1B_2 & 2B_1B_3 & 2B_2B_3 \end{bmatrix}^{\top} \tag{4.103}$$

The vector $\mathbf{E}_{\mathbf{v}}$ is defined as a function of the parameters of the matrix $\mathbf{E} = 2\mathbf{D}+\mathbf{D}^2$ with $E_{ij}$ the element from the $i$'th row and $j$'th column of $\mathbf{E}$. The vector $\mathbf{S}$ is function of the measured geomagnetic field where $B_i$ is the $i$'th element of $\vec{\mathbf{B}}_{M_k}$. The sensitivity matrix $\mathbf{H}_k(\hat{\mathbf{x}}_{k-1}^{+})$ can thus be given by

$$\mathbf{H}_k(\hat{\mathbf{x}}_{k-1}^{+}) = \left.\frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k-1}^{+},t_k} = \begin{bmatrix} 2(\vec{\mathbf{B}}_{M_k})^{\top}(\mathbf{I}_3 + \hat{\mathbf{D}}_{k-1}^{+}) - 2(\hat{\mathbf{b}}_{k-1}^{+})^{\top} & -\mathbf{S}_k^{\top}\mathbf{M}_k + 2\mathbf{J}_k \end{bmatrix} \tag{4.104}$$

where the $1\times 6$ row vector $\mathbf{J}_k$ and the $6\times 6$ matrix $\mathbf{M}_k$ are given respectively by

$$\mathbf{J}_k = \begin{bmatrix} B_1\hat{b}_1 & B_2\hat{b}_2 & B_3\hat{b}_3 & B_1\hat{b}_2 + B_2\hat{b}_1 & B_1\hat{b}_3 + B_3\hat{b}_1 & B_2\hat{b}_3 + B_3\hat{b}_2 \end{bmatrix} \tag{4.105}$$

$$\mathbf{M}_k = 2\mathbf{I}_6 + \begin{bmatrix} 2\hat{D}_{11} & 0 & 0 & 2\hat{D}_{12} & 2\hat{D}_{13} & 0 \\ 0 & 2\hat{D}_{22} & 0 & 2\hat{D}_{12} & 0 & 2\hat{D}_{23} \\ 0 & 0 & 2\hat{D}_{33} & 0 & 2\hat{D}_{13} & 2\hat{D}_{23} \\ \hat{D}_{12} & \hat{D}_{12} & 0 & \hat{D}_{11}+\hat{D}_{22} & \hat{D}_{23} & \hat{D}_{13} \\ \hat{D}_{13} & 0 & \hat{D}_{13} & \hat{D}_{23} & \hat{D}_{11}+\hat{D}_{33} & \hat{D}_{12} \\ 0 & \hat{D}_{23} & \hat{D}_{23} & \hat{D}_{13} & \hat{D}_{12} & \hat{D}_{22}+\hat{D}_{33} \end{bmatrix} \tag{4.106}$$

with $\hat{b}_i$ the components of the estimated bias vector $\hat{\mathbf{b}}_{k-1}^{+}$ and with $\hat{D}_{ij}$ the components of the estimated matrix $\hat{\mathbf{D}}_{k-1}^{+}$. The vectors $\mathbf{D}_{\mathbf{v}}$ and $\mathbf{E}_{\mathbf{v}}$ defined in Equations (4.98) and (4.102) could have been defined as different functions of matrices $\mathbf{D}$ and $\mathbf{E}$ respectively. The filter equations, Eqs. (4.99), (4.100), and (4.101) would remain the same but the quantities $\mathbf{S}$, $\mathbf{J}$, and $\mathbf{M}$ would have got a different expression to accommodate the different vectors $\mathbf{D}_{\mathbf{v}}$ and $\mathbf{E}_{\mathbf{v}}$. Nonetheless, the estimated bias vector $\hat{\mathbf{b}}$ and the estimated scale factors and non-orthogonality corrections matrix $\hat{\mathbf{D}}$ would be the same. Lastly, an approximate estimation for the measured magnetic field $\hat{\mathbf{B}}_{M_k}$ can be given by neglecting the effect of $\mathcal{O}$:

$$\hat{\mathbf{B}}_{M_k} = (\mathbf{I}_3 + \hat{\mathbf{D}})\vec{\mathbf{B}}_{M_k} - \hat{\mathbf{b}} \tag{4.107}$$

where $\hat{\mathbf{B}}_{M_k} \equiv \mathbf{A}_{R_k}^{M}\vec{\mathbf{B}}_{R_k}$. The implementation of the MCEKF is given under the form of a flowchart in Appendix A by Figure A.4.

# Chapter 5

# Attitude Control

In this chapter, the different algorithms used to achieve 3-axis Earth-pointing stabilization as well as the detumbling controller used in the ORCASat are presented. In Section 5.1, a sliding mode controller will be developed. In Section 5.2, three linear controllers, which make use of the linear model of the ORCASat outlined in Section 3.5, are developed. In Section 5.3, the detumbling controller is outlined and finally, in Section 5.4, the implementation of the different controllers is described.

## 5.1 Sliding Mode Controller (SMC)

The objective of the pointing controllers is 3 axis-stabilization Earth pointing. This is done by making the body frame $B$ coincide with the orbit frame $O$, i.e. by making the angular velocity $\vec{\boldsymbol{\omega}}^{B/O}$ converge to $\vec{\mathbf{0}}$ and the attitude quaternion $\mathbf{q}_O^B$ converge to the identity quaternion $\mathbf{I}_q$. The attitude quaternion $\mathbf{q}_O^B$ represents the rotation error between the body frame and the desired orbit or LVLH frame being thus from here on denoted by $\mathbf{q}_e$. The quaternion error $\mathbf{q}_e$ is given by

$$\mathbf{q}_e \equiv \mathbf{q}_O^B = \mathbf{q}_I^B \otimes \left(\mathbf{q}_I^O\right)^{-1} = \left(\mathbf{q}_I^O\right)^{-1} \odot \mathbf{q}_I^B = \begin{bmatrix} q_{c_4} & q_{c_3} & -q_{c_2} & -q_{c_1} \\ -q_{c_3} & q_{c_4} & q_{c_1} & -q_{c_2} \\ q_{c_2} & -q_{c_1} & q_{c_4} & -q_{c_3} \\ q_{c_1} & q_{c_2} & q_{c_3} & q_{c_1} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{5.1}$$

where $q_i$ and $q_{c_i}$ with $i = 1, 2, 3, 4$ are respectively the components of $\mathbf{q}_I^B$ and $\mathbf{q}_I^O$. The $c$ from $q_{c_i}$ stands for commanded quaternion since $\mathbf{q}_I^O$ is the attitude quaternion that the spacecraft should follow. The commanded quaternion can be obtained from $\mathbf{q}_O^I$ using the inverse attitude quaternion property given by Equations (2.31) and (2.28). The attitude quaternion $\mathbf{q}_O^I$, in turn, can be obtained from the DCM $\mathbf{A}_O^I$ defined in Equation (2.5) by using Equation (2.37). In analogy with the error quaternion $\mathbf{q}_e$, the angular velocity $\vec{\boldsymbol{\omega}}^{B/O}$ is going to be denoted by $\vec{\boldsymbol{\omega}}_e$ since it represents the error of the body angular velocity with respect to the desired angular velocity of the orbit frame $O$. The error angular velocity $\vec{\boldsymbol{\omega}}_e$ is given in the body frame by

$$\vec{\boldsymbol{\omega}}_e \equiv \vec{\boldsymbol{\omega}}_B^{B/O} = \vec{\boldsymbol{\omega}}_B^{B/I} - \vec{\boldsymbol{\omega}}_B^{O/I} \tag{5.2}$$

where the angular velocity of the orbit frame with respect to the inertial frame $\vec{\boldsymbol{\omega}}_B^{O/I}$ is given by [9]

$$\vec{\boldsymbol{\omega}}_B^{O/I} = \mathbf{A}_O^B \vec{\boldsymbol{\omega}}_O^{O/I} = \mathbf{A}_O^B \begin{bmatrix} 0 \\ -\|\vec{\mathbf{r}}_I \times \vec{\mathbf{v}}_I\|/\|\vec{\mathbf{r}}_I\|^2 \\ \|\vec{\mathbf{r}}_I\|(\hat{\mathbf{o}}_{2_I} \cdot \dot{\vec{\mathbf{v}}}_I)/\|\vec{\mathbf{r}}_I \times \vec{\mathbf{v}}_I\| \end{bmatrix} \tag{5.3}$$

where $\mathbf{A}_O^B = \mathbf{A}_I^B \mathbf{A}_O^I$ with $\mathbf{A}_I^B$ obtained from $\mathbf{q}_I^B$ using Equation (2.36) and with $\mathbf{A}_O^I$ obtained from Equation (2.5). The vectors $\vec{\mathbf{r}}_I$ and $\vec{\mathbf{v}}_I$ are the spacecraft position and velocity in the ECI frame $I$ and $\hat{\mathbf{o}}_{2_I}$ is given by Equation (2.6b).

The sliding mode controller presented is adapted from the work developed in [40]. The design of this controller will be split in two phases. The first corresponds to the design of the sliding manifold, and the second to the design of the sliding condition. The sliding condition can be seen as a reaching phase where the controller makes the system to reach the sliding manifold from an arbitrary state. The sliding manifold in turn is the condition that makes the system to converge to the reference attitude.

### 5.1.1 Sliding Manifold Design

The sliding variable $\vec{\mathbf{s}}$ will be defined as [40]

$$\vec{\mathbf{s}} = \vec{\boldsymbol{\omega}}_e + \boldsymbol{K}\vec{\mathbf{q}}_e \tag{5.4}$$

where $\boldsymbol{K}$ is a $3 \times 3$ positive definite matrix. In this work, $\boldsymbol{K}$ will be defined as $\boldsymbol{K} = K\mathbf{I}_3$, with $K$ a positive gain, without any loss of generality. The sliding manifold $\mathcal{S}$ is defined as the set of values of $\vec{\mathbf{q}}_e$ and $\vec{\boldsymbol{\omega}}_e$ such that $\vec{\mathbf{s}} = \vec{\mathbf{0}}$:

$$\mathcal{S} = \{\vec{\mathbf{q}}_e, \vec{\boldsymbol{\omega}}_e : \vec{\mathbf{s}} = \vec{\mathbf{0}}\} \tag{5.5}$$

This sliding variable $\vec{\mathbf{s}}$ can be shown to guarantee the convergence of $\vec{\mathbf{q}}_e$ to zero and $q_{e_4}$ to one when the system is in the sliding manifold. By considering the following candidate Lyapunov function

$$\mathcal{V} = \vec{\mathbf{q}}_e^\top \vec{\mathbf{q}}_e + (1 - q_{e_4})^2 = 1 - q_{e_4}^2 + 1 - 2q_{e_4} + q_{e_4}^2 = 2(1 - q_{e_4}) \tag{5.6}$$

where the unit norm property of the attitude quaternion $\mathbf{q}^\top \mathbf{q} = 1$ was used, and taking its derivative with respect to time while using the quaternion kinematics from Equation (3.9) gives

$$\dot{\mathcal{V}} = -2\dot{q_{e_4}} = \vec{\boldsymbol{\omega}}_e \cdot \vec{\mathbf{q}}_e = \vec{\boldsymbol{\omega}}_e^\top \vec{\mathbf{q}}_e \tag{5.7}$$

When the system is in the sliding manifold ($\vec{\mathbf{s}} = 0$) Equation (5.4) writes

$$\vec{\boldsymbol{\omega}}_e = -\boldsymbol{K}\vec{\mathbf{q}}_e \tag{5.8}$$

By substituting the previous equation in Eq. (5.7), the derivative of the candidate Lyapunov function gives

$$\dot{\mathcal{V}} = -\vec{\mathbf{q}}_e^\top \boldsymbol{K} \vec{\mathbf{q}}_e \tag{5.9}$$

which is negative definite, proving, according to Theorem 3, that the equilibrium $\vec{\omega}_e = \vec{0}$ and $\mathbf{q}_e = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^\top$ is globally asymptotically stable if the spacecraft is on the sliding manifold $\mathcal{S}$.

### 5.1.2 Sliding Condition Design

As stated before, the objective of the sliding condition is to produce the desired control torque to turn the system's trajectory towards the sliding manifold. In order to determine the desired control torque, the representation of the spacecraft motion in the space of the sliding variable is going to be computed by differentiating the sliding variable $\vec{s}$ from Equation (5.4) with respect to time yielding

$$\dot{\vec{s}} = \dot{\vec{\omega}}_e + \boldsymbol{K}\dot{\vec{\mathbf{q}}}_e = \dot{\vec{\omega}}_B^{B/I} - \dot{\mathbf{A}}_O^B \vec{\omega}_O^{O/I} - \mathbf{A}_O^B \dot{\vec{\omega}}_O^{O/I} + \frac{1}{2}\boldsymbol{K}\left(\vec{\omega}_e q_{e_4} - \vec{\omega}_e \times \vec{\mathbf{q}}_e\right) \tag{5.10}$$

where Equations (5.2) and (3.9) were used. By multiplying each side of Equation (5.10) by the spacecraft's inertia matrix $\mathbf{J} \equiv \mathbf{J}_B$ and by using the ORCASat's dynamic equation, Eq. (3.26), while neglecting the disturbance torques and considering that the momentum wheel is in nominal mode, $\dot{\vec{\mathbf{h}}}_B^s = 0$, gives

$$\mathbf{J}\dot{\vec{\mathbf{s}}} = \left(\mathbf{J}\vec{\boldsymbol{\omega}}_B^{B/I} + \vec{\mathbf{h}}_B^s\right) \times \vec{\boldsymbol{\omega}}_B^{B/I} + \mathbf{J}\left[\vec{\boldsymbol{\omega}}_e \times\right]\mathbf{A}_O^B \vec{\boldsymbol{\omega}}_O^{O/I} - \mathbf{J}\mathbf{A}_O^B \dot{\vec{\boldsymbol{\omega}}}_O^{O/I} + \frac{1}{2}\mathbf{J}\boldsymbol{K}\left(\vec{\boldsymbol{\omega}}_e q_{e_4} - \vec{\boldsymbol{\omega}}_e \times \vec{\mathbf{q}}_e\right) + \vec{\mathbf{u}} \tag{5.11}$$

If the spacecraft is in the sliding manifold, in order to keep it there, i.e., to satisfy the condition $\dot{\vec{\mathbf{s}}} = \vec{\mathbf{0}}$, the control torque must be equal to an equivalent torque ($\vec{\mathbf{u}} = \vec{\boldsymbol{\tau}}_{eq}$) defined as

$$\vec{\boldsymbol{\tau}}_{eq} = -\left(\mathbf{J}\vec{\boldsymbol{\omega}}_B^{B/I} + \vec{\mathbf{h}}_B^s\right) \times \vec{\boldsymbol{\omega}}_B^{B/I} - \mathbf{J}\left[\vec{\boldsymbol{\omega}}_e \times\right]\mathbf{A}_O^B \vec{\boldsymbol{\omega}}_O^{O/I} + \mathbf{J}\mathbf{A}_O^B \dot{\vec{\boldsymbol{\omega}}}_O^{O/I} - \frac{1}{2}\mathbf{J}\boldsymbol{K}\left(\vec{\boldsymbol{\omega}}_e q_{e_4} - \vec{\boldsymbol{\omega}}_e \times \vec{\mathbf{q}}_e\right) \tag{5.12}$$

where $\dot{\vec{\boldsymbol{\omega}}}_O^{O/I}$ is calculated numerically from $\vec{\boldsymbol{\omega}}_O^{O/I}$. If the satellite is not in the sliding manifold, $\vec{\mathbf{s}} \neq \vec{\mathbf{0}}$, then the control torque must be equal to the equivalent torque from Equation (5.12) plus an extra term, the sliding condition, to make the sliding variable $\vec{\mathbf{s}}$ converge to zero. The desired torque is given by

$$\vec{\boldsymbol{\tau}}_{des} = \vec{\boldsymbol{\tau}}_{eq} - \lambda \vec{\mathbf{s}} \tag{5.13}$$

where $\lambda$ is a positive constant number.

The ORCASat is a magnetically actuated spacecraft and as such, it cannot produce torque independently in any direction, therefore, before investigating the stability of the sliding condition, the magnetic control torque produced by the magnetorquers is going to be developed. The control torque $\vec{\mathbf{u}}$ produced can be written as

$$\vec{\mathbf{u}} = \vec{\mathbf{m}}_{ctrl} \times \vec{\mathbf{B}}_B = \left(\vec{\mathbf{m}}^{\parallel} + \vec{\mathbf{m}}^{\perp}\right) \times \vec{\mathbf{B}}_B = \vec{\mathbf{m}}^{\perp} \times \vec{\mathbf{B}}_B \tag{5.14}$$

where $\vec{\mathbf{B}}_B$ is the local geomagnetic field written in body frame coordinates and $\vec{\mathbf{m}}_{ctrl}$ is the dipole moment generated by the magnetorquers. The vectors $\vec{\mathbf{m}}^{\parallel}$ and $\vec{\mathbf{m}}^{\perp}$ are the components of the dipole moment $\vec{\mathbf{m}}_{ctrl}$ parallel and perpendicular to $\vec{\mathbf{B}}_B$ respectively. It can be seen by Equation (5.14) that the control torque is always perpendicular to the local magnetic field $\vec{\mathbf{B}}_B$ and to the dipole moment $\vec{\mathbf{m}}_{ctrl}$ and that the only component of $\vec{\mathbf{m}}_{ctrl}$ responsible for producing torque is the component perpendicular

to $\vec{\mathbf{B}}_B$. By premultiplying the previous equation by the local geomagnetic field and using the triple cross product rule $\vec{\mathbf{a}} \times (\vec{\mathbf{b}} \times \vec{\mathbf{c}}) = \vec{\mathbf{b}}(\vec{\mathbf{a}} \cdot \vec{\mathbf{c}}) - \vec{\mathbf{c}}(\vec{\mathbf{a}} \cdot \vec{\mathbf{b}})$ gives

$$\vec{\mathbf{B}}_B \times \vec{\mathbf{u}} = \vec{\mathbf{B}}_B \times \left(\vec{\mathbf{m}}^{\perp} \times \vec{\mathbf{B}}_B\right) = \vec{\mathbf{m}}^{\perp}\left(\vec{\mathbf{B}}_B \cdot \vec{\mathbf{B}}_B\right) - \vec{\mathbf{B}}_B\left(\vec{\mathbf{B}}_B \cdot \vec{\mathbf{m}}^{\perp}\right) = \vec{\mathbf{m}}^{\perp}\left\|\vec{\mathbf{B}}_B\right\|^2 \tag{5.15}$$

By making $\vec{\mathbf{u}} = \vec{\boldsymbol{\tau}}_{des}$, an expression for the magnetorquers' magnetic dipole moment control law can be found:

$$\vec{\mathbf{m}}_{ctrl} = \vec{\mathbf{m}}^{\perp} = \frac{\vec{\mathbf{B}}_B \times \vec{\boldsymbol{\tau}}_{des}}{\left\|\vec{\mathbf{B}}_B\right\|^2} \tag{5.16}$$

It is possible to see that if the desired torque $\vec{\boldsymbol{\tau}}_{des}$ is along the same direction as the local geomagnetic field, the dipole moment $\vec{\mathbf{m}}_{ctrl} = \vec{\mathbf{0}}$ and no control torque $\vec{\mathbf{u}}$ will be generated due to the lack of controllability in the direction of the magnetic field. On the other hand, if the desired control torque is perpendicular to the local geomagnetic field, the control torque complies with the desired torque and Equation (5.16) is exact.

The component of the desired control torque which is parallel to the sliding variable $\vec{\mathbf{s}}$ is the only component responsible for decreasing the distance of the satellite trajectory to the sliding manifold, therefore, the control torque only needs to compensate for the component of the desired torque which is parallel to $\vec{\mathbf{s}}$ [40]. Consider the following Lyapunov candidate function:

$$\mathcal{V} = \frac{1}{2}\vec{\mathbf{s}}^{\top}\mathbf{J}\vec{\mathbf{s}} \tag{5.17}$$

whose derivative with respect to time is

$$\dot{\mathcal{V}} = \frac{1}{2}(\dot{\vec{\mathbf{s}}}^{\top}\mathbf{J}\vec{\mathbf{s}} + \vec{\mathbf{s}}^{\top}\mathbf{J}\dot{\vec{\mathbf{s}}}) = \frac{1}{2}\left[\left(\dot{\vec{\mathbf{s}}}^{\top}\mathbf{J}\vec{\mathbf{s}}\right)^{\top} + \vec{\mathbf{s}}^{\top}\mathbf{J}\dot{\vec{\mathbf{s}}}\right] = \vec{\mathbf{s}}^{\top}\mathbf{J}\dot{\vec{\mathbf{s}}} \tag{5.18}$$

where the second equality is true due to the fact that the product $\dot{\vec{\mathbf{s}}}^{\top}\mathbf{J}\vec{\mathbf{s}}$ is a scalar and the transpose of a scalar is equal to that same scalar. By using Equation (5.11) in Equation (5.18), the previous equation gets

$$\dot{\mathcal{V}} = \vec{\mathbf{s}}^{\top}(-\vec{\boldsymbol{\tau}}_{eq} + \vec{\mathbf{u}}) = \vec{\mathbf{s}}^{\top}(-\vec{\boldsymbol{\tau}}_{eq} + \vec{\boldsymbol{\tau}}_{eq}^{\parallel} - \lambda\vec{\mathbf{s}}) = \vec{\mathbf{s}}^{\top}(-\vec{\boldsymbol{\tau}}_{eq}^{\perp} - \lambda\vec{\mathbf{s}}) = -\vec{\mathbf{s}}^{\top}\lambda\vec{\mathbf{s}} \tag{5.19}$$

where the equivalent torque $\vec{\boldsymbol{\tau}}_{eq} = \vec{\boldsymbol{\tau}}_{eq}^{\parallel} + \vec{\boldsymbol{\tau}}_{eq}^{\perp}$ was decomposed into two components, a component $\vec{\boldsymbol{\tau}}_{eq}^{\parallel}$ parallel to the sliding variable $\vec{\mathbf{s}}$, and a component $\vec{\boldsymbol{\tau}}_{eq}^{\perp}$ perpendicular to the vector $\vec{\mathbf{s}}$. The control torque defined only compensates for the parallel component of the desired torque: $\vec{\mathbf{u}} = \vec{\boldsymbol{\tau}}_{des}^{\parallel} = \vec{\boldsymbol{\tau}}_{eq}^{\parallel} - \lambda\vec{\mathbf{s}}$. The time derivative of the candidate Lyapunov function from Equation (5.19) is negative definite, which according to Theorem $3$ proves that the equilibrium $\vec{\mathbf{s}} = \vec{\mathbf{0}}$ is global asymptotically stable when the control torque only compensates for the parallel component of the desired torque. The expression for the magnetorquers' magnetic dipole moment, Equation (5.16), is then rewritten as

$$\vec{\mathbf{m}}_{ctrl} = \frac{\vec{\mathbf{B}}_B \times \vec{\boldsymbol{\tau}}_{des}^{\parallel}}{\left\|\vec{\mathbf{B}}_B\right\|^2} \qquad \vec{\boldsymbol{\tau}}_{des}^{\parallel} = \frac{\vec{\boldsymbol{\tau}}_{des} \cdot \vec{\mathbf{s}}}{\left\|\vec{\mathbf{s}}\right\|^2}\vec{\mathbf{s}} = \frac{\vec{\boldsymbol{\tau}}_{eq} \cdot \vec{\mathbf{s}}}{\left\|\vec{\mathbf{s}}\right\|^2}\vec{\mathbf{s}} - \lambda\vec{\mathbf{s}} = \vec{\boldsymbol{\tau}}_{eq}^{\parallel} - \lambda\vec{\mathbf{s}} \tag{5.20}$$

The stability of the sliding condition from Equation (5.13) can now be investigated by using the magnetic actuation law from Equation (5.20). By considering again the candidate Lyapunov function from Equation (5.17) and by substituting Equation (5.11), while using the results from Equations (5.14) and (5.20), in the time derivative of the candidate Lyapunov function given by Equation (5.18) yields

$$\dot{\mathcal{V}} = \vec{\mathbf{s}}^{\top}(-\vec{\boldsymbol{\tau}}_{eq} + \vec{\mathbf{u}}) = \vec{\mathbf{s}} \cdot \left( -\vec{\boldsymbol{\tau}}_{eq} + \frac{\vec{\mathbf{B}}_B \times \vec{\boldsymbol{\tau}}^{\parallel}_{eq}}{\left\|\vec{\mathbf{B}}_B\right\|^2} \times \vec{\mathbf{B}}_B - \lambda \frac{\vec{\mathbf{B}}_B \times \vec{\mathbf{s}}}{\left\|\vec{\mathbf{B}}_B\right\|^2} \times \vec{\mathbf{B}}_B \right) \tag{5.21}$$

By using the scalar triple product or mixed product permutation property $\vec{\mathbf{a}} \cdot (\vec{\mathbf{b}} \times \vec{\mathbf{c}}) = \vec{\mathbf{b}} \cdot (\vec{\mathbf{c}} \times \vec{\mathbf{a}})$ in the last term of the previous equation gives

$$\dot{\mathcal{V}} = -\vec{\mathbf{s}} \cdot \vec{\boldsymbol{\tau}}_{eq} + \vec{\mathbf{s}} \cdot \frac{\vec{\mathbf{B}}_B \times \vec{\boldsymbol{\tau}}^{\parallel}_{eq}}{\left\|\vec{\mathbf{B}}_B\right\|^2} \times \vec{\mathbf{B}}_B - \frac{\lambda}{\left\|\vec{\mathbf{B}}\right\|^2} \left(\vec{\mathbf{B}}_B \times \vec{\mathbf{s}}\right) \cdot \left(\vec{\mathbf{B}}_B \times \vec{\mathbf{s}}\right) \tag{5.22}$$

The last term on the right-hand side, which corresponds to the term of the sliding condition, is negative semidefinite, being equal to zero when $\vec{\mathbf{B}}_B$ and $\vec{\mathbf{s}}$ are parallel. Since the geomagnetic field changes periodically in the orbit frame $O$, it can be proved that the sliding condition is globally asymptotically stable to the equilibrium $\vec{\mathbf{s}} = \vec{\mathbf{0}}$ [40]. Note that if the sliding condition was discontinuous, using the sign function for instance, such that $\vec{\boldsymbol{\tau}}^{\parallel}_{des} = \vec{\boldsymbol{\tau}}^{\parallel}_{eq} - \lambda \operatorname{sign}(\vec{\mathbf{s}})$, the term due to the sliding condition would write

$$-\frac{\lambda}{\left\|\vec{\mathbf{B}}_B\right\|^2} \left(\vec{\mathbf{B}}_B \times sign(\vec{\mathbf{s}})\right) \cdot \left(\vec{\mathbf{B}}_B \times \vec{\mathbf{s}}\right)$$

where it can be seen that there can be vectors $\vec{\mathbf{s}}$ and $\vec{\mathbf{B}}_B$ such that $(\vec{\mathbf{B}}_B \times \operatorname{sign}(\vec{\mathbf{s}})) \cdot (\vec{\mathbf{B}}_B \times \vec{\mathbf{s}}) < 0$ making $\dot{\mathcal{V}} > 0$ and therefore the sliding condition cannot be proved stable. The value of $\lambda$ should be chosen such that the sliding condition can be the dominant term of Equation (5.22) so that $\dot{\mathcal{V}}$ can be negative. In practice, in spite of the controller developed is not guaranteed to be Lyapunov stable, due to the periodic nature of the magnetic field and the fact that the first two terms of Equation (5.22) can be made relatively small by choosing an appropriate value for $\lambda$, this controller has proven to be stable in all the simulations tested as it will be seen in Chapter 7. It will also be seen in Chapter 7 that too low values of $\lambda$ will increase the oscillations of this controller as well as the pointing error due to the fact that, as explained before, the term corresponding to the sliding condition becomes less dominant as the value of $\lambda$ decreases.

## 5.2 Linear Quadratic Regulator (LQR)

In this section, three types of LQR controllers are going to be presented. The first to be presented is the Infinite Horizon Controller (IHC) (Section 5.2.1), the second is the Finite Horizon Controller (FHC) (Section 5.2.2) and the last controller presented is the Constant Gain Controller (CGC) (Section 5.2.3). All these algorithms will take advantage of the fact that the variation of the geomagnetic field in the orbital

frame $O$ is approximately periodic with a period equal to the spacecraft's orbital period $T_{orb}$ as seen in Figure 5.1a. These controllers will make use of the linear model developed in Section 3.5 and repeated here again as

$$\dot{\mathbf{x}} = \boldsymbol{A}\mathbf{x} + \begin{bmatrix} \mathbf{J}^{-1}\vec{\mathbf{u}} \\ \mathbf{0}_{3\times 3} \end{bmatrix} \tag{5.23}$$

where $\mathbf{x}$ and $\boldsymbol{A}$ are defined in Equations (3.67) and (3.69) respectively and the control torque $\vec{\mathbf{u}}$ in Equation (3.71). As seen before, in Equation (5.14), the magnetic moment $\vec{\mathbf{m}}^{\parallel}$ generated in the direction parallel to the local geomagnetic field vector has no influence in the satellite motion. As such, and following the recommendation encountered in reference [40], the magnetic dipole moment $\vec{\mathbf{m}}_{ctrl}$ is going to be mapped using a new control signal $\vec{\mathbf{m}}_{cmd}$ according to Equation (5.24), where the subscript $cmd$ stands for commanded dipole moment. This mapping will allow us to obtain a more power efficient controller.

$$\vec{\mathbf{m}}_{cmd} \mapsto \vec{\mathbf{m}}_{ctrl} : \vec{\mathbf{m}}_{ctrl} = \frac{\vec{\mathbf{m}}_{cmd} \times \vec{\mathbf{B}}_B}{\left\|\vec{\mathbf{B}}_B\right\|} = -\frac{\vec{\mathbf{B}}_B \times \vec{\mathbf{m}}_{cmd}}{\left\|\vec{\mathbf{B}}_B\right\|} \tag{5.24}$$

Using the new control signal, $\vec{\mathbf{m}}_{cmd}$, the control torque $\vec{\mathbf{u}}$ from Equation (3.71) can be written as

$$\vec{\mathbf{u}} = -\frac{1}{\left\|\vec{\mathbf{B}}_O\right\|}\left(\vec{\mathbf{B}}_O \times \vec{\mathbf{m}}_{cmd}\right) \times \vec{\mathbf{B}}_O = \frac{1}{\left\|\vec{\mathbf{B}}_O\right\|}\vec{\mathbf{B}}_O \times \left(\vec{\mathbf{B}}_O \times \vec{\mathbf{m}}_{cmd}\right) = \frac{1}{\left\|\vec{\mathbf{B}}_O\right\|}\left[\vec{\mathbf{B}}_O\times\right]\left[\vec{\mathbf{B}}_O\times\right]\vec{\mathbf{m}}_{cmd} \tag{5.25}$$

where $\vec{\mathbf{B}}_B$ was replaced by $\vec{\mathbf{B}}_O$ according to the development done in Equation (3.71). The linear model for the ORCASat is then given by

$$\dot{\mathbf{x}}(t) = \boldsymbol{A}\mathbf{x}(t) + \boldsymbol{B}(t)\vec{\boldsymbol{u}}(t) \tag{5.26}$$

where the control vector $\vec{\boldsymbol{u}}(t)$ and the control matrix $\boldsymbol{B}(t)$ are now given by

$$\vec{\boldsymbol{u}}(t) = \vec{\mathbf{m}}_{cmd}(t) \tag{5.27}$$

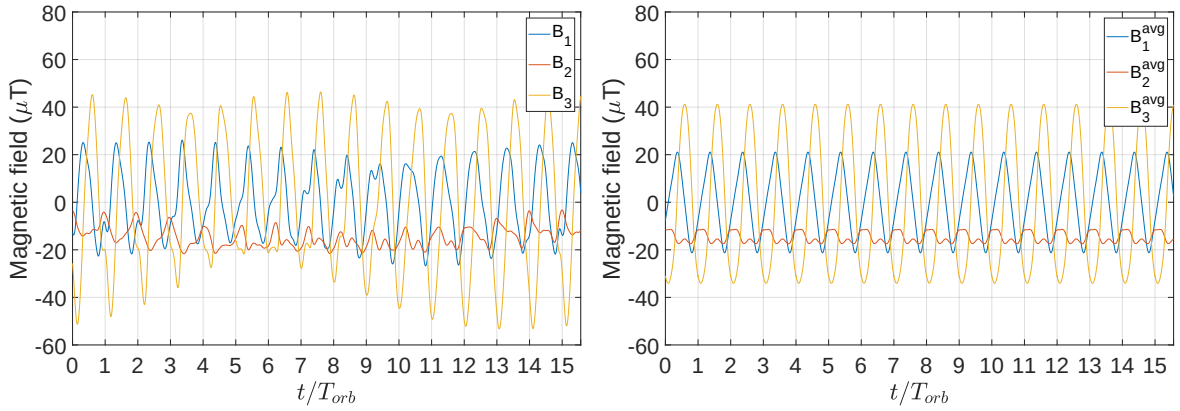
$$\boldsymbol{B}(t) = \begin{bmatrix} \dfrac{\mathbf{J}^{-1}}{\left\|\vec{\mathbf{B}}_O(t)\right\|}\left[\vec{\mathbf{B}}_O(t)\times\right]\left[\vec{\mathbf{B}}_O(t)\times\right] \\ \mathbf{0}_{3\times 3} \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{J}^{-1}}{\left\|\vec{\mathbf{B}}_O(t)\right\|}\begin{bmatrix} -B_2^2 - B_3^2 & B_1B_2 & B_1B_3 \\ B_1B_2 & -B_1^2 - B_3^2 & B_2B_3 \\ B_1B_3 & B_2B_3 & -B_1^2 - B_2^2 \end{bmatrix} \\ \mathbf{0}_{3\times 3} \end{bmatrix} \tag{5.28}$$

where $B_i$ are the components of the local geomagnetic field vector $\vec{\mathbf{B}}_O(t) = \begin{bmatrix} B_1(t) & B_2(t) & B_3(t) \end{bmatrix}^\top$.

### 5.2.1 Infinite Horizon Controller (IHC)

Due to the periodic nature of the geomagnetic field seen from the orbit frame, the linearized model of the satellite can be considered as periodic. It is, however, necessary to find an ideally periodic counterpart of the real geomagnetic field. This is done by averaging the geomagnetic field over $N = 15$ orbits which are contained in a $24\text{h}$ period

$$\vec{\mathbf{B}}_O^{avg}(t) = \frac{1}{N}\sum_{i=1}^{N}\vec{\mathbf{B}}_O^{i}(t) \tag{5.29}$$




(a) True geomagnetic field.

(b) Averaged geomagnetic field.

Figure 5.1: Geomagnetic field vector in the ORCASat's orbit viewed from the orbit frame $O$ during a $24$h period in $15/09/2019$. The true geomagnetic field was obtained using the $2015$ World Magnetic Model (WMM) [58].

where $\vec{\mathbf{B}}_O^i(t)$ is the magnetic field of each orbit within the $24$h period corresponding to the time interval $t \in [\tau + (i-1)T_{orb}; \tau + iT_{orb}[$. Figure 5.1b depicts the averaged geomagnetic field $\vec{\mathbf{B}}_O^{avg}(t)$. The resultant linear periodic system is given by

$$\dot{\mathbf{x}}(t) = \boldsymbol{A}\mathbf{x}(t) + \hat{\boldsymbol{B}}(t)\vec{\boldsymbol{u}}(t) \tag{5.30}$$

where $\hat{\boldsymbol{B}}(t)$ is given by Equation (5.28) after substituting the magnetic field $\vec{\mathbf{B}}_O(t)$ by the magnetic field $\vec{\mathbf{B}}_O^{avg}(t)$ from Equation (5.29). The IHC consists of a time-varying gain controller that relies on the minimization of the cost function

$$J\left(\vec{\boldsymbol{u}}(t)\right) = \frac{1}{2}\int_{\tau}^{\infty}\left[\mathbf{x}^{\top}(t)\boldsymbol{Q}(t)\mathbf{x}(t) + \vec{\boldsymbol{u}}^{\top}(t)\vec{\boldsymbol{u}}(t)\right]dt \tag{5.31}$$

where the error weighted matrix $\boldsymbol{Q}$ is a real positive semidefinite $n \times n$ matrix, being $n = 6$ the dimension of the state vector $\mathbf{x}$. The control law is then given by [59]

$$\vec{\boldsymbol{u}}(t) = \vec{\mathbf{m}}_{cmd}(t) = -\boldsymbol{K}(t)\mathbf{x}(t) = -\hat{\boldsymbol{B}}^{\top}(t)\hat{\boldsymbol{P}}(t)\mathbf{x}(t) \tag{5.32}$$

with

$$\hat{\boldsymbol{P}}(t) = \sum_{i=0}^{\infty}\boldsymbol{P}_{\infty}(t - iT_{orb}) \tag{5.33}$$

where, according to [40], the solution $\boldsymbol{P}_{\infty}(t)$ is obtained by solving the differential equation of the following iterative process for an arbitrary final condition:

$$\dot{\boldsymbol{P}}_{i+1}(t) = -\boldsymbol{P}_{i+1}(t)\boldsymbol{A}_i(t) - \boldsymbol{A}_i^{\top}(t)\boldsymbol{P}_{i+1}(t) - \boldsymbol{K}_i^{\top}(t)\boldsymbol{K}_i(t) - \boldsymbol{Q}(t) \tag{5.34}$$

with

$$\boldsymbol{K}_i(t) = \hat{\boldsymbol{B}}^{\top}(t)\boldsymbol{P}_i(t) \tag{5.35}$$

$$\boldsymbol{A}_i(t) = \boldsymbol{A} - \hat{\boldsymbol{B}}(t)\boldsymbol{K}_i(t) \tag{5.36}$$

The first iteration of Equation (5.34) is done by solving the regular Differential Riccati Equation (DRE) given by Equation (5.39) using as final condition $\boldsymbol{P}_{f_1} = \boldsymbol{Q}$. Due to the solution periodicity, the following iteration steps (using Equation (5.34)) were done using as final condition $\boldsymbol{P}_{f_{i+1}} = \boldsymbol{P}_i(\tau)$. Finally, the control dipole moment $\vec{\mathbf{m}}_{ctrl}$ is obtained from the control vector $\vec{\mathbf{m}}_{cmd}$ according to Equation (5.24).

### 5.2.2 Finite Horizon Controller (FHC)

Instead of using the periodic counterpart of the Earth's magnetic field vector, $\vec{\mathbf{B}}_O^{avg}(t)$, as done is the IHC, the FHC proposes using the real geomagnetic field. For each orbit the FHC is the time-varying gain controller that relies on the minimization of the following cost function:

$$J\left(\vec{\boldsymbol{u}}(t)\right) = \frac{1}{2}\mathbf{x}_f^\top \boldsymbol{P}_f \mathbf{x}_f + \frac{1}{2}\int_\tau^{\tau+T_{orb}} \left[\mathbf{x}^\top(t)\boldsymbol{Q}(t)\mathbf{x}(t) + \vec{\boldsymbol{u}}^\top(t)\vec{\boldsymbol{u}}(t)\right] dt \tag{5.37}$$

where $\mathbf{x}_f \equiv \mathbf{x}(\tau + T_{orb})$. The optimal control vector $\vec{\boldsymbol{u}}(t)$ is then given for each orbit ($t \in [\tau\,;\tau + T_{orb}[$) by

$$\vec{\boldsymbol{u}}(t) = \vec{\mathbf{m}}_{cmd}(t) = -\boldsymbol{K}(t)\mathbf{x}(t) = -\boldsymbol{B}^\top(t)\boldsymbol{P}(t)\mathbf{x}(t) \tag{5.38}$$

where $\boldsymbol{P}$ is the solution of the DRE

$$\dot{\boldsymbol{P}}(t) = -\boldsymbol{P}(t)\boldsymbol{A} - \boldsymbol{A}^\top\boldsymbol{P}(t) + \boldsymbol{P}(t)\boldsymbol{B}(t)\boldsymbol{B}(t)^\top\boldsymbol{P}(t) - \boldsymbol{Q}(t) \tag{5.39}$$

with final condition

$$\boldsymbol{P}(\tau + T_{orb}) = \boldsymbol{P}_f \tag{5.40}$$

The control dipole moment $\vec{\mathbf{m}}_{ctrl}$ is computed using the control vector $\vec{\mathbf{m}}_{cmd}$ according to Equation (5.24). The final condition can be seen as a design parameter. However, the difference $\boldsymbol{P}_f - \boldsymbol{P}(\tau)$ should be positive semidefinite to guarantee that $\boldsymbol{P}(t)$ is stabilizing and that the control law $\vec{\boldsymbol{u}}(t) = -\boldsymbol{B}^\top(t)\boldsymbol{P}(t)\mathbf{x}(t)$ is stable [40]. The method for finding $\boldsymbol{P}_f$ in this work consisted of solving the Algebraic Riccati Equation (ARE) present in Equation (5.45) for $t = \tau + T_{orb}$, i.e., by making $\bar{\boldsymbol{B}} = \boldsymbol{B}(\tau + T_{orb})$ in Equation (5.45). The solution obtained for the Riccati matrix, $\boldsymbol{P}_{ARE}$, is then multiplied by a positive scalar gain $K$:

$$\boldsymbol{P}_f = K\boldsymbol{P}_{ARE} \tag{5.41}$$

### 5.2.3 Constant Gain Controller (CGC)

The CGC is developed with the aim of avoiding all the necessary computations required by the two previous controllers. The first step of this controller consists of converting the linear periodic time-variant system defined in Equation (5.30) to a linear time-invariant system. This is done by averaging the periodic system matrix $\boldsymbol{A}(t)$ and the periodic control matrix $\hat{\boldsymbol{B}}(t)$ for one orbit of period $T_{orb}$ giving

$$\dot{\mathbf{x}}(t) = \left(\frac{1}{T_{orb}}\int_\tau^{\tau+T_{orb}} \boldsymbol{A}(t)dt\right)\mathbf{x}(t) + \left(\frac{1}{T_{orb}}\int_\tau^{\tau+T_{orb}} \hat{\boldsymbol{B}}(t)dt\right)\vec{\boldsymbol{u}} = \bar{\boldsymbol{A}}\mathbf{x}(t) + \bar{\boldsymbol{B}}\vec{\boldsymbol{u}}(t) \tag{5.42}$$

In the ORCASat case, since the state matrix $\boldsymbol{A}$ is already time-invariant, $\bar{\boldsymbol{A}} = \boldsymbol{A}$. The control vector is obtained by minimizing the following cost function

$$J\left(\vec{\boldsymbol{u}}(t)\right) = \frac{1}{2}\int_{\tau}^{\infty}\left[\mathbf{x}(t)^{\top}\boldsymbol{Q}\mathbf{x}(t) + \vec{\boldsymbol{u}}(t)^{\top}\vec{\boldsymbol{u}}(t)\right]dt \tag{5.43}$$

The optimal control vector $\vec{\boldsymbol{u}}(t)$ is then given by

$$\vec{\boldsymbol{u}}(t) = \vec{\mathbf{m}}_{cmd}(t) = -\boldsymbol{K}\mathbf{x}(t) = -\bar{\boldsymbol{B}}^{\top}\boldsymbol{P}\mathbf{x}(t) \tag{5.44}$$

where $\boldsymbol{P}$ is obtained by solving the ARE

$$\boldsymbol{P}\bar{\boldsymbol{A}} + \bar{\boldsymbol{A}}^{\top}\boldsymbol{P} - \boldsymbol{P}\bar{\boldsymbol{B}}\bar{\boldsymbol{B}}^{\top}\boldsymbol{P} + \boldsymbol{Q} = 0 \tag{5.45}$$

Finally, the control dipole moment $\vec{\mathbf{m}}_{ctrl}$ is computed according to Equation (5.24).

## 5.3 Detumbling Controller

The proposed control law for the detumbling controller is found in [60]. The control dipole moment $\vec{\mathbf{m}}_{ctrl}$ is given by

$$\vec{\mathbf{m}}_{ctrl} = -\frac{K_d}{\left\|\vec{\mathbf{B}}_B\right\|^2}\vec{\mathbf{B}}_B \times \vec{\boldsymbol{\omega}}_B^{B/I} \tag{5.46}$$

where $K_d$ is a positive scalar gain. The resulting control torque $\vec{\mathbf{u}}$ is then given by replacing the dipole moment $\vec{\mathbf{m}}_{ctrl}$ from Equation (5.46) into Equation (3.40) yielding

$$\vec{\mathbf{u}} = -\frac{K_d}{\left\|\vec{\mathbf{B}}_B\right\|^2}\left(\vec{\mathbf{B}}_B \times \vec{\boldsymbol{\omega}}_B^{B/I}\right) \times \vec{\mathbf{B}}_B = -K_d\left(\mathbf{I}_3 - \hat{\mathbf{B}}\hat{\mathbf{B}}^{\top}\right)\vec{\boldsymbol{\omega}}_B^{B/I} \tag{5.47}$$

where the last equality was obtained by expanding the triple cross product and by defining $\hat{\mathbf{B}} \equiv \frac{\vec{\mathbf{B}}_B}{\left\|\vec{\mathbf{B}}_B\right\|}$. Reference [60] also provides a method for computing the gain $K_d$:

$$K_d = \frac{4\pi}{T_{orb}}(1 + \sin\xi)J_{min} \tag{5.48}$$

where $T_{orb}$ is the orbital period in seconds, $\xi$ represents the inclination of the spacecraft orbit relative to the geomagnetic equatorial plane and $J_{min}$ is the minimum principal moment of inertia of the spacecraft. The stability of this controller can be proved by using the following candidate Lyapunov function and its respective time derivative:

$$\mathcal{V} = \frac{1}{2}\vec{\boldsymbol{\omega}}_B^{B/I} \cdot \left(\mathbf{J}_B\vec{\boldsymbol{\omega}}_B^{B/I}\right) \qquad\qquad \dot{\mathcal{V}} = \vec{\boldsymbol{\omega}}_B^{B/I} \cdot \left(\mathbf{J}_B\dot{\vec{\boldsymbol{\omega}}}_B^{B/I}\right) \tag{5.49}$$

where $\dot{\mathcal{V}}$ was obtained similarly to Equation (5.18). By using the dynamics equation, Eq. (3.26), $\dot{\mathcal{V}}$ gives:

$$\dot{\mathcal{V}} = \vec{\boldsymbol{\omega}}_B^{B/I} \cdot \left[ (\mathbf{J}_B \vec{\boldsymbol{\omega}}_B^{B/I} + \vec{\mathbf{h}}_B^s) \times \vec{\boldsymbol{\omega}}_B^{B/I} + \vec{\mathbf{u}} \right] = \vec{\boldsymbol{\omega}}_B^{B/I} \cdot \vec{\mathbf{u}} = \left( \vec{\boldsymbol{\omega}}_B^{B/I} \right)^\top \vec{\mathbf{u}} \tag{5.50}$$

Using Equation (5.47) in the previous equation, the time derivative of $\mathcal{V}$ is

$$\dot{\mathcal{V}} = -K_d (\vec{\boldsymbol{\omega}}_B^{B/I})^\top (\mathbf{I}_3 - \hat{\mathbf{B}}\hat{\mathbf{B}}^\top) \vec{\boldsymbol{\omega}}_B^{B/I} \tag{5.51}$$

It is possible to see that $\dot{\mathcal{V}}$ is negative semi-definite, having value $0$ in the equilibrium point $\vec{\boldsymbol{\omega}}_B^{B/I} = \vec{\mathbf{0}}$ and when the spacecraft angular velocity $\vec{\boldsymbol{\omega}}_B^{B/I}$ is parallel to $\vec{\mathbf{B}}$, $\vec{\boldsymbol{\omega}}_B^{B/I} = \omega \hat{\mathbf{B}}$. In this last case, the magnetic coils cannot deliver any torque component about the current spacecraft's spin axis and thus they are not able to decrease its angular speed. In practice, this last case is not a concern and the control law can be proved global asymptotic stable [60].

## 5.4 Controllers Implementation

The differential equations given in this chapter to obtain the LQR gains, Equations (5.34) and (5.39), were integrated backwards using the built-in Matlab ode solver `ode45`. The solution was computed for every time point of the simulated geomagnetic field. The Matlab function `interp1` was also used in order for the `ode45` solver to interpolate the time-dependent geomagnetic field during its internal steps. The Riccati matrix solution $\boldsymbol{P}$ of the ARE represented by Equation (5.45), used for obtaining the gain $\boldsymbol{K}$ of the CGC, was computed using the built-in Matlab function `care`. The $\boldsymbol{Q}$ matrix selected for the LQR controllers is given by

$$\boldsymbol{Q} = K_Q \operatorname{diag}\left( \begin{bmatrix} 1000 & 1000 & 1000 & 1 & 1 & 1 \end{bmatrix} \right) \tag{5.52}$$

where $K_Q$ is a positive scalar gain and the diagonal matrix was chosen empirically. The geomagnetic field $\vec{\mathbf{B}}_O$ was obtained in Matlab/Simulink using the $2015$ WMM block by propagating the orbit of the ORCASat. The orbital parameters are given in Table 7.1. All the Earth-pointing controllers used the minimum rotation method where the quaternion error $\mathbf{q}_e$ representing the minimum rotation path is the one that has $q_{e_4} > 0$. This can be better understood by noting that the attitude quaternion $\mathbf{q}(\vec{\mathbf{e}}, \theta)$ corresponding to a short $\theta^\circ$ rotation, with $0 < \theta < 180^\circ$ , $q_{e_4} > 0$, physically represents the same attitude as the attitude quaternion $\mathbf{q}(-\vec{\mathbf{e}}, 360 - \theta)$ corresponding to a long $(360 - \theta)^\circ$ rotation in the opposite direction. Using Equation (2.16) it can be seen that $\mathbf{q}(\vec{\mathbf{e}}, \theta) = -\mathbf{q}(-\vec{\mathbf{e}}, 360 - \theta)$. Therefore, if $q_{e_4} < 0$, then the following operation is done:

$$\mathbf{q}_e^{new} = -\mathbf{q}_e \tag{5.53}$$

where $\mathbf{q}_e^{new}$ is the new attitude quaternion representing the attitude error. Although it has not been done in this work, for simplicity reasons when testing the controllers in Matlab/Simulink, in a real application, the gain matrix $\boldsymbol{K}$ of the IHC and FHC should be parameterized by the mean anomaly $M$ such that $\boldsymbol{K} = \boldsymbol{K}(M)$. The implementation of the different control algorithms is summarized in Appendix A by Figures A.2 and A.3 while the code used for the determination of the LQR gains is given in Appendix E.

# Chapter 6

# Simulation Environment

In order to test and develop the ADCS of the ORCASat, a simulator was developed in Matlab/Simulink. This simulator must be able to generate a truth-model capable of describing the satellite's orbit motion and attitude including realistic environmental models. It must also include sensor and actuator models that connect the estimation and control algorithms to the data generated by the truth-model. The conception of this simulator had major contributions from previous Master's students, namely Duarte Rondão in [49] and Bernardo Lobo-Fernandes in [48]. In this thesis, the Simulink model was changed and improved. The most important alterations are the following: The division between the ADCS software and the simulator environment; Improvements on the sun sensor hardware model; Bug-fixing and performance improvements; Implementation of the ADCS operation modes and routines; Implementation of the transient-state for the momentum wheel; Adaptation of the attitude estimation algorithms for multiple sun sensor readings.

The simulator is divided into two major groups. The first group is responsible for the generation of the data feeding the ADCS and comprises the Spacecraft Mechanics Simulator (SMS) block, described in Section 6.1, the Sensor Model block, and the Actuator Model block, which are described in Section 6.2. The second major group, which will be described in Section 6.3, is the ADCS environment where the estimation and control algorithms were implemented. The ADCS environment also includes an onboard Orbital Propagator as well as a block responsible for the transition between the different stages of the ADCS, the ADCS Controller block.

## 6.1 Spacecraft Mechanics Simulator

The SMS block is responsible for applying the theory described in Chapter 3 to compute the motion and propagate the attitude of the spacecraft in time. The orbital dynamics equation and the attitude kinematics and dynamics equations, Equations (3.6), (3.9) and (3.26) respectively, are integrated giving the orbital position and velocity of the spacecraft as well as its angular rates and attitude. The Julian date and the different reference frames described in Section 2.1 are also computed in this block as well as the different environmental models, such as the Earth's magnetic field, the sun vector, the atmospheric

model and the gravity model. The spacecraft's perturbation torques and forces originated from these environmental models are also computed in the SMS block. This block is also responsible for determining if the ORCASat is in eclipse or not by making use of a ray-sphere intersection method as described in [49]. This method returns a Boolean flag, *Lit*, which will have value $0$ if the CubeSat is in eclipse and value $1$ otherwise. The code used to implement this method is given in Appendix A by Figure A.1. The different environmental models used and the respective references are displayed in Table 6.1.

Table 6.1: Environmental models used in the ADCS simulator.

| Variable to compute | Model used | Reference |
|---|---|---|
| Earth's magnetic field $\vec{\mathbf{B}}$ | World Magnetic Model (WMM) 2015 | [58] |
| Sun vector $\vec{\mathbf{s}}_{\odot}$ | Jet Propulsion Laboratory (JPL) ephemeris model DE432t | [61] |
| Atmospheric Density $\rho$ | NRLMSISE-00[1] atmosphere model | [62] |
| Earth's gravity acceleration $\boldsymbol{\nabla} U$ | Earth Gravitational Model 2008 (EGM2008) | [63] |

## 6.2 Sensors and Actuators Models

### 6.2.1 Hardware Selection

The selection of the hardware to equip the ORCASat was made according to the trade-off analysis in [64]. The chosen option consisted of a balanced approach between using Commercial Off-The-Shelf (COTS) components and in-house made components. The philosophy of this choice consisted of potential cost savings by developing some of the components in-house that have similar fidelity to more costly commercial options. More details about the hardware selection can be found in [48].

#### Sensors

The sensor suite selected for the ORCASat includes four sun sensors, a magnetometer, a gyroscope, and a GNSS receiver.

The chosen sun sensor is the Hyperion SS200. This sensor has flight heritage since 2018 and a field of view up to $110^{\circ}$ allowing for a smooth transition between measurements of two sun sensors in adjacent faces of the CubeSat [65]. The noise standard deviation is $0.5^{\circ}$ in the $\theta$ and $\phi$ angles shown in Figure 6.2 and the sampling rate selected is $10$Hz although it is capable of up to $100$Hz. As stated before, four sun sensors will be used in the ORCASat. This was the minimum number of sensors determined by a simulation in STK[2] to maintain full sensor visibility while in nadir pointing orientation for a one year mission [66]. These sensors will be placed in the faces normal to the $\pm\hat{\mathbf{b}}_1$ and $\pm\hat{\mathbf{b}}_2$ body frame axes defined in Figure 2.2.

The magnetometer and the gyroscope are included in the iNEMO LSM9DS1 IMU from STMicroelectronics. This system provides a 3D digital linear acceleration sensor, a 3D digital angular rate sensor, and a 3D digital magnetic sensor. The linear acceleration sensor won't be used in the case of the

[1] US Naval Research Laboratory Mass Spectrometer and Incoherent Scatter Radar Exosphere (NRLMSISE)
[2] `https://www.agi.com/home`

ORCASat. The main characteristics taken from the datasheet [67] are the magnetometer's measurement range of $\pm 4$G and the sensitivity of $0.14$mG/LSB, the gyro's measurement range of $\pm 245\,°\mathrm{s}^{-1}$ and the gyro's sensitivity of $8.75$mdps/LSB. Both of these sensors have a $16$bit data output. The incorporation of these sensors in the ORCASat body was not defined at the time of the simulations presented in the next chapter.

The GNSS receiver chosen for the ORCASat is the NovAtel OEM719 Multi-Frequency GNSS Receiver. This component was chosen due to its reasonable cost when compared to its competitors and because it can be unlocked for orbital operations [48].

#### Actuators

The magnetorquer set will be built in-house, providing a cost-saving opportunity and will be constituted by three orthogonal elements with equivalent characteristics oriented in each of the satellite body axes. Two of the elements are ferromagnetic magnetorquers while the other one is an air-core magnetorquer. These magnetorquers should provide a dipole moment of $0.25\,\mathrm{A\,m}^2$ in each axis. Initial studies have shown that the specifications given are feasible to be accomplished by both types of magnetorquers [48].

The chosen momentum wheel for the ORCASat is the Hyperion RW210 with $3\,\mathrm{mN\,m\,s}$ of total momentum storage. This wheel can also be found in $1.5\,\mathrm{mN\,m\,s}$ and $6\,\mathrm{mN\,m\,s}$ variants and has flight heritage since 2017 [68]. The momentum wheel will be used to provide a momentum bias of $3\,\mathrm{mN\,m\,s}$ along the $-\hat{\mathbf{b}}_2$ axis of the ORCASat and thus helping to stabilize the CubeSat in the orbital plane. The wheel's rotation rate at nominal speed is $15000$rpm and it is capable of a maximum torque of $0.1\,\mathrm{mN\,m}$.

### 6.2.2 Hardware Models

While the true sun vector, the geomagnetic field, and the angular rates resolved in the body frame are computed in the Spacecraft Mechanics Simulator block, the simulated real output from the sun sensors, magnetometers, and gyroscopes are computed in the Sensor Model block using hardware models. Similarly, the real limitations of the actuators are simulated in the Actuator Model block. The hardware models follow from [49].

Figure 6.1 depicts the sun sensor's hardware model. The input of the sun sensor model is the unit

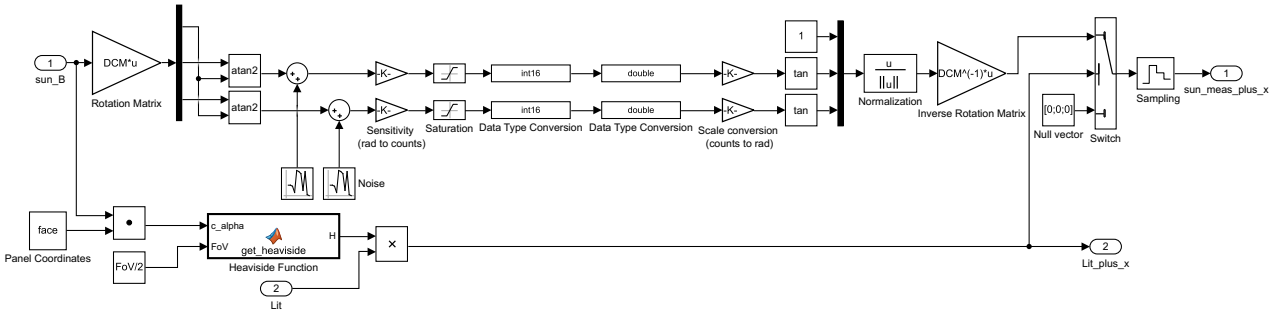



Figure 6.1: Matlab/Simulink sun sensor model.

sun vector computed in body frame coordinates by the Spacecraft Mechanics Simulator block. This unit vector is then transformed into the local sun sensor frame by using the appropriate rotation matrix and then it is converted into the two angles that define the sun vector orientation with respect to the normal of the sun sensor's face. This last transformation is given by

$$x = \frac{1}{\sqrt{1 + \tan^2(\theta) + \tan^2(\phi)}} \qquad y = \frac{\tan(\theta)}{\sqrt{1 + \tan^2(\theta) + \tan^2(\phi)}} \qquad z = \frac{\tan(\phi)}{\sqrt{1 + \tan^2(\theta) + \tan^2(\phi)}} \tag{6.1}$$

where the Cartesian coordinates $x$, $y$, $z$, and the angles $\theta$ and $\phi$ are defined according to Figure 6.2.

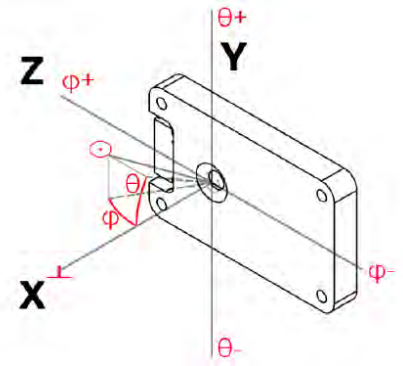



Figure 6.2: Sun sensor axes.

Both angles are corrupted by a zero-mean Gaussian white noise process. Similarly to the gyroscope and magnetometer models, the sun sensor is a digital sensor and thus it is subjected to quantization error. This quantization error is simulated using the sensitivity, saturation, data type conversion, and scale conversion blocks. The sensitivity block is responsible for converting the angle readings into unitless integer counts which are the data effectively measured and stored by the digital sensor and that depend on the number of bits of the Analog-to-Digital Converter (ADC) used to represent the analog value. The counts are then converted back to radians and transformed back to cartesian coordinates in the sun sensor frame. The input is then transformed back into the CubeSat body frame coordinate system. A switching logic was employed to detect if the sensor is illuminated by the sun or not. This switching logic is determined by the *Lit* flag introduced in Section 6.1 and by the angle between the sun vector and the normal to the sun sensor's face, the $x$-axis from Figure 6.2. If the *Lit* flag is zero, meaning that the satellite is in eclipse, or the angle between the normal to the sun sensor's face ($x$-axis from Figure 6.2) and the sun vector is bigger than half of the sensor's field of view, or both of the previous conditions simultaneously, the output of the sensor is a vector of zeros, else the output of the sun sensor is the perturbed sun vector. Lastly, the signal is sampled using a zero-order hold block. Since the ORCASat has four sun sensors, four block systems like the one in Figure 6.1 were implemented. Each one of these blocks uses different values for the panel coordinates and for the DCM. The different white noise generator blocks (Figure 6.1) should use different seeds. If the same seeds are used, the noise of each component $\theta$ and $\phi$ is not independent of each other. The seeds from the other white noise generator blocks present in the Simulink environment should also be different from each other for the same reason.

The magnetometer model is shown in Figure 6.3. The input of this model is the geomagnetic field

computed in the body frame $B$ by the SMS block. This input is first subjected to scale factor and misalignment errors and after it is corrupted by white noise and static bias. The Three-Axis Magnetometer (TAM) used in the ORCASat is a digital sensor and thus it is also subjected to quantization errors. Lastly, a zero-order hold block was added to simulate the sampling frequency of the sensor.

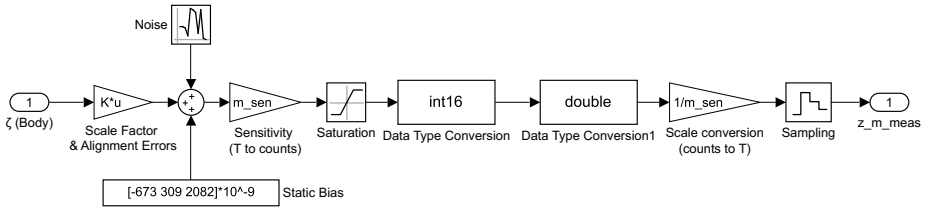



Figure 6.3: Matlab/Simulink magnetometer model.

The gyroscope model is detailed in Figure 6.4. The input of this model is the angular rate of the body frame with respect to the inertial frame given in body frame coordinates by the SMS block. Similarly to the magnetometer, the gyroscope is subjected to scale factor and misalignment errors being then corrupted by static bias and by noise. The gyroscope is affected by two different noise sources, the ARW and the RRW. The ARW and RRW noise parameters are multiplied by $1/\sqrt{\Delta t}$, where $\Delta t$ is the sampling rate of the gyroscope, giving the standard deviation driving the respective noise blocks responsible for computing a normally distributed random number with zero mean and unit standard deviation. Similarly to the other sensors, the gyroscope is also subjected to quantization errors. Finally, the input is sampled using the zero-order hold block according to the sampling rate of the sensor.

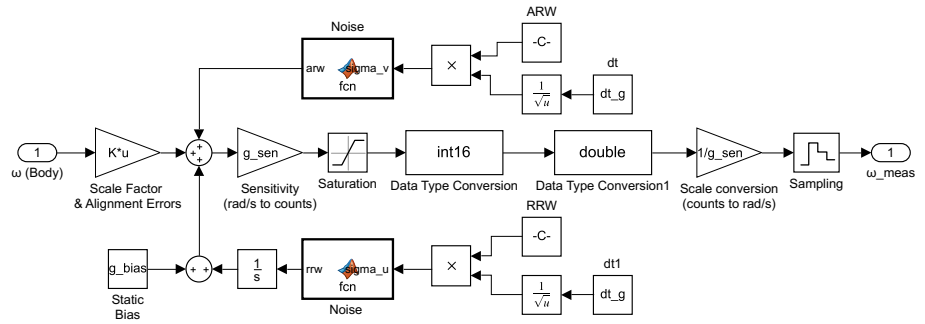



Figure 6.4: Matlab/Simulink gyroscope model.

The uncertainties associated with the GNSS receiver are not modelled in this work since they are typically very small. For instance, the accuracy of the selected GNSS receiver in Single Point L1 is $1.5\text{m}$ RMS according to the product sheet [69]. The magnetorquers were assumed to respond instantaneously and as previously said they are constrained to a maximum dipole moment of $0.25\text{A m}^2$ in each direction of the body axes which is enforced by using a saturation block. The momentum wheel is modelled using an integration block with saturation which is fed by the desired angular acceleration. The integration stops once the wheel achieves its nominal angular rate. Disturbance forces due to static and dynamic imbalance were not modelled. The torque produced by the wheel in frame $B$ is given by Equation (3.30).

The values for the number of bits of the ADC of the sun sensor and for its sensitivity were not included in the datasheet. Similarly, the values for the scale factor and misalignment matrix and for the static bias

of both the magnetometer and the gyro plus the noise level of the magnetometer and the ARW and RRW of the gyro were not included in their datasheet. Representative values for the missing model parameters were taken from reference [49]. The values used for the different parameters of the hardware models are compiled in Appendix B.

## 6.3 ADCS Environment

The ADCS environment is composed by the Attitude Estimator block, by the Attitude Controller block and by the onboard Orbital Propagator block. The Attitude Estimator block is fed by the onboard Orbital Propagator and by the Sensor Model block. It is in this block where the different estimation algorithms from Chapter 4 are implemented. The four different sun vectors coming from the Sensor Model block are fused together by averaging the sun vectors of the different sun sensors that provide data in each time instant. The Attitude Controller block, in turn, is fed by the onboard Orbital Propagator and by the Attitude Estimator block. It is in the Attitude Controller block were the selected nadir pointing controller (see Section 7.2.2) and detumbling controller are implemented. This block is also responsible for computing the quaternion error $\mathbf{q}_e \equiv \mathbf{q}_O^B$ by implementing Equation (5.1) and the error angular velocity $\vec{\omega}_e \equiv \vec{\omega}_B^{B/O}$ by implementing Equations (5.2) and (5.3). The dipole moment generated from the Attitude Controller block is then fed into the Actuator Model block, which is responsible for computing the controller torques acting on the satellite. The onboard Orbital Propagator consists of a modified version of the SMS block and it will be responsible for feeding the Attitude Estimator block and the Attitude Controller block as stated before, and for estimating the orbit of the satellite. Unlike the SMS block, in the onboard Orbital Propagator, the torques produced by the environmental disturbances are not computed. Under the onboard Orbital Propagator, in the block used to compute the geomagnetic field model, a zero-mean Gaussian white noise vector with a standard deviation of $165\mathrm{nT}$ was added to simulate the uncertainties from the WMM. The value of the standard deviation was taken from [58]. This noise block should, however, be removed when compiling the code for the actual ADCS board. The fact that the onboard Orbital Propagator uses the estimated attitude quaternion from the Attitude Estimator block makes it diverge with time from the actual orbital position. This block includes a resettable orbit integrator responsible for resetting the propagated orbital position, velocity and time once a new reading from the GNSS receiver is obtained. During the simulations, this reading (orbital position and velocity) comes directly from the SMS block and it is updated every $24\mathrm{h}$ of simulation run-time. The onboard Orbital Propagator will be initialized in two stages. The first stage is responsible for the environmental models (gravity, atmospheric density, geomagnetic field and the sun vector) while the second stage is responsible for computing the perturbation forces due to atmospheric drag and solar radiation pressure. The requirement of the two stages stems from the fact that the perturbation forces due to atmospheric drag and solar radiation pressure require the knowledge of the estimated attitude quaternion. The ADCS environment also includes a block where the operation routine of the ADCS and its state transitions were implemented, the ADCS Controller block. This block should be compiled to $C$ and implemented in the ORCASat's On-Board Computer (OBC). The operation of this block is explained in Appendix C.

# Chapter 7

# Simulations Results

In this chapter, the different components of the ADCS will be tested and discussed using the Simulink model of the satellite. All the simulations were performed using the eighth-order Dormand-Prince RK8(7) solver with a fixed time-step of $0.1$s. The general simulation parameters used are shown in Table 7.1. The value set for $C_D$ was chosen according to similar CubeSat projects, such as the AAUSAT project [45] and the CubeSTAR project [70]. Reference [11] also proposes the same value of $C_D$. As stated before, the ORCASat was modeled as a set of $6$ plates. The faces normal to $\pm\hat{\mathbf{b}}_1$ (Fig. 2.2) have dimensions $10\text{cm} \times 10\text{cm}$, while the other faces have dimensions of $20\text{cm} \times 10\text{cm}$. The absorption coefficient of the different panels was assumed to be equal to $1$ ($\sigma^i_{abs} = 1$) since most of the CubeSat's surface is covered with solar cells. The value selected for the parasitic dipole moment was chosen according to the value measured for the 3U Space Dart CubeSat in reference [71]. All results obtained for steady-state were calculated using simulations with a run-time of $24$h to ensure that unfavorable geomagnetic field geometry does not cause a problem at some point, as stated in [9]. In the simulations performed, the magnetometer operates simultaneously with the magnetorquers. This is a simplification and cannot be done in the real CubeSat since the magnetorquers will corrupt the magnetometer readings. In the real CubeSat, the magnetorquers should operate in the first third of each ADCS cycle while the magnetometer should work in the last third of each cycle, as suggested in talks with Alfred Ng, Deputy Director at CSA. Preliminary simulations using this architecture showed little performance degradation.

Table 7.1: General simulation parameters.

| Parameters | Symbol | Value | Unit |
|---|---|---|---|
| Epoch | - | 12:00:00 15/09/2019 | UTC |
| Initial Position in ECI coordinates | $\vec{\mathbf{r}}_I$ | $(-4123.9936; -2987.4334; -4463.0619)$ | km |
| Initial Velocity in ECI coordinates | $\vec{\mathbf{v}}_I$ | $(6.026; -3.455; -3.263)$ | km/s |
| Orbital period | $T_{orb}$ | $5549.7$ | s |
| Mass of the satellite | $m$ | $3.6$ | kg |
| Satellite inertia matrix in frame $B$ | $\mathbf{J}_B$ | $\text{diag}\begin{pmatrix}0.003 & 0.007 & 0.008\end{pmatrix}$ | kg m$^2$ |
| Drag coefficient | $C_D$ | $2$ | - |
| Parasitic dipole moment in frame $B$ | $\vec{\mathbf{m}}_B$ | $(0.00707; 0; 0.00707)$ | A m$^2$ |
| Position of the centroid in frame $B$ | $\vec{\mathbf{c}}_B$ | $(-4; -2; -2)$ | cm |
| Solar radiation pressure | $p_\odot$ | $4.5 \times 10^{-6}$ | Pa |

## 7.1 Detumbling Controller

The gain $K_d$ of the detumbling controller was chosen according to Equation (5.48), being equal to

$$K_d = 1.21 \times 10^{-5} \text{kg m}^2 \text{ s}^{-1}$$

where the orbital period used was $T_{orb} = 5549.7$s, $J_{min} = 0.003\text{kg m}^2$ and the angle $\xi$ was chosen to be equal to the orbit inclination $i = 51.6^\circ$ since the orbital insertion is an uncertainty and thus the angle $\xi$ can be in the interval $[51.6^\circ - 9.7^\circ\,; 51.6^\circ + 9.7^\circ] = [41.9^\circ\,; 61.3^\circ]$, being the value $9.7^\circ$ the inclination of the magnetic north. The initial angular velocity used in the different cases devised to test the detumbling controller is

$$\vec{\omega}_0 = \begin{bmatrix} 0.907 & 0.907 & 0.907 \end{bmatrix}^\top (\text{rad s}^{-1})$$

corresponding to a magnitude of $\|\vec{\omega}_0\| = 1.57\text{rad s}^{-1} = 90\,^\circ\text{s}^{-1}$. Due to the difficulty in precisely determine the exact inertia matrix of the satellite and in order to better evaluate the performance of the detumbling controller, two different inertia matrices were considered, the "true" inertia matrix and a perturbed inertia matrix. The new inertia matrix will be obtained in the body frame by a $zyx$ rotation of $28^\circ$, $-21^\circ$, and $29^\circ$ respectively. These angles were obtained randomly but selected in order to have a relatively large difference from the true inertia matrix. The two inertia matrices are given in $\text{kg m}^2$ by

$$\mathbf{J}_{nominal} = \begin{bmatrix} 0.003 & 0 & 0 \\ 0 & 0.007 & 0 \\ 0 & 0 & 0.008 \end{bmatrix} \quad \mathbf{J}_{perturbed} = \begin{bmatrix} 0.004410529 & 0.002021869 & 0.000454652 \\ 0.002021869 & 0.005932414 & 0.000258699 \\ 0.000454652 & 0.0002586989 & 0.007657058 \end{bmatrix}$$

The effect of the wheel momentum bias was also accounted for in testing the performance of the detumbling controller. In total four different cases were considered. These cases are summarized in Table 7.2. The convergence of the detumbling controller for cases 1 and 2 is plotted in Figure 7.1 where the evolution of the magnitude of the angular velocity ($\|\vec{\omega}\|$) is shown. The time the controller took to converge below $\|\vec{\omega}\| = 0.03\text{rad s}^{-1}$ is shown in Table 7.3. The $\|\vec{\omega}\| = 0.03\text{rad s}^{-1}$ threshold was considered since both the selected controller for the satellite and the estimation algorithms were shown to be able to converge below this threshold as it will be seen in Sections 7.2.3 and 7.3.2 respectively. The steady-state behaviour of the detumbling controller for the four cases considered is shown in Figure 7.2. The mean and maximum angular speeds for the different cases of the steady-state behaviour of the detumbling controller are shown in Table 7.3. The simulation time is $116400$s but the values and the plots referring to the steady-state behaviour correspond only to the last $86400\text{s} = 24\text{h}$ of the simulation to ensure the proper convergence of the controller. These simulations use the complete satellite Simulink model but without the Attitude Estimator block, i.e., the angular velocity of the ORCASat and the geomagnetic field are fed to the detumbling controller directly from the Sensor Model block.

It is possible to see that the detumbling controller was able to decrease the angular velocity of the satellite from $\|\vec{\omega}_0\| = 90\,^\circ\text{s}^{-1}$ to $\|\vec{\omega}\| = 0.03\text{rad s}^{-1} \simeq 1.72\,^\circ\text{s}^{-1}$ in about one orbit in the true inertia matrix case, case 1. In the perturbed inertia matrix case, case 2, the convergence time was smaller. This is

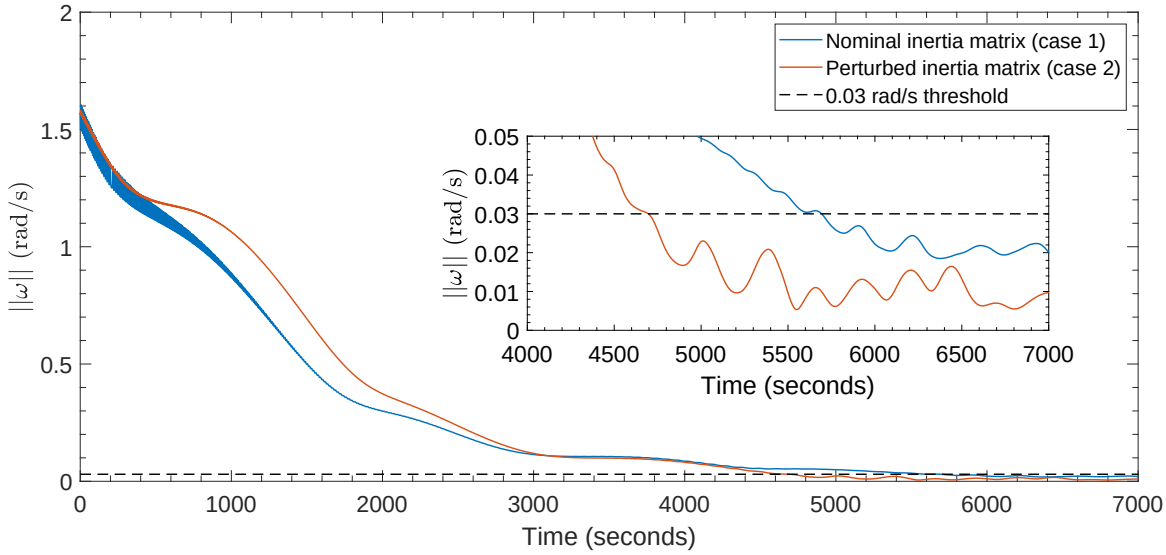
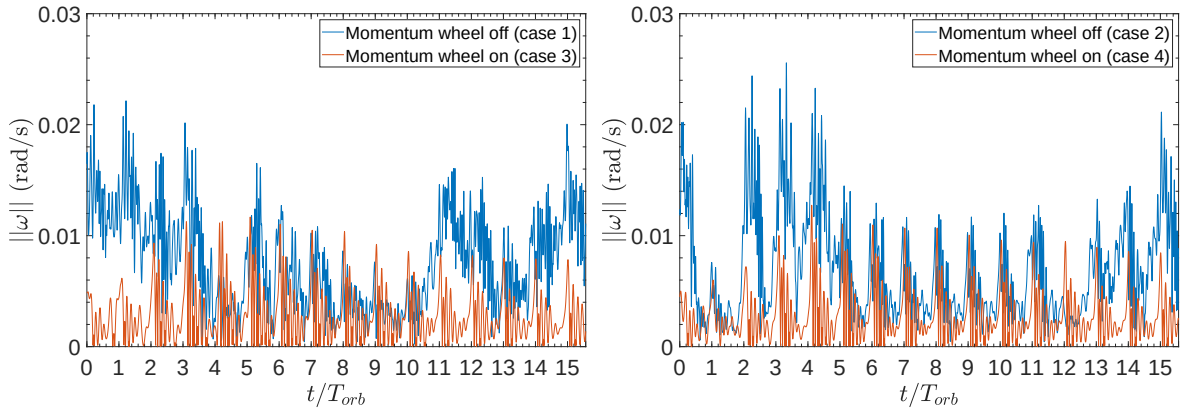



Figure 7.1: Convergence performance of the detumbling controller.




(a) True inertia matrix.

(b) Perturbed inertia matrix.

Figure 7.2: Steady-state performance of the detumbling controller.

likely due to a random effect since in the first $3000$s of the simulation the convergence was being done at a slower rate than for the case of the true inertia matrix. The effect of the perturbed inertia matrix in the steady-state of the detumbling controller is small but tends to decrease the mean angular speed of the satellite. From case 1 to case 2 there's a reduction of about $10\%$ while from case 3 to case 4 this reduction is just $3\%$. Although the $10\%$ reduction of the mean angular speed might seem significant it is not since this reduction has order of hundredths of degrees per second. The magnitude of this effect is smaller when the momentum wheel is on due to the extra gyroscopic stiffness that the momentum wheel provides. The maximum angular speeds however are larger in the cases where the perturbed inertia matrix was used where there is an increase of about $16\%$ from case 1 to case 2 and $9\%$ from case 3 to case 4. The detumbling controller was however able to maintain the angular speed bellow the $0.03\text{rad}\,\text{s}^{-1}$ threshold in all the cases considered especially in the cases where the momentum wheel is on. In this case there is a reduction in the maximum angular speed of about $50\%$ and a reduction of the mean angular speed of about $60\%$ when compared to the cases where the momentum wheel was off.

Table 7.2: Detumbling simulation parameters.

| Case | Wheel momentum (N m s) | Inertia matrix |
|---|---|---|
| 1 | $\begin{bmatrix}0 & 0 & 0\end{bmatrix}$ | $\mathbf{J}_{nominal}$ |
| 2 | $\begin{bmatrix}0 & 0 & 0\end{bmatrix}$ | $\mathbf{J}_{perturbed}$ |
| 3 | $\begin{bmatrix}0 & -0.003 & 0\end{bmatrix}$ | $\mathbf{J}_{nominal}$ |
| 4 | $\begin{bmatrix}0 & -0.003 & 0\end{bmatrix}$ | $\mathbf{J}_{perturbed}$ |

Table 7.3: Detumbling controller performance metrics.

| Case | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Mean Angular Speed (rad s$^{-1}$) | 0.0078 | 0.0070 | 0.0029 | 0.0028 |
| Mean Angular Speed (°s$^{-1}$) | 0.45 | 0.40 | 0.16 | 0.16 |
| Max. Angular Speed (rad s$^{-1}$) | 0.0221 | 0.0256 | 0.0117 | 0.0127 |
| Max. Angular Speed (°s$^{-1}$) | 1.27 | 1.47 | 0.67 | 0.73 |
| Convergence Time (s) | 5693 | 4697 | - | - |
| Convergence Time ($t/T_{orb}$) | 1.03 | 0.85 | - | - |

## 7.2 Pointing Controller Algorithms

This section is going to be used to analyse and compare the performance of the different control algorithms. These algorithms will be tested and compared regarding their pointing accuracy and total magnetic dipole moment produced, which is related to the power consumed. Based on the steady-state response of the different controllers, the ORCASat nominal controller will be selected. The performance of the selected controller will then be tested regarding its robustness to initial tumbling speeds, the ability to converge from different initial pointing errors, the influence of the satellite's inertial matrix uncertainties, and the effect of the momentum wheel performance degradation. The controllers will be tested using the complete satellite Simulink model, which includes environmental perturbations and the attitude estimation algorithms. The attitude estimation algorithms, however, do not include the MCEKF. The parameters used in the different estimation algorithms are displayed in Table 7.11. The simulation time is $116400$s but the values and the plots referring to the steady-state behaviour correspond only to the last $86400\text{s} = 24\text{h}$ of the simulation to ensure the proper convergence of the controllers.

### 7.2.1 Controllers Overview

The SMC is very sensitive to the value of the gain $\lambda$ (see Section 5.1.2). By decreasing this value one can get a more efficient controller, i.e., a controller that uses less power to control the spacecraft for about the same level of pointing accuracy, if at the same time the proportional gain $K$ is increased. The $\lambda$ gain, however, cannot be decreased infinitely, as explained in Section 5.1.2. If $\lambda$ is to low, the controller can't keep itself in the sliding manifold and so any increase in the $K$ gain won't provide a better pointing accuracy. For a fixed $\lambda$, increasing $K$ beyond a certain value makes the maximum values of the pointing error start to increase, producing a less smooth controller with increasing oscillations as $K$ increases.

The set of gains which provided the smallest maximum error are $K = 0.005$ and $\lambda = 0.02$. The maximum error is $3.9^\circ$ for a total mean dipole moment of $0.030$A m$^2$. The steady-state response is shown on Figure D.2a. It is clear that this controller respects the $10^\circ$ maximum error constrain by a significant margin and thus it is possible to choose a different set of gains that can have looser constraints in order to produce less dipole moment. The new set of gains is $K = 0.015$ and $\lambda = 0.0007$. The maximum error is $9.0^\circ$ for a mean total dipole moment of $0,0116$A m$^2$, which is about $2.5$ times smaller than the dipole moment used by the minimum maximum error case. The steady-state response for this set of gains is plotted in Figure D.1a. The values obtained for the most efficient and best-performing gain for

this controller and for the following controllers are summarized in Table 7.4.

The most efficient gain obtained for the IHC, within the $10^\circ$ pointing requirement, used $K_Q = 0.05$ and produced a mean total dipole moment of $0.0106$A m$^2$ for a maximum pointing error of $9.6^\circ$. The behaviour is shown in Figure D.1d. The controller gain which achieved the smallest maximum error was determined using $K_Q = 5$ and produced a mean total dipole moment of $0.0202$A m$^2$, which is approximately double the value obtained for the most efficient gain. The maximum error obtained was $3.3^\circ$. The steady-state behaviour is shown in Fig. D.2d. This controller, similarly to the FHC and to the CGC, after achieving the minimum maximum error, increasing $K_Q$ won't have a big impact in the controller's pointing performance, being the main consequence an increase in the dipole moment used.

In the FHC it was found that increasing the gain $K$ used in Equation (5.41) usually tends to increase the maximum pointing error. This effect, however, is not very significant. For instance for $K_Q = 0.5$, the maximum error obtained using $K = 1$, $K = 100$, and $K = 1000$ is $3.5^\circ$, $3.5^\circ$ and $3.9^\circ$ respectively. For a higher value of $K_Q$, for instance, $K_Q = 7$, for the same values of $K$ as in the previous example, the maximum error obtained was $3.4^\circ$, $3.7^\circ$, and $4.3^\circ$ respectively. The mean pointing errors obtained using the different values of $K$ have the same order up to $2$ decimal points. All the results presented in this work for the FHC were obtained using $K = 1$. The most efficient gain found for the FHC, within the pointing requirements, was determined using $K_Q = 0.04$ and used a mean total dipole moment of $0.0105$A m$^2$ for a maximum pointing error of $9.3^\circ$. The smallest maximum pointing error was found to be $3.3^\circ$ for a mean total dipole moment of $0.0173$A m$^2$ using $K_Q = 3$. The previous values correspond to a $60\%$ increase in the total dipole moment for almost a $3$ times smaller maximum pointing error. The behaviour of the controller using the two different gains is represented in Figures D.1c and D.2c respectively.

The most efficient gain found for the CGC, which still met the pointing error requirements from Section 1.3, was determined using $K_Q = 0.09$, and achieved a maximum pointing error of $8.9^\circ$ while using a mean total dipole moment of $0.0103$A m$^2$. The steady-state behaviour is shown in Figure D.1b. The smallest maximum pointing error achieved with this controller was $3.4^\circ$ for a total dipole moment of $0.0155$A m$^2$ using $K_Q = 3$. The previous values represent a $50\%$ increase in the total dipole moment for a reduction by a factor of $2.6$ of the maximum pointing error. The behaviour is shown in Figure D.2b.

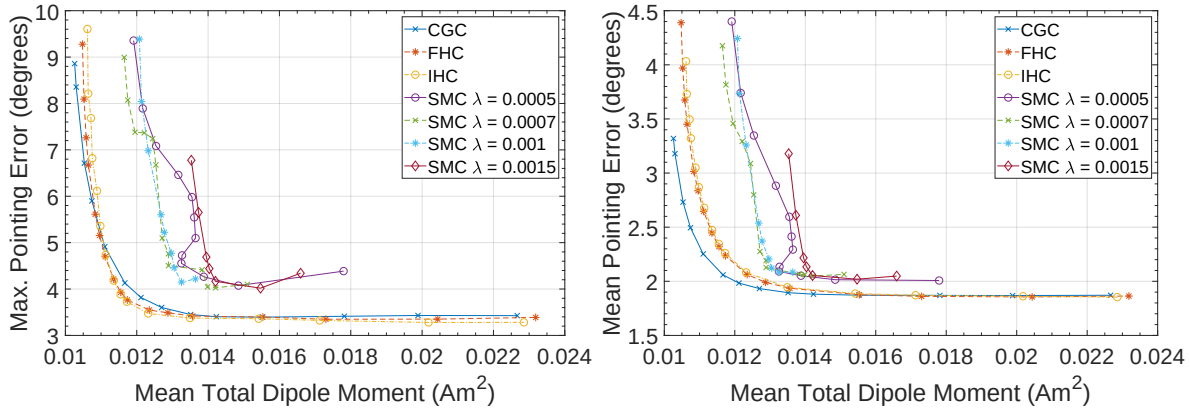
Table 7.4: Performance metrics of the different nadir-pointing controllers using the most efficient and best-performing gains.

| | | SMC | IHC | FHC | CGC |
|---|---|---|---|---|---|
| Most efficient gain | Maximum Error (degrees) | 9.0 | 9.6 | 9.3 | 8.9 |
| | Total Dipole Moment (A m$^2$) | 0.0116 | 0.0106 | 0.105 | 0.103 |
| Best-performing gain | Maximum Error (degrees) | 3.9 | 3.3 | 3.3 | 3.4 |
| | Total Dipole Moment (A m$^2$) | 0.0300 | 0.0202 | 0.0173 | 0.0155 |

### 7.2.2 Controllers Comparison, Discussion and Selection

The lowest maximum error is achieved by the IHC with a value of $3.28^\circ$, while the FHC and CGC achieve an error of $3.35^\circ$ and $3.39^\circ$ respectively. On the other hand, the IHC is the one that requires the

most dipole moment in order to achieve its lowest maximum error, being followed by the FHC and by the CGC. The lowest maximum error achieved using the SMC was $3.86^\circ$, which is greater than the values obtained using the LQR controllers, while still using a relatively more amount of power. When using the most efficient gains, this difference is much smaller but the LQR controllers still have an advantage in terms of efficiency. The same conclusions can be seen by looking at Figure 7.3a, which plots the




(a) Maximum pointing error.

(b) Mean pointing error.

Figure 7.3: Pointing controllers efficiency comparison (pointing error vs mean total dipole moment).

maximum pointing error achieved versus the total dipole moment used for a variety of increasing values of $K_Q$ (LQR) and $K$ (SMC). It can be seen that any of the LQR controllers are more efficient than the best cases for the SMC, i.e, the LQR controllers require less dipole moment than the SMC for the same maximum pointing error. It can also be seen in this figure that as $\lambda$ starts to decrease beyond a certain value the evolution of the maximum error with the total dipole moment starts getting less predictable unlike for higher values of $\lambda$. As stated before, it can thus be seen that a proper choice for the value of $\lambda$ can help to significantly increase the efficiency of the SMC, but reducing this gain beyond a certain value can have counterproductive results. From Figure 7.3a, it is also visible that for smaller pointing errors (less than $5^\circ$) the IHC and FHC are slightly more efficient than the CGC, although this difference vanishes when the gain further increases. For higher errors the inverse happens, the CGC can reach the same maximum pointing error while being more energy efficient than the FHC and IHC, although this difference is very small. Figure 7.3a also allows us to see that for higher errors, generally above the $5^\circ$ mark, by increasing the gain of all the different controllers, a relatively small increase of the total dipole moment allows us to achieve a much better pointing error. For instance, in the case of the CGC, increasing $K_Q$ can lead to an increase in the mean total dipole moment smaller than $9\%$ while providing a $45\%$ reduction of the maximum pointing error when considering the maximum errors $8.9^\circ$ and $4.9^\circ$.

Figure 7.3b provides some more insight into the behaviour of the different controllers. This figure plots the mean pointing error versus the mean total dipole moment used. It is possible to see that unlike in Figure 7.3a, the CGC is the controller which for the same amount of total dipole moment provides the smallest error. For larger gains, the difference between the CGC and the remaining LQR controllers vanishes. It can also be seen that the LQR controllers, once again, have an advantage in terms of efficiency and pointing accuracy in comparison with the SMC. In all cases, the difference between the

FHC and the IHC is negligible.

By comparing Figures 7.3a and 7.3b it is possible to see that after the SMC has reached the lowest value of its maximum error, the maximum error starts to increase at a larger rate than the mean pointing error with the continuous increase in gain. This shows that continuing to increase the gain $K$ of the SMC after a certain level increases the amplitude of the oscillations. In the LQR controllers, this effect is much less noticeable and can even be disregarded, being this one more advantage that the LQR controllers have with respect to the SMC.

According to the simulations performed, the proposed control algorithm for the ORCASat will be the CGC due to its simpler implementation compared to the IHC and FHC while providing a similar level of performance. The SMC is excluded since it has clearly worse performance. The gain chosen for the CGC was obtained using $K_Q = 0.7$. The maximum error obtained for this controller is $3.8^\circ$ and the mean error is $2.0^\circ$. The mean total dipole moment used is $0.0121\text{A}\,\text{m}^2$. This gain was chosen because it provides a low pointing error while not substantially increasing the total dipole moment. This controller must now be tested to ensure it can achieve its nominal pointing error from different initial attitudes (Section 7.2.3) and that it can resist to model uncertainties (Section 7.2.4).

### 7.2.3 Transient Behaviour of the Selected Controller

The transient behaviour of the controller will be tested in terms of the time it takes to converge to a pointing error below the $10^\circ$ limit error. Four different cases were devised combining two different initial pointing error angles, $90^\circ$ and $180^\circ$, about the $\hat{\mathbf{b}}_1$ axis with two different initial angular velocities $\vec{\omega}_0$. In configurations 1 and 2 $\vec{\omega}_0$ is representative of the angular velocity after detumbling and in cases 3 and 4 $\vec{\omega}_0$ corresponds to a higher angular rate case. The momentum wheel is on such as the attitude estimator. The magnetometer calibration filter is off. The vector $\hat{\mathbf{b}}_1$ was selected because it presents a challenging case for the attitude controller due to the gyroscopic stiffness provided by the momentum wheel about this axis. The different configurations used are shown in Table 7.5. The convergence time of the controller below the $10^\circ$ threshold is shown in Table 7.6 while its behaviour is given by Figure 7.4.

Table 7.5: Simulation configurations for the transient analysis.

| Configuration | Initial error (degrees) | Initial angular velocity $\vec{\omega}_0$ ($\text{rad}\,\text{s}^{-1}$) |
|---|---|---|
| 1 | 90 | $\begin{bmatrix} 6\times10^{-05} & 0.01 & 3\times10^{-05} \end{bmatrix}$ |
| 2 | 180 | $\begin{bmatrix} 6\times10^{-05} & 0.01 & 3\times10^{-05} \end{bmatrix}$ |
| 3 | 90 | $\begin{bmatrix} 0.05 & 0.05 & -0.05 \end{bmatrix}$ |
| 4 | 180 | $\begin{bmatrix} 0.05 & 0.05 & -0.05 \end{bmatrix}$ |

Table 7.6: CGC convergence times.

| Configuration | Time (s) | Time $(t/T_{orb})$ |
|---|---|---|
| 1 | 1697 | 0.31 |
| 2 | 1969 | 0.35 |
| 3 | 1055 | 0.19 |
| 4 | 1957 | 0.35 |

It is possible to verify that the controller not only was able to converge but it also converged in a relatively small time, less than half an orbit for all the different configurations presented. The controller was also able to overcome the initial wobbling due to the initial higher angular rates. The initial angular velocity from configurations 3 and 4 had the unexpected result of reducing the convergence times. This effect was significant in configuration 3, although it might just be a random effect since the reduction in the convergence time was negligible in configuration 4 and because the convergence in configuration 1 was being done at a faster rate in the first $900\text{s}$ of the simulation than in configuration 3.

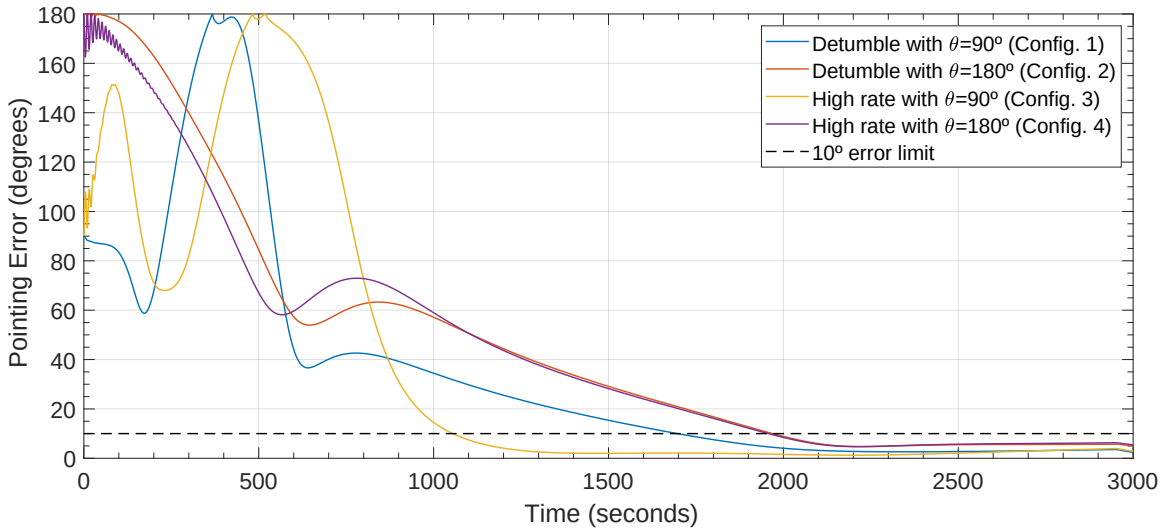



Figure 7.4: Transient behaviour of the selected nadir-pointing controller for the ORCASat.

### 7.2.4 Robustness of the Selected Controller to Model Uncertainties

The robustness of the controller to model uncertainties will be tested using 3 different cases applied to the steady-state response of the controller and applied to the $4$ different configurations shown in Table 7.5 for the transient behaviour. In case 1, the ORCASat principal axes of inertia will be rotated from the body frame $B$ defined in Section 2.1.4. This new inertia matrix is the perturbed inertia matrix $\mathbf{J}_{perturbed}$ from Section 7.1. In case 2 the momentum wheel is going to have its momentum storage reduced by a third of its nominal value going from $0.003$N m s to $0.002$N m s. Case 3 consists of using the perturbed inertia matrix from case 1 together with the reduced wheel momentum storage from case 2. The values used for the wheel angular momentum and for the ORCASat's inertia matrix in each of the cases considered are summarized in Table 7.7.

Table 7.7: Model uncertainty cases for the analysis of the ORCASat's nadir-pointing controller.

| Case | Wheel angular momentum vector (N m s) | Inertia matrix (kg m$^2$) |
|---|---|---|
| 1 | $\begin{bmatrix}0 & -0.003 & 0\end{bmatrix}$ | $\mathbf{J}_{perturbed}$ |
| 2 | $\begin{bmatrix}0 & -0.002 & 0\end{bmatrix}$ | $\mathbf{J}_{nominal}$ |
| 3 | $\begin{bmatrix}0 & -0.002 & 0\end{bmatrix}$ | $\mathbf{J}_{perturbed}$ |

#### Transient Behaviour

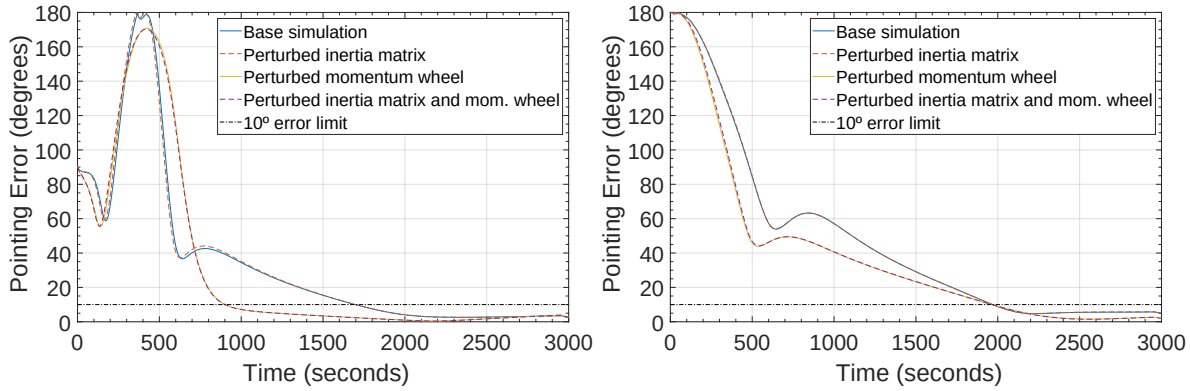
The convergence times for the $4$ different configurations used to test the transient behaviour of the selected controller for the ORCASat under the 3 different cases from Table 7.7 are shown in Table 7.8. The behaviour of the controller is plotted in Figure 7.5.

It is possible to verify that the controller still presents good convergence properties, in all the different cases considered, converging in less than half an orbit in all the cases and configurations devised. By looking at Figure 7.5 and by comparing the values from Tables 7.6 and 7.8, it is possible to see that the perturbed inertia matrix has very little impact on the convergence times and in the overall transient be-

haviour of the controller. The reduced wheel angular momentum has an impact on the overall transient behaviour of the controller by reducing the amplitude of the overshoots. This can be seen better in the smaller initial angle configurations (configurations 1 and 3) than in the high angle configurations (configurations 2 and 4). In terms of convergence time its effect can be positive (configuration 1), negative (configuration 3), or virtually none (configurations 2 and 4).

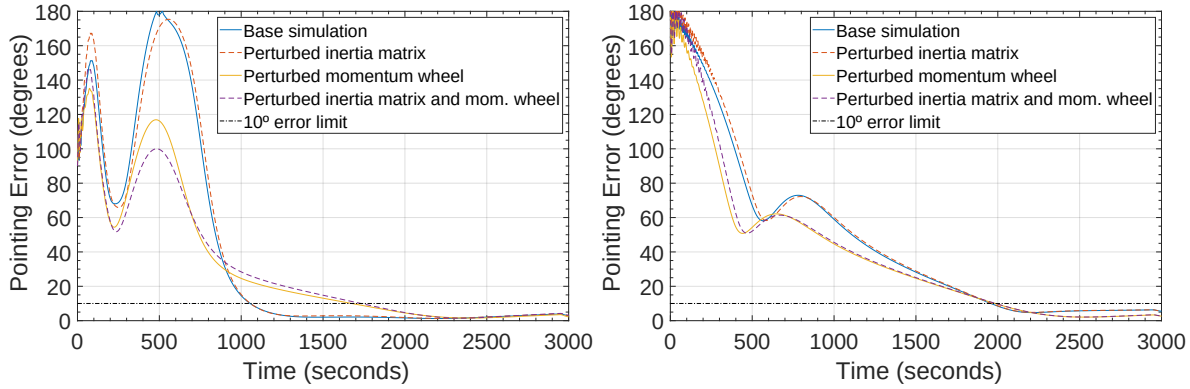
Table 7.8: Convergence times of the ORCASat's pointing controller under the $3$ model uncertainty cases.

| Configuration | 1 | | | 2 | | | 3 | | | 4 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Case | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| Time (s) | 1698 | 903 | 901 | 1967 | 1965 | 1965 | 1056 | 1675 | 1732 | 1962 | 1983 | 1987 |
| Time ($t/T_{orb}$) | 0.31 | 0.16 | 0.16 | 0.35 | 0.35 | 0.35 | 0.19 | 0.30 | 0.31 | 0.35 | 0.36 | 0.36 |




(a) Detumble mode with $90^\circ$ initial error (Config. 1).

(b) Detumble mode with $180^\circ$ initial error (Config. 2).




(c) Higher angular rate with $90^\circ$ initial error (Config. 3).

(d) Higher angular rate with $180^\circ$ initial error (Config. 4).

Figure 7.5: Transient behaviour of the ORCASat's controller under the $3$ model uncertainty cases.

### Nominal Behaviour

The performance metrics for the nominal behaviour of the controller are expressed in Table 7.9 while in Figure 7.6 it is shown the pointing error of the 3 last orbits of the simulation.

The nominal behaviour follows the same trend seen in the transient behaviour. The inertia matrix uncertainty has a very small impact in the overall performance of the controller and, in this case, it even

improved the maximum and mean pointing errors by a very small amount, less than a $1.5\%$ reduction of the maximum and mean values of the pointing error for case 1 relative to the base simulation and for case 3 relative to case 2. In practical terms, this difference can be disregarded since it is of the order of hundredths of a degree. The difference in the total dipole moment is negligible. The momentum wheel performance degradation is the perturbation which has the most impact in the performance of the controller, especially in the maximum error where there is an increase of $0.65^\circ$ from base simulation to case 2, corresponding to a $17\%$ increase in error, and an increase of $0.64^\circ$ from case 1 to case 3, corresponding also to a $17\%$ increase in error. The performance of the controller is still very good, not reaching even half of the maximum $10^\circ$ error limit. The mean error only increases $0.12^\circ$ $(6\%)$ from the base simulation to case 2 and $0.13^\circ$ $(6\%)$ from case 1 to case 3. The increase of the mean total dipole moment produced by the magnetorquers due to the momentum wheel performance degradation is even smaller being less than $2\%$ from the base simulation to case 2 and from case 1 to case 3.

Table 7.9: Performance metrics for the nominal behaviour of the ORCASat's nadir-pointing controller.

| Case | Maximum Error (degrees) | Mean Error (degrees) | Dipole Moment (A $m^2$) |
|---|---|---|---|
| Base simulation | 3.82 | 1.98 | 0.01211 |
| 1 | 3.76 | 1.95 | 0.01213 |
| 2 | 4.47 | 2.10 | 0.01233 |
| 3 | 4.40 | 2.08 | 0.01232 |

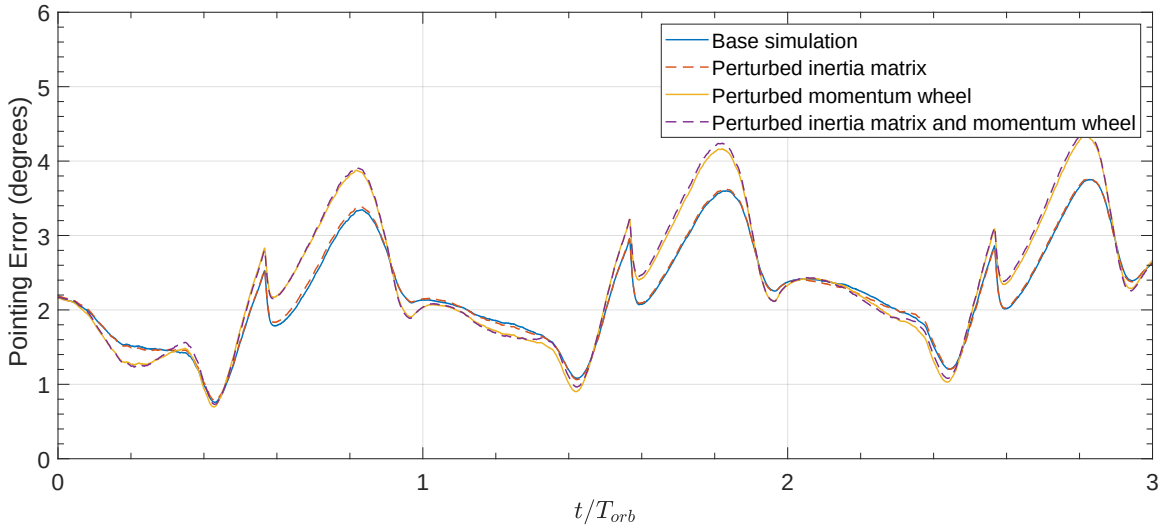



Figure 7.6: Effect of the model uncertainties on the nominal behaviour of the ORCASat's nadir-pointing controller (last 3 orbits of the simulation).

## 7.3 Estimation Algorithms

In this section, the performance of different algorithms responsible for determining/estimating the attitude of the ORCASat are going to be analysed. First, the nominal performance of the MEKF will be analysed and the impact of the MCEKF in the MEKF performance will be evaluated. Later, the

Table 7.10: TAM bias and $\mathbf{D}$ matrix values used in the simulations of the estimation algorithms.

| Scenario | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $b_1$ (nT) | $-610$ | $5000$ | $-610$ | $5000$ |
| $b_2$ (nT) | $258$ | $3000$ | $258$ | $3000$ |
| $b_3$ (nT) | $1793$ | $6000$ | $1793$ | $6000$ |
| $D_{11}$ | $-0.0438$ | $0.05$ | $-0.0418$ | $-0.0418$ |
| $D_{22}$ | $-0.1111$ | $0.1$ | $-0.1110$ | $-0.1110$ |
| $D_{33}$ | $-0.1387$ | $0.05$ | $-0.1334$ | $-0.1334$ |
| $D_{12}$ | $0.0002$ | $0.05$ | $-0.0778$ | $-0.0778$ |
| $D_{21}$ | $0.0052$ | $0.05$ | $0.0718$ | $0.0718$ |
| $D_{13}$ | $0.0161$ | $0.05$ | $-0.0400$ | $-0.0400$ |
| $D_{31}$ | $0.0002$ | $0.05$ | $-0.0672$ | $-0.0672$ |
| $D_{23}$ | $-0.0064$ | $0.05$ | $-0.0636$ | $-0.0636$ |
| $D_{32}$ | $0.0000$ | $0.05$ | $0$ | $0$ |

Table 7.11: Estimator parameters.

| Estimator | Parameter | Value |
|---|---|---|
| MEKF | Gyro ARW ($\sigma_v$) | $3.493\,08 \times 10^{-8}\,\mathrm{rad\,s^{-1/2}}$ |
| | Gyro RRW ($\sigma_u$) | $2.908\,88 \times 10^{-5}\,\mathrm{rad\,s^{-3/2}}$ |
| | TAM SD ($\sigma_m$) | $1.5 \times 10^{-5}\,\mathrm{T}$ |
| | Sun Sensor SD ($\sigma_s$) | $0.5°$ |
| QUEST | TAM SD ($\sigma_{mag}$) | 0.029 |
| | Sun Sensor SD ($\sigma_{ss}$) | 0.008 |
| MCEKF | Measurement noise covariance ($\Sigma$) | $300\mathbf{I}_3\mathrm{nT}$ |

performance of the QUEST algorithm will be evaluated and compared to the performance of the MEKF. Lastly, the convergence of the MEKF will be analysed as well as the influence that QUEST has on the MEKF's convergence time. The convergence properties of the MCEKF will be shown in the end.

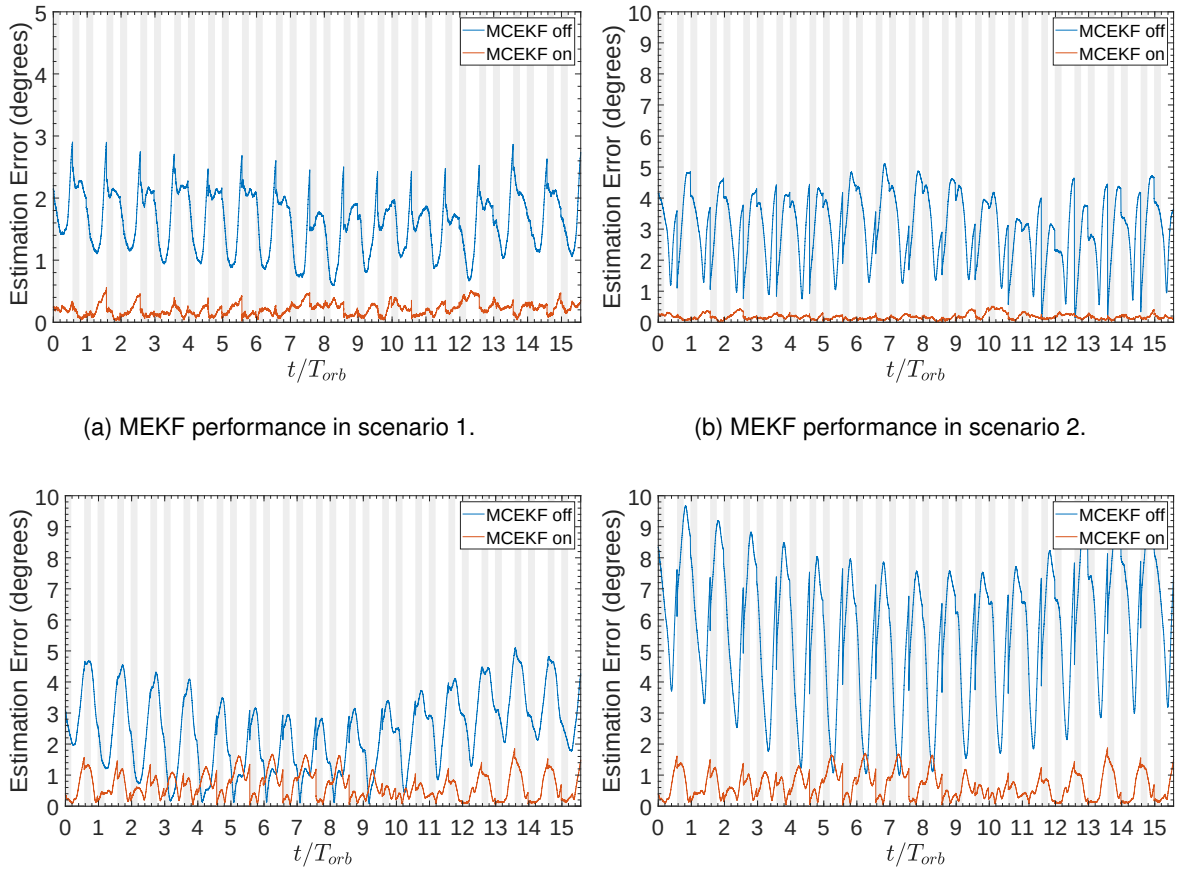
### 7.3.1 Nominal Behaviour of MEKF and QUEST and the Effect of MCEKF

Four different scenarios were drawn to test the performance of the MCEKF and the performance of the MEKF and QUEST as well as the effect that the MCEKF has on the MEKF and QUEST performance. All the simulations use the full satellite Simulink model with some modifications in the Sensor Model block as it will be seen further. The simulations also use the chosen controller for the ORCASat. The simulation time is $202800\mathrm{s}$ but the values and the plots referring to the steady-state behaviour correspond only to the last $86400\mathrm{s} = 24\mathrm{h}$ of the simulation to ensure the proper convergence of the algorithms. The 1st scenario uses the simulated data from magnetometer sensor used in the ORCASat. The $2$nd scenario was taken from [9] and uses a larger bias and a different scale factor and misalignment matrix $\mathbf{D}$. The $3$rd and $4$th scenarios use a different $\mathbf{D}$ matrix from the first two scenarios together with the bias from the 1st scenario and the bias from the $2$nd scenario respectively. This new matrix was designed to have larger off-diagonal terms to better evaluate the impact of the non-symmetry of the $\mathbf{D}$ matrix in the performance of the MCEKF. The values used for the bias vector $\vec{\mathbf{b}}$ and for $\mathbf{D}$ in the different scenarios are summarized in Table 7.10. The geomagnetic field generated $\vec{\mathbf{B}}_{gen}(t)$ for scenarios 2, 3 and 4 is given by

$$\vec{\mathbf{B}}_{gen}(t) = (\mathbf{I}_3 + \mathbf{D})^{-1}(\vec{\mathbf{B}}_B(t) + \vec{\mathbf{b}} + \vec{\boldsymbol{\nu}}(t))$$

where $\vec{\mathbf{B}}_B(t)$ is the geomagnetic field given in body frame coordinates from the SMS block. The covariance of the noise vector $\vec{\boldsymbol{\nu}}(t)$ is isotropic with a standard deviation of $50\mathrm{nT}$. This value was taken from [9]. The values used in scenario 1 for $\vec{\mathbf{b}}$ and $\mathbf{D}$ are given in Table 7.10 in accordance with the previous equation. The parameters used in the different estimators are displayed in Table 7.11. The results for the nominal behaviour of the MEKF with and without the effect of the MCEKF are plotted in Figure 7.7. The gray bars shown in the following plots represent the orbit regions where the sun sensors provide

data to the estimators. Table 7.12 provides a synthesis of the results for a better understanding.




(a) MEKF performance in scenario 1.

(b) MEKF performance in scenario 2.

(c) MEKF performance in scenario 3.

(d) MEKF performance in scenario 4.

Figure 7.7: Estimation error of the MEKF with and without the inclusion of the MCEKF. The grey vertical bars represent the orbit regions where the sun sensors provide data to the estimators.

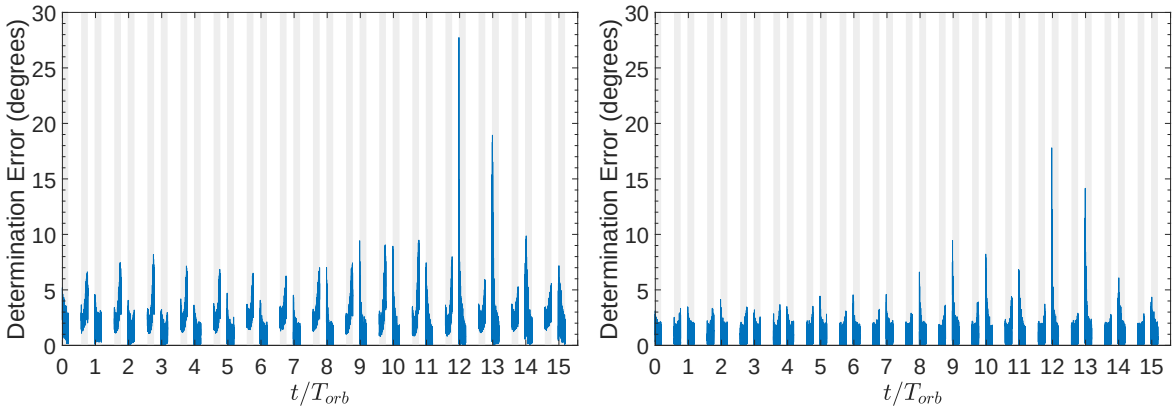
Table 7.12: Performance metrics of the MEKF in the different scenarios.

| | | Scenario 1 | Scenario 2 | Scenario 3 | Scenario 4 |
|---|---|---|---|---|---|
| Mean Estimation Error (degrees) | MCEKF off | 1.64 | 3.05 | 2.42 | 5.70 |
| | MCEKF on | 0,21 | 0,18 | 0.65 | 0.67 |
| Maximum Estimation Error (degrees) | MCEKF off | 2.90 | 5.11 | 5.11 | 9.67 |
| | MCEKF on | 0.56 | 0.52 | 1.85 | 1.88 |

As it is possible to see, the performance of the MEKF alone, using the simulated real sensor values, does not meet the $2^\circ$ attitude knowledge requirement presented in Section 1.3. Although the mean estimation error is below the $2^\circ$ threshold, the maximum error is $2.9^\circ$, which is $45\%$ higher than the limit error. The inclusion of the MCEKF allows to achieve much better results and to meet the $2^\circ$ requirement. Even in the worst-case scenario, scenario 4, where the maximum estimation error using only the MEKF is close to $10^\circ$, the introduction of the MCEKF has allowed it to be reduced below the $2^\circ$ limit error.

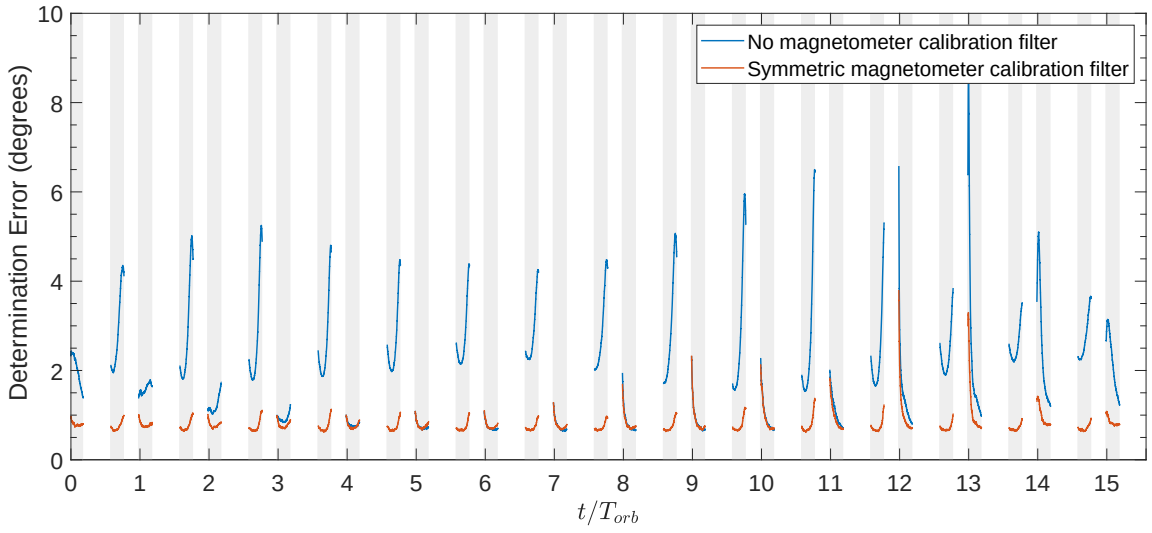
The importance of the magnetometer calibration can be clearly seen from the results obtained. It can also be concluded, from the results obtained, that the axes of the magnetometer should be closely

aligned with the axes of the spacecraft due to the effect that the $\mathbf{D}$ matrix has in the performance of the MCEKF and consequently in the performance of the MEKF. The effect of the $\mathbf{D}$ matrix is much higher than the effect of the bias of the magnetometer in the MCEKF. This can be seen by comparing the performance metrics of scenarios 3 and 4 when the MCEKF is on, where a $4$ times increase in the magnitude of the bias vector only leads to a $1.5\%$ increase of the maximum estimation error. Now by comparing scenario 2 with scenario 4, where the bias is the same, the $\mathbf{D}$ matrix with larger off-diagonal elements from scenario 4 leads to an increase of the maximum error by a factor of $3.6$. From Figure 7.7 it can also be seen, as expected, that the estimation error during eclipse is larger than the estimation error when the sun sensor measurements are available and that this difference grows with the disturbance applied to the true geomagnetic field, highlighting once again the need for an accurate measurement of the geomagnetic field and the introduction of the MCEKF.




(a) QUEST performance with the MCEKF off.

(b) QUEST performance with the MCEKF on.




(c) Moving average of the determination error with a window size of $600$ samples ($60$s).

Figure 7.8: Determination error of the QUEST algorithm in scenario 1. The grey vertical bars represent the orbit regions where the sun sensors provide data to the estimators.

The performance of the QUEST algorithm in scenario 1 is shown in Figure 7.8 while Table 7.13 provides its performance metrics. On the top, Subfigures 7.8a and 7.8b plot the direct output of the

determination error from the QUEST algorithm, while, on the bottom, Subfigure 7.8c shows the attitude determination error filtered using a moving average with a window size of $600$ samples which corresponds to a $1$ minute window size. Similarly to the case of the MEKF, the introduction of the MCEKF significantly improves the accuracy of the attitude knowledge. By comparing scenario 1 from Table 7.12 with Table 7.13 it is possible to conclude that the MEKF provides better results, both with and without the MCEKF, specially when looking at the maximum error metric. The MEKF is also able to provide attitude knowledge during eclipse while the QUEST algorithm is not. The output attitude from the QUEST algorithm is very noisy, thus the use of the moving average filter to present the determination error, while the MEKF provides a virtually noise free reading. Even with the introduction of the magnetometer calibration filter, the QUEST algorithm cannot satisfy the $2^\circ$ attitude knowledge requirement, and thus it will only be used for providing an initial attitude quaternion to initialize the MEKF as well as for sanity check.

Table 7.13: QUEST performance metrics in scenario 1 using the moving average filter.

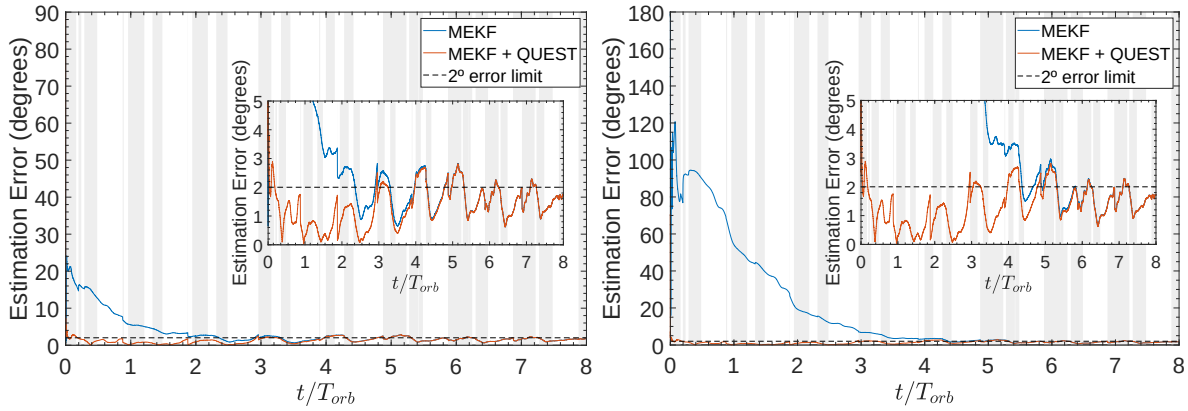
| | Mean Error (degrees) | Maximum Error (degrees) |
|---|---|---|
| MCEKF off | 2.04 | 8.69 |
| MCEKF on | 0.81 | 3.79 |

### 7.3.2 Transient Behaviour of MEKF and MCEKF and the Effect of QUEST

#### Multiplicative Extended Kalman Filter (MEKF)

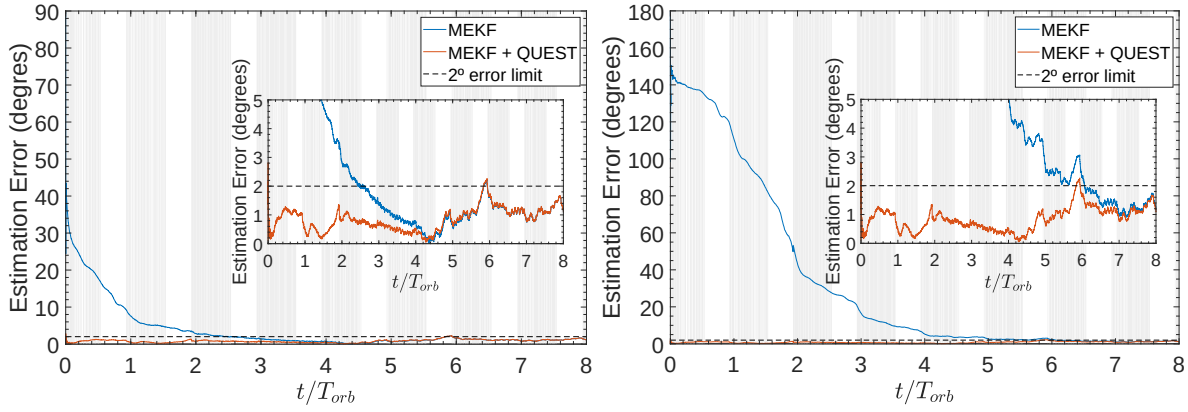
The convergence of the MEKF will be studied using four different cases. The $1$st and $2$nd cases will be used to simulate a real-world scenario, where the MEKF should acquire the attitude of the ORCASat while in detumbling mode, using the controller from Section 7.1, with the momentum wheel on. In the $3$rd and $4$th cases the simulation will be initialized using a higher angular velocity and no controller will be used to damp those angular rates. These last two cases will test the ability of the MEKF to achieve convergence while subjected to a higher angular rate. In cases 1 and 2, the angular rate has a mean value of $0.0029\text{rad}\,\text{s}^{-1}$, a maximum value of $0.0129\text{rad}\,\text{s}^{-1}$, and a minimum value of $1.23 \times 10^{-5}\text{rad}\,\text{s}^{-1}$, while in cases 3 and 4 the mean magnitude of the angular velocity is $0.0482\text{rad}\,\text{s}^{-1}$, the maximum value is $0.0559\text{rad}\,\text{s}^{-1}$, and the minimum value is $0.0425\text{rad}\,\text{s}^{-1}$. The angular rate profiles are shown in Figure D.3. The initial estimation error for cases 1 and 3 is $90^\circ$ while for cases 2 and 4 is $180^\circ$. These errors are prescribed about the unit vector $\hat{\mathbf{v}} = \begin{bmatrix}\sqrt{3}/3 & \sqrt{3}/3 & \sqrt{3}/3\end{bmatrix}^\top$ for all cases. The initial gyro bias value is $\begin{bmatrix}-0.02 & -0.02 & -0.02\end{bmatrix}^\top \text{rad}\,\text{s}^{-1}$ for all cases corresponding to an initial bias error of $\begin{bmatrix}-0.0271 & -0.0209 & -0.0321\end{bmatrix}^\top \text{rad}\,\text{s}^{-1}$ whose magnitude is more than $3$ times bigger than the magnitude of the bias considered for the gyroscope sensor (Equation (B.3b)). The initial covariance matrix is $\mathbf{P}_0 = \text{blkdiag}\left(\begin{bmatrix}10\mathbf{I}_3 & \mathbf{I}_3\end{bmatrix} \times 10^{-3}\right)$, where blkdiag denotes a block diagonal matrix. The evolution of the estimation error is shown in Figure 7.9, while Table 7.14 provides a summary of the convergence times.

In can be seen that the MEKF was able to converge in all the cases presented, that higher initial attitude errors lead to longer convergence times and that a higher initial rate also leads to longer convergence times. The MEKF algorithm in spite of achieving convergence even from a high $180^\circ$ error




(a) Detumble mode with $90^\circ$ initial error (Case 1).

(b) Detumble mode with $180^\circ$ initial error (Case 2).




(c) Higher angular rate with $90^\circ$ initial error (Case 3).

(d) Higher angular rate with $180^\circ$ initial error (Case 4).

Figure 7.9: Convergence performance of the MEKF. The grey vertical bars represent the orbit regions where the sun sensors provide data to the estimators.

Table 7.14: MEKF convergence times.

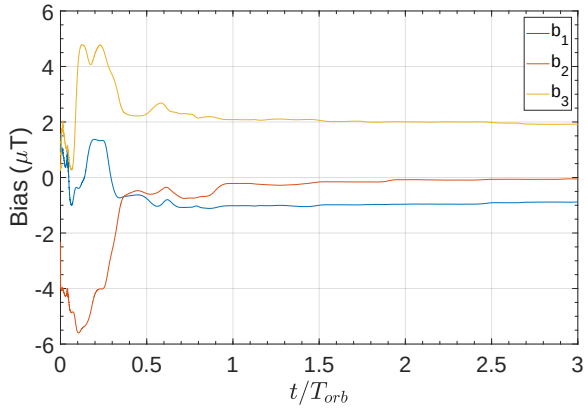
| | Time to reach $2°$ error $(t/T_{orb})$ | | Time to reach $2°$ error $(\mathrm{s})$ | |
|---|---|---|---|---|
| Case | MEKF | MEKF+QUEST | MEKF | MEKF+QUEST |
| 1 | 2.36 | 0.04 | 13080.8 | 219.9 |
| 2 | 4.37 | 0.04 | 24250.6 | 219.9 |
| 3 | 2.44 | 0 | 13524.4 | 0 |
| 4 | 6.06 | 0 | 33626.9 | 0 |

it can take a relatively long time to do so, up to $4.4$ orbits in normal detumble conditions and up to $6$ orbits in the higher angular rate case. The use of the QUEST algorithm to initialize the MEKF allows us to reduce those convergence times to virtually no time, as seen in Table 7.14. In cases 1 and 2 the attitude quaternion determined by QUEST and used to initialize the MEKF had a determination error of $3.63^\circ$ while in cases 3 and 4 this error was only $0.10^\circ$. From Figure 7.9 it can be seen that the $2^\circ$ limit does not exactly correspond to the full convergence of the filter. The time that it takes for the filter to fully converge and to enter in its nominal behaviour is larger, as seen by the intersection between the orange and blue lines. In practice, this does not present a concern since the error is already small and the difference between the blue and orange lines is also very small.

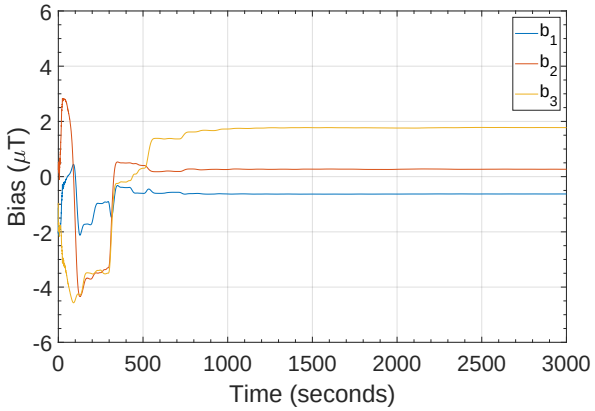
**Magnetometer Calibration Extended Kalman Filter (MCEKF)**

The convergence of the MCEKF will be tested using the values for the magnetometer bias and scale factor and misalignment matrix from scenarios 1 and 4 given by Table 7.10, representing a real-world case scenario and a hypothetically worst-case scenario for the magnetometer values. These two different sets of values will then be combined with the two different angular velocities from the transient simulations of the MEKF which represent the detumbling mode conditions when the momentum wheel is on and an initial higher angular rate case. The calibration filter is initialized with every component of the state vector $\mathbf{x}$ (Equation (4.97)) set to zero and with an initial covariance matrix $\mathbf{P}_0 = \text{blkdiag}\left(\left[\mathbf{I}_3 \quad 10^6\mathbf{I}_6\right] \times 10^{-12}\right)$. The MCEKF estimates for the bias vector are shown in Figure 7.10. Similar results are obtained for the $\mathbf{D}$ matrix parameters.

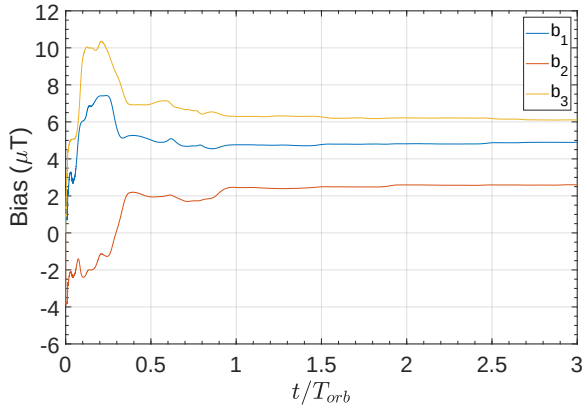
The MCEKF was able to converge in all the cases devised. By comparing Figure 7.10a with Figure 7.10c and Figure 7.10b with Figure 7.10d it can be seen that the time the filter takes to converge is approximately similar independently on the set of values used for the magnetometer bias and for $\mathbf{D}$. The MCEKF takes approximately $1$ orbit to converge under detumbling conditions, and approximately $0.15$ orbits or $800\text{s}$ ($\simeq 14\text{min}$) to converge under the higher angular rate conditions. Unlike the MEKF the MCEKF appears to have a faster convergence for higher angular rates.



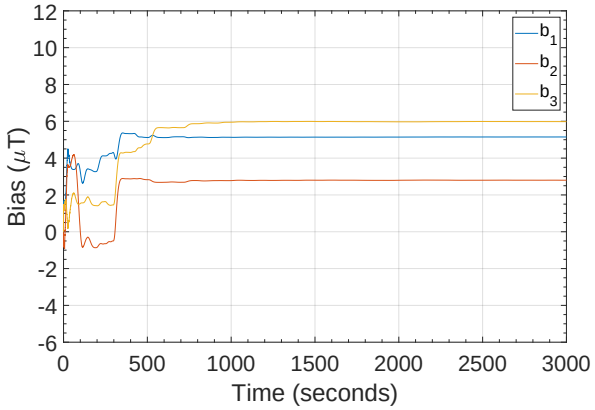

(a) Scenario 1 values under detumbling conditions.




(b) Scenario 1 values under higher angular rates.




(c) Scenario 4 values under detumbling conditions.




(d) Scenario 4 values under higher angular rates.

Figure 7.10: Convergence performance of the MCEKF - estimated magnetometer bias vector $\hat{\mathbf{b}}$.

# Chapter 8

# Conclusions

## 8.1 Achievements

The goal of this thesis was to develop an ADCS capable of meeting the requirements outlined in Section 1.3. To accomplish this goal, the performance of the attitude estimator was tested under different scenarios and an algorithm for on-line calibration of the magnetometer was proposed and analyzed. Four different attitude control algorithms were presented and their steady-state performance was tested and compared among them. Next, based on the simulations performed, the ORCASat's nadir-pointing controller was selected and its performance was tested under different conditions. The performance of the detumbling algorithm was also analyzed under different scenarios. The different performance analyses were done using a realistic orbital simulator developed in Matlab/Simulink.

The detumbling algorithm was proven effective in reducing the angular speed of the ORCASat from angular rates as high as $90\,°\mathrm{s}^{-1}$ to less than $2\,°\mathrm{s}^{-1}$ in about $1$ orbit. The detumbling controller was also proved to be capable of achieving angular rates low enough for the pointing controller and the attitude estimator to converge to their nominal states. The momentum wheel was shown to have a positive effect in decreasing the angular rate values under steady-state conditions, while the inertia matrix uncertainty was shown to have very little impact in the steady and transient-state performance of the controller, decreasing very slightly the average angular rate, while at the same time increasing its maximum value.

Next, the nominal performance of the different control algorithms was presented and a comparison between the efficiency in steady-state conditions of the different controllers was done. It was seen that the linear controllers provide a better performance than the SMC, not only in terms of efficiency but also in terms of lower mean and lower maximum pointing errors. Based on the simulations performed in steady-state conditions, the nominal controller for the ORCASat and its respective gain were selected. The selected controller was the CGC using $K_Q = 0.7$. This controller was selected because it presented a very similar performance to that of the other linear controllers while having at the same time a simpler implementation. The maximum and mean pointing errors, and the mean total dipole moment obtained for the selected controller and respective gain, while using the MEKF without the implementation of the MCEKF, was $3.8^\circ$, $2.0^\circ$, and $0.0121\mathrm{A\,m}^2$, respectively. The transient behaviour of the ORCASat's

controller was then tested using different initial pointing errors and angular velocities. The controller was shown to converge in all the cases considered in less than $0.4$ orbits. The ORCASat's controller was also tested to model uncertainties, namely the inertia matrix uncertainty and the performance degradation of the momentum wheel. The inertia matrix uncertainty was shown to have very little impact in the steady-state and in the transient-state performance of the controller, while the momentum wheel degradation was responsible for an increase in the pointing error under steady-state conditions as well as for affecting the convergence times in the transient-state. Nonetheless, the selected controller and gain were still able to converge below the $10^\circ$ limit in less than $0.5$ orbits and to achieve a steady-state pointing error below $5^\circ$, while including the performance of the MEKF, thus meeting the ADCS pointing requirements.

Subsequently, the attitude estimators and the MCEKF were tested. It was shown that the performance of the MEKF was much better than QUEST's. Nonetheless, the QUEST algorithm will be incorporated in the ORCASat, since it allows the MEKF to reduce its convergence time from almost $4.4$ orbits, under detumbling conditions with an initial attitude error of $180^\circ$, to approximately $0.04$ orbits. The performance of the MEKF alone did not meet the attitude knowledge requirement, even in the best (real) case scenario. The introduction of the MCEKF improved considerably the performance of the MEKF by reducing the maximum and mean estimation errors of the MEKF from $2.9^\circ$ to $0.56^\circ$, and from $1.64^\circ$ to $0.21^\circ$ respectively, under steady-state conditions in the true case scenario. Even in a worst-case scenario where the MEKF showed a maximum estimation error of almost $10^\circ$, the inclusion of the MCEKF allowed the MEKF to meet the ADCS $2^\circ$ attitude knowledge requirement. Simulations were also performed to test the convergence of the MCEKF, showing that this filter benefits more from slightly higher angular rates and that the values of the magnetometer bias and $\mathbf{D}$ do not have a major impact in the convergence time of this filter. Under a true case scenario, the MCEKF took approximately $1$ orbit to converge, being this algorithm, from the ones tested, the one that takes longer to converge.

## 8.2 Future Work

The next step in the development of the ADCS of the ORCASat will be the conversion of the ADCS blocks from the Simulink environment to $C$, and in the actual implementation of the software in the ADCS board itself. Software-In-the-Loop (SIL) and Hardware-In-the-Loop (HIL) testing methodologies should be implemented to make sure the software implemented in the ADCS board is consistent with the Matlab/Simulink simulations, and that the software is capable of interacting with the sensors and the actuators of the ORCASat and perform as designed. In-depth simulations must be done to ensure that the ADCS controller block introduced in Appendix C commands the ADCS states as desired. The ORCASat uses an orbital propagator which together with the GNSS receiver propagate the orbital position. The use of a simplified general perturbations model SGP4 or SGP8, to predict the position and velocity of the ORCASat, together with the NORAD Two-Line Element Sets, should be considered as an alternative to the present implementation. This solution not only does not require the use of a GNSS receiver but it also has proven to be very robust, and implementations in $C$ can be found online in CelesTrak's website [72] and in NASA's GitHub [73].

# Bibliography


[1] C. S. Agency. What is a cubesat. URL http://www.asc-csa.gc.ca/eng/satellites/cubesat/what-is-a-cubesat.asp. Last visited on 08/10/2019.

[2] Cubesat market by application (earth observation traffic monitoring, communication, science technology and education), end user (government military, commercial, non-profit organizations), size, subsystem, and region - global forecast to 2023. MarketsandMarkets, December 2018. URL https://www.marketsandmarkets.com/Market-Reports/cubesat-market-58068326.html. Last visited on 14/07/2020.

[3] E. Kulu. Nanosatellite and cubesat database. URL https://www.nanosats.eu/database. Last visited on 09/10/2019.

[4] C. P. S. University. Cubesat design specification rev. 13. URL https://static1.squarespace.com/static/5418c831e4b0fa4ecac1bacd/t/56e9b62337013b6c063a655a/1458157095454/cds_rev13_final2.pdf. Last visited on 09/10/2019.

[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, 2017.

[6] Ubc orbit projects. URL https://www.ubcorbit.com/projects. Last visited on 14/07/2020.

[7] ORCASat. URL https://www.orcasat.ca/orcasat. Last visited on 14/07/2020.

[8] CHIME telescope. URL https://chime-experiment.ca/cosmology. Last visited on 14/07/2020.

[9] F. L. Markley and J. L. Crassidis. *Fundamentals of spacecraft attitude determination and control*, volume 33. Springer, 2014.

[10] J. Crassidis and J. Junkins. *Optimal Estimation of Dynamic Systems*. Chapman & Hall/CRC Applied Mathematics & Nonlinear Science. CRC Press, 2011. ISBN 9781439839867. URL https://books.google.pt/books?id=CmjRBQAAQBAJ. Last visited on 14/07/2020.

[11] J. R. Wertz. *Spacecraft attitude determination and control*, volume 73. Springer Science & Business Media, 2012.

[12] B. Wie. *Space Vehicle Dynamics and Control*. AIAA education series. American Institute of Aeronautics and Astronautics, 1998. ISBN 9781563472619. URL https://books.google.pt/books?id=n97tEQvNyVgC. Last visited on 14/07/2020.

[13] H. D. BLACK. A passive system for determining the attitude of a satellite. *AIAA Journal*, 2(7): 1350–1351, 1964. doi: 10.2514/3.2555. URL https://doi.org/10.2514/3.2555. Last visited on 14/07/2020.

[14] M. D. SHUSTER and S. D. OH. Three-axis attitude determination from vector observations. *Journal of Guidance and Control*, 4(1):70–77, 1981. doi: 10.2514/3.19717. URL https://doi.org/10.2514/3.19717. Last visited on 14/07/2020.

[15] G. Wahba. A Least Squares Estimate of Satellite Attitude. *SIAM Review*, 7:409–409, July 1965. doi: 10.1137/1007077. Last visited on 14/07/2020.

[16] J. E. Keat. Analysis of Least-Squares Attitude Determination Routine DOAO, Feb. 1977. URL https://doi.org/10.5281/zenodo.32291. Scan by Charles Karney of a copy provided by the author. Last visited on 31/07/2020.

[17] M. SHUSTER. *Approximate algorithms for fast optimal attitude computation*. doi: 10.2514/6.1978-1249. URL https://arc.aiaa.org/doi/abs/10.2514/6.1978-1249. Last visited on 14/07/2020.

[18] F. L. Markley and D. Mortari. Quaternion attitude estimation using vector observations. *Journal of the Astronautical Sciences*, 48(2):359–380, 2000.

[19] J. L. Crassidis, F. L. Markley, and Y. Cheng. Survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*, 30(1):12–28, 2007. doi: 10.2514/1.22452. URL https://doi.org/10.2514/1.22452. Last visited on 14/07/2020.

[20] J. Farrell. Attitude determination by kalman filtering. *Automatica*, 6(3):419 – 430, 1970. ISSN 0005-1098. doi: https://doi.org/10.1016/0005-1098(70)90057-9. URL http://www.sciencedirect.com/science/article/pii/0005109870900579. Last visited on 14/07/2020.

[21] M. D. Shuster. The quest for better attitudes. *The Journal of the Astronautical Sciences*, 54(3-4): 657–683, 2006.

[22] N. Ibrahim. Attitude and orbit control of small satellites for autonomous terrestrial target tracking. Master's thesis, 2013.

[23] K. Svartveit. Attitude determination of the ncube satellite. *NTNU, June*, 2003.

[24] B. O. Sunde. Sensor modelling and attitude determination for micro-satellite. *MSc. thesis, Norwegian University of Science and Technology*, (2005), 2005.

[25] B. Johnston-Lemke. High performance attitude determination and control for nanosatellite missions. Master's thesis, 2011.

[26] K. Krogh and E. Schreder. Attitude determination for aau cubesat. Master's thesis, 2002.

[27] T. Xiang, T. Meng, H. Wang, K. Han, and Z.-H. Jin. Design and on-orbit performance of the attitude determination and control system for the zdps-1a pico-satellite. *Acta Astronautica*, 77:182 – 196, 2012. ISSN 0094-5765. doi: https://doi.org/10.1016/j.actaastro.2012.03.023. URL http://www.sciencedirect.com/science/article/pii/S0094576512000951. Last visited on 14/07/2020.

[28] D. Messmann, T. Grübler, F. Coelho, T. Ohlenforst, J. Bruegge, F. Mauracher, M. Doetterl, S. Plamauer, P. Schnierle, T. Kale, M. Seifert, A. Fuhrmann, E. Karagiannis, A. Ulanowski, T. Lausenhammer, A. Meraner, and M. Langer. Advances in the development of the attitude determination and control system of the cubesat move-ii. 07 2017.

[29] E. D. Wise. Design, analysis, and testing of a precision guidance, navigation, and control system for a dual-spinning cubesat. Master's thesis, Massachusetts Institute of Technology, 2013.

[30] B. Wie and P. M. Barba. Quaternion feedback for spacecraft large angle maneuvers. *Journal of Guidance, Control, and Dynamics*, 8(3):360–365, 1985. doi: 10.2514/3.19988. URL https://doi.org/10.2514/3.19988. Last visited on 14/07/2020.

[31] H. C. Polat, J. Virgili-Llop, and M. Romano. Survey, statistical analysis and classification of launched cubesat missions with emphasis on the attitude control method. *Journal of small satellites*, 5(3): 513–530, 2016.

[32] E. J. Øverby. Attitude control for the norwegian student satellite ncube. 2004.

[33] B. Graf. Swisscube control algorithm design and validation. *Master's thesis, Ecole Polytechnique Federale de Lausanne*, 2007.

[34] N. Jovanovic. Aalto-2 satellite attitude control system. G2 pro gradu, diplomityö, 2014-08-25. URL http://urn.fi/URN:NBN:fi:aalto-201409012592. Last visited on 14/07/2020.

[35] K. L. Makovec. A nonlinear magnetic controller for three-axis stability of nanosatellites. Master's thesis, 2001-07-12. URL https://vtechworks.lib.vt.edu/handle/10919/34131. Last visited on 14/07/2020.

[36] M. T. Nehrenz. Initial design and simulation of the attitude determination and control system for lightsail-1. 2010.

[37] R. J. Sellers. A gravity gradient, momentum-biased attitude control system for a cubesat. Master's thesis, 2013.

[38] M. Lovera and A. Astolfi. Global magnetic attitude control of spacecraft. In *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, volume 1, pages 267–272. IEEE, 2004.

[39] K. L. Musser and W. L. Ebert. Autonomous spacecraft attitude control using magnetic torquing only. 1989.

[40] R. Wisniewski. *Satellite attitude control using only electromagnetic actuation*. PhD thesis, Aalborg University. Department of Control Engineering, 1996.

[41] M. Lovera and A. Astolfi. Spacecraft attitude control using magnetic actuators. *Automatica*, 40(8): 1405–1414, 2004.

[42] P. Wang and Y. B. Shtessel. Satellite attitude control using only magnetorquers. In *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No. 98CH36207)*, volume 1, pages 222–226. IEEE, 1998.

[43] J. Gießelmann. Development of an active magnetic attitude determination and control system for picosatellites on highly inclined circular low earth orbits. 2006.

[44] S. B. Cotten. Design, analysis, implementation, and testing of the thermal control, and attitude determination and control systems for the canx-7 nanosatellite mission. Master's thesis, 2014.

[45] R. Holst. Satellite attitude control: Using magnetorquers with magnetic dipole moment calcellation. Master's thesis, 2014.

[46] A. M. Oluwatosin, Y. Hamam, and K. Djouani. Attitude control of a cubesat in a circular orbit using reaction wheels. In *2013 Africon*, pages 1–8, 2013.

[47] D. Hegel. Flexcore: Low-cost attitude determination and control enabling high-performance small spacecraft. 2016.

[48] B. Lobo-Fernandes. Design and analysis of an attitude determination and control system for the orcasat nanosatellite. Master's thesis, Instituto Superior Técnico, June 2019.

[49] D. Rondão. Modelling and simulation of the ecosat-iii attitude determination and control system. Master's thesis, Instituto Superior Técnico, April 2016.

[50] K. Wakker. *Fundamentals of Astrodynamics*. 01 2015. ISBN 978-94-6186-419-2.

[51] J. Stuelpnagel. On the parametrization of the three-dimensional rotation group. *SIAM Review*, 6 (4):422–430, 1964. ISSN 00361445. URL `http://www.jstor.org/stable/2027966`. Last visited on 18/10/2019.

[52] F. L. Markley. Spacecraft Attitude Representations. *NASA*, Jan. 1999. URL `https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19990110711.pdf`. Last visited on 18/09/2019.

[53] F. L. Markley. Unit quaternion from rotation matrix. *Journal of Guidance, Control, and Dynamics*, 31 (2):440–442, 2008. doi: 10.2514/1.31730. URL `https://doi.org/10.2514/1.31730`. Last visited on 14/07/2020.

[54] J. Wertz, N. C. for Space Technology (U.S.), and N. R. L. (U.S.). *Mission Geometry: Orbit and Constellation Design and Management : Spacecraft Orbit and Attitude Systems*. Space technology library. Microcosm Press, 2001. ISBN 9781881883074.

[55] M. J. Sidi. *Spacecraft dynamics and control: a practical engineering approach*, volume 7. Cambridge university press, 1997.

[56] J. L. Junkins and J. D. Turner. *Optimal spacecraft rotational maneuvers*. Elsevier, 2012.

[57] P. Hughes. *Spacecraft attitude dynamics*. J. Wiley, 1986. ISBN 9780471818427.

[58] S. M. P. A. C. B. M. N. B. H. A. W. V. R. S. M. Chulliat, A. and A. Thomson. The us/uk world magnetic model for 2015-2020: Technical report. *National Geophysical Data Center, NOAA*. doi: 10.7289/V5TB14V7. URL https://data.nodc.noaa.gov/cgi-bin/iso?id=gov.noaa.ngdc:WMM2015_Technical_Report. Last visited on 14/07/2020.

[59] D. S. Naidu. *Optimal control systems*. CRC press, 2002.

[60] G. Avanzini and F. Giulietti. Magnetic detumbling of a rigid spacecraft. *Journal of Guidance, Control, and Dynamics*, 35(4):1326–1334, 2012. doi: 10.2514/1.53074. URL https://doi.org/10.2514/1.53074. Last visited on 31/07/2020.

[61] JPL. Planetary ephemerides. URL https://ssd.jpl.nasa.gov/?ephemerides. Last visited on 14/07/2020.

[62] J. M. Picone, A. E. Hedin, D. P. Drob, and A. C. Aikin. Nrlmsise-00 empirical model of the atmosphere: Statistical comparisons and scientific issues. *Journal of Geophysical Research: Space Physics*, 107(A12):SIA 15–1–SIA 15–16, 2002. doi: 10.1029/2002JA009430. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2002JA009430. Last visited on 14/07/2020.

[63] N. K. Pavlis, S. A. Holmes, S. C. Kenyon, and J. K. Factor. The development and evaluation of the earth gravitational model 2008 (egm2008). *Journal of Geophysical Research: Solid Earth*, 117(B4), 2012. doi: 10.1029/2011JB008916. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2011JB008916. Last visited on 14/07/2020.

[64] A. Doknjas, B. Edwards, B. Lobo-Fernandes, P. Ogilvie, R. Arthurs, and A. Zoltan. Orcasat 2021 - mission concept document. ORCASat Documentation, January 2019.

[65] H. Technologies. Ss200 sun sensor specifications, 2019. URL https://hyperiontechnologies.nl/wp-content/uploads/2019/11/HT_SS200.pdf. Last visited on 14/07/2020.

[66] J. Guido and C. Brisebois. Sun sensor report. ORCASat Documentation, March 2019.

[67] STMicroelectronics. inemo lsm9ds1 datasheet, 2015. URL https://www.st.com/resource/en/datasheet/lsm9ds1.pdf. Last visited on 14/07/2020.

[68] H. Technologies. Rw210 reaction wheel specifications, 2019. URL https://hyperiontechnologies.nl/wp-content/uploads/2019/11/HT_RW210.pdf. Last visited on 14/07/2020.

[69] NovAtel. Receivers oem719 product sheet, November 2019. URL https://hexagondownloads.blob.core.windows.net/public/Novatel/assets/Documents/Papers/OEM719-Product-Sheet/OEM719-Product-Sheet.pdf. Last visited on 14/07/2020.

[70] F. G. Stray. Attitude control of a nano satellite. Master's thesis, 2010.

[71] J. Armstrong, C. Casey, G. Creamer, and G. Dutchover. Pointing control for low altitude triple cubesat space darts. 2009.

[72] T. Kelso. Revisiting spacetrack report 3, 2006. URL https://celestrak.com/publications/AIAA/2006-6753/. Last visited on 14/07/2020.

[73] D. Vallado. Sgp4 c source code, 2013. URL https://github.com/nasa/ECI/tree/master/examples/sgp4Prop/fsw/src. Last visited on 14/07/2020.

# Appendix A

# Algorithms Implementation

In this chapter, the implemented algorithm for eclipse calculation (Figure A.1), as well as the flowcharts corresponding to the different control algorithms (Figures A.2 and A.3) and estimation filters (Figures A.4, A.5, and A.6), are presented.

The eclipse calculation algorithm is employed to compute whether the ORCASat is in eclipse or not by using a ray-sphere intersection method [49]. This code was implemented inside a Matlab/Simulink block. It takes as input the Sun vector *sun_pos*, the Earth's radius *R*, and the spacecraft position vector with respect to the Earth's center *sat_pos*. It returns a Boolean flag, *Lit*, whose value is $0$ when the spacecraft is in eclipse and $1$ otherwise.

```
function [Lit] = fcn(sun_pos, sat_pos)

% Math for determining ray-sphere intersection:
% http://www.lighthouse3d.com/tutorials/maths/ray-sphere-intersection/

R = 6378.1; % Earth's radius km
sat_pos_km = sat_pos/1000;  % Convertion to km
sv = sun_pos - sat_pos_km;  % Vector from the satellite to the Sun (km)
sv_norm = sv/norm(sv);  % Compute the unit Sun vector

% Project the vector from the satellite to the Earth's center onto the unit Sun
% vector -> pc
sat_earth = -1*sat_pos_km;
dot_prod = dot(sv_norm, sat_earth);
pc = dot_prod/norm(sv_norm)*sv_norm;

% Visibility check:
if (dot_prod <= 0)  % If projection length is negative, we are in front
    Lit=1;          % of the Earth and therefore implicitly have visibility
else    % Projection is positive therefore we may be in the dark
    if (norm(sat_earth - pc) > R)    % Compare the distance sat_earth
        Lit = 1;                     % perpendicular to sv with the Earth's radius
    else
        Lit = 0;
    end
end
```

Figure A.1: Matlab implementation of the eclipse calculation algorithm.

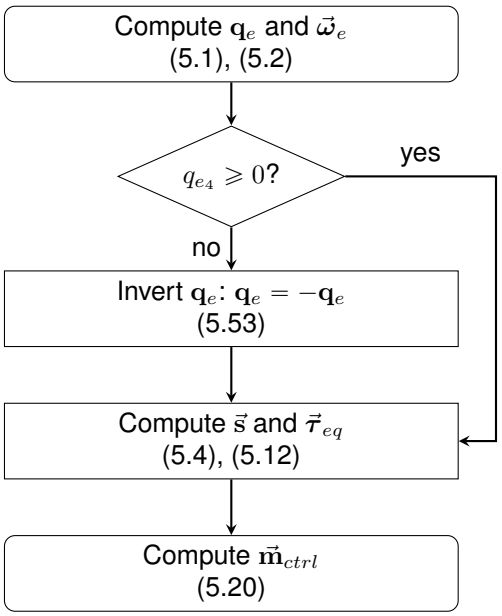



Figure A.2: SMC flowchart.

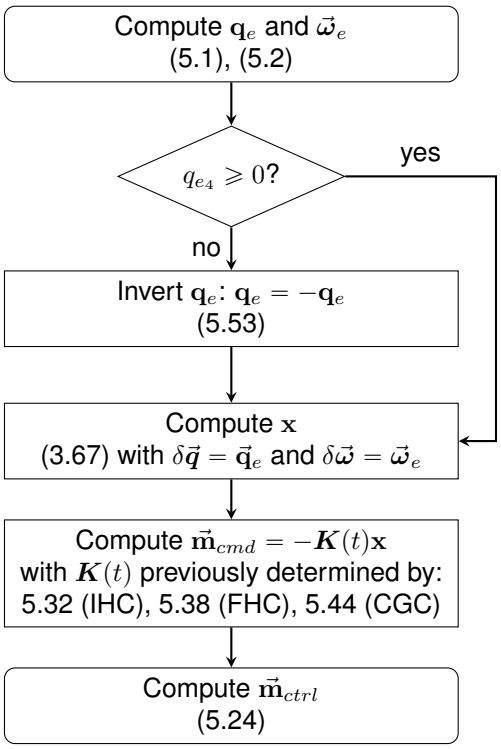



Figure A.3: IHC, FHC and CGC flowchart.

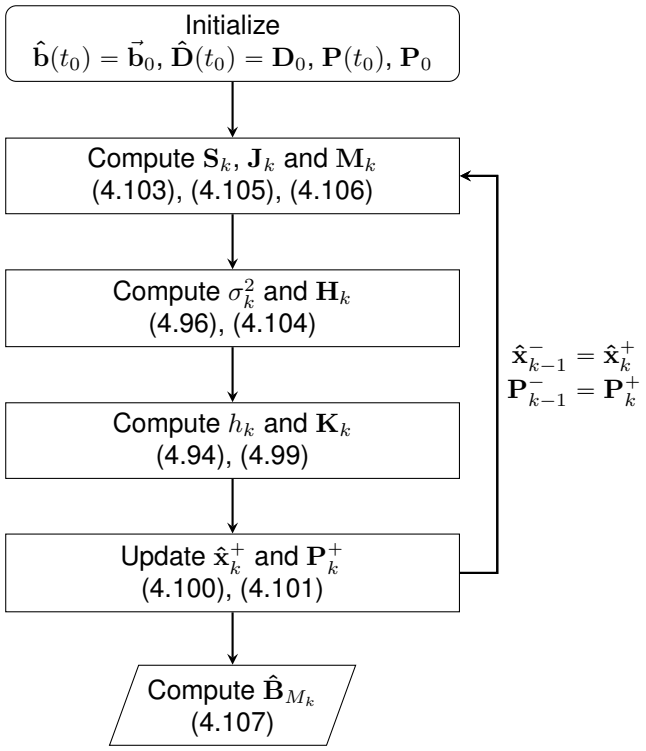



Figure A.4: MCEKF flowchart.

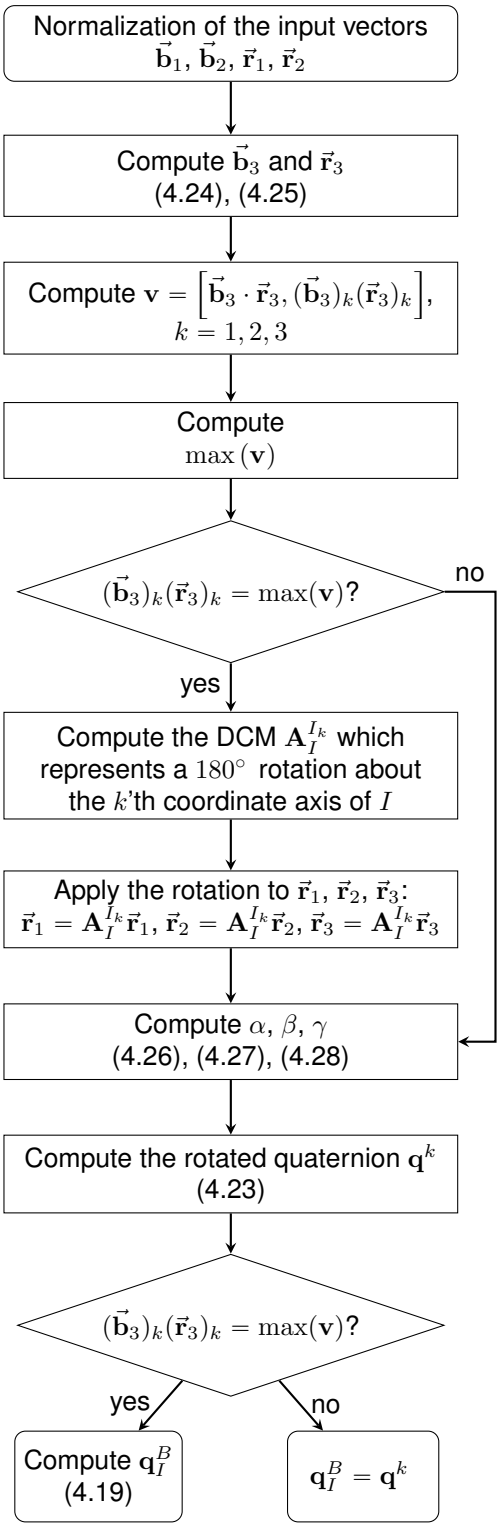



Figure A.5: QUEST flowchart.

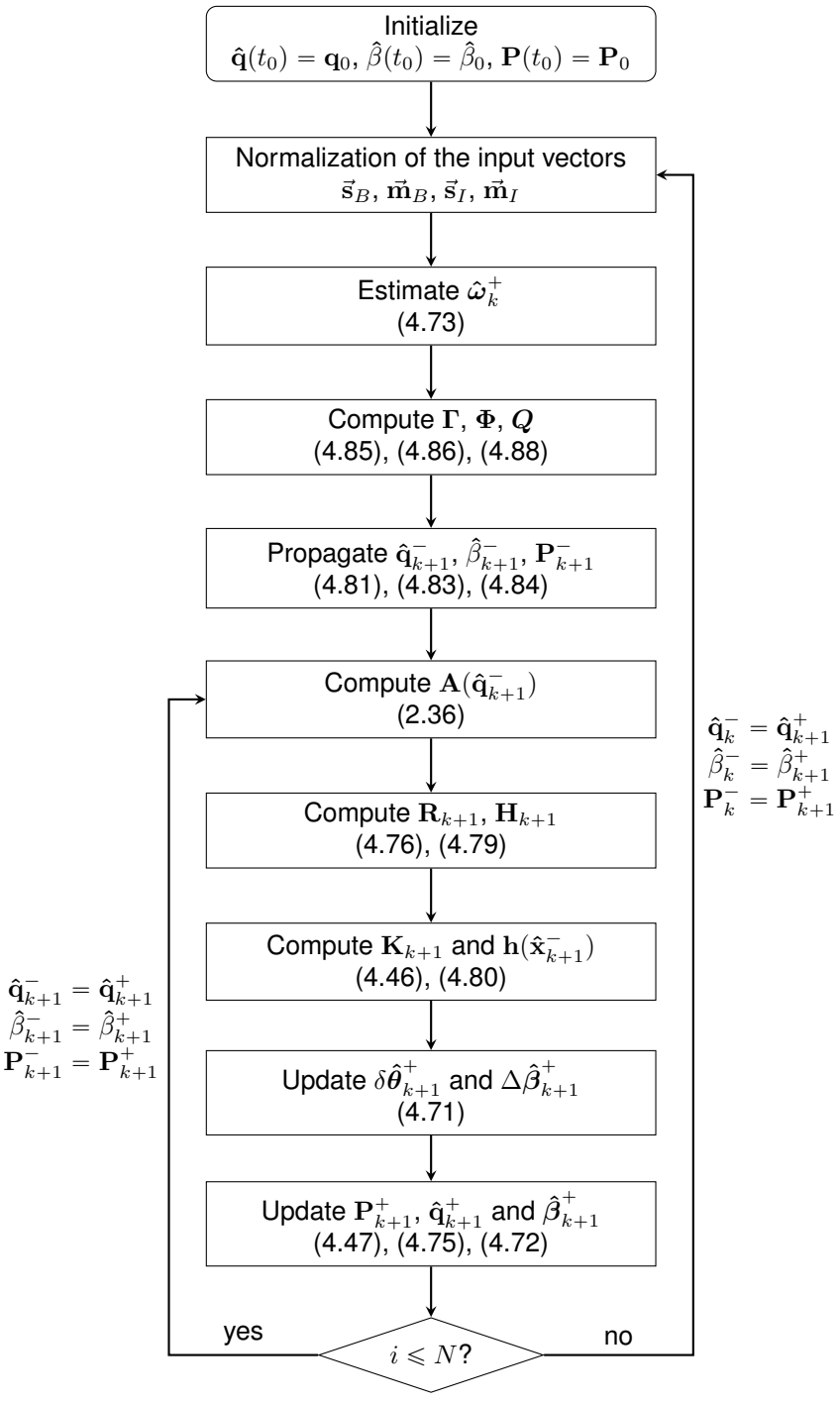



Figure A.6: MEKF flowchart.

# Appendix B

# Hardware Model Parameters

This appendix compiles the model parameters taken from the different sensor datasheets [65, 67] and from reference [49].

The magnetometer and gyroscope scale factors and misalignments matrix is given by

$$\mathbf{K} = \begin{bmatrix} g_1 \sin(90+\theta_1)\cos\phi_1 & g_1 \sin(90+\theta_1)\sin\phi_1 & g_1 \cos(90+\theta_1) \\ g_2 \sin(90+\theta_2)\cos(90+\phi_2) & g_2 \sin(90+\theta_2)\sin(90+\phi_2) & g_2 \cos(90+\theta_2) \\ g_3 \sin\theta_3 \cos\phi_3 & g_3 \sin\theta_3 \sin\phi_3 & g_3 \cos\theta_3 \end{bmatrix} \tag{B.1}$$

where the scale factors $g_1$, $g_2$, and $g_3$ and the misalignment angles $\theta_i$ and $\phi_i$, with $i = 1, 2, 3$ are given in Table B.1 for the magnetometer and in Equation (B.2) for the gyroscope. The static bias of the magnetometer, $\vec{\mathbf{b}}_{mag}$, and the static bias of the gyroscope, $\vec{\mathbf{b}}_{gyro}$, are given in Equation (B.3). The other parameters can be found in Table B.2 for the sun sensor, in Table B.3 for the magnetometer and in Table B.4 for the gyroscope.

Table B.1: Magnetometer scale factors and misalignment parameters.

| Sensor | Scale Factors | | | Misalignment Angles (degrees) | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $g_1$ | $g_2$ | $g_3$ | $\theta_1$ | $\theta_2$ | $\theta_3$ | $\phi_1$ | $\phi_2$ | $\phi_3$ |
| Magnetometer | 1.046 | 1.125 | 1.161 | 1.07 | −0.43 | −0.01 | −0.01 | 0.31 | 0.00 |

$$g_i = 1.00015, \quad i = 1, 2, 3 \tag{B.2a}$$

$$\theta_i = \phi_i = 0.00572957795^\circ, \quad i = 1, 2, 3 \tag{B.2b}$$

$$\vec{\mathbf{b}}_{mag} = \begin{bmatrix} -673 & 309 & 2082 \end{bmatrix} (\mathsf{nT}) \tag{B.3a}$$

$$\vec{\mathbf{b}}_{gyro} = \begin{bmatrix} 0.0071331 & 0.0008607 & 0.0120975 \end{bmatrix} (\mathsf{rad\,s^{-1}}) \tag{B.3b}$$

Table B.2: Sun sensor model parameters.

| Parameter | Value |
|---|---|
| Field of View | $110^\circ$ |
| Noise standard deviation | $0.5^\circ$ |
| Sensitivity | $125^\circ$ /LSB |
| Analog-to-digital resolution | $16$bit |
| Sampling rate | $10$Hz |

Table B.3: Magnetometer model parameters.

| Parameter | Value |
|---|---|
| Noise standard deviation | $15$nT |
| Sensitivity | $14$nT/LSB |
| Analog-to-digital resolution | $16$bit |
| Sampling rate | $10$Hz |

Table B.4: Gyroscope model parameters.

| Parameter | Value |
|---|---|
| Sensitivity | $8.75$mdps/LSB |
| Analog-to-digital resolution | $16$bit |
| Sampling rate | $10$Hz |
| ARW | $0.1°\text{h}^{-1/2}$ |
| RRW | $0.4323°\text{h}^{-3/2}$ |

# Appendix C

# ADCS Controller Block

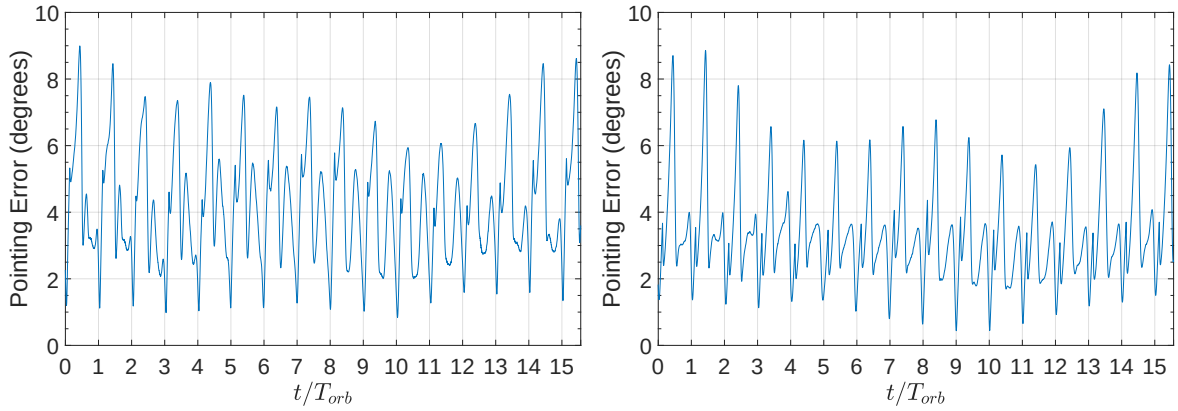
The ADCS Controller block is responsible for the proper initialization of the ADCS and for making it to transition to nominal mode. When the ADCS is first turned on, it will start with the detumbling controller and will keep using it until entering in nominal mode. The detumbling controller is responsible for reducing the angular speed of the ORCASat to $\|\vec{\omega}\| \leqslant 0.05\text{rad}\,\text{s}^{-1}$. This threshold value was proven possible to be reached using the detumbling controller presented in Section 5.3, as it can be seen in Section 7.1. Once this threshold is reached, the ADCS should activate the momentum wheel whose angular velocity will increase linearly with time (constant torque applied to the wheel) until it reaches its nominal value. While this procedure is done, the satellite's angular velocity should not exceed $\|\vec{\omega}\| = 0.1\text{rad}\,\text{s}^{-1}$. If this angular rate is exceeded, the wheel should stop accelerating and keep its current rotational speed until $\|\vec{\omega}\| \leqslant 0.05\text{rad}\,\text{s}^{-1}$ again. This method allows for a better control of the wheel speeding up process without destabilizing the CubeSat. The angular acceleration computed for the CubeSat's momentum wheel corresponds to $5\text{rad}\,\text{s}^{-2}$. This angular acceleration produces a torque along the wheel axis slightly less than $10\mu\text{N}\,\text{m}$, which corresponds to $10\%$ of the maximum torque the wheel can provide. Once the momentum wheel is at its nominal speed, the CubeSat's angular rate should decrease below $0.03\text{rad}\,\text{s}^{-1}$ ($\|\vec{\omega}\| \leqslant 0.03\text{rad}\,\text{s}^{-1}$) before the next initialization stage. This threshold was selected since both the estimators (MEKF and MCEKF) and the nadir pointing controller implemented were found to converge below this value of angular velocity, as can be seen in Chapter 7. The detumbling controller was also found to be capable of reaching this angular rate as seen in Section 7.1. In order for the ORCASat to acquire its attitude, it will require the current time, the orbital position, and the orbital velocity. This data will be obtained via the GNSS receiver. The first fix of the GNSS receiver can take up too $15\text{min}$ and it is one of the components which requires more power to operate. In order to avoid the ORCASat getting the GNSS fix and then entering in eclipse, where it won't be able to get its attitude, it was decided that the GNSS receiver will only start if the CubeSat knows that it just left eclipse and will enter in a sunlit area of the orbit. This step was accomplished by using a $1500\text{s}$ watchdog timer which starts counting every time none of the four sun sensors detect the presence of the sun. Once the watchdog reaches $t = 1500\text{s}$, a flag is set to one, indicating that the CubeSat is in an eclipse region of the orbit. In this way, energy is saved in the process of acquiring the attitude of the ORCASat. Once

the ORCASat gets its first GNSS fix, the first stage of the onboard orbital propagator together with the Attitude Estimator block will be initiated as soon as at least one of the sun sensors start providing data. In order to ensure proper convergence of the MEKF, a $5\text{min}$ or $300\text{s}$ watchdog timer is used. This amount of time was proven to be sufficient for the convergence of the MEKF as it can be seen in Section 7.3.2. If during this time the MEKF stops from having sun sensor data, the attitude estimator (MEKF) will be turned off and it will restart once the sun sensors start providing data again. Once the attitude determination is completed, the ORCASat will enter nominal mode and the controller will switch from detumble mode to nadir pointing mode. The second stage of the orbital propagator, responsible for computing the aerodynamic and solar radiation pressure disturbance forces, will be also switched on, now that it has an accurate attitude reading from the Attitude Estimator block. In nominal mode, a routine was implemented to get a new reading from the GNSS receiver every $24\text{h}$ in order to update the orbital propagator with new values of position, velocity, and time. Once the MCEKF converges, which will be assumed to be true after $1.5$ orbits, a new routine will be started. This new routine will be responsible for verifying the error between MEKF and the QUEST attitude estimations every time the sun sensors provide data. This error should be kept under $5^\circ$ using a moving average with a window size of $60\text{s}$. If the average error is larger than the $5^\circ$ threshold, a flag is set to one and it should be communicated to the ORCASat's ground station.

Preliminary simulations showed that the sequence described successfully managed to automatically bring the ORCASat from an arbitrary attitude and angular rate to nominal mode.

# Appendix D

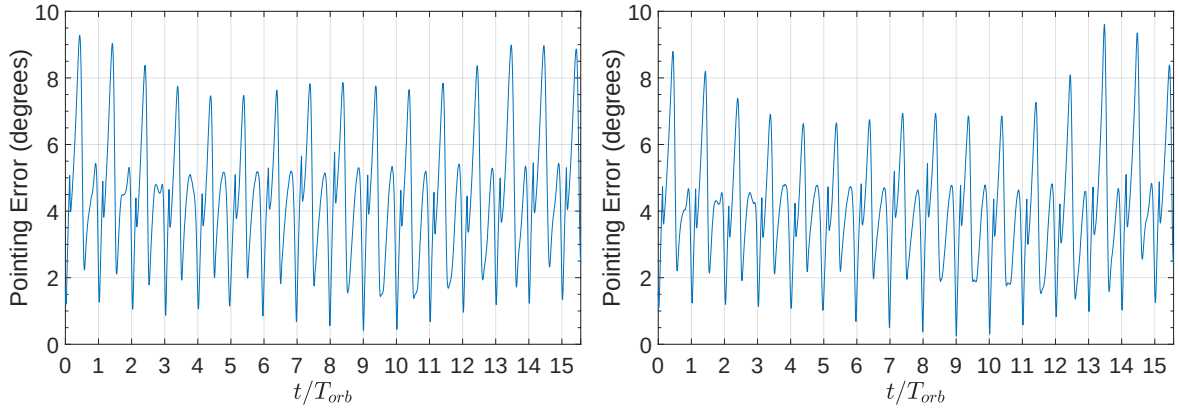
# Additional Figures




(a) SMC - $K = 0.015$ and $\lambda = 0.0007$.




(b) CGC - $K_Q = 0.09$.

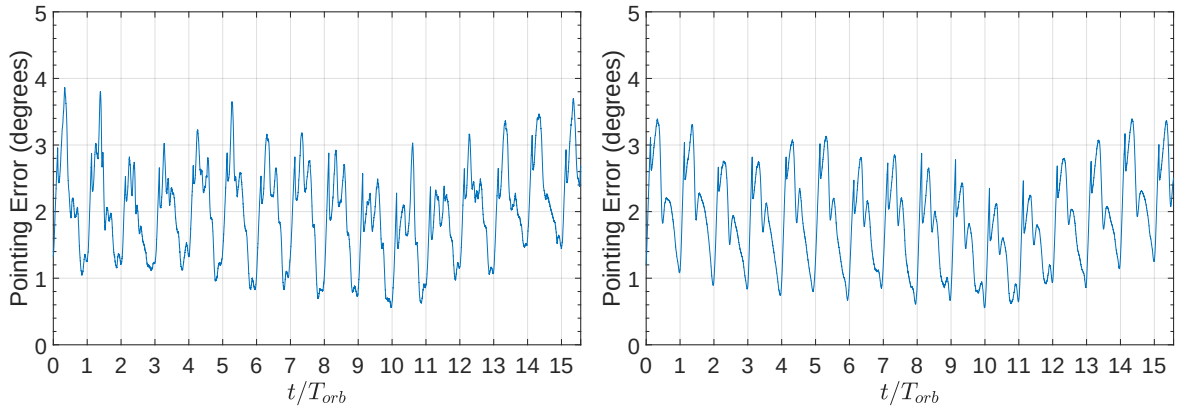



(c) FHC - $K_Q = 0.04$.



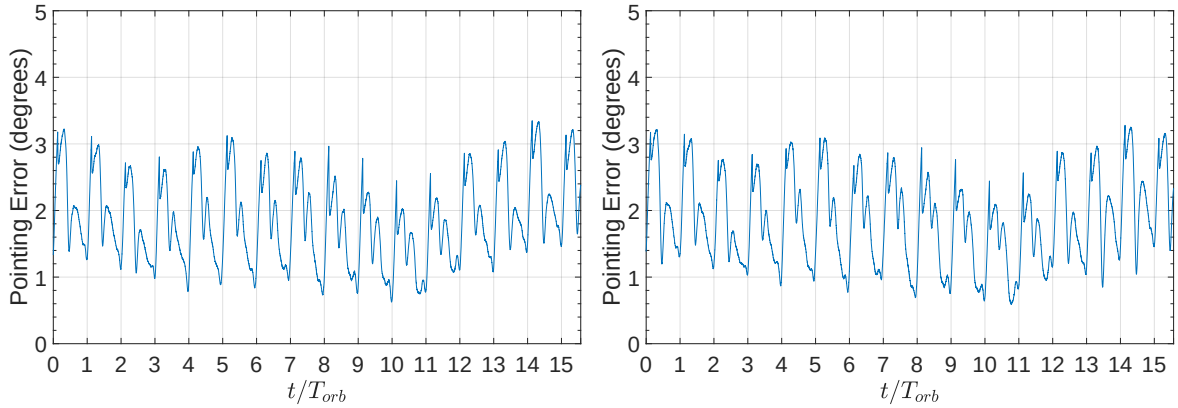

(d) IHC - $K_Q = 0.05$.

Figure D.1: Performance of the nadir-pointing controllers using the gain that provides the best efficiency.




(a) SMC - $K = 0.005$ and $\lambda = 0.02$.

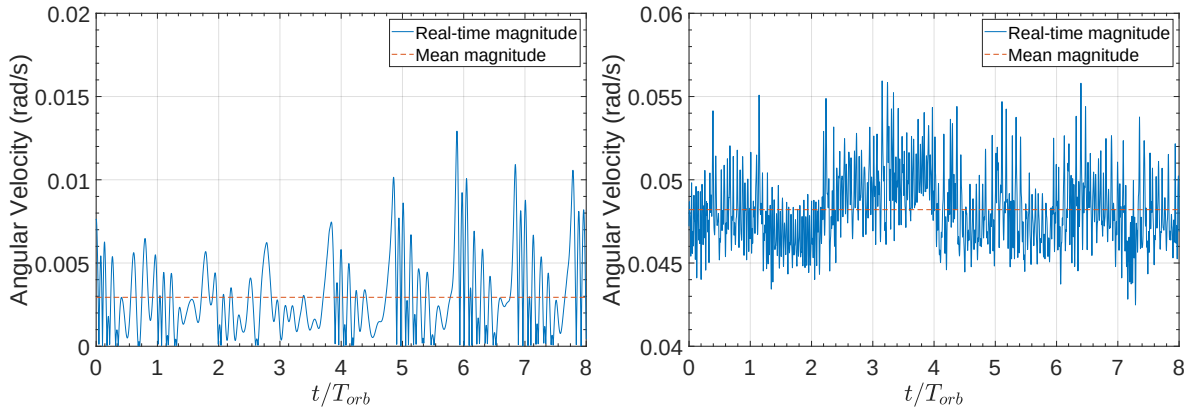
(b) CGC - $K_Q = 3$.




(c) FHC - $K_Q = 3$.

(d) IHC - $K_Q = 5$.

Figure D.2: Performance of the nadir-pointing controllers using the gain that provides the smallest maximum error.




(a) Angular velocity profile in cases 1 and 2.

(b) Angular velocity profile in cases 3 and 4.

Figure D.3: Angular velocity profiles used in the analysis of the transient behaviour of MEKF and MCEKF.

# Appendix E

# Matlab Scripts

## E.1 Constant Gain Controller

```
clear all                                   % Load the simulink simulation containing
load('campo_magnetico_200000s_10122019')    % the magnetic field
v = [6.02605329; -3.45486820; -3.26308364]*10^3;        % Velocity vector
r = [-4.1239936011; -2.9874334602; -4.4630619183]*10^6; % Position vector
mu = 3.98600442e14;                                     % Standard gravitational parameter
a = mu*norm(r)/(2*mu - norm(r)*norm(v)^2);              % Semi-major axis
T = 2*pi*a^(3/2)/sqrt(mu);                              % Orbital period
n_orbits = floor(86400/T);
step = 0.1;     % Simulink simulation step size
w0 = 2*pi/T;    % Orbit angular velocity
len = round(T/step);
%% Getting the magnetic field from the simulation data
B = zeros(len, 3, n_orbits);
for i=1:1:n_orbits
    B(:,:,i) = zeta_o.data(1+len*(i-1):len*i,:);
end
[lines_B,rows_B,dim3_B] = size(B);
%% Average of the magnetic field per orbit
B_avg = zeros(lines_B,rows_B);
for i=1:1:n_orbits
    B_avg(:,:) = B_avg(:,:) + B(:,:,i);
end
B_avg = B_avg/n_orbits;
%% Spacecraft Parameters
J = [0.003 0 0; 0 0.007 0; 0 0 0.008];  % Inertia tensor
J1 = J(1,1);                            % Prinicpal inertia moments
J2 = J(2,2);
J3 = J(3,3);
h0 = 0.003;         % Momentum wheel angular momentum about its the spin axis
kx = (J2-J3)/J1;
ky = (J3-J1)/J2;
kz = (J1-J2)/J3;
%% System matrix A
A_matrix = [0,0,w0*(1-kx)-h0/J1,-2*w0*(w0*kx+h0/J1),0,0;0,0,0,0,0,0;...
h0/J3-w0*(1+kz),0,0,0,0,2*w0*(w0*kz-h0/J3);0.5,0,0,0,0,0;0,0.5,0,0,0,0;0,0,0.5,0,0,0];
%% Control Matrix B for each time step
bx = B_avg(:,1);
by = B_avg(:,2);
bz = B_avg(:,3);
B_matrix = zeros(6,3,length(B_avg));
for i=1:1:length(B_avg)
    B_matrix(:,:,i)=[(J^(-1))*(1/sqrt(bx(i)*bx(i)+by(i)*by(i)+bz(i)*bz(i)))*...
    [-by(i)*by(i)-bz(i)*bz(i),bx(i)*by(i),bx(i)*bz(i);bx(i)*by(i),-bx(i)*bx(i)-...
    bz(i)*bz(i),by(i)*bz(i);bx(i)*bz(i),by(i)*bz(i),-bx(i)*bx(i)-by(i)*by(i)];zeros(3)];
end
%% ---> Constant Gain Controller (CGC) <---
B_cgc = zeros(6,3); % Control Matrix B for the constant gain controller
for i=1:1:len
    B_cgc = B_cgc + B_matrix(:,:,i);
end
B_cgc = B_cgc/len;
gain_Q = [0.01, 0.05, 0.1];     % Q matrix
diag_matrix = diag([10^3, 10^3, 10^3, 1, 1, 1]);
Q = zeros(6,6,length(gain_Q));  % Q = gain_Q*diag_matrix
for i =1:1:length(gain_Q)
    Q(:,:,i) = gain_Q(i)*diag_matrix;
end
n_iterations = length(gain_Q);  % For computing the solution for each Q matrix
P_Q = zeros(6,6,n_iterations);
```

```
K_Q = zeros(3,6,n_iterations);
for k=1:1:n_iterations  % Riccati matrix P and controller gain K -> matlab care function
    [P_Q(:,:,k) ,¬,K_Q(:,:,k)] = care(A_matrix, B_cgc, Q(:,:,k));
    gainQ = gain_Q(k);
    Kcgc = K_Q(:,:,k);
    Pcgc = P_Q(:,:,k);
    Qcgc = Q(:,:,k);
% save(['K_CGC_', num2str(gain_Q(k)) 'Qdif2_', num2str(k)],'Kcgc','gainQ','Pcgc','Qcgc')
end
```

## E.2 Infinite Horizon Controller

```
clear all                                   % Load the simulink simulation
load('campo_magnetico_200000s_10122019')    % containing the magnetic field
v = [6.02605329; -3.45486820; -3.26308364]*10^3;        % Velocity vector
r = [-4.1239936011; -2.9874334602; -4.4630619183]*10^6; % Position vector
mu = 3.98600442e14;                                     % Standard gravitational parameter
a = mu*norm(r)/(2*mu - norm(r)*norm(v)^2);              % Semi-major axis
T = 2*pi*a^(3/2)/sqrt(mu);                              % Orbital period
n_orbits = floor(120000/T);
step = 0.1;     % Simulink simulation step size
w0 = 2*pi/T;    % Orbit angular velocity
len = round(T/step);
t = 0:step:len*step-step;               % Time vector - used for interpolation
t_integration = len*step-step:-step:0;  % Integration time vector (Backwards integration)
%% Getting the magnetic field from the simulation data
B = zeros(len, 3, n_orbits);
for i=1:1:n_orbits
    B(:,:,i) = zeta_o.data(1+len*(i-1):len*i,:);
end
[lines_B,rows_B,dim3_B] = size(B);
%% Average of the magnetic field per orbit
B_avg = zeros(lines_B,rows_B);
for i=1:1:n_orbits
    B_avg(:,:) = B_avg(:,:) + B(:,:,i);
end
B_avg = B_avg/n_orbits;
%% Spacecraft Parameters
J = [0.003 0 0; 0 0.007 0; 0 0 0.008];  % Inertia tensor
J1 = J(1,1);                            % Prinicpal inertia moments
J2 = J(2,2);
J3 = J(3,3);
h0 = 0.003;     % Momentum wheel angular momentum about its the spin axis
kx = (J2-J3)/J1;
ky = (J3-J1)/J2;
kz = (J1-J2)/J3;
%% System matrix A
A_matrix = [0,0,w0*(1-kx)-h0/J1,-2*w0*(w0*kx+h0/J1),0,0;0,0,0,0,0,0;
h0/J3-w0*(1+kz),0,0,0,0,2*w0*(w0*kz-h0/J3);0.5,0,0,0,0,0;0,0.5,0,0,0,0;0,0,0.5,0,0,0];
%% Control Matrix B for each time step
bx = B_avg(:,1);
by = B_avg(:,2);
bz = B_avg(:,3);
B_matrix = zeros(6,3,len);
for i=1:1:len
    B_matrix(:,:,i) = [(J^(-1))*(1/sqrt(bx(i)*bx(i)+by(i)*by(i)+bz(i)*bz(i)))*...
    [-by(i)*by(i)-bz(i)*bz(i),bx(i)*by(i),bx(i)*bz(i);bx(i)*by(i),-bx(i)*bx(i)-...
    bz(i)*bz(i), by(i)*bz(i);bx(i)*bz(i),by(i)*bz(i),-bx(i)*bx(i)-by(i)*by(i)];zeros(3)];
end
%% ---> Infinite Horizon Controller (IHC) <---
gain = [0.01, 0.05, 0.1];   % Q matrix
diag_matrix = diag([10^3, 10^3, 10^3, 1, 1, 1]);
Q = zeros(6,6,length(gain));
for i =1:1:length(gain)     % Q = gain*diag_matrix
    Q(:,:,i) = gain(i)*diag_matrix;
end
B_matrix(:,:,end+1) = B_matrix(:,:,1);  % Periodic B_matrix - last value = first value
t_integration = [t_integration(1) + step, t_integration];
t(end+1) = t(end)+ step;
len = length(B_matrix);
P_IHC_new = zeros(6,6,len,length(gain));
K_IHC_new = zeros(3,6,len,length(gain));
B_matrix_ode = zeros(len,18);   % Converting the B_matrix to a vector (required in ode45)
for i=1:1:6
    for j=1:1:3
        B_matrix_ode(:,(i-1)*3+j) = B_matrix(i,j,:);
    end
end
n_cicles = 15;                              % Maximum number of iterations
dif = zeros(len,36,n_cicles, length(gain)); % Stores the difference between iterations
relative_error = zeros(len,36,n_cicles, length(gain));
for kk=1:1:length(gain) % Computed the solution for every gain defined in line 49
    Pf = Q(:,:,kk);
    Qaux = Q(:,:,kk);
    gainQ = gain(kk)
    len = length(B_matrix);
    % The value of Q entering in the ode45 function is represented by Pf - First Iteration
```

```
    [t_ode, Pode] = ode45(@(t_ode,P_ode) myode(t,P_ode,A_matrix,...
        B_matrix_ode,Qaux,t_ode), t_integration, Pf);   % (Backwards integration)
    AuxPode = Pode; % Change elements order - The first element now corresponds to t=0;
                % decreases the interpolation time in the myode2 function by a factor ...
                    of 5
    for i=1:1:length(Pode)
        Pode(i,:) = AuxPode(end-(i-1),:);
    end
    AUX(:,:,1,kk) = AuxPode;    % AUX keeps the inverse order of Pode of every iteration
    for i=1:1:n_cicles          % Next iterations
        Pf = reshape(Pode(1,:),size(eye(6)));   % The first index corresponds
        % to t=0 - Periodic condition, final value is equal to the first
        [t_ode2, Pode2] = ode45(@(t_ode2,P_ode2) myode2(t,P_ode2,...
            A_matrix,B_matrix_ode,Qaux,t_ode2,Pode), t_integration, Pf);
        Pode = Pode2;
        AuxPode = Pode;
        for iii=1:1:length(Pode) % Change elements order - same as in line 80
            Pode(iii,:) = AuxPode(end-(iii-1),:);
        end
        AUX(:,:,i+1,kk) = Pode2;
        dif(:,:,i,kk) = AUX(:,:,i,kk) - AUX(:,:,i+1,kk);
        m_AUX_i_plus_1 = max(abs(AUX(:,:,i+1,kk))); % Used to compute the relative_error
        for k=1:1:length(dif(1,:,i,kk))
           relative_error(:,k,i,kk) = abs(dif(:,k,i,kk))/m_AUX_i_plus_1(k);
        end
        m = max(max(relative_error(:,:,i,kk)));
%         AuxReshape = reshape(Pode(1,:), size(eye(6)));
        if m<0.01 % 1 per cent
            disp(['Threshold reached',' - Iteration ',num2str(i),' m=', num2str(m)])
            break
        end
        disp(['Threshold not reached',' - Iteration ', num2str(i), ' m=', num2str(m)])
    end
    for i=1:1:len   % convert Pode to a square matrix
        P_IHC_new(:,:,i,kk) = reshape(Pode(i,:), size(eye(6)));
    end
    for i=1:1:len   % compute the controller gain matrix
        K_IHC_new(:,:,i,kk) = (B_matrix(:,:,i)')*P_IHC_new(:,:,i,kk);
    end
    len = len-1;
    K_IHC_ext_new = K_IHC_new(:,:,1:end-1,kk); % Periodic extention of the Controller gain
    P_IHC_ext_new = P_IHC_new(:,:,1:end-1,kk);
    for i=1:1:2*n_orbits-1
        P_IHC_ext_new(:,:,i*len + 1:(i+1)*len) = P_IHC_new(:,:,1:end-1,kk);
        K_IHC_ext_new(:,:,i*len + 1:(i+1)*len) = K_IHC_new(:,:,1:end-1,kk);
    end
    t_ext = 0:step:(length(K_IHC_ext_new)*step-step);
    K_series_IHC_new = timeseries(K_IHC_ext_new, t_ext);
%     save(['K_series_IHC10_newMF_ode', num2str(kk)], 'K_series_IHC_new', 'Qaux', ...
    'diag_matrix', 'gainQ')
end
%% ode45 functions
function dPdt = myode(t,P,A,B_matrix_ode,Q,t_ode)
    B = interp1(t, B_matrix_ode, t_ode);
    B_aux = zeros(6,3);
    for i=1:1:6
        for j=1:1:3
            B_aux(i,j) = B((i-1)*3+j);
        end
    end
    P2 = reshape(P,size(A));
    dPdt2 = -P2*A - (A')*P2 + P2*B_aux*(B_aux')*P2 - Q;
    dPdt = reshape(dPdt2,size(P));
end

function dPdt = myode2(t,P,A,B_matrix_ode,Q,t_ode,Pode)
    B = interp1(t, B_matrix_ode, t_ode);
    B_aux = zeros(6,3);
    for i=1:1:6
        for j=1:1:3
            B_aux(i,j) = B((i-1)*3+j);
        end
    end
    P2 = reshape(P,size(A));
    Pi = interp1(t, Pode, t_ode);
    Pi = reshape(Pi,size(A));
    K = (B_aux')*Pi;
    A = A-B_aux*K;
    dPdt2 = -P2*A - (A')*P2 - (K')*K - Q;
    dPdt = reshape(dPdt2,size(P));
end
```

## E.3 Finite Horizon Controller

```
clear all                                   % Load the simulink file
load('campo_magnetico_200000s_10122019')    % containing the magnetic field
v = [6.02605329; -3.45486820; -3.26308364]*10^3;        % Velocity vector
```

```
r = [-4.1239936011; -2.9874334602; -4.4630619183]*10^6; % Position vector
mu = 3.98600442e14;                                     % Standard gravitational parameter
a = mu*norm(r)/(2*mu - norm(r)*norm(v)^2);              % Semi-major axis
T = 2*pi*a^(3/2)/sqrt(mu);                              % Orbital period
step = 0.1;     % Simulink simulation step size
w0 = 2*pi/T;    % Orbit angular velocity
len = round(T/step);
n_orbits = floor(120000/T);
%% Getting the magnetic field from the simulation data
B = zeros(len, 3, n_orbits);
for i=1:1:n_orbits
    B(:,:,i) = zeta_o.data(1+len*(i-1):len*i,:);
end
B(1,:,end+1) = zeta_o.data(1+len*(n_orbits+1 - 1),:);
[lines_B,rows_B,dim3_B] = size(B);
%% Spacecraft Parameters
J = [0.003 0 0; 0 0.007 0; 0 0 0.008];  % Inertia tensor
J1 = J(1,1);                            % Prinicpal inertia moments
J2 = J(2,2);
J3 = J(3,3);
h0 = 0.003;     % Momentum wheel angular momentum about its the spin axis
kx = (J2-J3)/J1;
ky = (J3-J1)/J2;
kz = (J1-J2)/J3;
%% System matrix A
A_matrix = [0,0,w0*(1-kx)-h0/J1,-2*w0*(w0*kx+h0/J1),0,0;0,0,0,0,0,0;
    h0/J3-w0*(1+kz),0,0,0,0,2*w0*(w0*kz-h0/J3);0.5,0,0,0,0,0;0,0.5,0,0,0,0;0,0,0.5,0,0,0];
%% ---> Finite Horizon Controller (FHC) <---
gain_Q = [150, 200, 250];          % Q matrix
diag_matrix = diag([10^3, 10^3, 10^3, 1, 1, 1]);
for i =1:1:length(gain_Q)
    Q(:,:,i) = gain_Q(i)*diag_matrix;
end
gain_Pf = [2, 2.5, 3];              % Gain of Pf
K_finite = zeros(3,6,len*n_orbits);
P_finite_ode = zeros(6,6,len*n_orbits);
t_integration = len*step:-step:0;   % Integration time vector (Backwards integration)
t = 0:step:len*step;                % Time vector - used for interpolation
for kkk = 1:1:length(Q)
    Q_aux = Q(:,:,kkk);
    for kk=1:1:length(gain_Pf)
        for k=1:1:n_orbits
            bx = [B(:,1,k); B(1,1,k+1)];
            by = [B(:,2,k); B(1,2,k+1)];
            bz = [B(:,3,k); B(1,3,k+1)];
            B_matrix = zeros(6,3,len+1);    % Control matrix of the system
            for i=1:1:len+1
            B_matrix(:,:,i) = [(J^(-1))*(1/sqrt(bx(i)*bx(i)+by(i)*by(i)+bz(i)*bz(i)))*...
            [-by(i)*by(i)-bz(i)*bz(i),bx(i)*by(i),bx(i)*bz(i);bx(i)*by(i),-bx(i)*bx(i)-...
    bz(i)*bz(i),by(i)*bz(i);bx(i)*bz(i),by(i)*bz(i),-bx(i)*bx(i)-by(i)*by(i)];zeros(3)];
            end
            [Pf,~,~] = care(A_matrix, B_matrix(:,:,end),Q(:,:,kkk));    % Final condition
            Pf = gain_Pf(kk)*Pf;                                        % Final condition
            B_matrix_ode = zeros((len+1),18);
            for i=1:1:6         % Converting the B_matrix to a vector (required in ode45)
                for j=1:1:3
                    B_matrix_ode(:,(i-1)*3+j) = B_matrix(i,j,:);
                end
            end
            [t_ode, Pode] = ode45(@(t_ode,P_ode) myode(t,P_ode,A_matrix,...
                B_matrix_ode,Q_aux,t_ode), t_integration, Pf);  % (Backwards integration)
            auxODE = Pode;
            Pode = zeros(6,6,len+1);    % Convert Pode to a square matrix
            for i=1:1:len+1
                Pode(:,:,i) = reshape(auxODE(len+1 - (i-1),:),size(eye(6)));
            end
            for i=1:1:len               % Global P matrix - all orbits considered
                P_finite_ode(:,:,i+(k-1)*len) = Pode(:,:,i);
            end
            for i=1:1:len               % Global gain matrix
                K_finite(:,:,i+(k-1)*len) = ((B_matrix(:,:,i))')*Pode(:,:,i);
            end
            Pf_aux(:,:,k) = Pf;
        end
        K_finite_series = timeseries(K_finite, 0:step:(length(K_finite)*step-step));
        gainPf = gain_Pf(kk)
        gainQ = gain_Q(kkk)
        save(['K_series_FHC', ' Pf', num2str(kk), ' Q', num2str(kkk)], ...
            'K_finite_series', 'gainPf', 'Q_aux', 'Pf_aux', 'gainQ');
    end
end
%% ode45 functions
function dPdt = myode(t,P,A,B_matrix_ode,Q,t_ode)
B = interp1(t, B_matrix_ode, t_ode);
B_aux = zeros(6,3);
for i=1:1:6
    for j=1:1:3
        B_aux(i,j) = B((i-1)*3+j);
    end
end
P2 = reshape(P,size(A));
dPdt2 = -P2*A - (A')*P2 + P2*B_aux*(B_aux')*P2 - Q;
dPdt = reshape(dPdt2,size(P));
end
```