



**IMPLEMENTATION OF ATTITUDE DETERMINATION
TECHNIQUES FOR A SMALL SATELLITE
THREE-AXIS SIMULATOR**

FERNANDO CARDOSO GUIMARÃES

**DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**IMPLEMENTATION OF ATTITUDE DETERMINATION
TECHNIQUES FOR A SMALL SATELLITE
THREE-AXIS SIMULATOR**

FERNANDO CARDOSO GUIMARÃES

Orientador: PROF. DR. RENATO ALVES BORGES, ENE/UNB

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA

**PUBLICAÇÃO PPGENE.DM - 703/2018
BRASÍLIA-DF, 30 DE JULHO DE 2018.**

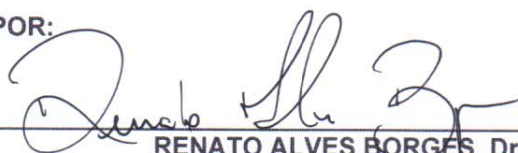
UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

IMPLEMENTATION OF ATTITUDE DETERMINATION TECHNIQUES
FOR A SMALL SATELLITE THREE-AXIS SIMULATOR

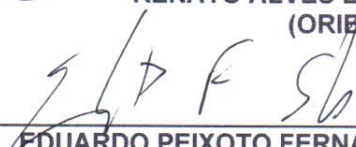
FERNANDO CARDOSO GUIMARÃES

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE.

APROVADA POR:



RENATO ALVES BORGES, Dr., ENE/UNB
(ORIENTADOR)



EDUARDO PEIXOTO FERNANDES DA SILVA, Dr., ENE/UNB
(EXAMINADOR INTERNO VINCULADO AO PROGRAMA)



THYRSO VILLELA NETO, Dr., INPE/CGEE
(EXAMINADOR EXTERNO)

Brasília, 30 de julho de 2018.

FICHA CATALOGRÁFICA

GUIMARÃES, FERNANDO CARDOSO

IMPLEMENTATION OF ATTITUDE DETERMINATION TECHNIQUES FOR A SMALL SATELLITE THREE-AXIS SIMULATOR [Distrito Federal] 2018

xiii, 111p., 201x297 mm

(ENE/FT/UnB, Mestre, Engenharia Elétrica, 2018)

Dissertação de Mestrado - Universidade de Brasília

Faculdade de Tecnologia - Departamento de Engenharia Elétrica

- | | |
|------------------------------------|-----------------------------------|
| 1. Air-bearing satellite simulator | 2. Kalman Filtering |
| 3. Cubesats | 4. Unscented Quaternion Estimator |
| 5. Attitude determination | 6. Onboard computer |
| I. ENE/FT/UnB | II. Título (série) |

REFERÊNCIA BIBLIOGRÁFICA

GUIMARÃES, F. C. (2018) IMPLEMENTATION OF ATTITUDE DETERMINATION TECHNIQUES FOR A SMALL SATELLITE THREE-AXIS SIMULATOR. Dissertação de Mestrado em Engenharia Elétrica, Publicação PPGENE.DM-703/2018, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 111p.

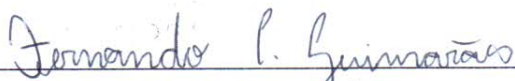
CESSÃO DE DIREITOS

AUTOR: FERNANDO CARDOSO GUIMARÃES

TÍTULO: IMPLEMENTATION OF ATTITUDE DETERMINATION TECHNIQUES FOR A SMALL SATELLITE THREE-AXIS SIMULATOR.

GRAU: Mestre ANO: 2018

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de Mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor se reserva a outros direitos de publicação e nenhuma parte desta dissertação de Mestrado pode ser reproduzida sem a autorização por escrito do autor.



FERNANDO CARDOSO GUIMARÃES

EQN 406/407 BLOCO A ASA NORTE BRASÍLIA-DF.

Dedico este trabalho à minha amada família.

Agradecimentos

Agradeço primeiro a Jesus Cristo pela oportunidade do estudo e do trabalho dignificantes. Agradeço em seguida ao Espiritismo, em cujo seio fui criado. Desconheço outra doutrina igualmente capaz de unir fé e razão, Ciência e Religião. *Fé inabalável só o é a que pode encarar frente a frente a razão em todas as épocas da humanidade*, dizia Allan Kardec há mais de cento e cinquenta anos. Nos momentos de dificuldade, de alegria ou de dúvida, essa doutrina, ancorada nos ensinamentos de Jesus, constitui o esteio que me sustenta e norteia os passos.

Agradeço à minha amada família, de quem sempre tive o apoio e amor incondicionais em todas as circunstâncias da minha vida. Aos meus pais, Joel e Suely, presto meus agradecimentos e homenagens. Às minhas irmãs, Bárbara e Stephanie, também agradeço pelo carinho e amizade. Em relação a essa dissertação de Mestrado, também devo agradecer-las pelo auxílio com os softwares CAD para a criação das figuras. Agradeço também ao meu cunhado Willian pelo importante apoio que ele constitui para mim e minhas irmãs nessa cidade que nos recebe já há algum tempo.

Agradeço ao meu orientador Renato Alves Borges pela orientação, dedicação e pelo entusiasmo com que nos compele às atividades de pesquisa. Agradeço também em especial ao professor Simone Battistini e à professora Chantal Cappelletti pelo apoio imprescindível que nos dão, especialmente tendo em vista que muitos de nós não possuem formação na área Aeroespacial, grupo no qual me incluo. Agradeço ainda aos professores do Departamento de Engenharia Elétrica, especialmente a Geovany Araújo Borges, João Yoshiyuki Ishihara, Adolfo Bauchspiess e Francisco Assis de Oliveira Nascimento, pelo exemplo e inspiração.

Devo também agradecer especialmente aos meus colegas do LAICA. O ambiente experimental com o qual nos envolvemos exige um enorme esforço coletivo sem o qual nenhum de nós seria capaz de produzir resultados satisfatórios. Assim, agradeço nominalmente aos colegas de Mestrado Rodrigo Cardoso da Silva e João Victor Lopes de Loiola, e aos colegas da Engenharia Aeroespacial Igor Seiiti Kinoshita e Lucas Silva. Também devo agradecer aos demais colegas do LARA pelas discussões profundas, e outras nem tanto, mas que muitas vezes são muito importantes para descontrair e relaxar.

Por fim, agradeço aos competentes funcionários do ENE e da Universidade de Brasília, bem como ao apoio financeiro da CAPES, sem o qual esse Mestrado nunca poderia realizar-se.

Abstract

The design and launch of pico- and nanosatellites – which in this work are generically called small satellites – have been received a great impulse over the last few decades. This new trend is now possible due to the advances in microelectronics and the reduction of the micro-electromechanical systems prices. The CubeSat standard introduction in the mid 1990s decisively contributed to these breakthrough changes in the field. Aiming to design and test technologies to be applied to small satellites, the *Laboratório de Inovação e Ciências Aeroespaciais* at *Universidade de Brasília* is developing a small satellite three-axis simulator composed by an air-bearing based table and a Helmholtz cage, besides other components. The attitude determination is a problem of a major importance for a spacecraft. In this work an attitude determination method based on computer vision was implemented and tested at laboratory. Using the Euler angles to evaluate its performance, this system produced results with standard deviations smaller than 0.5° for all the three angles. Other methods implemented were the TRIAD and USQUE algorithms developed in C language. The algorithms were executed at 25 MHz in a MSP430 of the CubeSat onboard computer ABACUS, being compared to their execution in other platforms. The execution time for TRIAD, corresponding to a single sampling interval, was of about 1.7ms. For USQUE the execution time was of about 64.5ms. The TRIAD would be very suitable for a spacecraft during the detumbling process, while the USQUE would be more appropriate afterwards, when the satellite is stabilized. The present masters dissertation have made great contributions to increase the laboratory capabilities. Furthermore, the complete and concise theoretical chapters of this work concerning several attitude representations and related topics can be useful as a reference for new students.

Resumo

O projeto e lançamento de pico e nanossatélites – os quais são chamados genericamente de pequenos satélites nesse trabalho – tem sido bastante impulsionados nas últimas décadas. Essa nova tendência é atualmente possível devido aos avanços da microeletrônica e à redução no preço dos sistemas microeletromecânicos. A introdução do padrão CubeSat em meados dos anos 1990 contribuiu decisivamente para as mudanças revolucionárias ocorridas nessa área. Com o objetivo de desenvolver e testar tecnologias a serem aplicadas a pico e nanosatellites, o Laboratório de Inovação e Ciências Aeroespaciais da Universidade de Brasília está desenvolvendo um simulador de três eixos para pequenos satélites. O simulador é composto principalmente por uma mesa com rolamento a ar e uma gaiola de Helmholtz, além de outros componentes. A determinação de atitude é um problema de grande importância para um satélite. Neste trabalho um sistema de determinação de atitude baseado em visão computacional foi implementado e testado em laboratório. Usando ângulos de Euler para avaliar seu desempenho, esse sistema produziu resultados com desvios padrão inferiores a 0.5° para todos os três ângulos. Outros métodos implementados foram os algoritmos TRIAD e USQUE desenvolvidos na linguagem C. Esses algoritmos foram executados a 25 MHz em um MSP430 do computador de bordo para CubeSats ABACUS, sendo comparados com sua execução em outras plataformas. O tempo de execução do TRIAD, correspondente a um intervalo de amostragem, foi de aproximadamente 1,7ms. Para o USQUE o tempo de execução foi de aproximadamente 64,5ms. O algoritmo TRIAD seria bastante adequado para um satélite durante a fase de estabilização, enquanto o USQUE seria mais apropriado para a fase subsequente. Esta dissertação de Mestrado deu grandes contribuições para a melhoria dos recursos do laboratório. Ademais, os capítulos teóricos que tratam de representações de atitude e tópicos correlatos, escritos de forma completa e concisa, poderão ser úteis como uma referência para novos estudantes.

LIST OF CONTENTS

ABSTRACT	IV
RESUMO	V
1 INTRODUCTION	1
1.1 BACKGROUND AND MOTIVATION	1
1.2 PROBLEM STATEMENT	4
1.3 GOALS AND CONTRIBUTIONS	7
1.4 PRESENTATION OF THE MANUSCRIPT	9
2 ATTITUDE REPRESENTATION	11
2.1 REFERENCE FRAMES	11
2.1.1 BODY-FIXED FRAME.....	12
2.1.2 NED.....	12
2.1.3 ORBIT REFERENCE FRAME	13
2.1.4 ECEF	14
2.1.5 ECI.....	14
2.1.6 ATTITUDE DEFINITION.....	15
2.2 ROTATIONS IN THREE-DIMENSIONAL SPACE	16
2.2.1 NOTATION AND CONVENTIONS	16
2.2.2 ROTATIONS IN \mathbb{R}^2	18
2.2.3 ROTATION MATRICES.....	21
2.2.4 FIXED AXIS OF ROTATION	24
2.2.5 ROTATION ANGLE ABOUT THE FIXED AXIS OF ROTATION	27
2.3 ATTITUDE REPRESENTATIONS.....	27
2.3.1 DIRECTION-COSINE MATRIX	28
2.3.2 EULER ANGLES	30
2.3.3 EULER SYMMETRIC PARAMETERS AND QUATERNIONS.....	32
2.3.4 RODRIGUES PARAMETERS AND GIBBS VECTOR	37
2.3.5 ROTATION VECTOR	38
2.3.6 ATTITUDE ERROR	38
2.3.7 ATTITUDE KINEMATICS	39

3	ATTITUDE DETERMINATION METHODS	41
3.1	DETERMINISTIC METHODS FOR ATTITUDE DETERMINATION	41
3.1.1	TRIAD ALGORITHM	41
3.2	STOCHASTIC METHODS FOR ATTITUDE ESTIMATION	43
3.2.1	RANDOM VARIABLES AND STOCHASTIC PROCESSES	43
3.2.2	KALMAN FILTER	46
3.2.3	UNSCENTED TRANSFORMATION	48
3.2.4	UNSCENTED QUATERNION ESTIMATOR	49
4	SMALL SATELLITE THREE-AXIS SIMULATOR FACILITY	59
4.1	FACILITY OVERVIEW	59
4.2	THREE-AXIS AIR-BEARING TABLE.....	61
4.3	EARTH MAGNETIC FIELD SIMULATOR	63
4.4	ATTITUDE DETERMINATION WITH COMPUTER VISION.....	66
4.5	ABACUS ONBOARD COMPUTER	71
5	EXPERIMENTAL PROCEDURES.....	74
5.1	LOCAL REFERENCE-SYSTEM	74
5.2	TESTING OF THE ATTITUDE DETERMINATION SYSTEM BASED ON COMPUTER VISION.....	76
5.3	ALGORITHMS IMPLEMENTATION IN C LANGUAGE	79
5.4	ALGORITHMS TESTS AND RESULTS	82
6	CONCLUSION	88
	REFERENCES.....	90

LIST OF FIGURES

1.1	Serpens 3U CubeSat. (Image from [UnB 2015])	3
1.2	General view of the LAICA facility main components.....	4
2.1	Body-system of a 3U CubeSat.	12
2.2	NED system representation relative to the Earth.....	13
2.3	Orbit reference frame representation relative to the Earth.....	13
2.4	Representation of the Earth-Centered-Earth-Fixed system.....	14
2.5	Representation of the Earth-Centered-Inertial system.....	15
2.6	Reference frames $\{i_1, j_1\}$ and $\{i_2, j_2\}$ related by the angle α	19
2.7	Reference frames $\{i_1, j_1\}$, $\{i_2, j_2\}$ and $\{i_3, j_3\}$ related by the angles α and β	20
2.8	Rotation of $\{i_1, j_1, k_1\}$ about k_1 trough an angle ψ	22
2.9	Angles between a body-system unit vector i_b and the reference-system vectors.	28
3.1	High-level view of the USQUE algorithm. The black arrows denote conversions between the attitude representations.....	50
4.1	LAICA facility overview.....	59
4.2	Double-view picture of the air-bearing table.....	61
4.3	Spherical base of the testbed air-bearing (left side) and the air compressor (right side).....	62
4.4	Rotating hemisphere of the testbed air-bearing (left side) and the two parts together (right side).	62
4.5	The gravitational torque applied to the testbed and its dependency on the CM and CR positions.....	63
4.6	Movable mass units of the air-bearing table.....	63
4.7	Helmholtz cage (center) and the three power supplies (left).	64
4.8	Model of one Helmholtz cage pair of coils. (Retrieved from [Loiola et al. 2018]).....	65
4.9	Values of the Y -component of B along the Y axis. (Retrieved from [Loiola et al. 2018]).....	66
4.10	Pinhole camera model representation. (Adapted from [Bradski and Kaehler 2008]).....	67
4.11	Modified version of the pinhole camera model representation. (Adapted from [Bradski and Kaehler 2008])	67

4.12	Chessboard pattern of fiducial markers used during the calibration procedure..	69
4.13	ArUco markers of 4 bits (left) and 6 (bits).	70
4.14	Camera attitude determination using ArUco markers of 6 bits.	71
4.15	Camera attitude relative to a board hold in different orientations.	71
4.16	Two views of LAICA ABACUS OBC 2017 mounted into a case provided by Gauss Srl.	72
4.17	Two views of LAICA ABACUS OBC 2017.	72
4.18	Body-system vectors of the ABACUS sensors.	73
4.19	JTAG programmer MSP-FET.	73
5.1	Apparatus adopted to physically establish the reference-system at the laboratory.	74
5.2	Plastic piece, the laser pointer and the protractor used to define the third orientation angle.	75
5.3	Laser pointing to the wall target.	75
5.4	ADCV system determining the attitude of the reference-system apparatus.	76
5.5	Protractor yaw angles <i>versus</i> ADCV yaw angles.	77
5.6	Protractor roll angles <i>versus</i> ADCV roll angles.	77
5.7	Protractor pitch angles <i>versus</i> ADCV pitch angles.	78
5.8	Protractor yaw angles <i>versus</i> ADCV yaw angles in steps of 1°.	78
5.9	Schematic representation of the TRIAD and USQUE implementation and the two libraries developed.	79
5.10	Schematic representation of the space occupied by a symmetric matrix in memory in the Linear Algebra library.	80
5.11	Logical representation of a symmetric matrix in the Linear Algebra library.	80
5.12	Communication architecture used in the experiments with ABACUS.	82
5.13	ABACUS, the CP2102 converter and the Raspberry mounted in a single structure.	83
5.14	ABACUS-Raspberry structure with an ArUco marker in detail (left) and mounted in the air-bearing table (right).	83
5.15	Measurements obtained by ABACUS gyro.	84
5.16	Measurements obtained by ABACUS accelerometer.	85
5.17	Measurements obtained by ABACUS magnetometer.	86
5.18	Values of the table DCM main diagonal.	86
5.19	q_1 component of the table quaternion of rotation obtained by USQUE.	86
5.20	Elements of the covariance matrix P correspondent to the table quaternion of rotation.	87
5.21	Gyro bias values estimated by USQUE.	87
5.22	Elements of the covariance matrix P correspondent to the gyro bias.	87

LIST OF TABLES

4.1	Helmholtz cage specifications.	66
4.2	Characteristics of the web camera utilized with the ADCV system.	71
4.3	Characteristics of ABACUS IMU.	72
4.4	MSP430 related features of ABACUS.	73
5.1	Summary of the ADCV tests results.	78
5.2	Algorithms performances and characteristics.	85

ACRONYMS

ADCS	Attitude determination and control systems
ADCV	Attitude determination system based on computer vision
CCD	Charge-coupled-device
CCS	Code Composer Studio
CDF	Conditional density function
CM	Center of mass
COTS	Commercial-off-the-shelf
CR	Center of rotation
DCM	Direct-cosine matrix
ECEF	Earth-Centered-Earth-Fixed
ECI	Earth-Centered-Inertial
EKF	Extended Kalman Filter
EMFS	Earth Magnetic Field Simulator
GRP	Generalized Rodrigues parameters
IMU	Inertial measurement unit
ISS	International Space Station
KF	Kalman filtering
LAICA	Laboratório de Inovação e Ciências Aeroespaciais
MCU	Microcontroller unit
MEMS	micro-electromechanical systems
MMU	Movable mass units

MRP	Modified Rodrigues parameters
NED	North-East-Down
OBC	Onboard computer
ORF	Orbit reference frame
PC	Personal computer
PDF	Probability density function
RPS	Rotations per second
UKF	Unscented Kalman Filter
UnB	Universidade de Brasília
USQUE	Unscented Quaternion Estimator
UT	Unscented transformation
WMM	World Magnetic Model

LIST OF SYMBOLS

\mathcal{B}	Basis of the vector space \mathbb{R}^3
X, Y, Z	Cartesian coordinate axes
O	Origin point of a reference frame
x, y, z	Components of a vector representation
i, j, k	Orthonormal vectors of a reference frame
\mathcal{F}	Orthonormal basis of the vector space \mathbb{R}^3
R_1	Rotation matrix corresponding to a rotation about i
R_2	Rotation matrix corresponding to a rotation about j
R_3	Rotation matrix corresponding to a rotation about k
ϕ	Euler angle roll
θ	Euler angle pitch
ψ	Euler angle yaw
$\ \bullet\ $	Euclidean norm of a vector \bullet
A	Generic rotation matrix
φ	Rotation angle about the fixed axis of rotation
n_A	Unit vector parallel to the fixed axis of rotation
s_\bullet	$\sin(\bullet)$
c_\bullet	$\cos(\bullet)$
$\eta_1, \eta_2, \eta_3, \eta_4$	Euler-symmetric parameters

\mathbf{q}	Quaternion of rotation
\otimes	Quaternion multiplication operator
ρ_1, ρ_2, ρ_3	Rodrigues parameters
$\boldsymbol{\rho}$	Gibbs vector
$\boldsymbol{\varrho}$	Vector associated to the Modified Rodrigues Parameters
\boldsymbol{p}	Vector associated to the Generalized Rodrigues Parameters
$\boldsymbol{\varphi}$	Rotation vector
$\boldsymbol{\omega}$	Angular velocity vector
$\boldsymbol{\beta}$	Bias vector of a rate gyro
p_X	Probability density function
σ	Standard deviation
σ^2	Variance
$\overline{\mathbf{X}}$	Mean of a random vector \mathbf{X}
P	Covariance matrix
$:=$	Estimation of
K	Kalman gain
Q	Covariance matrix of the process noise
R	Covariance matrix of the measurement noise
$\boldsymbol{\mathcal{X}}$	Sigma point associated to the state-space vector
\mathbf{X}_k^-	Predicted value of \mathbf{X} at a instant k
\mathbf{X}_k^+	Corrected value of \mathbf{X} at a instant k
\mathbf{B}	Magnetic field vector
γ	Constructive parameter of the Helmholtz cage

Chapter 1

Introduction

Since the launch of Sputnik-1 in 1957, artificial satellites have become indispensable instruments to the modern world. Communications, navigation and geographic localization, climate research and weather forecast, military and defense programs, imagery and mapping of the Earth, astronomical observations and other scientific studies, are some of the applications made possible by satellites.

Nowadays, when anyone has in their hands most of the services satellites enable, their importance continues to grow as the number of institutions from several countries pursuing the technology to develop and launch such spacecraft. In this context, the *Laboratório de Inovação e Ciências Aeroespaciais* (LAICA) at Universidade de Brasília (UnB) has joined Brazil's efforts to promote and dominate this field technologies.

The present masters dissertation concerns some of the researches conducted at LAICA using its small satellite three-axis simulator facility. The next sections will give further details about the new trend in the small satellite category, the specific problems this manuscript addresses and its contributions.

1.1 Background and motivation

The advances in microelectronics over the last years made possible the development of very small satellites with high technology and lower costs when compared to conventional satellites. The interest for this satellite category received a considerable impulse in the second half of the 1990s when Stanford University and California Polytechnic State University introduced the CubeSat standard [Heidt et al. 2000]. A CubeSat is a cubic shaped satellite of 10 cm side which corresponds to a 1U (one unit) size definition. The CubeSat concept naturally leads to modular designs, thus the extension of the 1U size to 2U, 3U and even larger sizes are common, with each of the modules stacked together or tied in other configurations [NASA 2015].

The general convention for satellite classification is as follows:

- Conventional satellite: > 100 kg;
- Microsatellite: 10 - 100 kg;
- Nanosatellite: 1 - 10 kg;
- Picosatellite: 0.01 - 1 kg.

Other classes exist but for those there are disagreements about the masses ranges; only the above last three classes seem to be well-established. CubeSats normally fall into the pico- and nanosatellite categories. Therefore, although the term "small satellites" can be interpreted in a wide sense, in this manuscript it should be understood as pico- and nanosatellites, and also as the microsatellites which come closer to the latter.

The advances in microelectronics that brought those small satellites into reality include the diminution in electronic components size and the increased performance of microprocessors, in opposition to their decreased power consumption. In addition, the utilization of *commercial-off-the-shelf* (COTS) components and the price reduction of MEMS (micro-electromechanical systems) sensors turned the projects costs extremely small.

The benefits of COTS components utilization come with the price of the great vulnerability of the satellite systems. Conventional satellites, on the other hand, employ space qualified components able to resist against radiation, high energy particles and large temperature variations for long periods of time. Hence, small satellite missions are not expected to have a long duration. Indeed, the lifetime of pico- and nanosatellites are normally less than one year[Bouwmeester and Guo 2010].

Small satellites are also limited in their communication capabilities. This is mainly due to power constraints and antenna gain. Furthermore, ground station tracking is difficult because most of the small satellites are not capable to perform a high precision attitude control [Bouwmeester and Guo 2010]. The main reasons for the last statement is the technology employed, commonly a passive or active magnetic actuators which are fine for simple detumbling¹ but inadequate to precise control. Despite of the adopted technique, an effective closed-loop control is only possible if the attitude is accurately estimated, and this is a major issue for any spacecraft.

The aforementioned restrictions limit the small satellite applicability. However, the interest for this technology continues to grow as long as the limitations are well known and there is still room for improvements [Bouwmeester et al. 2008]. Therefore, several works concerning specific and general issues of small satellites projects have been published [Oliveira et al. 2013, Bouwmeester et al. 2008, Ovchinnikov et al. 2007] and a great number of missions have been planned and executed. Indeed, in the recent work [Villela et al. 2018] is reported that in 2012 the hundredth CubeSat was launched, and probably, the thousandth will be launched by 2018, a 10-fold increase in five years only.

¹Spacecraft detumbling refers to the process of reducing its angular velocity after the orbital insertion.

The purposes and motivation of the first CubeSats projects were mainly educational. Nowadays, these tiny satellite are employed in several scientific applications, specially astronomy and space weather. Also, the use of CubeSats for Earth remote sensing is currently increasing [Villela et al. 2018].

Among all the small satellite missions executed it is specially worth to mention the Brazilian Serpens mission. This was a result of a consortium between the Brazilian Space Agency (AEB) and a few universities under the UnB coordination [AEB 2018, UnB 2015]. The Serpens 1 nanosatellite developed was a 3U CubeSat launched in 2015. The spacecraft orbited the Earth during six months as it collected and transmitted environmental data until its disintegration in atmosphere [FAPESP 2016]. Figure 1.1 exhibits the Serpens CubeSat before its launch.



Figure 1.1: Serpens 3U CubeSat. (Image from [UnB 2015])

In [Santilli et al. 2018] an interesting possible application for CubeSats constellations was purposed partially based on the results of Serpens mission. Some of the professors and students involved with the project are affiliated to LAICA where the well-succeed Serpens mission continues to motivate the researches conducted. Also, a newer version of the onboard computer (OBC) used in Serpens 1 and available at LAICA was employed in the present work to implement some attitude determination algorithms.

Aiming to develop, test and implement attitude determination and control systems (ADCS) , besides other technologies, a three-axis small satellite simulator facility is under development at LAICA. The main components of the simulator are a three-axis air-bearing table and a Helmholtz cage. The former is a three-degree-of-freedom rotating platform which replicates the frictionless conditions experimented by orbiting spacecraft; the latter simulates the Earth magnetic field a satellite can use to determine and control its own attitude. Figure 1.2 shows a general view of the main components of LAICA facility. A more detailed description will be given in Chapter 4.

Facilities similar to LAICA's demand intense research themselves and are subject to several academic works [Silva et al. 2018, Chesi et al. 2013, Samuels 2014]. Nevertheless, the

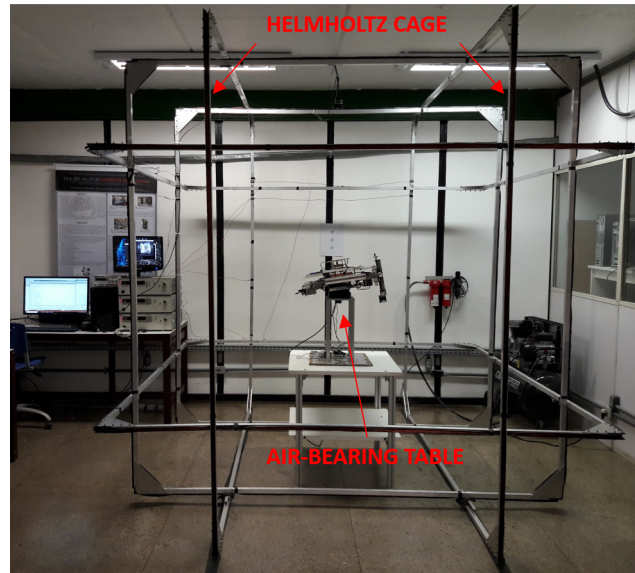


Figure 1.2: General view of the LAICA facility main components.

development of techniques and algorithms that can be applied in real life missions are the main goals to be attained. The present dissertation goals take place between those two lines, the facility itself and the implementation of practical satellite technologies, as will become clear in the rest of this chapter.

In order to conclude this section, an important remark is made: the capacity to design and build satellites, once restricted to big national space agencies and research institutes only, are now extendible to universities and even small companies such as startups. A reality now possible due to these breakthrough advances in the small satellite field.

1.2 Problem statement

In the last section was reported that most of the small satellites employ limited methods for attitude control. Additionally, since the closed-loop control system is suppose to decrease the error in the spacecraft attitude, it must be accurately determined, otherwise the actuation will be ineffective regardless of the control algorithm applied. Therefore, the attitude determination of a satellite is a problem of major importance.

The attitude of a body is its orientation in the three-dimensional space. Of course, this concept can be applied to several objects such as aircraft, terrestrial vehicles, robot moving parts and spacecraft. In Chapter 2, a mathematical definition for attitude will be given, for now, a qualitative one is sufficient to understand its relevance.

Is considered herein, as an example, a satellite at a 400 km high orbit² around Earth with an instrument which should be pointed to a specific location on Earth's surface. Simple

²This is approximately the same altitude of the International Space Station (ISS) from where the Serpens 1 nanosatellite was sent into orbit.

calculations show a pointing error of 2° (in any direction within a cone with an aperture angle of 4°) would imply in a circular region of about $400 \cdot (2 + 2) \cdot \pi/180 = 30$ km in diameter where the instrument could possibly point to. A deviation of this magnitude could pose significant difficulties to communications, imagery or other applications.

The sensors commonly used for attitude determination are magnetometers, accelerometers, rate gyros³, sun sensors and star sensors. Of course, some of them are more suitable or exclusive of certain applications. Accelerometers, magnetometers and gyros are commonly used in robotics, aircraft, smartphones and any other device meant to work on Earth's surface or atmosphere. Magnetometers and gyros are also employed in spacecraft as well as sun sensors and star sensors. The last two are more suitable for aerospace applications whilst accelerometers can not be easily employed in orbiting objects, considering they measure the (apparent) gravitational acceleration perceived by a body, which is practically zero for those spacecraft.

The measurements nature of each sensor can be used to classify them in two groups [B6 2007]:

- **Proprioceptive sensors**; which measure inner states of the body they are attached to, such as angular velocity and acceleration, without referring to the external world. Inertial sensors normally fall into this group as they measure quantities related to the inertial forces that appear to exist in the non-inertial frame of the sensor body.
- **Exteroceptive sensors**; which measure quantities related to references outside the sensor body, such as external fields (gravitational, magnetic) and light.

Some sensors can act as proprioceptive sensors or as exteroceptive depending on the application. Accelerometers, as an example, can be used as proprioceptive sensors since they measure their own accelerations without referring to any external reference. However, when employed in attitude determination, accelerometers are used to measure the external field of gravitational acceleration⁴, in which case they act as exteroceptive sensors.

Estimates based on proprioceptive sensors normally tend to exhibit large errors after some time. For instance, given an initial angular position of a body, the angular velocity measured by a gyro could be integrated in time, resulting in the body angular position. However, signals provided by real life sensors have random components and are commonly biased. An integration in time of the signal bias component may result in an unbounded error, turning the estimation inaccurate. Intuitively, these errors should be expected if proprioceptive sensor measurements are used to estimate quantities defined using external references.

³Rate gyros are gyroscopes which measures the time rate of angular changes. Hereinafter in this manuscript, they will be simply referred as gyros.

⁴The relationship between acceleration and gravitational field become patent when dealing with accelerometers. In fact, this connection led to Einstein's first insights concerning what would become his General Theory of Relativity.

Exteroceptive sensors measurements on the other hand, can be directly applied to estimate externally referenced quantities. Even though possessing a random behaviour and bias, because there is no need for time integration, the errors remain bounded. However, these sensors are dependent of external information sources, which can be untrustable or simply unavailable. Therefore, a satellite sun sensor is useless if the Sun is occluded by the Earth or the Moon; or a magnetometer measurements can be highly inaccurate if the sensor is exposed to spurious magnetic fields.

The advantages of each kind of sensor can be combined to produce more precise and accurate measurements. Kalman filtering (KF) approaches accomplish this task by dividing the estimation process in a prediction phase and a correction phase. In the first one, given the current variable estimate, the filter will use proprioceptive measurements to predict the variable value in a future instant. Considering a integration in time is often necessary with proprioceptive measurements, this phase requires a dynamic model formulation for the process under analysis. In the second phase, exteroceptive measurements are used to correct the predicted estimate and this often requires a change of scale or in the reference frame. The resulting estimated variable is a pondered value between the predicted estimation and the exteroceptive measurements. Qualitatively speaking, if exteroceptive measurements are corrupted by external perturbations, the proprioceptive measurements can attenuate these effects; at the same time, exteroceptive measurements compensate the proprioceptive predictions deviations as if the filter prediction part was reset by the correction part.

The methods applied for attitude determination can be classified as deterministic and optimal [Shuster 2004]. The first class takes a minimal set of measurements to obtain the attitude, which means the number of observations is the same as the unknown attitude parameters [Wertz 1978]. The second class take a number of measurements greater than the minimal set to minimize some cost function such that optimality is achieved in a statistical sense, as referred by G. M. Lerner in [Wertz 1978]. The deterministic methods do not take into account the random nature of measurements in their calculations, although a probabilistic analysis is always possible, assuming the obtained attitude parameters are function of random variables, (i.e., the measurements). However, in this analysis the random nature of the observations is considered *a posteriori* only, the deterministic algorithms do not take it into account in their theoretical models. Optimal methods in general model the sensors and processes noises and include the variables statistical parameters, such as mean and covariance, in their minimization procedures of the loss function.

The KF is an optimal method widely applied in several fields. Though linear in its original formulation [Aguirre 2015], nonlinear extensions of the KF were formulated such as the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF) . The EKF has been used for attitude determination for a long time [Markley 2003] and is the workhorse of real-time implementations for spacecraft applications [Crassidis et al. 2007]. The more recent UKF has been adapted for attitude estimation problems either [Crassidis 2003, Cheon and Kim 2007, Menegaz, H. M. T. 2016]. One of these proposed methods, the Un-

scented Quaternion Estimator (USQUE) , was implemented and tested in the present work at LAICA facility. The TRIAD algorithm and its variations is one of the most employed deterministic methods [Shuster 2004] and was also implemented at LAICA. Further details concerning the TRIAD algorithm and the UKF for attitude determination, with emphasis in USQUE, will be given in Chapters 3 and 5.

1.3 Goals and contributions

The present dissertation objectives consist in implementing attitude determination techniques to be applied at LAICA facility. The algorithms, software and hardware shall constitute complete attitude determination systems for the experiments to be performed at the laboratory. As explained in Section 1.1, the simulator platform itself demands intensive research and the techniques implemented aims at providing an accurate attitude determination method for the platform. Nevertheless, the main goals pursued at LAICA are the development and testing of small satellite technologies. Therefore, the techniques implemented in this work shall be at least partly suitable for practical satellites. The following observations will clarify these two complementary goals:

- As explained in Section 1.2, some of the sensors employed in attitude determination on Earth differ from those used in Space.
- Hence, methods suitable for the laboratory platform can not always be directly applied to satellites.
- In its current state, LAICA infrastructure still does not allow the test of some common satellite attitude determination systems.
- The TRIAD algorithm requires two vector observations to determine the attitude of a body. In spacecraft, magnetometers, sun sensors and star trackers normally are used to provide at least one of these vectors but currently the last two can not be tested at LAICA.
- On the other hand, using the Helmholtz cage, magnetometers and gyros, the USQUE algorithm can be implemented and tested.

Besides the TRIAD and USQUE algorithms, an attitude determination system based on computer vision (ADCV) was implemented at LAICA facility. This system employs a web camera external to ABACUS onboard computer and connected to a computer at the facility. The ADCV is useful for algorithms evaluation, providing independent means of comparison to the other methods.

Even though TRIAD algorithm implemented at LAICA still can not determine attitude using sun sensors or star trackers, accelerometers can replace them for testing purposes.

While serving to effectively determine the air-bearing platform attitude, the TRIAD can be almost completely evaluated – the difference would lie in the software layer responsible for collecting the sensors data.

The USQUE algorithm can be implemented and tested at LAICA. This method can be directly applied to satellites if their OBC have enough processing power. Regardless of hardware limitations though, an implementation in C language of this algorithm allows its compilation for several microprocessors architectures.

Summarizing the aforementioned observations, the ADCV is an attitude determination system adequate for laboratory implementation; the TRIAD algorithm is effective to determine the platform and a spacecraft attitude as long as the proper sensors are employed; the USQUE algorithm is capable of determining the platform and a satellite attitude. Therefore, the present work contributes to increase the LAICA facility capabilities, specially when considering the several practical contributions which were necessary for the experiments execution. The present work also initiates the implementation and testing of small satellite technologies by utilizing the CubeSat onboard computer ABACUS. Moreover, the relevance of the TRIAD and USQUE implementation is highlighted by the fact that neither of these algorithms was embedded on the Serpens CubeSat due to the short development time available until launch. Hence, the implementation of these algorithms on an OBC of the same Serpens OBC family represents a significant improvement for future missions. Additionally, simulations performed during the development of that CubeSat suggested good results would have been obtained if TRIAD and USQUE had been embedded on Serpens [Oliveira et al. 2014].

The first studies concerning the air-bearing table, the Helmholtz cage and the general problem of attitude determination led to the published works listed below:

- **Guimarães, F. C., da Silva, R. C., de Loiola, J. V. L., Borges, G. A., Borges, R. A., Battistini, S., Cappelletti, C. *Aplicação do Filtro de Kalman para a Determinação de Atitude de Plataforma de Testes de Pequenos Satélites. XIII Simpósio Brasileiro de Automação Inteligente - SBAI, October 2017* [Guimarães et al. 2017].** In this work an EKF approach was implemented to determine the air-bearing platform attitude. The prediction part of the algorithm used a three-axis gyro measurements integrated in time to obtain a first attitude estimate, represented by the three Euler angles⁵. The correction was made using an attitude measurement given by a deterministic algorithm, which used the three-axis accelerometer and magnetometer measurements as the two vector observations. The results were satisfactory in spite of some limitations. These includes the absence of a gyro bias estimation and the simplistic model adopted for the local Earth's magnetic field. A gyro bias estimation and the direct inclusion of magnetometer and accelerometer measurements into the filter structure could have yielded better results. However, the improvements attained since this publication allowed to overcome such limitations in this present dissertation.

⁵The Euler angles representation is defined in Chapter 2.

- **Loiola, J. V. L., van der Ploeg, L. C., Silva, R. C., Guimarães, F. C., Borges, R. A., Borges, G. A., Battistini, S., Cappelletti, C. *3 axis simulator of the Earth magnetic field*. IEEE Aerospace Conference, March 2018 [Loiola et al. 2018].** In this work the Earth magnetic field simulator (EMFS) under development at LAICA was described. The simulator is composed by the Helmholtz cage, software and sensors, and is capable of simulating the Earth magnetic field in the same way a satellite would perceive while orbiting the planet. Since this publication, the EMFS capabilities have been extended, although some limitations still exist. The current state of the simulator is described in Chapter 4.
- **Silva, R. C., Guimarães, F. C., Loiola, J. V. L., Borges, R. A., Battistini, S., Cappelletti, C. *Tabletop testbed for attitude determination and control of nanosatellites*. Journal of Aerospace Engineering, 2018 [Silva et al. 2018].** In this work – accepted for publication – the three-axis air-bearing table and the Earth magnetic field simulator at LAICA were described in details. Several balancing procedures were compared with the LSM method employed in [Silva et al. 2016]. In addition, some results of the tests applied to the EMFS were provided. Chapter 4 presents the concept of balancing, which concerns to three-axis air-bearing tables.

Besides this manuscript, the main results obtained in the present dissertation have not been published yet. At least two papers, concerning the ADCV system and the USQUE implementation and test, respectively, are intended to be submitted soon.

1.4 Presentation of the manuscript

The present dissertation is organized in five chapters, including the present introductory chapter.

In Chapter 2 some basic concepts are provided. They concern the necessary mathematical framework for a formal definition of attitude and the directly related topics about rotations in the three-dimensional space. In other words, the extensive material of that chapter consists in a survey of several attitude representations and the related fundamental concepts, which were extracted from classical papers and books of the field and concisely presented in the manuscript.

In Chapter 3 the deterministic and stochastic methods of attitude determination implemented in this work are described. The fundamental concepts of Chapters 2 and 3 are certainly quite useful for new students which are beginning their theoretical studies.

Chapter 4 presents the experimental elements dealt with in this present dissertation context. An overview of the LAICA facility is exhibited, followed by detailed descriptions of the three-axis air-bearing table and the Helmholtz cage. Then, the implemented system for

attitude determination at the laboratory using computer vision is presented. The chapter ends with a description of the nanosatellite OBC ABACUS where some algorithms were tested.

Chapter 5 describes the experimental procedures performed at LAICA and the practical aspects involved in the implementation of the attitude determination techniques, including some software and hardware issues. In the end of the chapter the results obtained and their analysis are presented.

In Chapter 6 the conclusions are given.

Chapter 2

Attitude representation

This chapter contains the fundamental concepts concerning the attitude definition and characterization. The mathematical framework presented should be sufficient to understand the experimental procedures of this dissertation and their results, despite of some omitted points.

2.1 Reference frames

In Chapter 1 the attitude of a body was simply defined as its orientation in the three-dimensional space which, for the sake of a mathematical grounding, must be understood as being equivalent to the \mathbb{R}^3 set. In order to give a formal definition for the attitude of a body, the term *reference frame* – or *reference system* – shall be established first, considering the terminology vary among authors.

A reference frame in \mathbb{R}^3 is composed by a set of three linearly independent vectors belonging to a vector space, which turns to be the \mathbb{R}^3 itself. These vectors constitute a basis \mathcal{B} for \mathbb{R}^3 , thus any vector in this set can be expressed in terms of them. For convenience, normally the three vectors taken are mutually orthogonal and with unit norms, thereby constituting an orthonormal basis for \mathbb{R}^3 . Besides the three vectors set, an origin point \mathbf{O} for the reference system must be specified. In diagram representations, the origins of the three reference vectors are conveniently fixed in \mathbf{O} . Additionally, a right-handed reference frame can be associated to the well-known Cartesian coordinate system represented by the axes \mathbf{X} , \mathbf{Y} and \mathbf{Z} . The definition of a right-handed frame and the notation adopted in this work will be given in Section 2.2.

Beyond the mathematical formalism, in practice a reference frame must be defined using physical quantities or well-known objects, such as gravity, magnetic fields, the Earth, the stars, the Sun, etc. The way a reference frame is defined can affect the variables measurements referenced in this system. In general, these effects depend on whether the reference frame is *inertial* or *non-inertial*.

Inertial reference frames are a class of non-accelerated frames, which means they can not rotate but only translate with a constant velocity. On the contrary, non-inertial reference frames are accelerated frames, either rotating or translating with a variable velocity. The consequence of these definitions is the acceleration of a body measured in an inertial frame are independent of the frame movement. Indeed, Newton’s laws are only valid in an inertial frame. In a non-inertial frame the accelerations depend on the frame movement and a body fixed relative to this system perceives them as forces, called inertial or fictitious forces. On the other hand, an inertial frame is an ideal concept which nowadays still leads to unsettled questions about relative and absolute references¹. In real-life applications, is only possible to define reference systems approximately inertial, which means the frame-dependent accelerations are negligible for certain purposes.

2.1.1 Body-fixed frame

There are some reference frames frequently utilized in applied problems. The first to be mentioned is essential for the attitude definition and is designated as the body-fixed frame, body-frame or body-system. This reference system origin normally is placed at the body center and the frame is fixed relative to it, as shown in Figure 2.1, where the body is a 3U CubeSat. The body-system is obviously non-inertial and the accelerations and angular velocities measured by sensors mounted in the body will be affected by the body-frame movement.

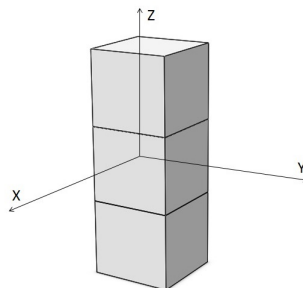


Figure 2.1: Body-system of a 3U CubeSat.

2.1.2 NED

Another non-inertial frame, commonly used in devices on Earth, is the North-East-Down (NED) system [Cai et al. 2011] depicted in Figure 2.2. In this frame the Z axis points to the center of mass (CM) of the Earth, the X axis points to the geodetic north and the Y axis completes the right-handed frame. The system origin can be arbitrarily placed at any point on the Earth’s surface or above it. The origin point are usually placed at the center of the

¹These questions are related to the secular debate between ‘substantialism’ and ‘relationism’.

body for which the attitude must be determined. It can also be placed at some point on the Earth's surface while the body translates above, such as in aircraft or spacecraft applications.

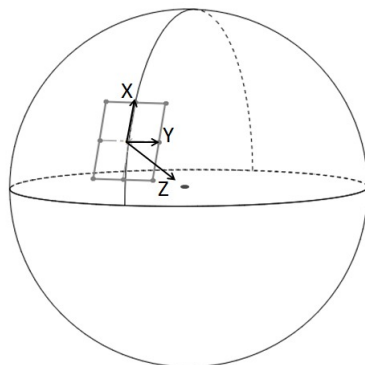


Figure 2.2: NED system representation relative to the Earth.

2.1.3 Orbit reference frame

The orbit reference frame (ORF) is applied to orbiting objects, including spacecraft. This system is represented in Figure 2.3, where the origin is placed at the CM of the body and the Z axis points toward the CM of the Earth. The Y axis is parallel to the negative orbit normal direction. This is the same direction of the vector $\mathbf{k} \times \mathbf{v}$, where \mathbf{v} is the spacecraft velocity vector, \mathbf{k} is the unit vector parallel to Z , and the symbol \times denotes the vector product. The X axis is defined in order that it completes the right-handed frame, it is also parallel to \mathbf{v} for circular orbits [Sidi 1997]. The direction of the Z axis in the ORF also defines the spacecraft nadir direction. A common task of a satellite ADCS consists in pointing an instrument toward the nadir direction.

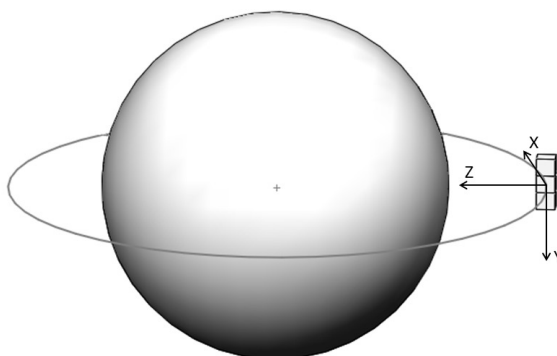


Figure 2.3: Orbit reference frame representation relative to the Earth.

2.1.4 ECEF

The Earth-Centered-Earth-Fixed (ECEF) system, as the name suggests, is a reference frame fixed relative to the Earth such that its origin is placed at center of the planet and the X axis lies in the equator plane, intersecting the prime meridian in Greenwich (i.e., at 0° latitude and 0° longitude). The Z axis is parallel to the spin axis of the Earth, pointing to the north pole. The Y axis is chosen in order that it completes the right-handed frame [Cai et al. 2011]. A representation of the ECEF system is exhibited in Figure 2.4.

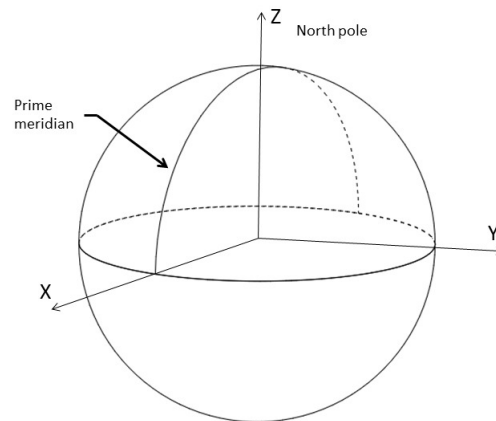


Figure 2.4: Representation of the Earth-Centered-Earth-Fixed system.

2.1.5 ECI

The last three reference systems are non-inertial as is the body-system. All of them are rotating frames, the NED and ORF systems translate with or around the Earth's surface, and the ECEF translates around the Sun; curved trajectories always implies in acceleration. For a great number of cases the acceleration effects in these frames can be ignored. Nevertheless, an inertial frame definition such that the system does not rotate relative to the solar system is useful, specially for spacecraft applications. Figure 2.5 shows a representation of the Earth-Centered-Inertial (ECI) reference system.

The origin point of ECI is fixed at the Earth center and the Z axis is parallel to the spin axis of the planet, pointing to the north pole. The equatorial plane of the Earth is inclined relative to its orbit plane around the Sun – by an angle of about 23.5° – called the ecliptic plane. Therefore, the intersection of these two planes is a straight line passing through the Earth's center and orthogonal to its spin axis. This line direction is approximately constant relative to the distant stars, which are the effective inertial reference used by the ECI system. Thus, the X axis is parallel to this intersection line, pointing toward the Sun at the vernal equinox, which marks the first day of spring in the northern hemisphere and the first day of autumn in the southern hemisphere. For this reason, X is also called the vernal equinox vector.

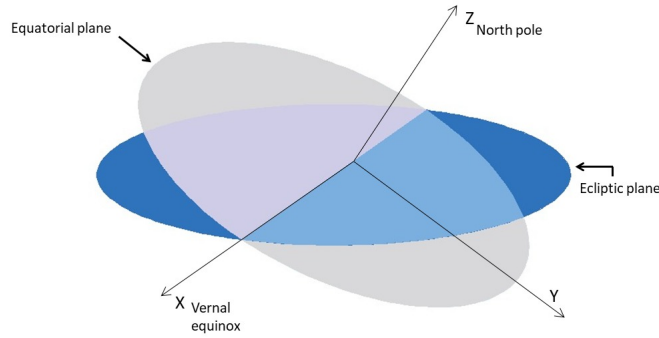


Figure 2.5: Representation of the Earth-Centered-Inertial system.

Considering the Earth’s precessional motion due to the Sun and the nutational motion due to the Moon², the Z and X axes defined above are not fixed relative to the celestial inertial reference. Consequently, these axes directions are associated to a certain date. Nowadays, the date commonly used is specified by the J2000 standard: January 1, 2000.

Referring to the observations made concerning inertial reference frames, the ECI system is not truly inertial, specially due to the translational movement of the Earth around the Sun. However, considering this motion have a long period, the ECI system can be considered inertial for most of the practical applications.

Summarizing what was exposed, the origin point of the ECI system is fixed at the Earth center; the Z axis is parallel to the spin axis of the Earth, pointing toward the north pole; X axis is the vernal equinox vector and the direction of both axes are associated to a specific date; the Y axis is defined in order that it completes the right-handed frame [Sidi 1997, Wertz 1978, Schutz et al. 2004].

2.1.6 Attitude definition

Once the commonly used reference frames have been presented, the attitude of a body is defined as the rotation operation applied to a reference frame – not fixed to it – that makes its X , Y , Z axes parallel to the X , Y , Z axes of the body-frame, respectively.

Given two reference frames, there is an infinite number of rotation operations that can align one frame to the other. This variety allows different mathematical representations for the attitude. Despite of the infinitely large number of possible rotation operations, the overall process can be unequivocally defined in a way that it depends on the two reference frames only. Section 2.3 will present the way this unique characterization can be obtained, and some of the different forms of attitude representation.

Before concluding this section, the terminology shall be clarified. The two reference

²The period of the precessional motion of the Earth is of 25,800 years. The nutational motion period is of 18.6 years.

frames necessary for the attitude definition have asymmetrical roles. The body-frame is always present, whereas the other frame is chosen accordingly to the application. Moreover, the rotation operation is applied to this last frame which act as a reference for the attitude determination. Therefore, since the body-frame is also a reference frame, in order to avoid ambiguities with the word *reference* the terminology must be settled. Hereafter, the term *reference-system* will be exclusively used to designate the frame not fixed relative to the body, such as NED, ECEF, ORF or ECI. Analogously, the term *body-system* will be preferred to designate the body-fixed frame, although the other terms might be used.

2.2 Rotations in three-dimensional space

In this section the first equations of this manuscript will be given. Firstly, the adopted notation shall be presented.

2.2.1 Notation and conventions

- Scalars are represented by minuscule italic letters: a, b, α, β .
- Vectors are represented by bold italic letters. Unless otherwise stated, these characters will also be minuscule: $\mathbf{x}, \mathbf{y}, \mathbf{u}, \mathbf{v}$.
- Matrices are represented by capital letters: A, B, Ω, Ψ .
- A matrix entries are represented by the minuscule version of the letter which represents the matrix. The row and column indices will accompany the small case character as subscripts: a_{ij} is the element of the i^{th} row and the j^{th} column of the matrix A .
- Angles are represented by minuscule Greek italic letters: $\phi, \theta, \psi, \alpha$.
- Points and Cartesian coordinate axes are represented by capital bold italic letters: $\mathbf{X}, \mathbf{Y}, \mathbf{O}, \mathbf{P}$.

Besides the above notations, some conventions must also be established.

A vector $\mathbf{u} \in \mathbb{R}^n$ is represented by the ordered set of scalars $\{u_1, u_2, \dots, u_n\}$ (with $u_i \in \mathbb{R}$) such that $\mathbf{u} = (u_1, u_2, \dots, u_n)$.

The vector space \mathbb{R}^3 possess a standard basis correspondent to the unit vectors $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$. It also has the element $(0, 0, 0)$ as an origin point. The three unit vectors along with the origin can be seen as a standard reference frame. However, in an experimental point of view, the standard basis is no significant unless it is associated to a physically established reference system, such as those presented in Section 2.1. Therefore, the fact of any basis for \mathbb{R}^3 can be expressed in terms of the standard basis has no practical

meaning. Defining such expression would be equivalent to establish the attitude between the two bases *a priori*, which is the actual problem to be addressed. In view of these observations, in this work the concept of standard basis will be dropped. Thus, any basis of \mathbb{R}^3 will be treated as being conceptually equivalent to each other, reflecting the mathematical equivalence between reference frames in attitude determination problems.

Considering the correspondence between the reference vectors and the bases for \mathbb{R}^3 , a convention shall be established in order to settle the concepts. In the present dissertation, the reference vectors of an orthonormal basis for \mathbb{R}^3 will be represented by the unit vectors \mathbf{i} , \mathbf{j} and \mathbf{k} . A subscript character might be used with each unit vector to distinguish them from the unit vectors of another frame. Therefore, a vector $\mathbf{u}_r \in \mathbb{R}^3$ expressed in a basis $\mathcal{B}_r = \{\mathbf{i}_r, \mathbf{j}_r, \mathbf{k}_r\} \subset \mathbb{R}^3$ can be written in terms of the unit vectors, as shown in Equation 2.1.

$$\mathbf{u}_r = x_r \mathbf{i}_r + y_r \mathbf{j}_r + z_r \mathbf{k}_r . \quad (2.1)$$

A column or row matrix provides a convenient way to represent vectors defined as in Equation 2.1. For consistency, only a column matrix representation will be used in this dissertation, such that $\mathbf{u}_r = [x_r \ y_r \ z_r]^T$. The superscript T denotes the matrix transposition operation. Also, the notation for vectors, with bold minuscule italic letters, must prevail over the matrix notation.

In order to avoid ambiguities with the commonly used notation (x, y, z) , in this work the column vector notation will be exclusively used to define the way a vector is expressed in a certain basis, not the standard basis. Therefore, the definition in Equation 2.2 is given.

$$[x_r \ y_r \ z_r]^T \stackrel{\text{def}}{=} x_r \mathbf{i}_r + y_r \mathbf{j}_r + z_r \mathbf{k}_r = \mathbf{u}_r . \quad (2.2)$$

As a consequence of the definition in Equation 2.2, before introducing a column vector and its components, the basis in which it is expressed must be provided. The unit vectors of a frame can also be expressed as column vectors. For instance,

$$\mathbf{i}_r = 1 \cdot \mathbf{i}_r + 0 \cdot \mathbf{j}_r + 0 \cdot \mathbf{k}_r , \quad (2.3)$$

hence, $\mathbf{i}_r = [1 \ 0 \ 0]^T$ and of course, $\mathbf{j}_r = [0 \ 1 \ 0]^T$, and $\mathbf{k}_r = [0 \ 0 \ 1]^T$. The convenience of matrix notation will be clear in the next topics.

The definitions above implies that two different vector columns can represent the same vector. That is, the vector columns are different representations of the same vector in different bases of \mathbb{R}^3 . The term *abstract vector* is often used to designate the entity which remain invariable regardless of the basis, distinguishing this entity from its frame-dependent representations [Shuster 1993]. In applied studies an abstract vector is associated to physical quantities, such as velocity, acceleration, magnetic field, etc.

Some works use different letter fonts or other variations to distinguish an abstract vector from its representations in different bases. In the present dissertation, the abstract vector will follow the notation for vectors already established, also being designated by simple letters such as \mathbf{u} , \mathbf{x} , \mathbf{y} , but free of accents or other symbols. An abstract vector representations will be designated by the same letter, but with subscript or superscript symbols, or accents. The extra symbol of an abstract vector representation will refer to the basis in which it is expressed. Therefore, the vector \mathbf{u}_r in Equation 2.1 can be seen as a vector representation of an abstract vector \mathbf{u} in the basis \mathcal{B}_r . Additionally, in this manuscript the word *vector* used alone will designate an abstract vector representation.

Considering a vector components constitute an ordered set, their roles are not interchangeable. For vectors in \mathbb{R}^3 , this is associated to the right-handedness propriety. A reference frame $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\} \in \mathbb{R}^3$ is right-handed if

$$\mathbf{i} \times \mathbf{j} = \mathbf{k}, \quad \mathbf{j} \times \mathbf{k} = \mathbf{i}, \quad \mathbf{k} \times \mathbf{i} = \mathbf{j}. \quad (2.4)$$

If the unit vector \mathbf{i} is associated to the Cartesian \mathbf{X} axis, Equations 2.4 are equivalent to say that \mathbf{j} and \mathbf{k} are associated to the \mathbf{Y} and \mathbf{Z} axes, respectively.

The bases for \mathbb{R}^2 and \mathbb{R}^3 considered in the subsequent exposition will always refer to reference frames. Hence, they will always be orthonormal bases for those vector spaces. In order to delimit the concept of basis, relating it to a reference frame, hereafter the symbol \mathcal{F} will be used to designate an orthonormal basis for \mathbb{R}^2 or \mathbb{R}^3 . As a consequence of this convention, the terms *basis*, *reference frame*, and its previously defined synonymous, will be used interchangeably whenever no ambiguity is possible.

2.2.2 Rotations in \mathbb{R}^2

In order to specify how rotations can be applied to a reference frame in \mathbb{R}^3 , is useful to begin with rotations in \mathbb{R}^2 . Let $\mathcal{F}_1 = \{\mathbf{i}_1, \mathbf{j}_1\}$ and $\mathcal{F}_2 = \{\mathbf{i}_2, \mathbf{j}_2\}$ be two reference frames of \mathbb{R}^2 . An abstract vector \mathbf{u} , expressed in terms of the reference frame $\{\mathbf{i}_1, \mathbf{j}_1\}$ such that $\mathbf{u}_1 = [x_1 \ y_1]^T \in \mathbb{R}^2$, possesses a different set of components in the other basis \mathcal{F}_2 . Let $\mathbf{u}_2 = [x_2 \ y_2]^T \in \mathbb{R}^2$ be the representation of \mathbf{u} in \mathcal{F}_2 . The relationship between the two representations can be found by considering the frame $\{\mathbf{i}_1, \mathbf{j}_1\}$ is rotated about the origin through a positive angle α (i.e., a counter-clockwise rotation) until it reaches the second frame $\{\mathbf{i}_2, \mathbf{j}_2\}$, as shown in Figure 2.6, where the vector \mathbf{u} remain fixed.

The relationship between \mathbf{u}_1 and \mathbf{u}_2 is given by Equations 2.5 and 2.6.

$$x_2 = x_1 \cos \alpha + y_1 \sin \alpha, \quad (2.5)$$

$$y_2 = -x_1 \sin \alpha + y_1 \cos \alpha. \quad (2.6)$$

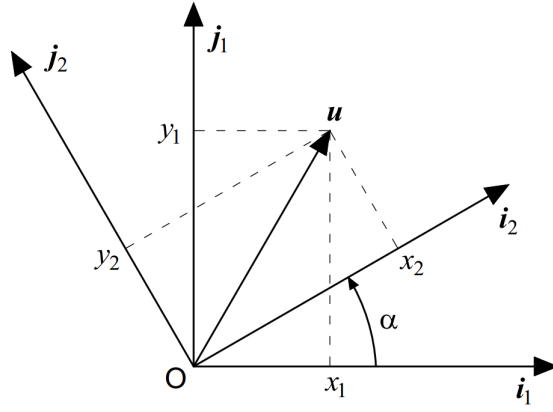


Figure 2.6: Reference frames $\{i_1, j_1\}$ and $\{i_2, j_2\}$ related by the angle α .

The rotation operation could have been done by rotating the $\{i_2, j_2\}$ frame about the origin through an angle $-\alpha$. The result would still be the overlapping between the two frames. In this case, the relationship between the two set of components can be found by swapping their subscripts and replacing α by $-\alpha$ in Equations 2.5 and 2.6. Equations 2.7 and 2.8 demonstrate these changes effects:

$$x_1 = x_2 \cos(-\alpha) + y_2 \sin(-\alpha) = x_2 \cos \alpha - y_2 \sin \alpha , \quad (2.7)$$

$$y_1 = -x_2 \sin(-\alpha) + y_2 \cos(-\alpha) = x_2 \sin \alpha + y_2 \cos \alpha . \quad (2.8)$$

Once α is fixed, the rotation operation is linear over a vector components. Thus, the last equations can be written using the matrix product as a linear operator. This results in more compact forms shown in Equations 2.9 and 2.10.

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} , \quad (2.9)$$

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} . \quad (2.10)$$

A noticeable fact is that the square matrix in Equation 2.10 is the transpose of the one in Equation 2.9. Furthermore, replacing the vector $[x_1 \ y_1]^T$ in Equation 2.9 by the right-hand expression in Equation 2.10 gives the following:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} . \quad (2.11)$$

Observing Equation 2.11, is clear that the product between the two square matrices is

equal to the identity matrix. Therefore, Equations 2.12 holds:

$$\begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix}^T = I_2 . \quad (2.12)$$

where the I_2 matrix is the identity matrix of order 2. The last Equation in 2.12 defines the orthogonality property for matrices. Thus, any square matrix A that satisfies the condition $AA^T = I$ is an orthogonal matrix. Also, it is easy to show that $\det(A) = \pm 1$ whenever A is orthogonal. Another property of an orthogonal matrix is that its inverse is its transpose, that is $A^{-1} = A^T$. The square matrices of Equations 2.9 and 2.10 have determinants equal to +1. More will be said later about the properties of orthogonal matrices and their relationship with rotations.

Figure 2.7 shows three reference frames $\mathcal{F}_1 = \{\mathbf{i}_1, \mathbf{j}_1\}$, $\mathcal{F}_2 = \{\mathbf{i}_2, \mathbf{j}_2\}$ and $\mathcal{F}_3 = \{\mathbf{i}_3, \mathbf{j}_3\}$, related by rotations about the origin through the positive angles α and β . The frame $\{\mathbf{i}_2, \mathbf{j}_2\}$ can be seen as $\{\mathbf{i}_1, \mathbf{j}_1\}$ rotated through the angle α , and the frame $\{\mathbf{i}_3, \mathbf{j}_3\}$ can be seen as $\{\mathbf{i}_2, \mathbf{j}_2\}$ rotated through the angle β . Equations 2.13 and 2.14 defines the relationship between the vectors components where, as expected, the vector $\mathbf{u}_3 = [x_3 \ y_3]^T$ is the same fixed vector \mathbf{u} expressed in the basis \mathcal{F}_3 .

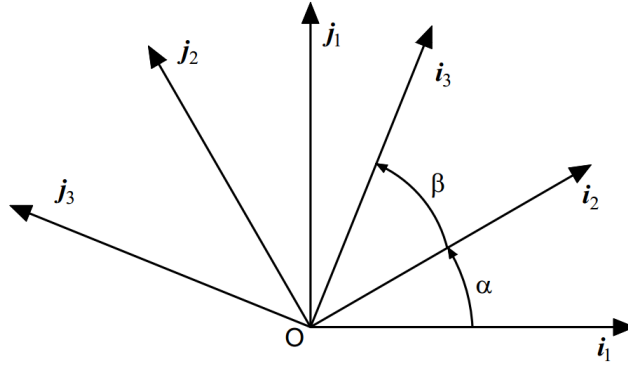


Figure 2.7: Reference frames $\{\mathbf{i}_1, \mathbf{j}_1\}$, $\{\mathbf{i}_2, \mathbf{j}_2\}$ and $\{\mathbf{i}_3, \mathbf{j}_3\}$ related by the angles α and β .

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} , \quad (2.13)$$

$$\begin{bmatrix} x_3 \\ y_3 \end{bmatrix} = \begin{bmatrix} \cos \beta & \sin \beta \\ -\sin \beta & \cos \beta \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} . \quad (2.14)$$

Clearly, the frames $\{\mathbf{i}_1, \mathbf{j}_1\}$ and $\{\mathbf{i}_3, \mathbf{j}_3\}$ could be directly related by rotating the former

through an angle $(\alpha + \beta)$. Thus, the relationship between the two frames is given by:

$$\begin{bmatrix} x_3 \\ y_3 \end{bmatrix} = \begin{bmatrix} \cos(\alpha + \beta) & \text{sen}(\alpha + \beta) \\ -\text{sen}(\alpha + \beta) & \cos(\alpha + \beta) \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}. \quad (2.15)$$

On the other hand, replacing the vector $[x_2 \ y_2]^T$ in Equation 2.14 by the right-hand expression in Equation 2.13 gives the following:

$$\begin{bmatrix} x_3 \\ y_3 \end{bmatrix} = \begin{bmatrix} \cos \beta & \text{sen} \beta \\ -\text{sen} \beta & \cos \beta \end{bmatrix} \begin{bmatrix} \cos \alpha & \text{sen} \alpha \\ -\text{sen} \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}. \quad (2.16)$$

Therefore, the square matrix in Equation 2.15 must be equivalent to the matrix product in Equation 2.16. This can be easily verified by multiplying the two matrices and using some trigonometric identities. Consequently, a sequence of rotations can simply be described by the product of the associated square matrices.

2.2.3 Rotation matrices

The results obtained for rotations in \mathbb{R}^2 can naturally be extended to \mathbb{R}^3 by observing that the reference frames $\{\mathbf{i}_1, \mathbf{j}_1\}$ and $\{\mathbf{i}_2, \mathbf{j}_2\}$ depicted in Figure 2.6 can be seen as subsets of reference frames in \mathbb{R}^3 . In fact, if $\{\mathbf{i}_1, \mathbf{j}_1\}$ is a subset of the frame $\{\mathbf{i}_1, \mathbf{j}_1, \mathbf{k}_1\}$, the rotation through the angle α in that picture can be seen as realized about the \mathbf{k}_1 vector.

Let $\mathbf{u}_1 = [x_1 \ y_1 \ z_1]^T \in \mathbb{R}^3$ and $\mathbf{u}_2 = [x_2 \ y_2 \ z_2]^T \in \mathbb{R}^3$ be the representations of an abstract vector \mathbf{u} in the bases $\mathcal{F}_1 = \{\mathbf{i}_1, \mathbf{j}_1, \mathbf{k}_1\}$ and $\mathcal{F}_2 = \{\mathbf{i}_2, \mathbf{j}_2, \mathbf{k}_2\}$, respectively. If the frame $\{\mathbf{i}_1, \mathbf{j}_1, \mathbf{k}_1\}$ rotates through an angle ψ about \mathbf{k}_1 , the z_1 component of \mathbf{u}_1 should remain unchanged. The rotation procedure is depicted in Figure 2.8. Thus, extending Equation 2.9 to \mathbb{R}^3 gives Equation 2.17:

$$\mathbf{u}_2 = R_3(\psi)\mathbf{u}_1, \quad (2.17)$$

where,

$$R_3(\psi) = \begin{bmatrix} \cos \psi & \text{sen} \psi & 0 \\ -\text{sen} \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.18)$$

The third column and the third row of the matrix in Equation 2.18 guarantees the z_1 component remain unaltered. Using the correspondences $\mathbf{i} - 1$, $\mathbf{j} - 2$ and $\mathbf{k} - 3$, the number 3 subscript in $R_3(\psi)$ indicates the rotation is taken about \mathbf{k} .

The matrix $R_3(\psi)$ which perform a change of basis, by means of a rotation of frames,

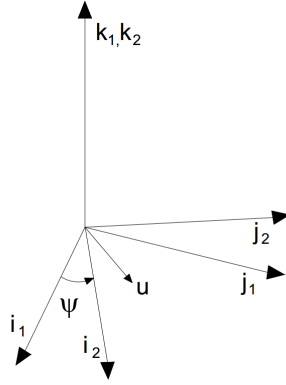


Figure 2.8: Rotation of $\{i_1, j_1, k_1\}$ about k_1 through an angle ψ .

is called a rotation matrix or rotation operator. Naturally, any matrix that realizes such a change of basis is called a rotation matrix. This is also true for different dimensions; the square matrices of order 2 in Equations 2.9 to 2.16 are also rotation matrices.

For a rotation about the i_1 vector through an angle ϕ the rotation matrix $R_1(\phi)$ is given by Equation 2.19.

$$R_1(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \text{sen } \phi \\ 0 & -\text{sen } \phi & \cos \phi \end{bmatrix}. \quad (2.19)$$

The matrix rotation operator for a rotation about the j_1 vector through an angle θ is given by Equation 2.20.

$$R_2(\theta) = \begin{bmatrix} \cos \theta & 0 & -\text{sen } \theta \\ 0 & 1 & 0 \\ \text{sen } \theta & 0 & \cos \theta \end{bmatrix}. \quad (2.20)$$

The rotation matrices of Equations 2.18 to 2.20 are orthogonal as well as the 2×2 rotation matrices were. Regardless of the rotation axis, every rotation matrix is an orthogonal matrix. This is due to the geometric characteristics which remain unchanged in spite of a rotation. These characteristics are the norm (or length) of vectors and the angle between vectors. These two parameters can be calculated using the inner product, as shown in Equations 2.21 and 2.22:

$$\|\mathbf{u}_1\|^2 = \mathbf{u}_1^T \mathbf{u}_1, \quad (2.21)$$

$$\cos \alpha = \frac{\|\mathbf{u}_1\| \|\mathbf{v}_1\| \mathbf{u}_1^T \mathbf{v}_1}{\|\mathbf{u}_1\| \|\mathbf{v}_1\|}, \quad (2.22)$$

where $\mathbf{u}_1, \mathbf{v}_1 \in \mathbb{R}^3$ are two vectors in the same orthonormal basis \mathcal{F}_1 , and α is the smaller

angle between them. The notation $\|\bullet\|$ represents the euclidean norm of a vector.

Let $A \in \mathbb{R}^3$ be an orthogonal matrix. Equations 2.23 demonstrate that when A is multiplied by u_1 , the result is a vector with the same length:

$$\begin{aligned} \|Au_1\|^2 &= (Au_1)^T(Au_1) = \\ (\mathbf{u}_1^T A^T)(Au_1) &= \mathbf{u}_1^T (A^T A) \mathbf{u}_1 = \\ \mathbf{u}_1^T I_3 \mathbf{u}_1 &= \mathbf{u}_1^T \mathbf{u}_1 = \|\mathbf{u}_1\|^2 . \end{aligned} \quad (2.23)$$

Since the vectors norm do not change when they are operated by an orthogonal matrix A , the preservation of the angle between two vectors u_1 and v_1 can be analysed using the inner product only, as stated by the definition in Equation 2.22. Equations 2.24 demonstrate the inner product remain unchanged despite of the pre-multiplication by A :

$$\begin{aligned} (Au_1)^T(Av_1) &= \\ (\mathbf{u}_1^T A^T)(Av_1) &= \mathbf{u}_1^T (A^T A) v_1 = \\ \mathbf{u}_1^T I_3 v_1 &= \mathbf{u}_1^T v_1 . \end{aligned} \quad (2.24)$$

A consequence of the results above is that every rotation matrix A must be orthogonal. The converse is not true. As mentioned before, for any orthogonal matrix B , $\det(B) = \pm 1$. In order to B be a rotation matrix, $\det(B)$ must be equal to $+1$ so that the right-handedness property is preserved. The matrix $R_1(\phi)$ defined in Equation 2.19 can be used to demonstrate this fact. Given a vector $\mathbf{u}_1 = [x_1 \ y_1 \ z_1]^T$ in a reference frame of \mathbb{R}^3 , Equations 2.25 show the effect of applying the rotation operator $R_1(0)$ to this frame has over \mathbf{u}_1 components. However, the sign of the 1 in the upper-left-hand corner in $R_1(0)$ is reversed. With this small alteration, the matrix determinant is equal to -1 .

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & \cos(0) & \sin(0) \\ 0 & -\sin(0) & \cos(0) \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} -x_1 \\ y_1 \\ z_1 \end{bmatrix} . \quad (2.25)$$

The supposed rotation operator reversed the sign of the \mathbf{u}_1 first component. Nevertheless, \mathbf{u}_1 should have remained unchanged since the rotation angle was 0. Seeing the reference frame as a rigid structure, such operation took one frame to another not by means of a rotation, but by deforming the original frame. Hence, the right-handed frame was taken into a left-handed frame. Considering the special properties of the group of orthogonal matrices with determinant equal to $+1$, it receives the name $SO(3)$. Hence, the $SO(3)$ group comprises all the orthogonal matrices of $\mathbb{R}^{3 \times 3}$ with determinant equal to $+1$. The matrices of

$SO(3)$ are also called proper real orthogonal matrices.

For rotations in \mathbb{R}^2 was shown that a sequence of rotations could be described by multiplying the correspondent rotation matrices. However, the rotations was taken about the same implicit axis Z^3 . For rotations in \mathbb{R}^3 the matrix product property still holds, even for rotations taken about different axes. The demonstration of this fact is similar to the \mathbb{R}^2 case, one must takes intermediate frames and replace the vectors obtained in each step. Equation 2.26 defines a matrix A as a product of several matrices of the form R_n , which means the rotations are applied about one of the frame vectors i , j and k .

$$A = R_2 R_1 R_1 R_3 R_2 . \quad (2.26)$$

In Equation 2.26 the rotation angles were omitted since they are not required for the subsequent analysis.

Using the intermediate reference frames, the matrix product property of sequential rotations in \mathbb{R}^3 arises as naturally as in the \mathbb{R}^2 case. However, unlike the 2-dimensional rotations, it is not obvious that the matrix A is a rotation matrix. This is mainly due to the fact that the intermediate angles – which do not belong to the same plane – can not be summed as in Equation 2.15. Despite of this impossibility, A is indeed a rotation matrix since it is an orthogonal matrix with $\det(A) = 1$. In fact, given two orthogonal matrices A and B , the product AB is also an orthogonal matrix as demonstrated in Equation 2.27.

$$\begin{aligned} (AB)(AB)^T &= (AB)(B^T A^T) \\ &= A(BB^T)A^T = AA^T = I_3 . \end{aligned} \quad (2.27)$$

Since each R_n matrix is an orthogonal matrix , the property demonstrated in Equation 2.27 implies that A is an orthogonal matrix. Also, once $\det(AB) = \det(A)\det(B)$ and $\det(R_n) = 1$, $\det(A) = 1$. Therefore, as explained earlier, the matrix A in Equation 2.26 is a rotation matrix.

2.2.4 Fixed axis of rotation

The matrix multiplication property for sequential rotations constitutes a procedure to obtain a rotation matrix with a different form than the R_n ones. However, given a frame in \mathbb{R}^3 , it should be possible to rotate it freely, not exclusively about the frame vectors i , j and k . The fact that not only matrices of the form R_n are rotation matrices, but every orthogonal matrix with determinant equal to 1, suggests such a freedom of movement. These observations lead to the question of whether a rotation matrix not in the shape of R_n could

³An axis can be mathematically defined using a vector parallel to it. Thus, in this work both words *axis* and *vector* will be used to describe a rotation.

represent a rotation about a vector different from i , j and k . This is indeed the case, as shown in the following.

Let $A \in \mathbb{R}^{3 \times 3}$ be a rotation matrix. The scalars λ which satisfy the Equation 2.28 are called *eigenvalues* of A .

$$A\mathbf{v}_1 = \lambda\mathbf{v}_1, \quad (2.28)$$

where $\mathbf{v}_1 \in \mathbb{R}^3$ is any vector such that $\mathbf{v}_1 \neq \mathbf{0}$ ⁴. It is called *eigenvector* of A . Equation 2.28 implies in:

$$A\mathbf{v}_1 - \lambda\mathbf{v}_1 = \mathbf{0}, \quad (2.29)$$

$$(A - \lambda I_3)\mathbf{v}_1 = \mathbf{0}. \quad (2.30)$$

In order to the linear system of Equation 2.30 have a non-trivial solution, since $\mathbf{v} \neq \mathbf{0}$, the Equation 2.31 must be satisfied.

$$\det(A - \lambda I_3) = 0. \quad (2.31)$$

The expansion of the determinant in Equation 2.31 results in a polynomial of the same order as A , in this case a polynomial of order 3. It is called the characteristic polynomial for A and Equation 2.31 is called a characteristic equation. Considering A is orthogonal, Equations 2.32 demonstrate that A has at least one eigenvalue equal to 1.

$$\begin{aligned} \det(A - I_3) &= 1 \cdot \det(A - I_3) = \det(A^T)\det(A - I_3) = \\ \det[A^T(A - I_3)] &= \det(A^T A - A^T I_3) = \\ \det(I_3 - A^T) &= \det[(I_3 - A)^T] = \\ \det(I_3 - A) &= \det[(-1)(A - I_3)] = \\ &= -\det(A - I_3). \end{aligned} \quad (2.32)$$

Therefore,

$$\det(A - I_3) = 0. \quad (2.33)$$

Comparing Equation 2.33 to Equation 2.31, the former states that $\lambda = 1$ is an eigenvalue

⁴A zero written in bold face represents a vector or matrix with all entries equal to 0.

of A . Replacing this eigenvalue in Equation 2.28, results in

$$A\mathbf{v}_1 = \mathbf{v}_1 . \quad (2.34)$$

Since A is a rotation matrix, the left-hand side of Equation 2.34 is equivalent to apply a rotation to the reference frame in which \mathbf{v}_1 is expressed. It implies that there exists a vector \mathbf{v}_1 which remain unaltered after the change of reference frames performed by A . This is only possible if \mathbf{v}_1 is parallel to the axis of rotation.

The expected result was thereby obtained. Given an arbitrary rotation matrix A , there indeed exists a single axis about which the rotation operation is taken. This axis is parallel to the eigenvector corresponding to the eigenvalue $\lambda = 1$, and it is called the fixed axis of rotation.

The result presented is highlighted when compared to the sequential process of multiplying R_n matrices. A sequence of rotations as the one exemplified in Equation 2.26 is equivalent to take a single rotation about the \mathbf{v}_1 vector defined in Equation 2.34.

Once the existence of the fixed axis has been established, it can be found by solving the homogeneous linear system described in Equation 2.30 with $\lambda = 1$. That is, one must solve the linear system of Equation 2.35.

$$(A - I_3)\mathbf{v}_1 = \mathbf{0} . \quad (2.35)$$

However, directly solving the system in Equation 2.35 using common approaches do not take into account the orthogonality of A . This property establishes several relationships between A entries, leading to a much simpler solution for the fixed axis. This simpler solution is demonstrated in [Kuipers 1999] and it is shown in Equation 2.36.

$$\mathbf{v}_1 = \begin{bmatrix} a_{23} - a_{32} \\ a_{31} - a_{13} \\ a_{12} - a_{21} \end{bmatrix} . \quad (2.36)$$

Applying Equation 2.36 to the matrices R_n is a good exercise to verify this equation validity. For instance, applying Equation 2.36 to the matrix $R_1(\phi)$ defined in Equation 2.19 results in

$$\mathbf{v}_1 = \begin{bmatrix} 2\sin \phi \\ 0 \\ 0 \end{bmatrix} , \quad (2.37)$$

where the fixed axis obtained is parallel to the rotation axis of R_1 which is $\mathbf{i} = [1 \ 0 \ 0]^T$.

2.2.5 Rotation angle about the fixed axis of rotation

Once the fixed axis of rotation was obtained, the angle of rotation about this fixed axis must be found. In [Kuipers 1999] is demonstrated how this angle, designated as φ , can be calculated. Equation 2.38 shows the result.

$$\cos \varphi = \frac{\text{Tr}(A) - 1}{2}. \quad (2.38)$$

The function $\text{Tr}(A)$ in Equation 2.38 denotes the *trace* of A . Again, using Equation 2.38 with $R_1(\phi)$ is helpful. Since $\text{Tr}(R_1(\phi)) = 2\cos \phi + 1$, Equation 2.38 holds.

The vector found in Equation 2.37 is not an unit vector. In general, the application of Equation 2.36 will not result in an unit vector. Nevertheless, given the total rotation angle φ , the unit vector \mathbf{n}_A which defines the fixed axis of rotation – also designated as unit rotation vector – can be calculated using Equation 2.39 [Shuster 1993].

$$\mathbf{n}_A = \frac{1}{2\sin \varphi} \begin{bmatrix} a_{23} - a_{32} \\ a_{31} - a_{13} \\ a_{12} - a_{21} \end{bmatrix}. \quad (2.39)$$

In Equation 2.39, $\sin \varphi \neq 0$ in order that \mathbf{n}_A can be defined. Furthermore, it should be noticed that \mathbf{n}_A is a vector representation invariant under the change of basis realized by A .

The material presented in this section basically followed the same lines of the logical sequence in which this subject was presented in [Kuipers 1999]. However, the mathematical formalism was a few different. The concept of abstract vector and the explicit consideration of the reference frames, treating them as bases for \mathbb{R}^3 , make this work approach closer to the one adopted in [Shuster 1993]. As reported earlier, the concept of abstract vector was taken from [Shuster 1993] as well as the result in Equation 2.38. A more detailed exposition of this section subject can be found in the before-mentioned works.

2.3 Attitude representations

In the end of Section 2.1 the attitude of a body was defined as the rotation operation which takes the reference-system into the body-system. In Section 2.2 the rotation operation was characterized in several ways. Consequently, there is more than one way to mathematically describe the attitude of a body. An attitude representation is a specification about the manner the reference-system is rotated into the body-system. The subsequent sections will present a few of the most employed attitude representations in practice.

2.3.1 Direction-cosine matrix

In Section 2.2 was reported that any matrix A of the $SO(3)$ group is a rotation matrix. Moreover, the rotation operation that takes one reference frame into another represents a operation of a change of basis for \mathbb{R}^3 . Therefore, given two orthonormal bases $\mathcal{F}_1, \mathcal{F}_2 \subset \mathbb{R}^3$ and being A the change of basis operator from \mathcal{F}_1 to \mathcal{F}_2 , it can be proved that A is unique [Meyer 2000]. That is, A depends only on the two bases. For the attitude definition, it means the rotation matrix A depends only on the two reference frames: the one rotated (reference-system) and the target frame (body-system).

The uniqueness of A implies that it can be used to unambiguously describe the attitude of a body. On the basis of this fact, the rotation matrix A constitutes an attitude representation which naturally arose throughout the precedent analysis. Besides the rotation matrix term, A is also called *attitude matrix* or *direct-cosine matrix* (DCM).

The term direct-cosine matrix comes from the geometric interpretation associated to each of its entries. Figure 2.9 shows a reference-system $\mathcal{F}_r = \{\mathbf{i}_r, \mathbf{j}_r, \mathbf{k}_r\}$ and the unit vector \mathbf{i}_b of a body-system $\mathcal{F}_b = \{\mathbf{i}_b, \mathbf{j}_b, \mathbf{k}_b\}$, all parting from the same origin point.

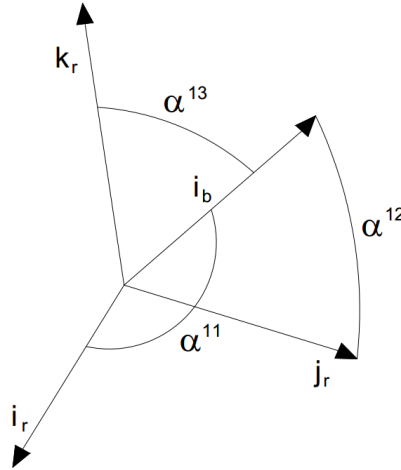


Figure 2.9: Angles between a body-system unit vector \mathbf{i}_b and the reference-system vectors.

The angles between \mathbf{i}_b and the reference-system unit vectors are also depicted in Figure 2.9. Equation 2.22 stated the relationship between the smaller angle formed by two vectors and the inner product operation. For unit vectors it assumes a simpler form given by Equation 2.40:

$$\cos \alpha = \langle \mathbf{u}_s, \mathbf{v}_s \rangle = \mathbf{u}_s^T \mathbf{v}_s, \quad (2.40)$$

where $\mathbf{u}_s, \mathbf{v}_s \in \mathbb{R}^3$ are unit vectors of the same basis \mathcal{F}_s . Also, in Equation 2.40 was defined a different notation, used in some fields, for the inner product ($\langle \bullet, \bullet \rangle$). Thus, Equation 2.40 implies that the angle between two unit vectors can be easily obtained using the inner product operation applied to them.

The derivation of the DCM A corresponding to the reference-system \mathcal{F}_r and the body-system \mathcal{F}_b requires a third basis such as \mathcal{F}_s mentioned above. The procedure of change of basis is well-known and will be omitted here, only the final result shall be provided. This derivation can be found in [Meyer 2000] for bases and vector spaces in general, and in [Kuipers 1999, Wertz 1978, Shuster 1993] for the applied problem of attitude determination.

In order to present an equation for A , the representations of the \mathcal{F}_r and \mathcal{F}_b reference vectors must be provided. Let $\mathbf{r}_s^1, \mathbf{r}_s^2$ and \mathbf{r}_s^3 be the vector representations in \mathcal{F}_s of $\mathbf{i}_r, \mathbf{j}_r$ and \mathbf{k}_r , respectively. The representations of the body-system frame vectors are designated in the same way. The vectors $\mathbf{b}_s^1, \mathbf{b}_s^2$ and \mathbf{b}_s^3 are the representations of $\mathbf{i}_b, \mathbf{j}_b$ and \mathbf{k}_b in \mathcal{F}_s , respectively. Equation 2.41 gives the definition of the attitude matrix in terms of the reference-system and body-system frame vectors representations.

$$A = \begin{bmatrix} \langle \mathbf{b}_s^1, \mathbf{r}_s^1 \rangle & \langle \mathbf{b}_s^1, \mathbf{r}_s^2 \rangle & \langle \mathbf{b}_s^1, \mathbf{r}_s^3 \rangle \\ \langle \mathbf{b}_s^2, \mathbf{r}_s^1 \rangle & \langle \mathbf{b}_s^2, \mathbf{r}_s^2 \rangle & \langle \mathbf{b}_s^2, \mathbf{r}_s^3 \rangle \\ \langle \mathbf{b}_s^3, \mathbf{r}_s^1 \rangle & \langle \mathbf{b}_s^3, \mathbf{r}_s^2 \rangle & \langle \mathbf{b}_s^3, \mathbf{r}_s^3 \rangle \end{bmatrix}. \quad (2.41)$$

Equation 2.41 clarifies the motivation for the term direction-cosine matrix. Referring to Equation 2.40, each entry of A is the cosine of the angle between two unit vectors, where one is taken from the body-system and the other from the reference-system. The matrix A is 3×3 as long as there are nine angles to take into account. Moreover, it must be noticed that the order $\mathbf{b} - \mathbf{r}$ used in Equation 2.41 does not need to be obeyed once the inner product is commutative. An ordered fashion is nevertheless helpful, working as a mnemonic device and yielding an elegant formulation.

The row-column pattern that emerges using the above-mentioned order $\mathbf{b} - \mathbf{r}$ in Equation 2.41 suggests a product in the form $[\mathbf{b}_s][\mathbf{r}_s]^T$. The terms $[\mathbf{b}_s]$ and $[\mathbf{r}_s]$ designate block matrices composed of the respective vector representations. Equation 2.42 demonstrates that such interpretation is correct.

$$A = \begin{bmatrix} (\mathbf{b}_s^1)^T \\ (\mathbf{b}_s^2)^T \\ (\mathbf{b}_s^3)^T \end{bmatrix} \left[\mathbf{r}_s^1 \mid \mathbf{r}_s^2 \mid \mathbf{r}_s^3 \right]. \quad (2.42)$$

The left-hand block matrix in Equation 2.42 is the above-mentioned $[\mathbf{b}_s]$ matrix, whereas $[\mathbf{r}_s]$ is the right-hand block matrix. Using the transposition operation, Equation 2.42 yields Equations 2.43.

$$A = \begin{bmatrix} \mathbf{b}_s^1 & \mathbf{b}_s^2 & \mathbf{b}_s^3 \end{bmatrix}^T \begin{bmatrix} \mathbf{r}_s^1 & \mathbf{r}_s^2 & \mathbf{r}_s^3 \end{bmatrix} = B_s^T R_s, \quad (2.43)$$

where the terms $B_s = [\mathbf{b}_s]^T$ and $R_s = [\mathbf{r}_s]$ establish a neater notation for the block matrices. Also, once is clear the frame vector representations are column vectors, the vertical lines were omitted in Equation 2.43.

Equation 2.43 is a remarkable result since it explicitly establishes the fact that the attitude depends only on the body-system and the reference-system. In addition, it must be noticed that the attitude do not depend on the third basis since it was arbitrarily chosen. In fact, the TRIAD algorithm derivation presented in Section 3.1 yields a result similar to Equation 2.43 without requiring a third basis.

The mathematical description of rotations using matrices makes the DCM the most natural attitude representation. However, it requires nine values to represent the attitude, whereas the rotations in \mathbb{R}^3 has three degrees of freedom. The constraints the orthogonality property of the DCM imposes on its entries implies that they are not independent of each other nevertheless. Thus, there are redundancy in the DCM representation. This characteristic makes it less attractive when compared to other forms of representation. Despite of this fact, the attitude matrix is commonly retrieved in many algorithms to establish the relationship between vectors measurements from different reference frames. The advantages the DCM has for mathematical manipulations is also worth to mention.

2.3.2 Euler angles

Some of the results presented thus far concerning rotations in \mathbb{R}^3 and the attitude problem were first obtained by Leonhard Euler (1707-1783). A theorem proved by him states that any reference frame can be rotated into another by means of a sequence of three rotations about the reference vectors. Of these three, two successive rotations can not be about the same axis. Moreover, only a twelve of the all possible sequences can be applied [Wertz 1978].

Using the already referred correspondences $i - 1$, $j - 2$, $k - 3$, and the rotation matrices R_n , in this work only the sequence designated as $3 - 2 - 1$ will be described. This sequence corresponds to a rotation about the k vector, followed by a rotation about the new j axis, and finalizing with a rotation about the newest x vector. The adjective *new* may be ambiguous. A more precise description requires two intermediate reference frames besides the reference-system $\mathcal{F}_r = \{\mathbf{i}_r, \mathbf{j}_r, \mathbf{k}_r\}$ and the body-system $\mathcal{F}_b = \{\mathbf{i}_b, \mathbf{j}_b, \mathbf{k}_b\}$. Designating them as $\mathcal{F}' = \{\mathbf{i}', \mathbf{j}', \mathbf{k}'\}$ and $\mathcal{F}'' = \{\mathbf{i}'', \mathbf{j}'', \mathbf{k}''\}$, the $3 - 2 - 1$ sequence goes as follows:

- a rotation about \mathbf{k}_r through an angle ψ , taking \mathcal{F}_r into \mathcal{F}' , is followed by
- a rotation about \mathbf{j}' through an angle θ , taking \mathcal{F}' into \mathcal{F}'' , which is followed by
- a rotation about \mathbf{i}'' through an angle ϕ , taking \mathcal{F}'' into \mathcal{F}_b .

The angles ϕ , θ and ψ are called Euler angles and they are the three parameters used to describe the attitude in this form of representation. The Greek letters ϕ , θ and ψ are traditionally adopted to represent the rotations about the \mathbf{i} , \mathbf{j} , and \mathbf{k} vectors, respectively, and can be used as alias to designate them. Additionally, the terms *roll*, *pitch* and *yaw* angles

are commonly used to refer to the same respective angles ϕ , θ and ψ ⁵.

The Euler angles representation can be mathematically formulated in terms of the rotation matrices R_n . The process is similar to the one applied to obtain Equation 2.16 in the \mathbb{R}^2 case. Each rotation is applied to take one frame to the next, until the last intermediate frame is taken into the body-system. The process result is equivalent to the direct application of the matrix product property for rotation sequences, as in the example of Equation 2.26. The final rotation matrix A is given by Equations 2.44.

$$A(\phi, \theta, \psi) = R_1(\phi)R_2(\theta)R_3(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.44)$$

The matrix multiplication in Equation 2.44 yields

$$A(\phi, \theta, \psi) = \begin{bmatrix} c_\psi c_\theta & s_\psi c_\theta & -s_\theta \\ c_\psi s_\theta s_\phi - s_\psi c_\phi & s_\psi s_\theta s_\phi + c_\psi c_\phi & c_\theta s_\phi \\ c_\psi s_\theta c_\phi + s_\psi s_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi & c_\theta c_\phi \end{bmatrix}, \quad (2.45)$$

where, the short notation $s_\bullet = \sin(\bullet)$ and $c_\bullet = \cos(\bullet)$ was employed in Equation 2.45.

The Euler angles representation uses the minimum set of three parameters to describe the attitude. On the other hand, singularities occur for certain angles, which means the Euler angles are not unique for such orientations. Therefore, if the body-system is static relative to the reference-system and the attitude is determined, the singularity implies that there are more than one triple (ϕ, θ, ψ) able to represent that attitude. If the body-system moves relative to the reference-system, some of the Euler angles time rates tend to infinity.

The singularities occur even limiting the angles ranges to avoid the trigonometric functions periodicity, which is always necessary when dealing with angles and rotations. The singularity is intrinsic to this form of representation, and any other that uses only the minimum set of three parameters. In fact, as demonstrated in [Stuelpnagel 1964], every parametrization of rotations in \mathbb{R}^3 , using only three parameters and which is global, thereby possessing a finite number of parameters associated to a single rotation, have singular points where the

⁵In [Kuipers 1999] the author emphasizes that the widespread terms row, pitch and yaw are improperly used in most of the cases. The argument is made in favour of the terms heading angle for ψ , elevation angle for θ , and bank angle for ϕ . However, these terms seem to be appropriate only for reference-systems not placed at the body center. This would be the case for a NED system with an origin point placed on the Earth's surface, while the body center translates above the earth as in an aircraft or spacecraft application. This interpretation is suggested by the referred work itself, where an aircraft application always emerges in the arguments. The utilization of the terms yaw, pitch and roll, does seem to be appropriate for body centred reference-systems such as ORF. These terms are also applicable when a NED system is placed at the body center, as in smartphones applications, or as in the modelling of the air-bearing satellite table at LAICA, described in Chapter 4.

three parameters are not uniquely defined. Thus, regardless of the sequence of rotation, a singularity always occurs for certain angles. They depend on the sequence of rotation. For the $3-2-1$ sequence the singularity occurs when $\theta = \pm 90^\circ$ or odd multiple of these values.

As an example of the singular points meaning, the triples $(-90^\circ, 90^\circ, 0^\circ)$ and $(0^\circ, 90^\circ, 90^\circ)$ represent the same attitude. Thus, if a body-system attitude is defined by $(0, 90^\circ, 90^\circ - \epsilon_\psi)$, with $0 < \epsilon_\psi \ll 1$, and it rotates to the attitude given by $(0, 90^\circ, 90^\circ + \epsilon_\psi)$, its change in orientation could be accounted as $(0^\circ, 0^\circ, 2\epsilon_\psi)$. However, since the intermediate attitude can also be given by $(-90^\circ, 90^\circ, 0^\circ)$, the frame movement could be described as going from $(-90^\circ + \epsilon_\phi, 90^\circ, 0^\circ)$ to $(0^\circ, 90^\circ, 90^\circ + \epsilon_\psi)$, where $0 < \epsilon_\phi \ll 1$. That is, the orientation change could be regarded as $(90^\circ - \epsilon_\phi, 0, 90^\circ + \epsilon_\psi)$. Therefore, the so called singularity is associated not only to the non-uniqueness of the parameters, but also to their discontinuity for small changes in attitude.

Other forms of attitude representation which use more parameters than the minimum avoid the singularity problem. In fact, one can observe that the Euler angles triples $(-90^\circ, 90^\circ, 0)$ and $(0, 90^\circ, 90^\circ)$ applied to Equation 2.45 result in the same DCM. This would be true for another attitude representation with a redundant number of parameters. On the basis of this fact, the Euler angles representation is not commonly used in any application where the body is capable to move freely in space. Nevertheless, they are useful to represent small variations in attitude – for a stabilized spacecraft, for example – or with objects that can not reach the singular angles. Indeed, the Euler angles are frequently used with the air-bearing satellite platform at LAICA, once its movements are mechanically bounded. Furthermore, this form of representation are often used for graphic visualization, once their geometric interpretation is straightforward.

2.3.3 Euler symmetric parameters and quaternions

The DCM representation is the same rotation matrix. The Euler angles are associated to sequential rotations about the reference vectors. Both representations come directly from two different ways to characterize a rotation in \mathbb{R}^3 . The last characterization presented in Section 2.2 was given in terms of the fixed axis of rotation and the total angle of rotation. These elements can be used as a form of attitude representation.

The unit rotation vector components plus the total rotation angle, result in four possible parameters for the attitude representation. This redundant number of variables can lead to an attitude representation free of singularities and with less parameters than the DCM. The Euler-Rodrigues symmetric parameters – more commonly known as the Euler symmetric parameters – consist in an attitude representation based on the four parameters η_1, η_2, η_3 and

η_4 , given by Equations 2.46:

$$\begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} = \boldsymbol{\eta} = \mathbf{n}_{br} \sin\left(\frac{\varphi}{2}\right), \quad \eta_4 = \cos\left(\frac{\varphi}{2}\right), \quad (2.46)$$

where \mathbf{n}_{br} defines the principal axis of the rotation which takes the reference-system \mathcal{F}_r to the body-system \mathcal{F}_b , and φ is the total rotation angle. Also, it must be clear that the column vector $[\eta_1 \ \eta_2 \ \eta_3]^T$ is a vector representation in \mathcal{F}_r or \mathcal{F}_b – it is invariant under the two bases.

The Euler-Rodrigues symmetric parameters are clearly not independent of each other, and this is a manifestation of the redundancy in the attitude parametrization. Indeed, since \mathbf{n}_{br} has unit norm, the four parameters satisfy the constraint stated by Equations 2.47.

$$\eta_1^2 + \eta_2^2 + \eta_3^2 + \eta_4^2 = \|\boldsymbol{\eta}\|^2 + \eta_4^2 = \|\mathbf{n}_{br}\|^2 \sin^2\left(\frac{\varphi}{2}\right) + \cos^2\left(\frac{\varphi}{2}\right) = 1. \quad (2.47)$$

The utilization of the Euler symmetric parameters as the four components of a quaternion leads to an extremely convenient form of attitude representation. A quaternion \mathbf{q} is an ordered 4-tuple of scalars such that $\mathbf{q} = (q_1, q_2, q_3, q_4)$, with $q_i \in \mathbb{R}$. Given two quaternions $\mathbf{q}' = (q'_1, q'_2, q'_3, q'_4)$ and $\mathbf{q} = (q_1, q_2, q_3, q_4)$, the quaternion addition operation is defined in Equation 2.48.

$$\mathbf{q} + \mathbf{q}' = (q_1 + q'_1, q_2 + q'_2, q_3 + q'_3, q_4 + q'_4). \quad (2.48)$$

The quaternion multiplication operation is designated by the symbol \otimes . If $\mathbf{q}'' = \mathbf{q}' \otimes \mathbf{q}$, the \mathbf{q}'' components are given by Equations 2.49.

$$\begin{aligned} q''_1 &= q_4 q'_1 + q'_4 q_1 - (q'_2 q_3 - q'_3 q_2), \\ q''_2 &= q_4 q'_2 + q'_4 q_2 - (q'_3 q_1 - q'_1 q_3), \\ q''_3 &= q_4 q'_3 + q'_4 q_3 - (q'_1 q_2 - q'_2 q_1), \\ q''_4 &= q'_4 q_4 - (q'_1 q_1 + q'_2 q_2 + q'_3 q_3). \end{aligned} \quad (2.49)$$

A more convenient formulation of Equation 2.49 can be found if the quaternion components q_1, q_2 and q_3 are regarded as a column vector $\bar{\mathbf{q}} = [q_1 \ q_2 \ q_3]^T$. This terminology can lead to confusion with the vector representation definition given in Equation 2.2. However, such notation is convenient when dealing with rotations. Hence, a quaternion can be seen as a block column matrix of order 4 such that $\mathbf{q} = [\bar{\mathbf{q}} \ q_4]^T$ ⁶. The term $\bar{\mathbf{q}}$ is called the vector

⁶The differentiation between a column vector matrix used in general and a vector representation in a certain basis, shall be clear from the context.

part of the quaternion, and the remaining component q_4 is its scalar part. Equations 2.50 establish the more compact definition of the quaternion product.

$$\mathbf{q}'' = \begin{bmatrix} \bar{\mathbf{q}}'' \\ q_4'' \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{q}}' \\ q_4' \end{bmatrix} \otimes \begin{bmatrix} \bar{\mathbf{q}} \\ q_4 \end{bmatrix} = \begin{bmatrix} q_4 \bar{\mathbf{q}}' + q_4' \bar{\mathbf{q}} - \bar{\mathbf{q}}' \times \bar{\mathbf{q}} \\ q_4' q_4 - \langle \bar{\mathbf{q}}', \bar{\mathbf{q}} \rangle \end{bmatrix}. \quad (2.50)$$

The definition in Equation 2.50 highlights the non-commutativity of the quaternion product provided that the vector product is non-commutative. The multiplication definition given is not the one traditionally used since Hamilton invented the quaternions in 1843 [Shuster 2008]. For the attitude representation point of view, the product operation stated by Equation 2.50 is convenient nevertheless.

The norm of a quaternion \mathbf{q} , denoted as $\|\mathbf{q}\|$, is defined by Equation 2.51.

$$\|\mathbf{q}\| = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2}. \quad (2.51)$$

The complex conjugate \mathbf{q}^* of a quaternion \mathbf{q} is defined as in Equation 2.52.

$$\mathbf{q}^* = \begin{bmatrix} -\bar{\mathbf{q}} \\ q_4 \end{bmatrix}. \quad (2.52)$$

The identity for quaternion addition is $\mathbf{0} = [0 \ 0 \ 0 \ 0]^T$, since $\mathbf{q} + \mathbf{0} = \mathbf{q}$. The identity for quaternion multiplication is $\mathbf{1} = [0 \ 0 \ 0 \ 1]^T$, because $\mathbf{q} \otimes \mathbf{1} = \mathbf{1} \otimes \mathbf{q} = \mathbf{q}$. Therefore, the multiplicative inverse \mathbf{q}^{-1} of \mathbf{q} is such that $\mathbf{q} \otimes \mathbf{q}^{-1} = \mathbf{q}^{-1} \otimes \mathbf{q} = \mathbf{1}$. Using the complex conjugate of a quaternion, \mathbf{q}^{-1} can be written as in Equation 2.53.

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{\|\mathbf{q}\|^2}. \quad (2.53)$$

As mentioned earlier, the quaternions are useful when their components are taken as the Euler symmetric parameters, as shown in Equations 2.54:

$$\mathbf{q} = \begin{bmatrix} \bar{\mathbf{q}} \\ q_4 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\eta} \\ \eta_4 \end{bmatrix}, \quad (2.54)$$

where the equality $\bar{\mathbf{q}} = \boldsymbol{\eta}$ states the relationship between the vector part of the quaternion and the unit rotation vector \mathbf{n}_{br} . This demonstrates the convenience of the notation $\bar{\mathbf{q}} = [q_1 \ q_2 \ q_3]^T$, provided that the quaternion vector part can be interpreted as a vector representation in \mathcal{F}_r and \mathcal{F}_b .

A quaternion defined as in Equation 2.54 is called a quaternion of rotation. The quaternions utilization as containers for the Euler symmetric parameters provides an useful algebraic apparatus for this form of attitude representation. In fact, quaternions are employed

as quaternions of rotations in such extent that in some works they seem to be synonymous. However, a quaternion is an algebraic concept by its own nature, as it was conceived by Hamilton.

Despite of the fact that a quaternion is not necessarily a quaternion of rotation, if its components satisfy a similar constraint of the one given by Equations 2.47, such quaternion can be associated to a rotation. Therefore, the constraint in Equations 2.47 when applied to a quaternion of rotation gives $\|\mathbf{q}\| = 1$. Thus, a quaternion of unit norm, or simply a unit quaternion, can always be associated to a rotation in the same way a proper real orthogonal matrix can. An important consequence of the unit norm constraint is $\mathbf{q}^{-1} = \mathbf{q}^*$, which is analogous to the property $A^{-1} = A^T$ of orthogonal matrices⁷.

There are several ways to write the attitude matrix as a quaternion of rotation function. In order to give the first formulation, the matrices $\Xi(\mathbf{q}) \in \mathbb{R}^{4 \times 3}$ and $\Psi(\mathbf{q}) \in \mathbb{R}^{4 \times 3}$ are defined by Equations 2.55 and 2.56.

$$\Xi(\mathbf{q}) = \begin{bmatrix} q_4 I_3 + [\bar{\mathbf{q}} \times] \\ -\bar{\mathbf{q}}^T \end{bmatrix}, \quad (2.55)$$

$$\Psi(\mathbf{q}) = \begin{bmatrix} q_4 I_3 - [\bar{\mathbf{q}} \times] \\ -\bar{\mathbf{q}}^T \end{bmatrix}. \quad (2.56)$$

The term $[\mathbf{u} \times]$, where $\mathbf{u} = [u_1 \ u_2 \ u_3]^T$, is a skew-symmetric matrix defined as

$$[\mathbf{u} \times] = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}. \quad (2.57)$$

The matrix $[\mathbf{u} \times]$ given by Equation 2.57 can be used to define the vector product as a matrix operation. Let $\mathbf{v} = [v_1 \ v_2 \ v_3]^T$ be a vector of \mathbb{R}^3 , Equation 2.58 demonstrates this vector product property.

$$[\mathbf{u} \times] \mathbf{v} = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} u_2 v_3 - u_3 v_2 \\ u_3 v_1 - u_1 v_3 \\ u_1 v_2 - u_2 v_1 \end{bmatrix} = \mathbf{u} \times \mathbf{v}. \quad (2.58)$$

Given the above definitions, the attitude matrix $A(\mathbf{q})$ written as a quaternion of rotation function can be defined as in Equation 2.59.

$$A(\mathbf{q}) = \Xi^T(\mathbf{q}) \Psi(\mathbf{q}). \quad (2.59)$$

⁷This analogy is even more remarkable with an unitary complex matrix H which has the property $H^{-1} = H^*$, where H^* is its conjugate transpose.

Expanding the terms in Equation 2.59 and applying the matrix product yields a second formulation for $A(\mathbf{q})$, stated by Equation 2.60.

$$A(\mathbf{q}) = (q_4^2 - \|\bar{\mathbf{q}}\|^2)I_3 + 2\overline{\mathbf{q}\mathbf{q}^T} - 2q_4[\bar{\mathbf{q}}\times]. \quad (2.60)$$

Given an attitude matrix A , Equation 2.60 shows the two quaternions \mathbf{q} and $-\mathbf{q}$ can be associated to A . Hence, the quaternions of rotations are not unique, corresponding to a 2-1 attitude parametrization. However, unlike the Euler angles, such non-uniqueness does not mean the Euler symmetric parameters possess singular values because it does not imply discontinuities. Indeed, the four-dimensional quaternion of rotation representation is locally homeomorphic⁸ with the $SO(3)$ group [Stuelpnagel 1964].

The last formulation of $A(\mathbf{q})$ is obtained by writing the Equation 2.60 in terms of the quaternion components and performing the matrix operations. This procedure result is shown in Equation 2.61.

$$A(\mathbf{q}) = \begin{bmatrix} 2(q_1^2 + q_4^2) - 1 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & 2(q_2^2 + q_4^2) - 1 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & 2(q_3^2 + q_4^2) - 1 \end{bmatrix}. \quad (2.61)$$

The description of sequential rotations can be obtained by multiplying the two correspondent quaternions of rotation. Let \mathbf{q}_{*1} and \mathbf{q}_{2*} be two quaternions of rotation. The first represents the rotation which takes a reference frame \mathcal{F}_1 to the frame \mathcal{F}_* ; the second quaternion parametrizes the rotation which takes \mathcal{F}_* to the reference frame \mathcal{F}_2 . The product property of rotation matrices states that $A(\mathbf{q}_{2*})A(\mathbf{q}_{*1})$ is the rotation matrix of the single equivalent rotation of \mathcal{F}_1 into \mathcal{F}_2 . The corresponding rule for the quaternions of rotation is given by Equations 2.62:

$$A(\mathbf{q}_{2*})A(\mathbf{q}_{*1}) = A(\mathbf{q}_{2*} \otimes \mathbf{q}_{*1}) = A(\mathbf{q}_{21}), \quad (2.62)$$

where \mathbf{q}_{21} is the quaternion of rotation which parametrizes the single rotation that takes \mathcal{F}_1 to \mathcal{F}_2 .

The quaternion of rotation is an attitude representation free of singularities and has less parameters than the DCM. As a consequence, the number of operations in several equations – such as Equation 2.62 – is considerably smaller. Also, the microprocessors finite-precision arithmetic tends to violate the unit norm and orthogonality constraints in software implementations of the equations. The first constraint is easier to attend than the second, which is another advantage worth to mention. When compared to the Euler angles, besides the absence of singularities, the quaternions of rotations do not require the computation of trigonometric

⁸In a simplest definition, a continuous mapping between two sets is a homeomorphism if it is invertible and its inverse is also continuous [Do Carmo 2010].

functions, which imposes a high computational burden. Owing to these characteristics, the quaternions of rotation have been the most employed form of attitude representation over the last few decades [Crassidis et al. 2007].

2.3.4 Rodrigues parameters and Gibbs vector

The Rodrigues parameters constitute a form of attitude representation based on the minimum set of three parameters. This means that singularities exist for certain attitudes. This characteristic along with other disadvantages limit their practical utilization. However, the Rodrigues parameters are used as intermediate parametrization in several algorithms, including USQUE filter. Thus, it will be briefly described.

The Rodrigues parameters ρ_1 , ρ_2 , and ρ_3 are simply obtained by dividing the Euler symmetric parameters η_1 , η_2 and η_3 by η_4 , as shown in Equations 2.63.

$$\boldsymbol{\rho} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \end{bmatrix} = \frac{\boldsymbol{\eta}}{\eta_4} = \frac{1}{\eta_4} \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} . \quad (2.63)$$

The vector $\boldsymbol{\rho}$ defined in Equation 2.63 is also called the Gibbs vector. Of course, the Gibbs vector can be written as in Equation 2.64.

$$\boldsymbol{\rho} = \mathbf{n}_{br} \tan\left(\frac{\varphi}{2}\right) . \quad (2.64)$$

The quaternion of rotation $\mathbf{q} = [\boldsymbol{\eta} \quad \eta_4]^T$ can be retrieved from Gibbs vector applying Equations 3.27.

$$\mathbf{q} = \begin{bmatrix} \bar{\mathbf{q}} \\ q_4 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\eta} \\ \eta_4 \end{bmatrix} = \frac{1}{\sqrt{1 + \|\boldsymbol{\rho}\|^2}} \begin{bmatrix} \boldsymbol{\rho} \\ 1 \end{bmatrix} . \quad (2.65)$$

The Gibbs vector is clearly not defined for odd multiples of 180° . This singular point can be moved to the odd multiples of 360° changing Equation 2.63 as in Equation 2.66.

$$\boldsymbol{\rho} = \frac{\bar{\mathbf{q}}}{1 + q_4} . \quad (2.66)$$

The vector $\boldsymbol{\rho}$ components in Equation 2.66 are called the *modified Rodrigues parameters* (MRP). This modification applied to the Gibbs vector was generalized in [Schaub and Junkins 1996], considering the geometric interpretation inherent in MRP. In fact, the modified Rodrigues parameters can be seen as the result of a stereographic projection of a four-dimensional hypersphere with unit radius onto a three-dimensional hyperplane.

This interpretation was extended in the before-mentioned work resulting in Equation 2.67:

$$\mathbf{p} = \frac{\bar{\mathbf{q}}}{a + q_4}, \quad (2.67)$$

where $a \in \mathbb{R}$, $a \leq 1$. The vector \mathbf{p} components are designated as the *generalized Rodrigues parameters* (GRP). The quaternion of rotation can be retrieved from the GRP using Equations 2.68 and 2.69.

$$q_4 = \frac{-a\|\mathbf{p}\|^2 + \sqrt{1 + \|\mathbf{p}\|^2(1 - a^2)}}{1 + \|\mathbf{p}\|^2}, \quad (2.68)$$

$$\bar{\mathbf{q}} = (a + q_4)\mathbf{p}. \quad (2.69)$$

2.3.5 Rotation vector

The last form of attitude parametrization described in this manuscript is the rotation vector, which is a three-parameter representation. It has not been much applied except as an intermediate quantity used in conversions between other forms of attitude representation. Indeed, in this work the rotation vector is employed to carry the attitude data between software layers of the ADCV system. Equation 2.70 gives the rotation vector φ definition:

$$\varphi = \mathbf{n}_{br}\varphi, \quad (2.70)$$

where φ is in radians. Of course, \mathbf{n}_{br} is the unit rotation vector which defines the fixed axis of rotation, and φ is the total rotation angle. Limiting the range of φ in order that the parametrization is global, discontinuities occurs as expected. For instance, if $-\pi < \varphi \leq \pi$, the rotation vector direction changes abruptly when a small rotation from $(\pi - \epsilon)$ to $(-\pi + \epsilon)$ is taken, where $0 < \epsilon \leq 1$.

2.3.6 Attitude error

This topic is dedicated to the important problem of quantifying the attitude error. For attitude determination algorithms, an evaluation of the attitude error is important to increase or decrease the attitude estimate in the next time step. In closed-loop controllers the attitude error is used as an input to the control action.

The desired value for the attitude in attitude determination problems is the real value of the body orientation. Thus, the attitude error can be seen as being the rotation that takes the real attitude to the attitude estimate. Let \mathcal{F}_r and \mathcal{F}_b be the reference-system and the body-system. Also, let $\hat{\mathcal{F}}_b$ be the estimated reference frame for the body-system. Hence, the

attitude error can be defined as the rotation which takes \mathcal{F}_b to $\hat{\mathcal{F}}_b$. When the three reference frames are considered, there is a sequence of two rotations. Firstly, \mathcal{F}_r is rotated onto \mathcal{F}_b , which is then rotated onto $\hat{\mathcal{F}}_b$. The first rotation can be represented by a DCM A_{br} , the second rotation, which defines the attitude error, can be represented by the attitude error matrix δA . Using the product property for rotation sequences yields Equation 2.71:

$$\hat{A}_{br} = (\delta A)A_{br} , \quad (2.71)$$

where \hat{A}_{br} parametrizes the single rotation that takes \mathcal{F}_r directly to $\hat{\mathcal{F}}_b$, which is the attitude estimate for the body. Thus, the attitude error matrix can be given by Equation 2.72.

$$(\delta A) = \hat{A}_{br}A_{br}^{-1} = \hat{A}_{br}A_{br}^T . \quad (2.72)$$

Using the corresponding quaternions of rotation $\hat{\mathbf{q}}_{br}$, $\delta\mathbf{q}$, and \mathbf{q}_{br} , the property shown in Equation 2.62 results in Equations 2.73.

$$\hat{A}_{br} = (\delta A)A_{br} \Leftrightarrow \hat{\mathbf{q}}_{br} = \delta\mathbf{q} \otimes \mathbf{q}_{br} . \quad (2.73)$$

Therefore, the attitude estimate is related to the real attitude by a multiplicative error. Using the quaternion inverse, the quaternion product in Equation 2.73 yields

$$\delta\mathbf{q} = \hat{\mathbf{q}}_{br} \otimes \mathbf{q}_{br}^{-1} = \hat{\mathbf{q}}_{br} \otimes \mathbf{q}_{br}^* , \quad (2.74)$$

where the correspondence $\mathbf{q}^{-1} = \mathbf{q}^*$ was used in Equation 2.74.

Equation 2.74 is commonly employed in attitude determination algorithms such as EKF [Markley 2003] and USQUE [Crassidis 2003]. Furthermore, it must be noticed that when there is no estimation error, $\delta A = I_3$ and $\delta\mathbf{q} = [0 \ 0 \ 0 \ 1]^T = \mathbf{1}$.

2.3.7 Attitude kinematics

This section concludes with the relationship between the attitude parameters time rates and the angular velocities measured in the body-system. These quantities can be measured by proprioceptive sensors mounted in the body such as gyros. The fact that is possible to estimate how the attitude parameters change in time using the body angular velocities is quite intuitive. However, since the attitude is an exteroceptive quantity by its own, is expected that the relationship between the attitude parameters time rates and the angular velocities depends also on the current attitude. This is the case indeed, as shown by Equation 2.75 which expresses the relationship between the quaternion of rotation time derivative and

the vector of angular velocities $\boldsymbol{\omega}_b$ measured in the body-system.

$$\mathbf{q}'(t) = \frac{d\mathbf{q}(t)}{dt} = \frac{1}{2}\Xi(\mathbf{q}(t))\boldsymbol{\omega}_b(t) . \quad (2.75)$$

In Equation 2.75, the terms \mathbf{q}' and $\boldsymbol{\omega}_b$ must be seen as column matrices, being $\Xi(\mathbf{q})$ a 4×3 matrix. Equation 2.75 demonstrates that $\mathbf{q}'(t)$ can be calculated using the angular velocities measurements, being also dependent of the current attitude given by $\mathbf{q}(t)$ through $\Xi(\mathbf{q}(t))$.

The roles of \mathbf{q} and $\boldsymbol{\omega}_b$ in Equation 2.75 can be swapped using the matrix $\Omega(\boldsymbol{\omega}_b) \in \mathbb{R}^{4 \times 4}$ as in Equation 2.76.

$$\mathbf{q}'(t) = \frac{1}{2}\Omega(\boldsymbol{\omega}_b(t))\mathbf{q}(t) = \frac{1}{2} \begin{bmatrix} -[\boldsymbol{\omega}_b \times] & \boldsymbol{\omega}_b \\ -\boldsymbol{\omega}_b^T & 0 \end{bmatrix} \mathbf{q}(t) . \quad (2.76)$$

Regarding $\boldsymbol{\omega}_b$ as the vector part of a quaternion with scalar part equal to 0, Equations 2.75 and 2.76 can be rewritten in terms of the quaternion product as given by Equation 2.77.

$$\mathbf{q}'(t) = \frac{1}{2} \begin{bmatrix} \boldsymbol{\omega}_b \\ 0 \end{bmatrix} \otimes \mathbf{q}(t) . \quad (2.77)$$

The quaternion of rotation time derivative equations are essential to formulate dynamic models of rotating objects. The previous kinematics relationships can be derived by calculating the limit $\delta\mathbf{q}/\delta t \rightarrow \mathbf{q}'(t)$ with $\delta t \rightarrow 0$, where $\delta\mathbf{q}$ is the error quaternion described earlier. These derivations and further details about the kinematics equations of different attitude representations can be found in [Shuster 1993].

Chapter 3

Attitude determination methods

In Chapter 2 different forms of attitude representations were presented. This chapter describes two classes of methods employed to determine the attitude of a body using measurements provided by sensors. The attitude determination algorithms normally dependent on the attitude representation chosen. The TRIAD method presented in Section 3.1 utilizes the DCM parametrization, whereas the USQUE method presented in Section 3.2 is based on the quaternions of rotation.

3.1 Deterministic methods for attitude determination

As reported in Chapter 1, the attitude determination methods can be classified as deterministic and optimal. The deterministic methods employ a minimum number of measurements to determine the attitude. The most important consequence of the deterministic methods approaches is the absence of a stochastic modelling in the algorithms derivation. If the sensors are not precise in face of the environment noise, the deterministic methods may not be effective. Still, when such conditions are not encountered they can be useful, specially due to the soft computational effort these methods often require. Moreover, a previous treatment of the measurements, such as the application of simple low-pass filters, along with a well-executed calibration, can improve the deterministic algorithms results.

3.1.1 TRIAD algorithm

The TRIAD algorithm is a general method proposed in the short work [Black 1964] in the early days of the space race. It requires two sets of three mutually orthogonal unit vectors. The algorithm name¹ comes from the fact that each of the vectors set is called a triad. One of the sets is obtained using two linearly independent abstract vectors observations taken in the

¹The name TRIAD apparently was given to this algorithm years after it was proposed in 1964. In [Wertz 1978] it is simply referred as an algebraic method to solve the three-axis attitude determination problem.

reference-system. The other triad is associated to the correspondent vector representations measured in the body-system. Thus, let $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ be two abstract vectors, being $\boldsymbol{\mu}_r$ and $\boldsymbol{\nu}_r$ the correspondent vectors measured in the reference-system, and $\boldsymbol{\mu}_b$ and $\boldsymbol{\nu}_b$ the same abstract vectors represented in the body-system. Equations 3.1 shows how the triad in the reference-system can be stated.

$$\boldsymbol{u}_r = \frac{\boldsymbol{\mu}_r}{\|\boldsymbol{\mu}_r\|}, \quad \boldsymbol{v}_r = \frac{\boldsymbol{\mu}_r \times \boldsymbol{\nu}_r}{\|\boldsymbol{\mu}_r \times \boldsymbol{\nu}_r\|}, \quad \boldsymbol{w}_r = \boldsymbol{u}_r \times \boldsymbol{v}_r . \quad (3.1)$$

In Equation 3.1, $\{\boldsymbol{u}_r, \boldsymbol{v}_r, \boldsymbol{w}_r\}$ is the triad taken in the reference-system. Clearly, \boldsymbol{u}_r is orthogonal to $\boldsymbol{\mu}_r \times \boldsymbol{\nu}_r$; in the same way, \boldsymbol{w}_r is orthogonal to the unit vector $\boldsymbol{u}_r \times \boldsymbol{v}_r$. The correspondent triad expressed in the body-system is analogously defined, as in Equation 3.2.

$$\boldsymbol{u}_b = \frac{\boldsymbol{\mu}_b}{\|\boldsymbol{\mu}_b\|}, \quad \boldsymbol{v}_b = \frac{\boldsymbol{\mu}_b \times \boldsymbol{\nu}_b}{\|\boldsymbol{\mu}_b \times \boldsymbol{\nu}_b\|}, \quad \boldsymbol{w}_b = \boldsymbol{u}_b \times \boldsymbol{v}_b . \quad (3.2)$$

Let A be the attitude matrix associated to the body-system and the reference-system. Equations 3.3 demonstrate the relationship between the two triads.

$$\boldsymbol{u}_b = A\boldsymbol{u}_r, \quad \boldsymbol{v}_b = A\boldsymbol{v}_r, \quad \boldsymbol{w}_b = A\boldsymbol{w}_r . \quad (3.3)$$

Equations 3.3 can be written as the single Equation 3.4 by defining the two block matrices $M_r = [\boldsymbol{u}_r | \boldsymbol{v}_r | \boldsymbol{w}_r]$ and $M_b = [\boldsymbol{u}_b | \boldsymbol{v}_b | \boldsymbol{w}_b]$.

$$M_b = AM_r . \quad (3.4)$$

Thus, the DCM can be calculated using Equations 3.5.

$$A = M_b M_r^{-1} = M_b M_r^T . \quad (3.5)$$

The roles of the $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ vector representations are asymmetrical in the triads definitions. The observations of $\boldsymbol{\mu}$ are more relevant than the $\boldsymbol{\nu}$ ones. As pointed out in [Wertz 1978], this suggests that the vector representations of $\boldsymbol{\mu}$ should be taken as the more accurate measurements. On the other hand, if they have similar accuracy, the modifications given by Equations 3.6 can be utilized.

$$\boldsymbol{u}_b = \frac{\frac{\boldsymbol{\mu}_b}{\|\boldsymbol{\mu}_b\|} + \frac{\boldsymbol{\nu}_b}{\|\boldsymbol{\nu}_b\|}}{\left\| \frac{\boldsymbol{\mu}_b}{\|\boldsymbol{\mu}_b\|} + \frac{\boldsymbol{\nu}_b}{\|\boldsymbol{\nu}_b\|} \right\|}, \quad \boldsymbol{v}_b = \frac{\frac{\boldsymbol{\mu}_b}{\|\boldsymbol{\mu}_b\|} - \frac{\boldsymbol{\nu}_b}{\|\boldsymbol{\nu}_b\|}}{\left\| \frac{\boldsymbol{\mu}_b}{\|\boldsymbol{\mu}_b\|} - \frac{\boldsymbol{\nu}_b}{\|\boldsymbol{\nu}_b\|} \right\|}, \quad \boldsymbol{w}_b = \boldsymbol{u}_b \times \boldsymbol{v}_b . \quad (3.6)$$

The reference-system triad can be defined using the same procedure. However, the application of Equation 3.6 is more critical for the body-system vectors, which are measured by sensors mounted in the body. The vectors observed in the reference-system are normally pro-

vided by databases built with high-precision instruments and models of the Earth's magnetic and gravitational fields, astronomical records, etc.

The TRIAD method requires two sets of a minimum of two vectors. The attitude determination is a problem of three degrees of freedom nonetheless. In a first look the TRIAD method should not be considered deterministic since it requires a greater number of measures. As explained in [Shuster 2004] and [Wertz 1978], the TRIAD method effectively uses only three values to determine A provided that some irrelevant information is discarded in the triads construction and in Equation 3.5. Hence, the TRIAD algorithm is an authentic deterministic method.

In order to conclude this section, the similarity between Equations 3.5 and Equations 2.43 is pointed out. Nevertheless, even though the relationship $A = B_s^T R_s$ is meaningful from a theoretical point of view it is not practical. Indeed, it requires a third basis in addition to those associated to the reference-system and the body-system, leading to the problem of how the reference vectors can be expressed in it. That is, the first attitude determination problem originates other two, and the solution would be postponed indefinitely. Equations 3.5 do not demand a third basis, but it seems that they have the disadvantage of requiring sensors measurements. However, such disadvantage does not exist, once in practice the reference systems definitions – implicitly used in the product $B_s^T R_s$ – also depend on several instruments and observations. Chapter 5 describes the experimental apparatus utilized to establish a local reference-system at LAICA facility. The same chapter gives a description of the TRIAD software implementation realized in the present work.

3.2 Stochastic methods for attitude estimation

In Chapter 1 the optimal methods for attitude determination were defined as those in which a number of observations greater than the minimum necessary is used. This allows the variables statistical parameters, such as mean and standard deviation, to be estimated. These parameters can thus be used as metrics to minimize some cost function in order that the optimality is achieved. In view of this definition, the optimal methods can also be classified as stochastic methods in the sense that the variables are treated as random variables. Of course, this concept could be applied to any practical problem, thus the exposition given in the following topics is quite general. In the end of this section the USQUE filter will be described, a stochastic method specifically designed for the attitude determination problem.

3.2.1 Random variables and stochastic processes

Stochastic processes treat physical quantities as random variables. The term *random* does not mean the variables behaviour is unknowable or unpredictable. Randomness implies that the values assumed by physical quantities can not be known with absolute certainty but only

by means of probabilities. Nevertheless, in Engineering problems a good knowledge of the variables probabilistic behaviour may be sufficient to build functional, robust solutions, in spite of the uncertainties.

The probabilistic behaviour of a continuous random variable is modelled by a *probability density function* (PDF) p_X , defined by Equations 3.7.

$$\begin{aligned} p_X(x) &\geq 0; \\ \int_{-\infty}^{\infty} p_X(\tau) d\tau &= 1; \\ P(a < X \leq b) &= \int_a^b p_X(\tau) d\tau. \end{aligned} \tag{3.7}$$

In Equations 3.7 $P(x \in C)$ is the probability of the variable X be in $C \subset \mathbb{R}$, where the set C is considered left-open².

A PDF function is characterized by its statistical moments. The first moment \bar{x} of a random variable X is defined as $\bar{x} = E[X]$, where E designate the statistical expectation, or mean, of X . The second central moment of X is defined as $E[(X - \bar{x})^2]$ and corresponds to the variance σ^2 of X , being σ its standard deviation. A Gaussian or normal random variable X is modelled by the PDF defined in Equation 3.8.

$$p_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x - \bar{x})^2}{2\sigma^2}\right]. \tag{3.8}$$

The parameters \bar{x} and σ in Equation 3.8 are the first moment and the second central moment of X , respectively. Therefore, a Gaussian random variable is completely characterized by these two moments. The notation $X \sim N(\bar{x}, \sigma^2)$ is commonly used to indicate that X is a Gaussian random variable with mean \bar{x} and variance σ^2 .

Stochastic algorithms often consider the quantities are Gaussian. Such assumption is justified in a physical and mathematical point of view by the Central Limit Theorem³. Also, a Gaussian random variable requires only two moments to be completely characterized and this is typically the maximum that can be estimated in practice [Maybeck 1979]. Hence, a stochastic method designed to estimate a random variable $X \sim N(\bar{x}, \sigma^2)$ actually seeks estimates for \bar{x} and σ^2 . With these two parameters the probability of X be in a certain interval can be calculated. For instance, for a Gaussian random variable, $P(\bar{x} - 3\sigma < x \leq \bar{x} + 3\sigma) \approx 99.73\%$.

The interval $(\bar{x} - 3\sigma, \bar{x} + 3\sigma]$ is commonly used to evaluate the convergence of an algo-

²The set C is actually a Borel subset of \mathbb{R} [Hajek 2015].

³The description and derivation of the Central Limit Theorem can be found in [Hajek 2015] and several other books concerning probability theory.

rithm. Thus, if the real mean fell into this (estimated) interval the algorithm would converge⁴. However, the convergence is not the unique requisite. An optimal method must obtain an estimate of the mean with a minimum error, which implies the variance of the estimated variable is also minimum. For example, if an algorithm yields $\bar{x} = 10^\circ$ and $\sigma = 50^\circ$ as estimates for a body single-axis attitude, the fact that $P(10^\circ - 150^\circ < x \leq 10^\circ + 150^\circ) \approx 99.74\%$ is useless. On the other hand, if the values $\bar{x} = 10^\circ$ and $\sigma = 0.5^\circ$ are obtained, the probability $P(10^\circ - 1.5^\circ < x \leq 10^\circ + 1.5^\circ) \approx 99.74\%$ becomes meaningful.

A random vector \mathbf{X} is composed by multiple random variables. A n -size random vector is Gaussian if its PDF is in the form given in Equation 3.9.

$$p_X(\mathbf{x}) = \frac{1}{(2\pi)^{n/2} \sqrt{\det(C_X)}} \exp \left[\frac{-1}{2} (\mathbf{x} - \bar{\mathbf{x}}) C_X^{-1} (\mathbf{x} - \bar{\mathbf{x}})^T \right]. \quad (3.9)$$

The mean vector $\bar{\mathbf{x}}$ is defined in Equations 3.10. The $n \times n$ matrix C_X is the covariance matrix of X defined in Equations 3.11.

$$\bar{\mathbf{x}} = E[\mathbf{X}] = [E[X_1] \quad E[X_2] \quad \dots \quad E[X_n]]^T. \quad (3.10)$$

$$C_X = E[(\mathbf{X} - \bar{\mathbf{x}})(\mathbf{X} - \bar{\mathbf{x}})^T] = \begin{bmatrix} E[(X_1 - \bar{x}_1)^2] & \dots & E[(X_1 - \bar{x}_1)(X_N - \bar{x}_N)] \\ E[(X_2 - \bar{x}_2)(X_1 - \bar{x}_1)] & \dots & E[(X_2 - \bar{x}_2)(X_N - \bar{x}_N)] \\ \vdots & \dots & \vdots \\ E[(X_N - \bar{x}_N)(X_1 - \bar{x}_1)] & \dots & E[(X_N - \bar{x}_N)^2] \end{bmatrix}. \quad (3.11)$$

Thus, the covariance matrix can be written as in Equation 3.12:

$$C_X = \begin{bmatrix} \sigma_1^2 & \dots & \sigma_{1N} \\ \sigma_{21} & \dots & \sigma_{2N} \\ \vdots & \ddots & \vdots \\ \sigma_{N1} & \dots & \sigma_N^2 \end{bmatrix}, \quad (3.12)$$

where $\sigma_{ij} = \sigma_{ji}$. The covariance matrix is not only symmetric, but also positive semidefinite. If two components of \mathbf{X} are independent of each other, $\sigma_{ij} = 0$. Thus, if the components of \mathbf{X} are mutually independent, C_X is diagonal. Equations 3.10 and 3.12 show that a random vector can be used to estimate the mean and the variance of multiple random variables at the same time. This is the approach employed by state space estimation algorithms such as the KF.

⁴An interval of the form $(\bar{x} - \lambda, \bar{x} + \lambda]$ is called a confidence region for \bar{x} [Bar-Shalom et al. 2001].

A stochastic process consists in a random vector $\mathbf{X}(t)$ which changes with time. Therefore, at each time t_k , $\mathbf{X}(t_k)$ is a different random vector. Physical quantities such as position, velocity, and the attitude parameters can be modelled as stochastic processes. Hence, a stochastic method applied to estimate these variables must seek their statistical moments at each time. If at any given instant t_k the quantities are Gaussian random vectors, the algorithm must estimate the mean $\bar{\mathbf{x}}(t)$ and covariance $C_{\mathbf{X}}(t)$ for each t_k .

3.2.2 Kalman filter

The Kalman filter is an optimal method for stochastic estimation. It is optimal in the sense that it minimizes the variances of the estimation errors. In fact, the KF is the best linear estimator of such minimization problem as shown in [Simon 2006]. Also, the KF is a state space estimator such that the variables to be estimated constitute the state space vector, which is modelled as a random vector. The state space model adopted by the KF is defined as in Equations 3.13 and 3.14.

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k, t_k) , \quad (3.13)$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k, t_k) . \quad (3.14)$$

The vector \mathbf{x}_k in Equations 3.13 and 3.14 is the state space vector, where $\mathbf{x}_k = \mathbf{x}(t_k)$ and the variables are treated as discrete-time quantities ($k \in \mathbb{Z}$). The vector \mathbf{u}_k is the input of the system and \mathbf{y}_k is the output. The random vectors \mathbf{w}_k and \mathbf{v}_k correspond to the process and the measurement noises, respectively, both white, zero-mean and uncorrelated. The function \mathbf{f} relate quantities of different instants, it therefore corresponds to a dynamical model for \mathbf{x}_k . The function \mathbf{h} establishes the relationship between \mathbf{x}_k and the output \mathbf{y}_k , which usually comprises measurements from sensors. In its original formulation, the KF consider \mathbf{f} and \mathbf{h} linear. However, since the attitude determination is an inherently nonlinear problem, \mathbf{f} and \mathbf{h} will be considered nonlinear in this work.

The noises \mathbf{w}_k and \mathbf{v}_k can be considered additive in certain applications. Thus, the state space equations are written as in Equations 3.15 and 3.16.

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, t_k) + \mathbf{w}_k , \quad (3.15)$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, t_k) + \mathbf{v}_k . \quad (3.16)$$

The KF aims to estimate the mean and covariance of the conditional density function (CDF) of \mathbf{x}_k , conditioned on the measurements taken at consecutive instants $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_m$

($m \leq k$). This CDF is designated as $p_{\mathbf{x}_k|\mathbf{y}_1, \dots, \mathbf{y}_m}$. The input \mathbf{u}_k is not necessarily a random vector, if it is however, the CDF is also conditioned on \mathbf{u}_k . The form of this CDF is usually not known, even if \mathbf{x}_k is Gaussian. The propagated random vector \mathbf{x}_{k+1} will be also Gaussian only if \mathbf{f} is linear. As explained earlier, the random vectors are assumed to be Gaussian nevertheless, thus the problem becomes tractable and only the mean and the covariance are needed.

The estimation procedure of the KF is divided in a prediction and a correction parts. In the former, the algorithm calculates an estimate $\hat{\mathbf{x}}_k^-$ for the mean of $p_{\mathbf{x}_k|\mathbf{y}_1, \dots, \mathbf{y}_{k-1}}$, as in Equation 3.17.

$$\hat{\mathbf{x}}_k^- := E[\mathbf{x}_k | \mathbf{y}_1, \dots, \mathbf{y}_{k-1}]. \quad (3.17)$$

The symbol $:=$ indicates that $\hat{\mathbf{x}}_k^-$ is not the true expectation but an estimate for it. The term $\hat{\mathbf{x}}_k^-$ is an *a priori* estimate and in this part the KF work as a predictor ($m < k$). In the correction part, the algorithm calculates an *a posteriori* estimate $\hat{\mathbf{x}}_k^+$ for the mean of $p_{\mathbf{x}_k|\mathbf{y}_1, \dots, \mathbf{y}_k}$, as in Equation 3.18.

$$\hat{\mathbf{x}}_k^+ := E[\mathbf{x}_k | \mathbf{y}_1, \dots, \mathbf{y}_k]. \quad (3.18)$$

In Equation 3.18 the KF work as a filter ($m = k$). The covariances P_k^- and P_k^+ of the estimation errors of $\hat{\mathbf{x}}_k^-$ and $\hat{\mathbf{x}}_k^+$ are defined by Equations 3.19 and 3.20.

$$P_k^- := E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T], \quad (3.19)$$

$$P_k^+ := E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^+)(\mathbf{x}_k - \hat{\mathbf{x}}_k^+)^T]. \quad (3.20)$$

Equations 3.19 and 3.20 show that the covariance matrices P_k^- and P_k^+ are also estimates for the covariance matrices of $p_{\mathbf{x}_k|\mathbf{y}_1, \dots, \mathbf{y}_{k-1}}$ and $p_{\mathbf{x}_k|\mathbf{y}_1, \dots, \mathbf{y}_k}$. For the linear case $P_k^+ \leq P_k^-$ ⁵, thus the filter produces a better estimate by incorporating the current measurements \mathbf{y}_k . Additionally, the filter is optimal because it minimizes $Tr(P_k)$. For the nonlinear cases, these properties are approximately true. The matrix P_k^- is calculated in the prediction part of the KF, whereas P_k^+ is calculated in the correction part. Hence, the algorithm outputs are $\hat{\mathbf{x}}_k$ and P_k . The procedures stated by the KF will be describe in details in the context of the USQUE filter.

⁵The inequality $A \leq B$ means $B - A \geq \mathbf{0}$, thus the matrix $(B - A)$ is positive semidefinite.

3.2.3 Unscented transformation

The extended Kalman filter was proposed to solve nonlinear spacecraft navigation problems in the late 1960s, only a few years after Rudolf E. Kálmán (1930-2016) have established the algorithm which bears his name [Simon 2006]. The unscented Kalman filter is a completely different nonlinear approach proposed in the mid 1990s by S. J. Julier, J. K. Uhlmann and collaborators, who presented the new method in several papers and reports [Julier and Uhlmann 1994, Julier et al. 1995, Julier and Uhlmann 1997, Julier et al. 2000, Julier and Uhlmann 2004]. This topic focus on the UKF on which the USQUE filter is based.

The UKF uses the unscented transformation (UT) to estimate $\hat{\boldsymbol{x}}_k$ and P_k . The EKF analytically linearise the functions \boldsymbol{f} and \boldsymbol{h} to approximate the nonlinear problem to a linear one. On the other hand, the fundamental idea behind the UT is such that should be easier to statistically approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function [Julier and Uhlmann 1994]. Given several realizations of \boldsymbol{x} and \boldsymbol{y} , estimates for $\bar{\boldsymbol{x}}$ and σ^2 could be obtained using the well-know formulas of the *sample mean* and the *sample covariance* [Bar-Shalom et al. 2001]. However, dealing with a large number of samples at each time step is not practical [Aguirre 2015]. Thus, the UT provides a small set of representative samples, called sigma points, such that their sample mean approximates the mean of the original Gaussian PDF up to the third order – better than EKF – whereas their sample covariance matches the original covariance up to the second order. This is the same approximation obtained by EKF, but the UT has the advantage that it does not require the computation of the \boldsymbol{f} and \boldsymbol{h} Jacobian functions.

Given a Gaussian random vector \boldsymbol{x} with known mean $\bar{\boldsymbol{x}}$ and covariance P , the UT defines the sigma points $\mathcal{X}[i]$ defined by Equations 3.21:

$$\begin{aligned}\mathcal{X}[0] &= \bar{\boldsymbol{x}} ; \\ \mathcal{X}[i] &= \bar{\boldsymbol{x}} + \left[\sqrt{(n + \lambda)P} \right]_{*i}, \quad i = 1, 2, \dots, n ; \\ \mathcal{X}[n + i] &= \bar{\boldsymbol{x}} - \left[\sqrt{(n + \lambda)P} \right]_{*i}, \quad i = 1, 2, \dots, n ;\end{aligned}\tag{3.21}$$

where n is the size of $\bar{\boldsymbol{x}}$ and λ is a scalar parameter. The term \sqrt{A} denotes any matrix such that $(\sqrt{A})(\sqrt{A})^T = A$. Thus, the term $[\sqrt{A}]_{*i}$ designates the i^{th} -column of such matrix⁶. The Cholesky factorization is often applied to obtain \sqrt{A} .

Let \boldsymbol{z} be a random vector such that $\boldsymbol{z} = \boldsymbol{g}(\boldsymbol{x})$. The $(2n + 1)$ sigma points $\mathcal{X}[i]$ can be transformed into new sigma points $\mathcal{Z}[i]$ by the function \boldsymbol{g} using Equation 3.22.

$$\mathcal{Z}[i] = \boldsymbol{g}(\mathcal{X}[i]) .\tag{3.22}$$

⁶If the definition of \sqrt{A} is such that $(\sqrt{A})^T(\sqrt{A}) = A$, its rows must be taken to calculate the sigma points.

The mean and covariance of \mathbf{z} is subsequently approximated by $\hat{\mathbf{z}}$ and $P_{\mathbf{z}}$, which are respectively a weighted sample mean and a weighted sample covariance of the transformed sigma points. These estimates are shown in Equations 3.23 and 3.24:

$$\hat{\mathbf{z}} = \sum_{i=0}^{2n} W_i \boldsymbol{\chi}[i], \quad (3.23)$$

$$P_{\mathbf{z}} = \sum_{i=0}^{2n} W_i (\boldsymbol{\chi}[i] - \hat{\mathbf{z}})(\boldsymbol{\chi}[i] - \hat{\mathbf{z}})^T, \quad (3.24)$$

where the weights W_i are defined by Equations 3.25.

$$\begin{aligned} W_0 &= \frac{\lambda}{n + \lambda}, \\ W_i &= \frac{1}{2(n + \lambda)}. \end{aligned} \quad (3.25)$$

The scalar λ allows the filter to be more precisely adjusted. This constant can be used to reduce the effect of high order moments, which are discarded by the UT approximation [Julier and Uhlmann 1997]. The application of the UT to the Kalman filter will be demonstrated in the USQUE filter description.

3.2.4 Unscented quaternion estimator

The unscented quaternion estimator was proposed in [Crassidis 2003]. The estimator is an application of the UKF principles to the attitude determination problem. However, the direct utilization of the UT to the quaternion of rotation of a body violates the unit norm constraint, due to the summations of the weighted sample mean and the weighted sample covariance. In view of this fact, in USQUE the quaternion of rotation is not directly estimated but the attitude error, which is represented by a generalized Gibbs error vector. This form of attitude parametrization is not subjected to any similar constraint, thus the UT and the weighted sample formulas can be applied. On the other hand, the propagation in time – that is, the filter prediction part – is realized using the kinematics equations for the quaternions of rotation. The output equation is also applied using the quaternion of rotation. Consequently, conversions between these different forms of representation is required. When the overall procedure is considered, the USQUE algorithm works as if it was two filters such that each filter estimates different quantities; one estimates the Gibbs error vector, whereas the other estimates the quaternion of rotation. Hereafter, the quaternion of rotation will be simply called quaternion, as long as it is exclusively applied as a form of attitude representation. The USQUE procedure is depicted in Figure 3.1. The left side shows the Gibbs error vector evaluation, whereas the right side shows the quaternion evaluation.

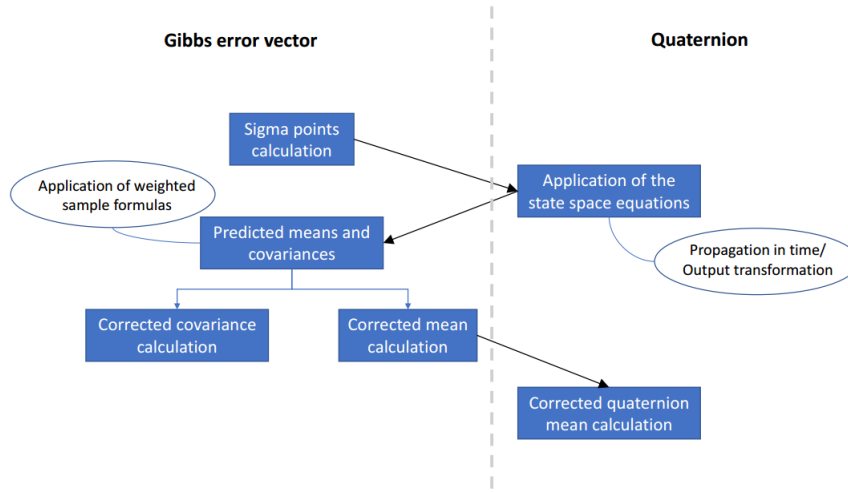


Figure 3.1: High-level view of the USQUE algorithm. The black arrows denote conversions between the attitude representations.

A qualitative description of the process shown in Figure 3.1 is given below, where the USQUE algorithm is compared to UKF.

1. The common UKF starts with the sigma points calculation. In USQUE this part is applied to the Gibbs error vector.
2. The UKF proceeds with the nonlinear transformations associated to the functions f and h from the state space model, resulting in a set of transformed sigma-point state space vectors and sigma-point outputs. In USQUE the nonlinear transformations are applied to quaternions, thus a conversion between the attitude representations is required, yielding a set of sigma-point quaternions and sigma-point outputs.
3. The UKF applies the weighted sample formulas to the transformed sigma points to estimate the predicted statistical moments of the state space vector and the output vector (*a priori* estimates). In USQUE these formulas can be applied to the Gibbs error vector only, therefore a new conversion is needed in order that the transformed sigma-point quaternions yields transformed sigma points for the Gibbs error vector. The before-mentioned statistical moments are subsequently calculated for the Gibbs error vector.
4. The UKF uses the statistical moments to compute the Kalman gain, which is employed to update – resulting in *a posteriori* estimates – the mean and covariance predicted in the correction part of the filter; this is the end of the UKF execution at that sample time. In USQUE the same procedure is applied to the predicted mean and covariance of the Gibbs error vector, but the filter proceeds with a last conversion to calculate the corrected quaternion; the execution for that sample time is thus finished.

After the above description, the details can be given. The USQUE filter takes a generalized Gibbs error vector δp as part of the state space vector x . Given the quaternion

$\mathbf{q} = [\bar{\mathbf{q}}^T \quad q_4]^T$ associated to a body attitude, the purposed definition for $\delta\mathbf{p}$ is stated in terms of the error quaternion $\delta\mathbf{q} = [\delta\bar{\mathbf{q}}^T \quad \delta q_4]^T$. The relationship between the two parametrization errors is shown in Equation 3.26:

$$\delta\mathbf{p} = c \cdot \frac{\delta\bar{\mathbf{q}}}{a + \delta q_4}, \quad (3.26)$$

where a is the stereographic projection parameter of the GRP, presented in Section 2.3, and c is a scale factor, which is allowed provided that the $\delta\mathbf{p}$ norm is not constrained. The value chosen for c is $c = 2(a + 1)$, thus $\|\delta\mathbf{p}\|$ is equal to φ for small angles, where φ is the total angle of rotation. The error quaternion $\delta\mathbf{q}$ is calculated from $\delta\mathbf{p}$ using Equations 3.27.

$$\begin{aligned} \delta q_4 &= \frac{-a\|\delta\mathbf{p}\|^2 + c\sqrt{c^2 + \|\delta\mathbf{p}\|^2(1 - a^2)}}{c^2 + \|\delta\mathbf{p}\|^2}, \\ \delta\bar{\mathbf{q}} &= \frac{(a + \delta q_4)\delta\mathbf{p}}{c}. \end{aligned} \quad (3.27)$$

In USQUE the input \mathbf{u} of the state space model corresponds to the vector of proprioceptive angular rates $\boldsymbol{\omega}$. These quantities are measured by a three-axis rate integrating gyro. The random behaviour of this sensor measurements is modelled by Equations 3.28.

$$\begin{aligned} \boldsymbol{\omega}(t) &= \boldsymbol{\omega}_0(t) + \boldsymbol{\beta}(t) + \boldsymbol{\xi}_v(t), \\ \boldsymbol{\beta}'(t) &= \boldsymbol{\xi}_u(t), \end{aligned} \quad (3.28)$$

where $\boldsymbol{\omega}_0(t)$ contains the real angular rates of the body and $\boldsymbol{\omega}(t)$ comprises the quantities measured by the sensor. In addition, $\boldsymbol{\beta}(t)$ is the sensor bias, whereas $\boldsymbol{\xi}_v(t)$ and $\boldsymbol{\xi}_u(t)$ are white, independent and Gaussian process noises, such that

$$\begin{aligned} E[\boldsymbol{\xi}_v(t)\boldsymbol{\xi}_v^T(\tau)] &= I_3\sigma_v^2\delta(t - \tau), \\ E[\boldsymbol{\xi}_u(t)\boldsymbol{\xi}_u^T(\tau)] &= I_3\sigma_u^2\delta(t - \tau), \end{aligned} \quad (3.29)$$

where $\delta(t - \tau)$ is the Dirac delta function. From the model above, the *a posteriori* estimate $\hat{\boldsymbol{\omega}}_k^+$ for the angular velocities and the *a priori* estimate $\hat{\boldsymbol{\beta}}_k^-$ for the bias can be computed using Equations 3.30.

$$\begin{aligned} \hat{\boldsymbol{\omega}}_k^+ &= \boldsymbol{\omega}_k - \hat{\boldsymbol{\beta}}_k^+, \\ \hat{\boldsymbol{\beta}}_{k+1}^- &= \hat{\boldsymbol{\beta}}_k^+. \end{aligned} \quad (3.30)$$

In Equations 3.30 ω_k is the measurement given by the gyro sensor. Thus, Equations 3.30 perform an online calibration of the gyro. The bias vector $\hat{\beta}_k^+$ is estimated by USQUE in order that the angular rates can be accurately incorporated to the model. Hence, the state space vector \hat{x}_k is defined by Equation 3.31.

$$\mathbf{x}_k = \begin{bmatrix} \delta p_k \\ \beta_k \end{bmatrix} = \mathbf{x}_k[0]. \quad (3.31)$$

Besides $\mathbf{x}[0]$, the other sigma points $\mathbf{x}[i]$ are partitioned into two sigma-point vectors as shown in Equation 3.32.

$$\mathbf{x}_k[i] = \begin{bmatrix} \mathbf{x}_k^{\delta p}[i] \\ \mathbf{x}_k^{\beta}[i] \end{bmatrix}. \quad (3.32)$$

The quaternion estimate \hat{q}_k^+ is propagated in time – yielding a predicted estimate – using Equations 3.33 and 3.34. Denoting the gyro sampling interval as Δt , these equations are the discrete-time equivalent of the kinematic equation given in Section 2.3 by Equation 2.76.

$$\hat{q}_{k+1}^- = \Omega(\hat{\omega}_k^+) \hat{q}_k^+, \quad (3.33)$$

$$\Omega(\hat{\omega}_k^+) = \begin{bmatrix} \cos(0.5\|\hat{\omega}_k^+\|\Delta t)I_3 - [\hat{\psi}_k^+ \times] & \hat{\psi}_k^+ \\ -\hat{\psi}_k^{+T} & \cos(0.5\|\hat{\omega}_k^+\|\Delta t) \end{bmatrix}. \quad (3.34)$$

In Equation 3.34, the vector $\hat{\psi}_k^+$ is parallel to $\hat{\omega}_k^+$, as shown in Equation 3.35.

$$\hat{\psi}_k^+ = \sin(0.5\|\hat{\omega}_k^+\|\Delta t) \frac{\hat{\omega}_k^+}{\|\hat{\omega}_k^+\|}. \quad (3.35)$$

The state vector sigma points $\mathbf{x}_k^{\delta p}[i]$ are converted into sigma-point error quaternions using Equation 3.27. The sigma-point error quaternions are used to calculate the sigma-point quaternions afterwards, using the multiplicative error property, as shown in Equations 3.36.

$$\begin{aligned} \hat{q}_k^+[0] &= \hat{q}_k^+, \\ \hat{q}_k^+[i] &= \delta q_k^+[i] \otimes \hat{q}_k^+. \end{aligned} \quad (3.36)$$

Equations 3.36 implies that $\delta q_k^+[0] = [0 \ 0 \ 0 \ 1]^T$, which in turn requires that $\mathbf{x}_k^{\delta p}[0]$ be equal to zero. This requirement is imposed at the beginning of the algorithm at each sample time

execution. After propagated in time, the sigma-point quaternions are utilized to calculate the predicted sigma-point error quaternions, using the quaternion inverse $(\delta \mathbf{q}_{k+1}^- [i])^{-1}$.

The quaternion and angular velocities equations presented above can be written in the form of the dynamic state space model equation, as shown in Equation 3.37.

$$\mathbf{q}_{k+1} = \mathbf{f}(\mathbf{q}_k, \boldsymbol{\omega}_k, t_k) + \mathbf{w}_k = \Omega(\boldsymbol{\omega}_k) \mathbf{q}_k + \mathbf{w}_k. \quad (3.37)$$

The additive process noise \mathbf{w}_k in USQUE is modelled as a zero-mean Gaussian stochastic process. That is, $\mathbf{w}_k \sim N(\mathbf{0}, Q_k)$, where Q_k is the covariance matrix of \mathbf{w}_k : $Q_k = E[\mathbf{w}_k \mathbf{w}_k^T]$. Therefore, the matrix Q_k can be computed using Equation 3.38.

$$Q_k = \begin{bmatrix} (\sigma_v^2 \Delta t + \frac{1}{3} \sigma_u^2 \Delta t^3) I_3 & -(\frac{1}{2} \sigma_u^2 \Delta t^2) I_3 \\ -(\frac{1}{2} \sigma_u^2 \Delta t^2) I_3 & (\sigma_u^2 \Delta t) I_3 \end{bmatrix}. \quad (3.38)$$

In UKF, when the process noise is additive, as in Equation 3.37, the matrix Q_k is simply added to the predicted covariance matrix P_k^- . However, in [Crassidis 2003] the authors propose a different procedure. Parting from the continuous-time process noise $\mathbf{w}(t)$, they compute a new covariance matrix \bar{Q}_k using an approximation for the integration over the sampling interval. The matrix \bar{Q}_k derivation depends on the attitude kinematic models and the matrix Q_k . Assuming the approximation $\|\Delta t \hat{\boldsymbol{\omega}}_k^+\| \ll 1$ is valid, the result obtained in that work is given by Equation 3.39:

$$\bar{Q}_k = \frac{\Delta t}{2} \begin{bmatrix} (\sigma_v^2 - \frac{1}{6} \sigma_u^2 \Delta t^2) I_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \sigma_u^2 I_3 \end{bmatrix}, \quad (3.39)$$

where $\mathbf{0}_3$ is a 3×3 matrix with all entries equal to 0. The matrix \bar{Q}_k is used in the USQUE algorithm instead of Q_k .

The output equation of the state space model in USQUE regards as the output \mathbf{y}_k the measurements obtained from N exteroceptive sensors mounted in the body. Let $\boldsymbol{\mu}_b^i$ be the 3×1 vector measurement provided by the i^{th} -sensor. Of course, $\boldsymbol{\mu}_b^i$ is expressed in the body-system. The correspondent vector expressed in the reference-system is designated as $\boldsymbol{\mu}_r^i$. Therefore, using the relationship between the quaternion of rotation and the attitude matrix, the vector measurement $\boldsymbol{\mu}_b^i$ is given by Equation 3.40.

$$\boldsymbol{\mu}_b^i = A(\mathbf{q}) \boldsymbol{\mu}_r^i + \mathbf{v}_b^i. \quad (3.40)$$

The additive measurement noise \mathbf{v}_b^i is modelled as a zero-mean Gaussian stochastic process such that $\mathbf{v}_b^i \sim N(\mathbf{0}, R_k^i)$, where the covariance matrix R_k^i is equal to $\sigma_i^2 I_3$. Equation 3.40 is very convenient since it can be applied to any three-axial sensor which provides vector observations. The measurements of the N sensors can be concatenated using block matrices,

as shown in Equation 3.41, which was written in the form of the state space output equation.

$$\mathbf{y}_k = \mathbf{h}(\mathbf{q}_k, t_k) + \mathbf{v}_{b_k} = \begin{bmatrix} A(\mathbf{q})\boldsymbol{\mu}_r^1 \\ A(\mathbf{q})\boldsymbol{\mu}_r^2 \\ \vdots \\ A(\mathbf{q})\boldsymbol{\mu}_r^N \end{bmatrix}_k + \begin{bmatrix} \mathbf{v}_b^1 \\ \mathbf{v}_b^2 \\ \vdots \\ \mathbf{v}_b^N \end{bmatrix}_k. \quad (3.41)$$

Obviously, the complete covariance matrix R_k of \mathbf{v}_b is such that $R_k = \text{diag}[\sigma_1^2 I_3 \ \sigma_2^2 I_3 \ \dots \ \sigma_N^2 I_3]^T$. The vectors $\boldsymbol{\mu}_r^i$ are considered constants, which means they are treated as parameters of the function \mathbf{h} . The validity of this assumption depends on the application. For instance, if a three-axis accelerometer is used as an exteroceptive sensor and the reference-system is fixed relative to Earth's surface, the hypothesis above is valid. On the other hand, for a spacecraft orbiting the Earth, if a three-axis magnetometer is employed, the magnetic field measured in any reference-system will be a function of the spacecraft position around Earth. In such case, the application of Equation 3.41 simply ignores the vectors $\boldsymbol{\mu}_r^i$ dynamics.

Once the basic definitions were provided, the systematic procedure in USQUE can be presented. The algorithm in what follows will be described as a list of instructions. This form of organization is useful for software implementation. The overall procedure is divided in three parts: initialization, prediction, and correction. The first part is executed only once. The other two parts are executed at each sample time. Thus, after the correction part is executed, the algorithm returns to the prediction part for the next instant.

INITIALIZATION

(Do once)

1. Choose initial values for $\hat{\mathbf{q}}_0^+$ and $\hat{\boldsymbol{\beta}}_0^+$.
2. Choose an initial value for the 6×6 covariance matrix P_0^+ . The upper-left 3×3 submatrix of P_0^+ is associated to attitude error angles, and the gyro parameters σ_u and σ_v .
3. Choose the stereographic projection parameter a and set $c = 2(a + 1)$.
4. Calculate the matrix \bar{Q}_k using σ_u , σ_v , and the gyro sampling interval Δt . Set this matrix as a constant object.
5. Set the matrix R_k as $R_k = \text{diag}[\sigma_1^2 I_3 \ \sigma_2^2 I_3 \ \dots \ \sigma_N^2 I_3]^T$, using the sensors variances σ_i^2 . Set this matrix as a constant object.
6. Choose the parameter λ .
7. Set $\mathbf{x}_0^+ = [\mathbf{0}^T \ \boldsymbol{\beta}_0^{+T}]^T$.

PREDICTION

(Do at each sample time)

1. Calculate the sigma points using Equations 3.42 (unlike the UT, the authors of [Crassidis 2003] recommend using the matrix \overline{Q}_k twice in USQUE).

$$\begin{aligned}\mathcal{X}_k[0] &= \hat{\mathbf{x}}_k^+ ; \\ \mathcal{X}_k[i] &= \hat{\mathbf{x}}_k^+ + \left[\sqrt{(n + \lambda)(P_k^+ + \overline{Q}_k)} \right]_{*i} , \quad i = 1, 2, \dots, n ; \\ \mathcal{X}_k[n + i] &= \hat{\mathbf{x}}_k^+ - \left[\sqrt{(n + \lambda)(P_k^+ + \overline{Q}_k)} \right]_{*i} , \quad i = 1, 2, \dots, n .\end{aligned}\tag{3.42}$$

2. Calculate the sigma-point error quaternions using Equations 3.43.

$$\begin{aligned}\delta q_{4k}^+[i] &= \frac{-a \left\| \mathcal{X}_k^{\delta p}[i] \right\|^2 + c \sqrt{c^2 + \left\| \mathcal{X}_k^{\delta p}[i] \right\|^2 (1 - a^2)}}{c^2 + \left\| \mathcal{X}_k^{\delta p}[i] \right\|^2} ; \\ \delta \overline{\mathbf{q}}_k^+[i] &= \frac{(a + \delta q_{4k}^+[i])}{c} \mathcal{X}_k^{\delta p}[i], \quad i = 1, 2, \dots, 2n .\end{aligned}\tag{3.43}$$

3. Calculate the sigma-point quaternions using Equations 3.44.

$$\begin{aligned}\hat{\mathbf{q}}_k^+[0] &= \hat{\mathbf{q}}_k^+ ; \\ \hat{\mathbf{q}}_k^+[i] &= \delta \mathbf{q}_k^+[i] \otimes \hat{\mathbf{q}}_k^+[0] .\end{aligned}\tag{3.44}$$

4. Calculate the sigma-point angular rates using Equation 3.45.

$$\hat{\omega}_k^+[i] = \omega_k - \mathcal{X}_k^\beta[i] .\tag{3.45}$$

5. Calculate the matrix $\Omega(\hat{\omega}_k^+[i])$ for each sigma-point angular velocity.

6. Propagate the sigma-point quaternions forward in time using Equation 3.46.

$$\hat{\mathbf{q}}_{k+1}^-[i] = \Omega(\hat{\omega}_k^+[i]) \hat{\mathbf{q}}_k^+[i] .\tag{3.46}$$

7. Calculate the propagated sigma-point error quaternions using Equation 3.47.

$$\delta \mathbf{q}_{k+1}^-[i] = \hat{\mathbf{q}}_{k+1}^-[i] \otimes (\hat{\mathbf{q}}_{k+1}^-[0])^{-1} .\tag{3.47}$$

8. Calculate the sigma-point outputs using Equations 3.48, where the attitude matrix A must be evaluated for each sigma-point quaternion.

$$\mathbf{y}_{k+1}[i] = \mathbf{h}(\mathbf{q}_{k+1}^-[i], t_k) = \begin{bmatrix} A(\mathbf{q}^-[i])\boldsymbol{\mu}_r^1 \\ A(\mathbf{q}^-[i])\boldsymbol{\mu}_r^2 \\ \vdots \\ A(\mathbf{q}^-[i])\boldsymbol{\mu}_r^N \end{bmatrix}_{k+1}. \quad (3.48)$$

9. Calculate the propagated sigma points using Equations 3.49.

$$\begin{aligned} \boldsymbol{\chi}_{k+1}^{\delta p}[0] &= \mathbf{0}; \\ \boldsymbol{\chi}_{k+1}^{\delta p}[i] &= c \cdot \frac{\delta \bar{\mathbf{q}}_{k+1}^-[i]}{a + \delta q_{4k+1}^-[i]}; \\ \boldsymbol{\chi}_{k+1}^\beta[i] &= \boldsymbol{\chi}_k^\beta[i]. \end{aligned} \quad (3.49)$$

10. Calculate the predicted mean for the state vector \mathbf{x}_{k+1} applying the weighted sample mean formula to the propagated sigma points, as in Equation 3.50.

$$\hat{\mathbf{x}}_{k+1}^- = \frac{1}{n + \lambda} \left(\lambda \boldsymbol{\chi}_{k+1}[0] + \frac{1}{2} \sum_{i=1}^{2n} \boldsymbol{\chi}_{k+1}[i] \right). \quad (3.50)$$

11. Calculate the predicted covariance matrix applying the weighted sample covariance formula to the propagated sigma points, as in Equation 3.51 (here the matrix $\bar{\mathbf{Q}}_k$ is added again).

$$\begin{aligned} P_{k+1}^- &= \frac{1}{n + \lambda} \left[\lambda (\boldsymbol{\chi}_{k+1}[0] - \hat{\mathbf{x}}_{k+1}^-) (\boldsymbol{\chi}_{k+1}[0] - \hat{\mathbf{x}}_{k+1}^-)^T \right. \\ &\quad \left. + \frac{1}{2} \sum_{i=1}^{2n} (\boldsymbol{\chi}_{k+1}[i] - \hat{\mathbf{x}}_{k+1}^-) (\boldsymbol{\chi}_{k+1}[i] - \hat{\mathbf{x}}_{k+1}^-)^T \right] + \bar{\mathbf{Q}}_k. \end{aligned} \quad (3.51)$$

12. Calculate the predicted output applying the weighted sample mean formula to the propagated sigma-point outputs, as in Equation 3.52, where a more compact form is written using the weights W_i defined earlier.

$$\hat{\mathbf{y}}_{k+1}^- = \sum_{i=0}^{2n} W_i \mathbf{y}_{k+1}[i]. \quad (3.52)$$

CORRECTION

(Do at each sample time)

1. Calculate the output covariance matrix P_{k+1}^{yy} applying the weighted sample covariance formula to the propagated sigma-point outputs, as in Equation 3.53.

$$P_{k+1}^{yy} = \sum_{i=0}^{2n} W_i (\mathcal{Y}_{k+1}[i] - \hat{\mathbf{y}}_{k+1}^-) (\mathcal{Y}_{k+1}[i] - \hat{\mathbf{y}}_{k+1}^-)^T. \quad (3.53)$$

2. Calculate the innovation $\boldsymbol{\vartheta}_{k+1}$ using the sensors measurements \mathbf{y}_{k+1} and the predicted output $\hat{\mathbf{y}}_{k+1}^-$ as in Equation 3.54.

$$\boldsymbol{\vartheta}_{k+1} = \mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}^-. \quad (3.54)$$

3. Calculate the innovation covariance matrix $P_{k+1}^{\vartheta\vartheta}$ using Equation 3.55.

$$P_{k+1}^{\vartheta\vartheta} = P_{k+1}^{yy} + R_{k+1}. \quad (3.55)$$

4. Calculate the cross-correlation matrix P_{k+1}^{xy} applying the weighted sample cross-correlation formula to the propagated sigma points, and the propagated sigma-point outputs, as in Equation 3.56.

$$P_{k+1}^{xy} = \sum_{i=0}^{2n} W_i (\mathcal{X}_{k+1}[i] - \hat{\mathbf{x}}_{k+1}^-) (\mathcal{Y}_{k+1}[i] - \hat{\mathbf{y}}_{k+1}^-)^T. \quad (3.56)$$

5. Calculate the Kalman gain K_{k+1} using Equation 3.57.

$$K_{k+1} = P_{k+1}^{xy} (P_{k+1}^{\vartheta\vartheta})^{-1}. \quad (3.57)$$

6. Use Equation 3.58 to update the state vector \mathbf{x}_{k+1}^- yielding the corrected value \mathbf{x}_{k+1}^+ .

$$\hat{\mathbf{x}}_{k+1}^+ = \hat{\mathbf{x}}_{k+1}^- + K_{k+1} \boldsymbol{\vartheta}_{k+1}. \quad (3.58)$$

7. Use Equation 3.59 to update the covariance matrix P_{k+1}^- yielding the correct value P_{k+1}^+ .

$$P_{k+1}^+ = P_{k+1}^- - K_{k+1} P_{k+1}^{\vartheta\vartheta} K_{k+1}^T. \quad (3.59)$$

8. The corrected state vector is such that $\hat{\mathbf{x}}_{k+1}^+ = \left[\delta \hat{\boldsymbol{\rho}}_{k+1}^{+T} \quad \hat{\boldsymbol{\beta}}_{k+1}^{+T} \right]^T$. Thus, calculate the

corrected error quaternions using Equations 3.60.

$$\begin{aligned}\delta q_{4_{k+1}}^+ &= \frac{-a \|\delta \hat{\mathbf{p}}_{k+1}^+\|^2 + c \sqrt{c^2 + \|\delta \hat{\mathbf{p}}_{k+1}^+\|^2} (1 - a^2)}{c^2 + \|\delta \hat{\mathbf{p}}_{k+1}^+\|^2}; \\ \delta \bar{\mathbf{q}}_{k+1}^+ &= \frac{(a + \delta q_{4_{k+1}}^+)}{c} \delta \hat{\mathbf{p}}_{k+1}^+.\end{aligned}\quad (3.60)$$

9. Use Equation 3.61 to update the quaternions $\hat{\mathbf{q}}_{k+1}^-$ yielding a corrected value $\hat{\mathbf{q}}_{k+1}^+$.

$$\hat{\mathbf{q}}_{k+1}^+ = \delta \mathbf{q}_{k+1}^+ \otimes \hat{\mathbf{q}}_{k+1}^- [0]. \quad (3.61)$$

10. Reset $\delta \hat{\mathbf{p}}_{k+1}^+$ to zero for the next propagation.

11. Go back to the **PREDICTION** part or finish the algorithm.

The present section is concluded with a few last comments. Equations 3.58 and 3.59 are the core of any KF filter approach. They consist in a simple linear update rule in which the predicted value and the output measurements are weighted accordingly to their probabilistic behaviour. The Kalman gain work as the weight in the update rule, which gives to the KF its remarkable properties. In USQUE the predicted value is computed using proprioceptive measurements. The estimates are then corrected by the exteroceptive measurements, as described in Chapter 1. Moreover, Equation 3.58 has the general form of a moving average filter, which is a low-pass filter. Therefore, the KF smooths the quantities comprised in the state vector as long as it minimizes the estimated covariance, thereby reducing the effect of high-frequency noises. The USQUE filter requires only one exteroceptive sensor, whereas the TRIAD algorithm demands two exteroceptive vector measurements. On the other hand, a gyro is needed in the prediction part of the USQUE algorithm. A deterministic method such as TRIAD can yield good results. However, when noise environments are excessive or the sensors have limited precision, a stochastic method such as USQUE can provide more accurate and precise results, but with the price of a higher computational burden.

Chapter 4

Small satellite three-axis simulator facility

This chapter presents the main components of LAICA facility. Of these, the most important is the three-axis air-bearing table, followed by the Helmholtz cage. However, the main practical contributions of the present dissertation concern the system of attitude determination based on computer vision, and the ABACUS onboard computer as well, both recently incorporated into the laboratory.

4.1 Facility overview

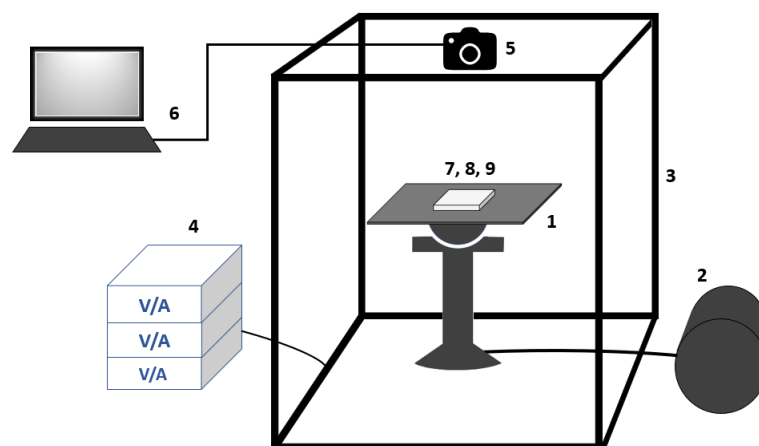


Figure 4.1: LAICA facility overview.

Figure 4.1 provides an overview of LAICA facility. This figure is a schematic representation of the Figure 1.2 presented in Chapter 1. The components represented in this figure are listed below:

1. Three-axis air-bearing table.
2. Air compressor.
3. Helmholtz cage.
4. Power supplies.
5. Web camera.
6. Personal computer (PC) .
7. ABACUS OBC.
8. Sensors.
9. Actuators.

The components above can be briefly described as in the following:

- the three-axis air-bearing table rotates over a thin layer of pressurized air provided by the compressor;
- the coils of the Helmholtz cage, fed by the power supplies, create a controlled magnetic field at its center;
- the camera placed above the table is employed to determine its attitude using computer vision software and markers;
- the ABACUS OBC is used to run attitude determination and control algorithms. It also reads sensors data and outputs control signals to actuators.
- The sensors are utilized to measure magnetic fields, accelerations and angular velocities. Some of them are embedded on ABACUS board.
- The actuators can control the air-bearing table attitude. Today, they consist of a set of reaction wheels and magnetorquers.
- The whole facility is integrated by two computers (only one is shown in Figure 4.1). The air-bearing table balancing algorithms, the control software of the Helmholtz cage, and the CV software are all executed in these computers. They also are employed to perform on-line and off-line analysis of the experiments.

The sensors utilized in this work belong to ABACUS board, thus they are described in Section 4.5. Furthermore, since the actuators are not employed in this dissertation, only a brief description of them is given in that same section in the context of ABACUS capabilities. The other components of the facility are also described in their respective sections.

4.2 Three-axis air-bearing table

The three-axis air-bearing table is a testbed for testing small satellite technologies. Air-bearing equipments have been used for testing and simulation purposes for over 50 years, since the beginning of the space race [Schwartz et al. 2003]. Air-bearings are utilized to simulate the virtually frictionless space environment a spacecraft encounter, in the same way spherical and cylindrical roller bearings are employed to reduce the friction between mechanical parts. Once in orbit, a satellite can not be fixed or receive maintenance. Furthermore, a satellite can be operated remotely only, thus it requires a high degree of autonomy. These capabilities can be design aided and safely tested by air-bearing platforms. In the small satellites context, such approach helps saving costs whilst preventing an expensive launch of an untested and possibly defective spacecraft.

The air-bearing testbeds can be planar or spherical. The former simulates the negligible forces imposed to translational motion in space, whereas the latter simulates the practically torque-free conditions of rotational motion in that environment. An example of a planar air-bearing platform can be found in [Ivanov et al. 2018]. An example of a highly compact spherical air-bearing can be found in [Chesi et al. 2013]. The LAICA facility testbed is a spherical air-bearing table with three degrees of freedom in rotational motion. Therefore, this platform is very suitable to develop and test techniques of attitude determination and control. Figure 4.2 shows a double-view picture of the LAICA air-bearing table. The left side of the figure shows the table from the top, whereas the right side shows it from the bottom.

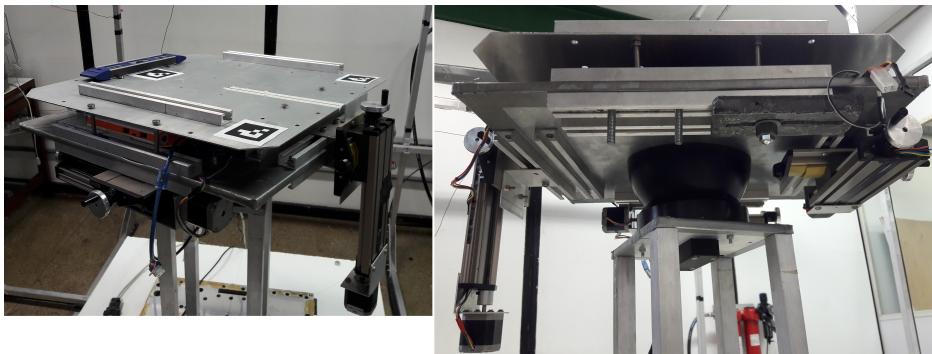


Figure 4.2: Double-view picture of the air-bearing table.

Spherical air-bearing platforms have two concentric spherical parts separated by a thin air film. One of them act as a cup shaped base supporting the other part, which can rotate almost freely. The spherical base has six small holes through which the pressurized air flows from the air compressor. Figure 4.3 shows the spherical base with the small holes and the air compressor. The left side of Figure 4.4 shows the rotating hemisphere of the air-bearing; the right side shows the two parts together.

In order to reproduce the low gravitational torque conditions encountered by orbiting spacecraft, the air-bearing table center of mass must be coincident with its center of rotation

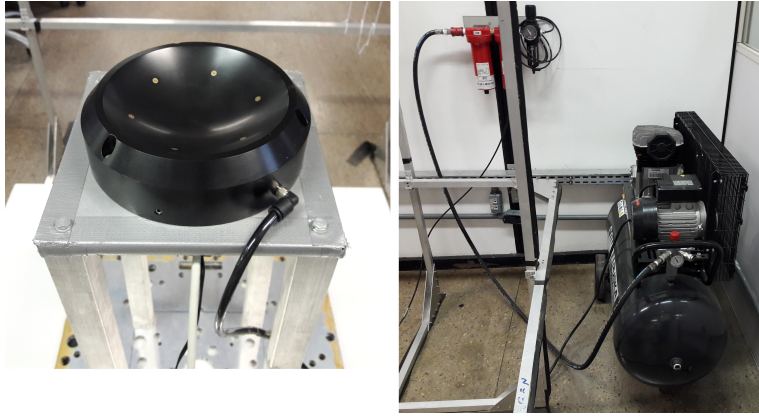


Figure 4.3: Spherical base of the testbed air-bearing (left side) and the air compressor (right side).

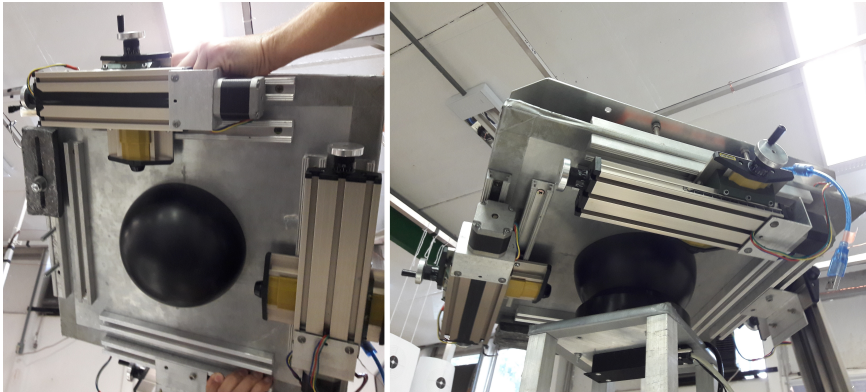


Figure 4.4: Rotating hemisphere of the testbed air-bearing (left side) and the two parts together (right side).

(CR) . Figure 4.5 depicts the relationship between the CM, the CR and the gravitational torque applied to the air-bearing table. In Figure 4.5 the gravitational torque is given by $\mathbf{r} \times M\mathbf{g}$, where M is the total mass of the platform, and \mathbf{r} is the vector parting from CR toward the CM – the point where the force due to gravity is applied. Of course, the components of the gravitational acceleration vector \mathbf{g} depend on the reference frame adopted. Therefore, the smaller $\|\mathbf{r}\|$ is, the smaller is the gravitational torque applied.

The testbed CM position can be adjusted using the movable mass units (MMU) . There are three of these devices attached to the table, and their CM can be dislocated along to the reference frame vectors of the platform. Thus, moving the CM of the three MMU allows the total CM of the testbed to be shifted relative to the CR, which is fixed and located at the center of the air-bearing hemisphere. Figure 4.6 shows the three MMU of the platform. In the top left corner these devices are shown together with their respective reference axis. The right side of the figure shows the MMU relative to \mathbf{Z} in details. In the bottom part of the figure the MMU relative to \mathbf{X} is depicted. The \mathbf{Y} MMU is similar to this last one.

The procedure of approaching the CM to the CR is called *balancing*. There are several algorithms employed to reduce the distance between these two points, one of them can be found in [Silva et al. 2016]. A detailed description of the LAICA air-bearing table is given

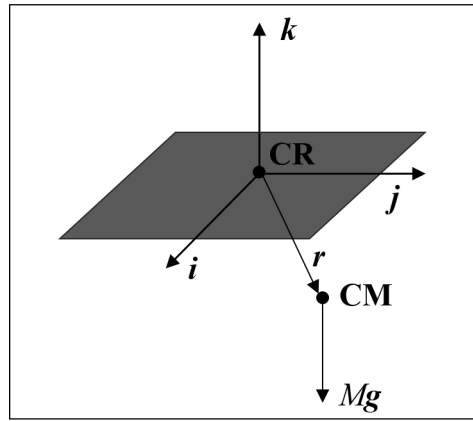


Figure 4.5: The gravitational torque applied to the testbed and its dependency on the CM and CR positions.

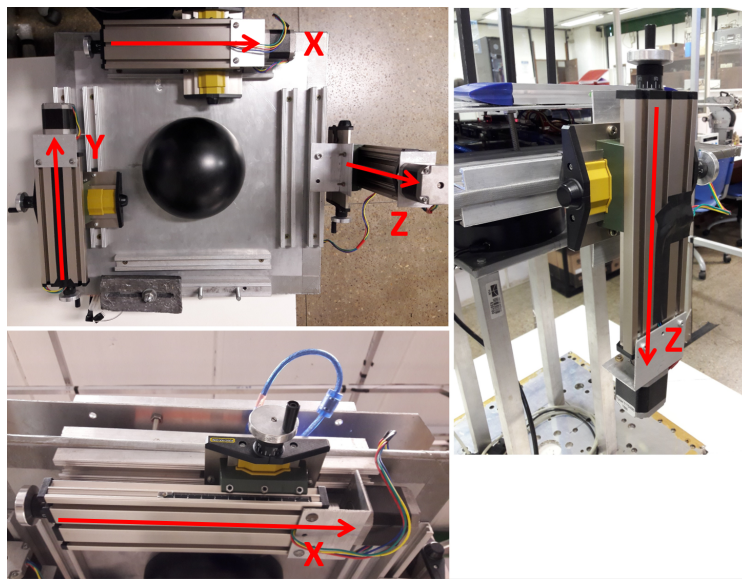


Figure 4.6: Movable mass units of the air-bearing table.

in [Silva and Rodrigues 2015]. That work provides details concerning the technical specifications of the air-bearing table components, including the air compressor and its accessories, besides the procedures to assemble the platform.

4.3 Earth Magnetic Field Simulator

The Earth magnetic field simulator at LAICA is composed by a Helmholtz cage, three power supplies, software applications and a three-axis magnetometer. The EMFS purpose consists in simulating the Earth's magnetic field perceived by a spacecraft orbiting the planet. The core of the EMFS is the Helmholtz cage, which is a cubic structure mounted using three parallel pairs of square coils. Each pair produces a magnetic field approximately constant along the axis orthogonal to the coils planes. Figure 4.7 shows the Helmholtz cage and the

power supplies on the left.

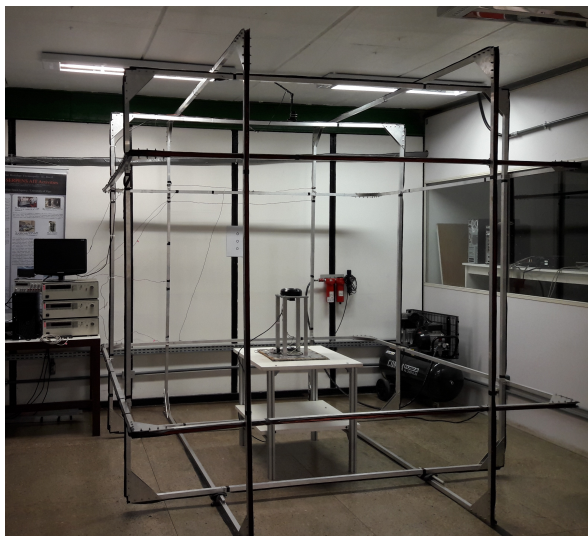


Figure 4.7: Helmholtz cage (center) and the three power supplies (left).

Each power supply is connected to one pair of coils, providing the electric current responsible for generating the magnetic field. The three power supplies are controlled by a single PC through GPIB interfaces. The software MATLAB[®] is utilized to send the commands to the power supplies and to close the Helmholtz cage control loop through a three-axis magnetometer.

Earth's magnetic field can be reproduced using the **wrldmagn** function of MATLAB. This function takes as inputs the latitude, the longitude, and the altitude associated to a pre-specified orbit. This function calculates the Earth's magnetic field using the World Magnetic Model (WMM) [NOAA 2015a]. This model is based on databases updated every 5 years, considering the magnetic field of the planet changes with time. The current database is valid within the period 2015-2020 [NOAA 2015b], and owing to this reason the **wrldmagn** function also takes as input the year when the simulation occurs.

Figure 4.8 shows a scheme representing one pair of coils and a Cartesian system fixed at the bottom coil center. The square coils, parallel to each other, have a side length of L and the distance between them is D . The two coils are connected in series, thus the current I through them is the same and flows in the same direction.

The magnetic field \mathbf{B} generated by the pair of coils in Figure 4.8 can be calculated using the Biot-Savart's law, as shown in [Batista et al. 2017] and [Loiola et al. 2018]. The Z -component of \mathbf{B} along the Z axis, $B(z)$, is given by Equation 4.1:

$$B(z) = \frac{4\mu_0 NI}{\pi L} \left[f\left(\frac{z}{L/2}\right) + f\left(\frac{D-z}{L/2}\right) \right], \quad (4.1)$$

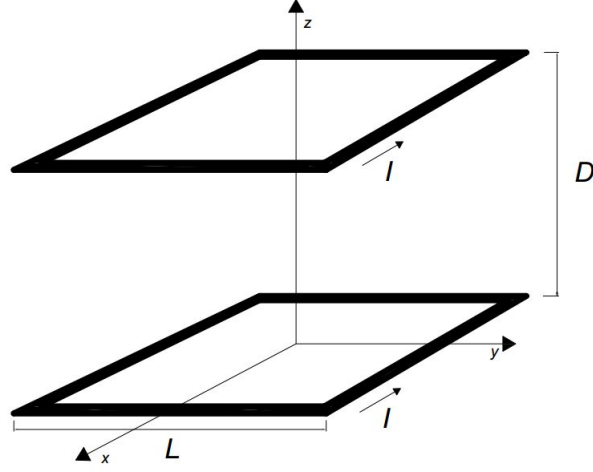


Figure 4.8: Model of one Helmholtz cage pair of coils. (Retrieved from [Loiola et al. 2018])

where the auxiliary function $f(\gamma)$ is defined in Equation 4.2.

$$f(\gamma) = \frac{1}{(\gamma^2 + 1)\sqrt{\gamma^2 + 2}}. \quad (4.2)$$

One of the requisites of the Helmholtz cage design consists in establishing a constant magnetic field within a certain region inside the cage. Considering the Taylor series expansion of $B(z)$ about the cage center $z_0 = D/2$, the approximation of this function as a constant can be analysed using the derivatives of $B(z)$. The odd order derivatives of $B(z)$ is equal to zero at z_0 . Since the concavity of functions is associated to their second order derivatives, if $B''(z_0) = 0$, the approximation of $B(z)$ as a constant function can be achieved. Equation 4.3 shows the expression for $B''(z_0)$.

$$\frac{d^2 B(z_0)}{dz^2} = \frac{8\mu_0 N I}{\pi L} \left(\frac{2}{L}\right)^2 \left[f''\left(\frac{D}{L}\right) \right]. \quad (4.3)$$

In Equation 4.3, N is the number of turns in each coil and μ_0 is the magnetic permeability of the medium. Therefore, in order that $B''(z_0) = 0$, the second derivative of the auxiliary function f must be equal to zero at D/L . Equation 4.4 exhibits the expansion of f'' .

$$\frac{d^2 f(\gamma)}{d\gamma^2} = \frac{2(6\gamma^6 + 18\gamma^4 + 11\gamma^2 - 5)}{(\gamma^2 + 1)^3(\gamma^2 + 2)^{5/2}}. \quad (4.4)$$

Consequently, f'' is zero if the sixth order polynomial in the numerator of Equation 4.4 is zero. This polynomial have only one real positive root given by $\gamma \approx 0.5445$. Therefore, if $\gamma = D/L = 0.5445$, the desired approximation can be achieved. The Helmholtz cage of LAICA was built observing the relationship $D = 0.5445L$, the results obtained were published in [Loiola et al. 2018]. One of the tests results are reproduced in Figure 4.9, it shows the relationship between the Y -component of B with the position y along the Y axis.

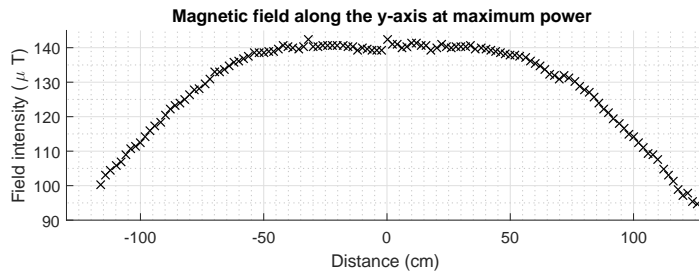


Figure 4.9: Values of the Y -component of B along the Y axis. (Retrieved from [Loiola et al. 2018])

The uniformity region of the magnetic field is of about $\pm 0.5\text{m}$ around the cage center, as can be seen in Figure 4.9. Thus, considering the three pairs of coils, there is an almost cubic region of 1m side length within which the magnetic field is approximately homogeneous.

This section concludes with the Helmholtz cage specifications, given in Table 4.1.

Pair axis	N	L (m)	Resistance (Ω)	Max. current (A)
X	20	2.5	6.6 ± 0.1	6.0
Y	20	2.5	6.8 ± 0.1	6.0
Z	20	2.5	6.9 ± 0.1	6.0

Table 4.1: Helmholtz cage specifications.

4.4 Attitude determination with computer vision

The attitude determination technique based on computer vision implemented at LAICA is a method utilized to determine the attitude of a planar object using fiducial markers and a camera. Images of a known pattern of fiducial markers printed on a plane are captured by a camera and processed in software. The markers pattern seems to be distorted in the images due to the relative position and orientation of the planar object with respect to the camera. Measuring these perspective projections distortions in the images allows the planar object attitude to be estimated.

The relationship between the three Cartesian coordinates of a point in the physical world and its correspondent two coordinates in an image can be found using the pinhole camera model. This model assumes that only a single light ray parting from a point in space reaches the camera image plane – or projective plane – through a single hole. In CCD (charge-coupled-device) cameras, the image plane coincides with the matrix of sensors which convert the light intensity in digital values [Forsyth and Ponce 2003]. With this assumption, the image is always focused and a first relationship between the coordinates can be obtained using a simple similarity between triangles. The pinhole model is depicted in Figure 4.10 where two frames are shown: the camera image frame $\mathcal{F}_i = \{\mathbf{i}_i, \mathbf{j}_i, \mathbf{k}_i\}$ and the physical

world frame $\mathcal{F}_r = \{\mathbf{i}_r, \mathbf{j}_r, \mathbf{k}_r\}$. Moreover, the optical axis part from the image plane toward the pinhole, being orthogonal to the former.

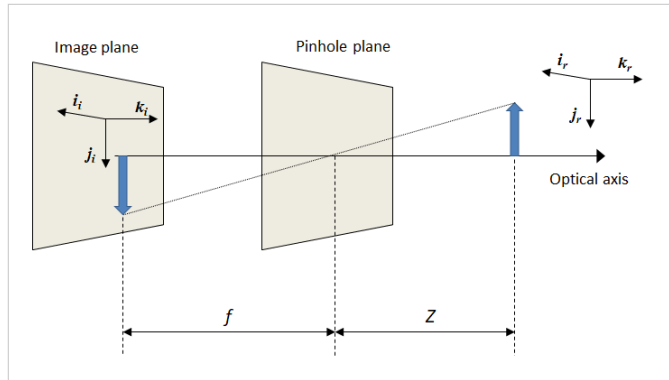


Figure 4.10: Pinhole camera model representation. (Adapted from [Bradski and Kaehler 2008])

Figure 4.10 shows that the image is inverted with respect to the object. In order to simplify the equations, the model shown in Figure 4.11 is used instead. In this model the light rays part from the object toward the pinhole, intersecting the image plane placed between them.

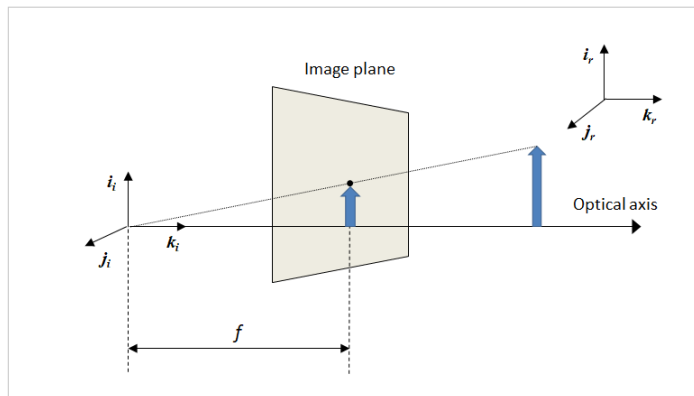


Figure 4.11: Modified version of the pinhole camera model representation. (Adapted from [Bradski and Kaehler 2008])

Let f be the focal length of the camera, which is the distance between the pinhole and the image plane. The relationship between the coordinates of points in physical world (x_r, y_r, z_r) and the image coordinates (x_i, y_i) is given by Equations 4.5.

$$\begin{aligned} x_i &= f \frac{x_r}{z_r}, \\ y_i &= f \frac{y_r}{z_r}. \end{aligned} \tag{4.5}$$

The focal camera focal length is not necessarily the same in \mathbf{i}_i and \mathbf{j}_i directions due to the fact that the CCD sensors are rectangular, not square. Also, in practice the pinhole and

the origin point of the image frame are not aligned to each other. Therefore, Equations 4.5 must incorporate the focal length f_x and the offset c_x in i_i direction, the focal length f_y and the offset c_y in j_i direction. These modifications result in Equations 4.6.

$$\begin{aligned}x_i &= f_x \frac{x_r}{z_r} + c_x, \\y_i &= f_y \frac{y_r}{z_r} + c_y.\end{aligned}\tag{4.6}$$

Equations 4.5 can be written using matrices, yielding Equation 4.7.

$$\begin{bmatrix}x_i \\y_i\end{bmatrix} = \frac{1}{z_r} \begin{bmatrix}f_x & 0 & c_x \\0 & f_y & c_y\end{bmatrix} \begin{bmatrix}x_r \\y_r\end{bmatrix}.\tag{4.7}$$

Real cameras do not use simple pinholes to capture images since the intensity of the light rays inside the camera would be very small. Cameras use lenses to focus the light, which allows the image to be rapidly formed due to the higher light intensity. On the other hand, in practice a lens introduces distortions in the images that must be corrected. There are two basic distortions that must be compensated. The first is the radial distortion which occurs due to the fact that spherical lenses do not have one single focal point. Thus, the further a point is from the optical axis the more it is dislocated in the image plane. The second is the tangential distortion which occurs because the image plane is not perfectly orthogonal to the optical axis. The radial distortion can be compensated using Equation 4.8, whereas the tangential distortion can be corrected using Equation 4.9 [Bradski and Kaehler 2008].

$$\begin{aligned}x_{comp} &= x(1 + k_1r^2 + k_2r^4 + k_3r^6), \\y_{comp} &= y(1 + k_1r^2 + k_2r^4 + k_3r^6);\end{aligned}\tag{4.8}$$

$$\begin{aligned}x_{comp} &= x + [2p_1y + p_2(r^2 + 2x^2)], \\y_{comp} &= y + [p_1(r^2 + 2y^2) + 2p_2x].\end{aligned}\tag{4.9}$$

In the last equations, $r = \sqrt{x^2 + y^2}$.

The four parameters f_x , c_x , f_y and c_y are called the camera intrinsic parameters. They are normally not provided by the camera manufacturer and neither are the distortion parameters k_1 , k_2 , k_3 , p_1 and p_2 . However, a calibration procedure allows these parameters to be estimated. The OpenCV software library provides functions that can be used to estimate the camera intrinsic and distortion parameters. Besides the functions, a complete calibration application is available, requiring the utilization of fiducial markers. A chessboard pattern

is often employed as a fiducial marker, where the corners of the black and white squares are identified by the application. The calibration procedure goes as follows: a chessboard pattern printed in a flat surface is positioned in front of a web camera in several orientations. The application periodically captures an image of the chessboard and, in the end, it calculates all of the nine unknown parameters of the camera. Figure 4.12 shows a sequence of images of the chessboard pattern during the calibration procedure. The bottom right part of the figure shows the exact moment when the application captures an image, exhibiting the correspondent negative counterpart of the original one.

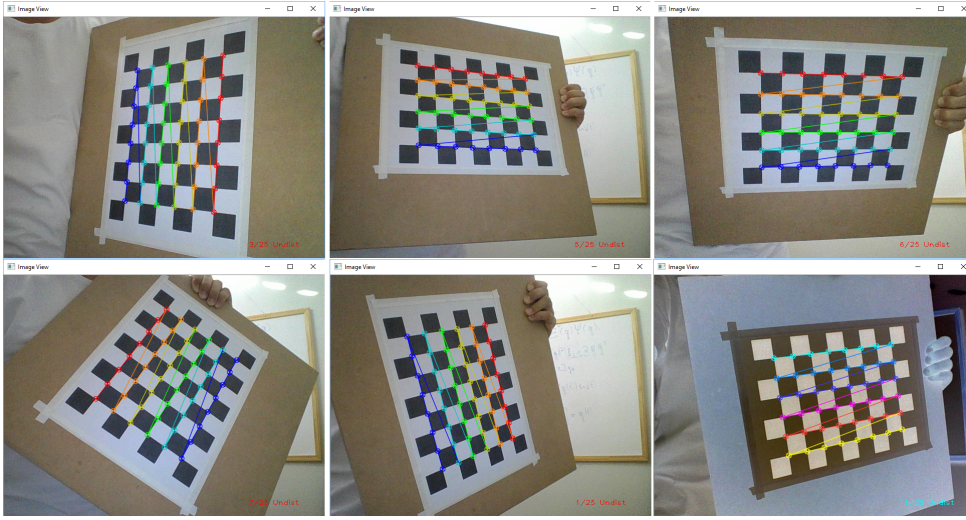


Figure 4.12: Chessboard pattern of fiducial markers used during the calibration procedure.

The previous analysis considered the frames \mathcal{F}_i and \mathcal{F}_r aligned to each other. When they are not, the relative orientation between them can be found. If the camera frame is regarded as the body-system and the physical world frame is the reference-system, this is equivalent of founding the camera attitude. Equation 4.10 shows the relationship between the coordinates of a point in the physical world and the correspondent coordinates in the image, where the two frames are rotated and translated relative to each other.

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = s \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} A & \mathbf{t} \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ z_r \\ 1 \end{bmatrix}. \quad (4.10)$$

In Equation 4.10, s is a scale factor and A is the rotation matrix which takes \mathcal{F}_r to \mathcal{F}_i . The vector \mathbf{t} is the translation vector representing the position of the \mathcal{F}_r origin point expressed in \mathcal{F}_i .

The attitude together with the translation vector comprise the camera *pose*, which is a concept widely employed in robotics. Given a set of points – provided by a fiducial marker – with known positions in \mathcal{F}_r and the correspondent points in \mathcal{F}_i , the camera pose can be estimated. The chessboard pattern can be used in the pose estimation, however, in this work

the fiducial markers provided by the ArUco library are utilized. This software is available as a free third-party library of OpenCV and is based on the fiducial markers system proposed in [Garrido-Jurado et al. 2014]. The authors themselves implemented the ArUco library, which provides several functions for generating different markers, identifying them, and estimating the camera pose, besides other features. In ArUco, the markers are grouped in *dictionaries* accordingly to some common characteristics. Figure 4.13 shows two markers from different dictionaries.



Figure 4.13: ArUco markers of 4 bits (left) and 6 (bits).

Each ArUco marker is a matrix of white and black squares. The marker on the left side of Figure 4.13 is a 4×4 marker with a border of one square in length, and the other markers from its dictionary possess the same characteristics. The size of the marker determines its number of bits. Therefore, the marker on the left side of Figure 4.13 has 4 bits, whereas the one on the right side has 6 bits. Moreover, each marker has an index number which is unique within the dictionary. The markers in Figure 4.13 are those of index 0 in their respective dictionaries.

The ArUco library provides a function to estimate the camera pose using a single marker, requiring the camera intrinsic parameters, which must be found first. If several markers - from a single dictionary - are exhibited in an image, this function outputs the estimated pose of each marker found. This feature allows the utilization of various markers to estimate the pose of a flat object. If the markers are placed and aligned on the object surface, the pose of each marker can be combined to estimate the object pose more accurately. Figure 4.14 exhibits the ArUco library utilization for pose determination. In the left side of this figure, the estimated attitude parameters – given as Euler angles and rotation vector components – were printed in the terminal window. The left side shows the 6-bits ArUco markers identified, while the 4-bit marker is not recognized since it belongs to a different dictionary. The frames in Figure 4.14 was drawn online using a function provided by the ArUco library, turning this software into an augmented reality application. Also, another function draws the index of each marker and a rectangle around it.

Figure 4.15 shows the utilization of the same markers to estimate the camera pose relative to the board, which is hold in several orientations.

The software of the ADCV system was implemented in C++ language using the ArUco library and the OpenCV functions. Furthermore, a socket communication was employed for online transmitting the attitude data to MATLAB, where they are stored and analysed. The camera of the ADCV system is the C170 web camera of Logitech – Table 4.2 describes some

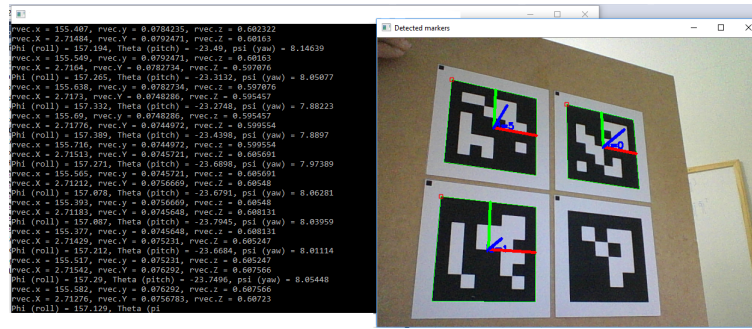


Figure 4.14: Camera attitude determination using ArUco markers of 6 bits.

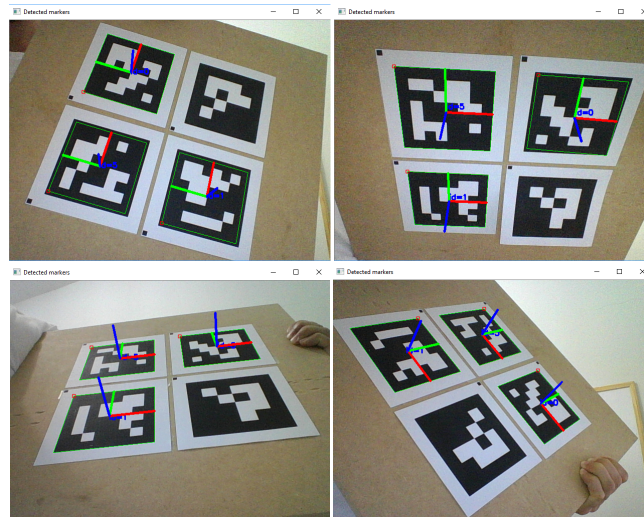


Figure 4.15: Camera attitude relative to a board hold in different orientations.

of its features. The tests applied to evaluate the ADCV system are described in Chapter 5.

Connection	USB 2.0
Diagonal field of view	58°
Focal length	2.3 mm
Resolution	1024 × 768 pixels

Table 4.2: Characteristics of the web camera utilized with the ADCV system.

4.5 ABACUS onboard computer

The onboard computer ABACUS was developed by the startup company Gauss Srl based in Rome, Italy. The company was founded by former members of the GAUSS research group at the *Scuola di Ingegneria Aerospaziale* of the *Sapienza - Università di Roma*. The ABACUS OBC was designed as a multi-purpose platform to attend different satellite projects needs. Figure 4.16 exhibits two views of the LAICA ABACUS OBC – version 2017 – used in the present dissertation.

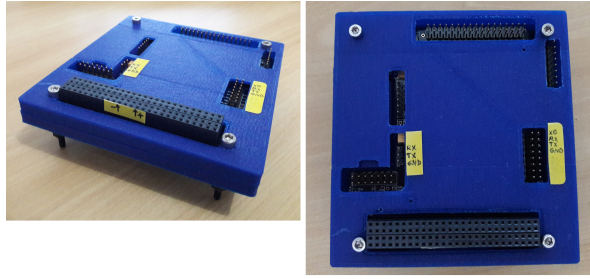


Figure 4.16: Two views of LAICA ABACUS OBC 2017 mounted into a case provided by Gauss Srl.

In Figure 4.16 the board is mounted into a case provided by the manufacturer. Figure 4.17 shows two views of the board, exposed this time.

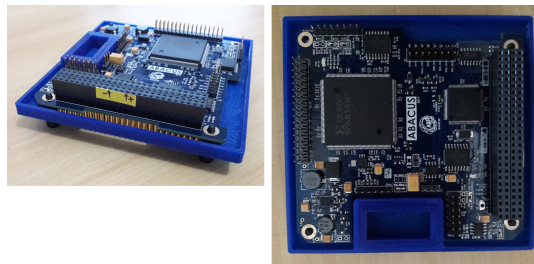


Figure 4.17: Two views of LAICA ABACUS OBC 2017.

The main components of ABACUS OBC are the MCU (microcontroller unit) MSP430F5438A-EP of Texas Instruments, the FPGA Spartan-3E of Xilinx, and the inertial measurement unit (IMU) MPU-9250 of InvenSense Inc. The last device is composed by a three-axis magnetometer, a three-axis gyro and a three-axis accelerometer, all of them encapsulated into a single chip, which is used to estimate the attitude of the board. Figure 4.18 depicts the reference frames of ABACUS IMU. The accelerometer and the gyro frames are aligned with each other, whereas the magnetometer frame is in an opposed direction due to the manner the chip is mounted on the board. Table 4.3 describes the main characteristics of the ABACUS IMU.

Parameter	Gyroscope	Accelerometer	Magnetometer
Resolution	16 bits	16 bits	14 bits
Full scale range	$\pm 250 \text{ }^\circ/\text{s}$	$\pm 2\text{g}$	$\pm 4800 \mu\text{T}$

Table 4.3: Characteristics of ABACUS IMU.

The ABACUS MSP430 is a 16-bit RISC MCU which can run in a frequency up to 25 MHz. This chip is used in this work to run the attitude algorithms written in C language. The FPGA Spartan-3E can run in a frequency up to 100 MHz, thus being suitable for heavy computational operations. Besides the higher processing speed, the FPGA can improve the performance of programs that can be parallelized, such as those which perform matrix algebra operations. Despite of these benefits, the FPGA is not employed in this dissertation,

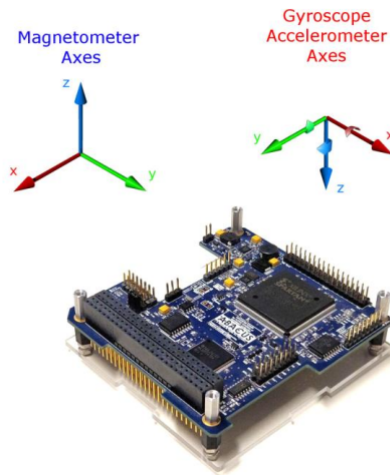


Figure 4.18: Body-system vectors of the ABACUS sensors.

however it is intended to be adopted in future works. Table 4.4 describes the main features of ABACUS related to the MSP430.



Figure 4.19: JTAG programmer MSP-FET.

The MSP430 MCU is programmed using the MSP-FET JTAG emulation tool of Texas Instruments, shown in Figure 4.19. The IDE (interface development environment) Code Composer Studio (CCS) of the same manufacturer is the environment where the applications written in C can be debugged, compiled and loaded into the board.

MCU	16 bits RISC 25 MHz
Memory	16 KB SRAM + 256 KB Flash
I/O	45 + 16 GPIOs
Communication	2 × I2C 2 × SPI 4 × COM ports
Storage	16 MB Flash (ext.)
Other	3-channel DMA

Table 4.4: MSP430 related features of ABACUS.

Chapter 5

Experimental procedures

This chapter presents the experimental procedures executed in the laboratory besides the practical issues concerning the algorithms implementation.

5.1 Local reference-system

Chapters 2 and 3 established that the reference-system is associated to reference objects or physical quantities, such as the magnetic field, gravitational acceleration, distant stars, etc. On the other hand, the body-system is associated to the sensors mounted on the body. For instance, the ABACUS IMU establishes the board body-system as depicted in Figure 4.18. Provided that the body attitude is measured with respect to the reference-system, in order to test attitude determination algorithms in laboratory, a local reference-system located at the facility must be established first.

In this work the reference-system is defined using the gravitational acceleration and a target fixed at one of the laboratory walls. Figure 5.1 shows an apparatus mounted to establish the reference-system at LAICA facility.

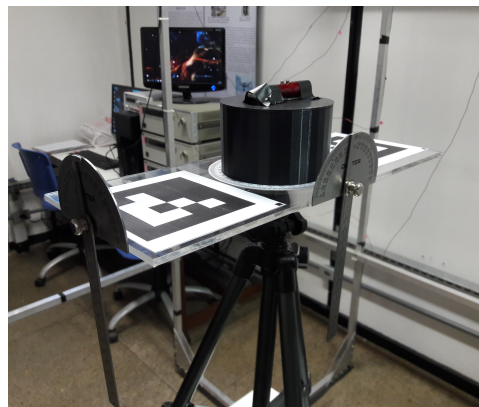


Figure 5.1: Apparatus adopted to physically establish the reference-system at the laboratory.

The device shown in Figure 5.1 is composed by a flat acrylic sheet fixed on a camera

tripod. Also, two protractors are mounted orthogonally to the acrylic surface, where two ArUco markers are affixed to. The tool used to measure the angles in each protractor acts as a pendulum, allowing the adjustment of the acrylic inclination accordingly to the gravitational acceleration. Therefore, two orientation angles can be established using these protractors. The other angle is defined using a target affixed at one of the laboratory walls and a common laser pointer. A cylindrical piece of plastic is mounted at the acrylic center and the laser pointer is fixed on its top. The plastic piece is free to rotate, thus it is used to indicate the third angle using a third protractor placed between the piece and the acrylic. Figure 5.2 contains a picture of the apparatus, where the plastic piece, the laser pointer and the third protractor are shown in details.

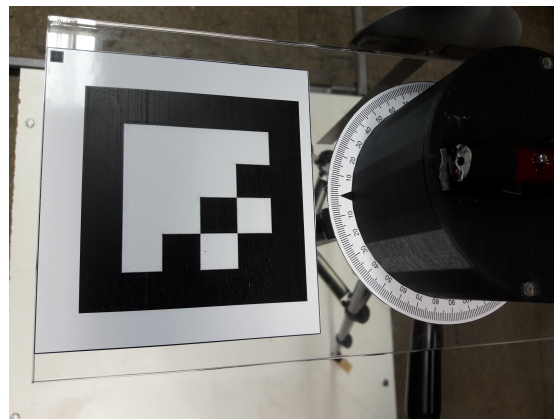


Figure 5.2: Plastic piece, the laser pointer and the protractor used to define the third orientation angle.

Rotating the acrylic in one direction and rotating the plastic piece in the opposite direction, in order that the laser keeps pointing to the target on the wall, allows the measurement of the third angle. The relative movement between the piece and the acrylic determines this angle. Figure 5.3 shows the laser pointing to the target.

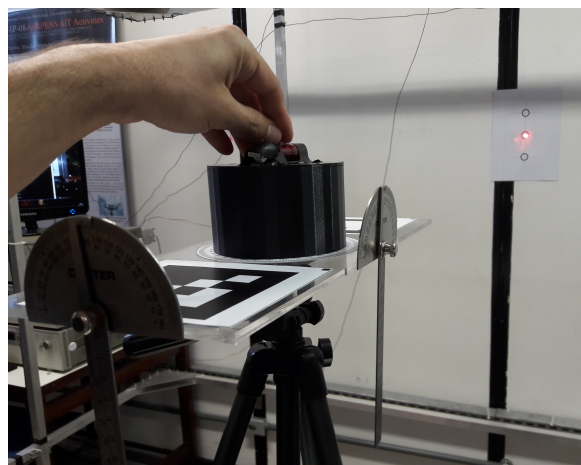


Figure 5.3: Laser pointing to the wall target.

The ArUco markers are used to align the camera frame to the physical reference-system. Hence, after the camera alignment, the ADCV reference-system can be used as the local

reference-system, replacing the apparatus. As a consequence of this definition, the X and Y axes of the local reference-system are orthogonal to the gravitational acceleration. The X points in the opposite direction of the wall target and the Z axis is parallel to the gravitational acceleration, but pointing in the opposite direction. Thus, the local reference-system is similar to the NED system.

5.2 Testing of the attitude determination system based on computer vision

The main purpose of the ADCV system is to determine the air-bearing table attitude. The camera is fixed on the laboratory ceiling above the air-bearing base. The performance of the ADCV system was evaluated using the apparatus described in Section 5.1. At each step of the experimental procedures, only one angle measured by the apparatus was changed. Therefore, the correspondence between the variable angle and the ADCV measures, given as Euler angles, could be easily established.

Figure 5.4 shows a picture of the computer screen while the ADCV system estimated the attitude of the reference-system apparatus.

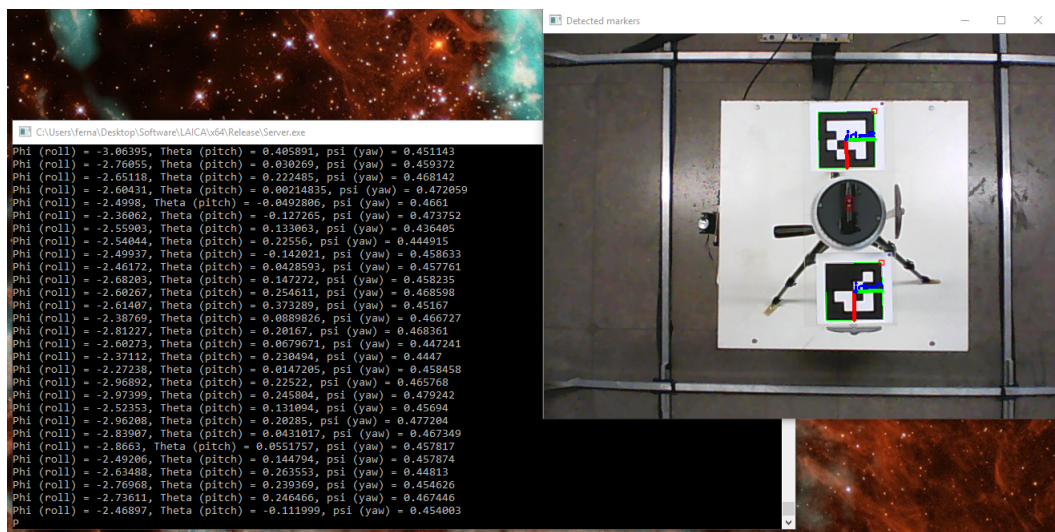


Figure 5.4: ADCV system determining the attitude of the reference-system apparatus.

In the first test, the apparatus yaw angle was changed in steps of 5° in the approximate range $[-25^\circ, 25^\circ]$. Figure 5.5 shows the results obtained. The red line in the graph of Figure 5.5 corresponds to a linear curve fitting given by $y = 0.9995x + 0.6549$. The mean of the standard deviation comprising all of the ADCV measurements is 0.0266° . Thus, the yaw angle determination is quite accurate.

Figure 5.6 shows the results obtained for the roll angles measured in the range $[-20^\circ, 20^\circ]$, also in steps of 5° . The linear curve shown is defined by $y = 1.0845x - 1.0935$. The standard deviation comprising all of the roll measurements is 0.4015° . As expected, the

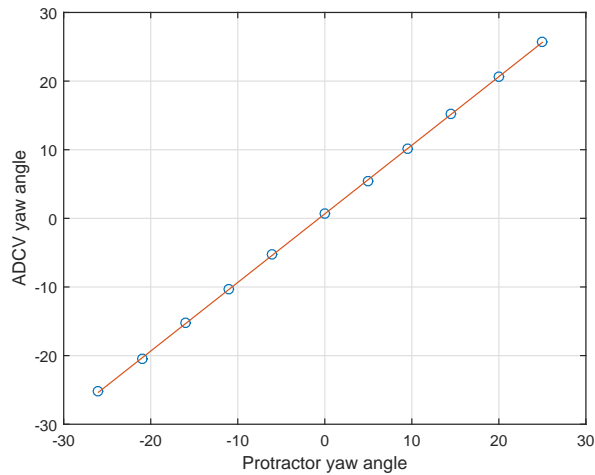


Figure 5.5: Protractor yaw angles *versus* ADCV yaw angles.

roll angles determined by the ADCV system have greater variance than the yaw angles due to the fact that these last ones correspond to rotations about the camera optical axis.

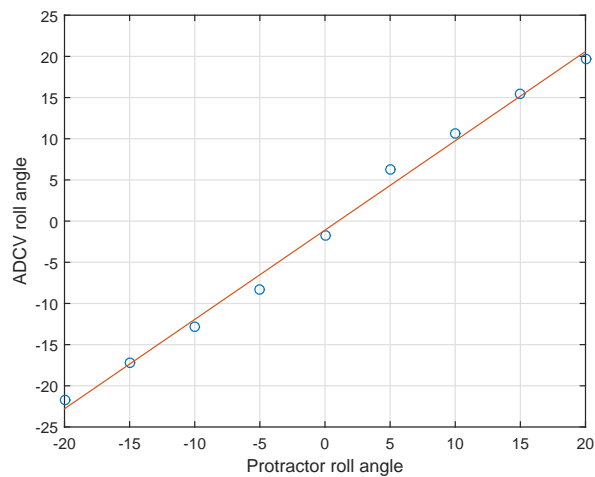


Figure 5.6: Protractor roll angles *versus* ADCV roll angles.

The pitch angles determination apparently has inferior quality. Figure 5.7 depicts the results, where the angles were measured in the range $[-15^\circ, 20^\circ]$. The linear curve shown is defined by $y = 0.8559x + 1.3574$. The standard deviation comprising all of the pitch measurements is 0.3246. The standard deviation of the pitch angle was smaller than the roll angle. On the other hand, the slope of the linear curve for the pitch angle was less closer to 1.

The last test applied to the ADCV system concerns the yaw angle also. However, in this case the angles were changed in steps of 1° in the range $[-5^\circ, 5^\circ]$. Figure 5.8 shows the graph obtained. The linear curve shown is defined by $y = 1.0055 * x + 0.7821$. The standard deviation comprising all of the yaw measurements is 0.0248° , which is approximately the same obtained in the first test. Table 5.1 summarize the results obtained with all the ADCV tests.

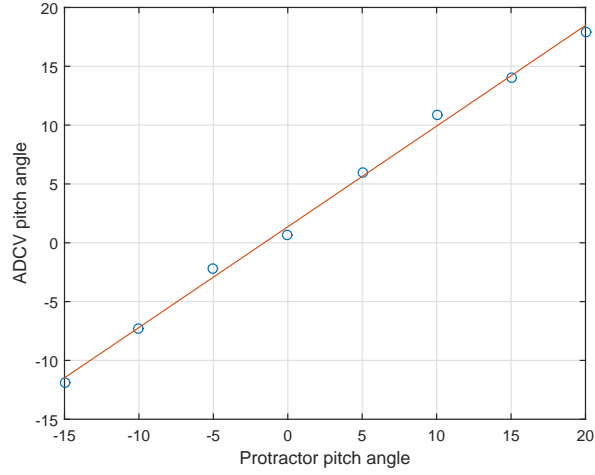


Figure 5.7: Protractor pitch angles *versus* ADCV pitch angles.

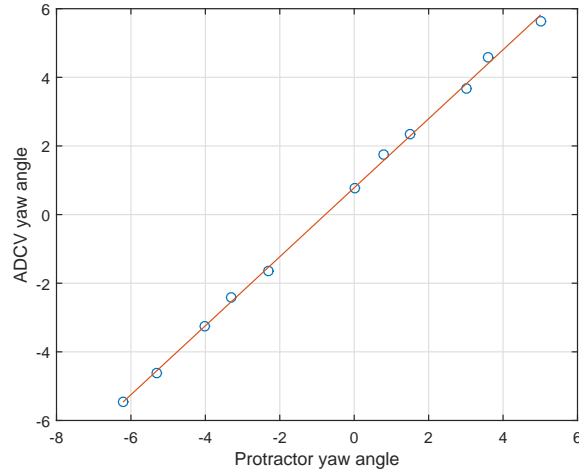


Figure 5.8: Protractor yaw angles *versus* ADCV yaw angles in steps of 1° .

Euler angle	Range	Step	Linear fitting	σ
Yaw	$[-25^\circ, 25^\circ]$	5°	$y = 0.9995x + 0.6549$	0.0266°
Row	$[-20^\circ, 20^\circ]$	5°	$y = 1.0845x - 1.0935$	0.4015°
Pitch	$[-15^\circ, 20^\circ]$	5°	$y = 0.8559x + 1.3574$	0.3246
Yaw (higher precision)	$[-5^\circ, 5^\circ]$	1°	$y = 1.0055 * x + 0.7821$	0.0248°

Table 5.1: Summary of the ADCV tests results.

The results presented in this section have demonstrated the high accuracy of ADCV. One of the difficulties posed by this system is the alignment of the camera accordingly to the tripod device. Even extremely small movements of the laser pointer led to great displacements with respect to the wall target. Also, the system is affected by different lighting conditions and the quality of the printed markers. Moreover, the tests performed were static, which means the markers were hold steady during the tests. Dynamic tests using the air-bearing table are intended to be realized in the future.

5.3 Algorithms implementation in C language

This section describes some practical aspects concerning the algorithms implementation in C language. The TRIAD algorithm and specially the USQUE filter employ several matrix operations. Unlike research tools such as MATLAB, such operations can not be readily performed in C language. Even though libraries are available to realize these operations, they are designed to be adopted in different situations, thereby not taking advantage of some specific characteristics of TRIAD and USQUE formulation. Moreover, in order to execute these algorithms in a MCU or limited processors, optimizations concerning the memory utilization and the execution speed must be realized. Such optimizations are only possible by designing libraries to attend the specific needs of the algorithms. Hence, two basic libraries were implemented in this work to be used as a background to TRIAD and USQUE algorithms. Figure 5.9 exhibits a diagram representing the algorithms and the libraries developed. The library in the bottom left was developed to perform the matrix operations, such as matrix product and addition, the Cholesky factorization, etc. The library represented in the bottom right of the figure contains functions to realize the conversion between attitude representations, quaternion operations and others.

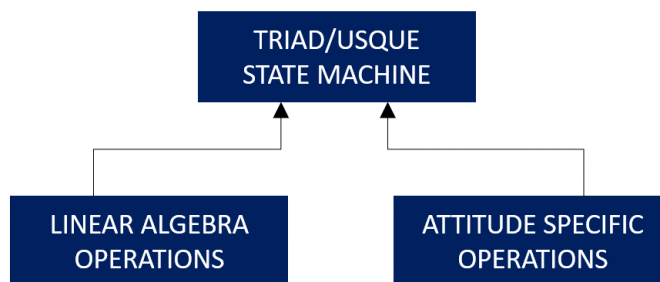


Figure 5.9: Schematic representation of the TRIAD and USQUE implementation and the two libraries developed.

In an algebraic point of view, one of the most distinguishable properties of USQUE and any other KF approach is the presence of several symmetric matrices. In order to save memory, the symmetric matrices of the Linear Algebra library are stored as lower triangular matrices, provided that the upper part of a symmetric matrix is identical to the lower part, except for its main diagonal. Figure 5.10 shows how the symmetric matrices are stored in memory. They are actually a sequence of contiguous address positions, where the first element of each line is taken at increased positions with respect to the first element of the previous line. Thus, a $N \times N$ symmetric matrix occupies only $(N+1)N/2$ memory positions. In Figure 5.10 the arrows represent C pointers; the matrix is stored as an array of these pointers.

Such definition of symmetric matrices as a sequence of bytes explores the benefits of the locality of access. This concept is based on the fact that if a memory position was accessed by the program, a contiguous position is likely to be accessed too. In processors

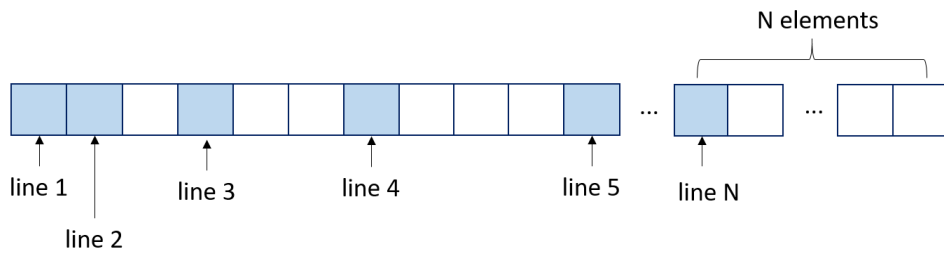


Figure 5.10: Schematic representation of the space occupied by a symmetric matrix in memory in the Linear Algebra library.

which possess cache memories the locality of access can reduce the execution time of a software [Patterson and Hennessy 2007]. Simple MCU such as the MSP430 do not have cache memories, however the algorithms implemented in C can be executed in several processors which can take advantage of locality of access. Figure 5.11 shows the same symmetric matrix represented in a logical manner. The blue arrows exemplify the manner in which the software must go through each of the matrix lines. Since the symmetric matrix lines can not be accessed in an usual way, several matrix operations functions were specifically designed to treat these matrices. Besides of saving space, the utilization of such definitions does not implies in a higher execution time for the matrix operations.

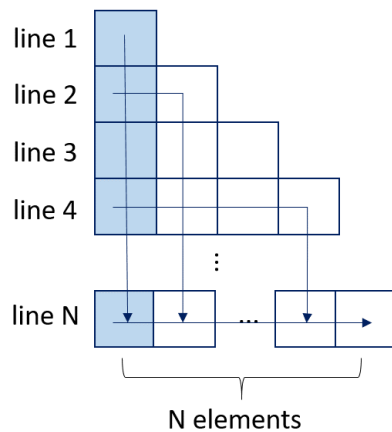


Figure 5.11: Logical representation of a symmetric matrix in the Linear Algebra library.

Of course, besides the above definition, the diagonal matrices are stored as vectors only in the library.

Some of the USQUE equations should not be directly applied in practical computations. Equation 5.1 reproduces the Kalman gain definition of the UKF, which is the same definition adopted in USQUE.

$$K_{k+1} = P_{k+1}^{xy} (P_{k+1}^{\partial\partial})^{-1} . \quad (5.1)$$

It is not recommended to calculate the inverse of $P_{k+1}^{\partial\partial}$ and then post multiply the result

with P_{k+1}^{xy} . An efficient way to compute the Kalman gain is by solving the linear system written in the unusual form $XA = B$. Therefore, the Kalman gain are the solutions of the linear system $K_{k+1}P_{k+1}^{\partial\partial} = P_{k+1}^{xy}$, which can be efficiently solved using the LU factorization of $P_{k+1}^{\partial\partial}$ [Meyer 2000]. This is the method adopted in the Linear Algebra library implemented.

Another KF equation that should not be directly applied in practice is reproduced in Equation 5.2.

$$P_{k+1}^+ = P_{k+1}^- - K_{k+1}P_{k+1}^{\partial\partial}K_{k+1}^T . \quad (5.2)$$

The Equation 5.2 used to update the covariance matrix is written in this way in papers and books to emphasize the fact that such operation preserves the positive semidefiniteness characteristic of P_{k+1} . In order to update P_{k+1} in software, a simpler version can be derived. Replacing the definition of the Kalman gain given in Equation 5.1 and considering that $P_{k+1}^{\partial\partial}$ is symmetric, the update rule of the covariance matrix can be written as in Equation 5.3.

$$P_{k+1}^+ = P_{k+1}^- - K_{k+1}(P_{k+1}^{xy})^T . \quad (5.3)$$

Numerical computations of P_{k+1} can violate its positive semidefiniteness characteristic. Is not easy to force such matrix to always be positive semidefinite, however a simple approach can be adopted to avoid this problem. The matrix P_{k+1} is made symmetric after each update. Thus, although this not ensures the positive semidefiniteness, since every of these matrices is symmetric, the operation to force it to be symmetric can be helpful. The procedure adopted is given by Equation 5.4, where the matrix B is always symmetric regardless of A .

$$B = \frac{(A + A^T)}{2} . \quad (5.4)$$

Another error that the finite-precision arithmetic of processors can introduce. It consists in the violation of the unit norm constraint for quaternions of rotation or the orthogonality property of the attitude matrices. In order to ensure these constraints are obeyed, the brute-force normalization defined by Equation 5.5 is applied to the quaternions estimates. A similar procedure for DCM is not easy to apply. Is recommended that the DCM is converted into a quaternion and the normalization is applied in the sequence. If is necessary, the normalized quaternion can then be converted back into a DCM.

All the functions of the implemented libraries were validated using MATLAB. That is, the same operations performed by the C software was realized in MATLAB and the results were compared.

$$\mathbf{q} = \frac{\mathbf{q}'}{\|\mathbf{q}'\|} . \quad (5.5)$$

5.4 Algorithms tests and results

Figure 5.12 illustrates the methodology adopted in the experiments using the ABACUS OBC. The air-bearing table requires that the devices mounted in it use wireless communication, considering the table must be free to rotate. The ABACUS board does not possess a built-in wireless communication peripheral, thus a Raspberry Pi 3 Model B board was utilized to establish such interface. This board have a built-in WiFi chip able to connect to the laboratory network. The connection between ABACUS and the Raspberry is not direct, nevertheless. It is established through the UART-USB converter CP2102. In the software point of view, a software was implemented in Raspberry using the *termios.h* library to access the CP2102 device driver. In ABACUS the functions provided by the Gauss Srl manufacturer was used to transmit and receive data. A socket network application was also implemented in C for the Linux Operating System of the Raspberry. The socket network allows the data to be directly transmitted to MATLAB and commands to be received from it.

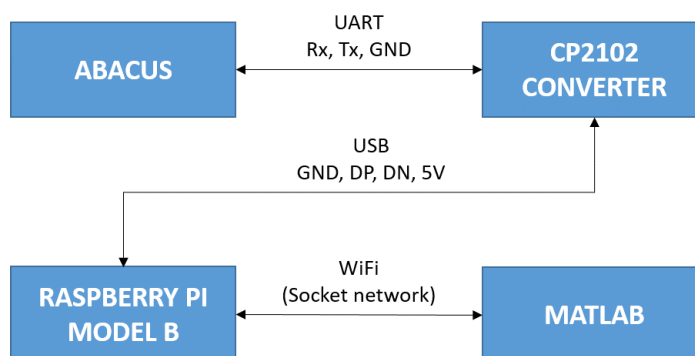


Figure 5.12: Communication architecture used in the experiments with ABACUS.

Figure 5.13 exhibits a picture of ABACUS, the CP2102 converter and the Raspberry mounted together in a single structure. The Raspberry was power supplied by a battery mounted in the table. Two pins of the Raspberry provided the power source to feed ABACUS. The red and black wires shown in Figure 5.13 provides the power source to ABACUS. The purple, green and blue wires establish the UART connection between the ABACUS UART and the CP2102 converter, which in turns is connected to one of the Raspberry USB ports.

The left side of Figure 5.14 shows the ABACUS-Raspberry structure with an ArUco marker fixed on its top. The right side of this figure shows the ABACUS-Raspberry structure mounted on the air-bearing table surface, which allows the ADCV to estimate the device attitude.

Before the tests realization the air-bearing table was manually balanced and the X coil of the Helmholtz cage was turned on with 6A. Then, the table was put into motion in order that it could rotate slowly.

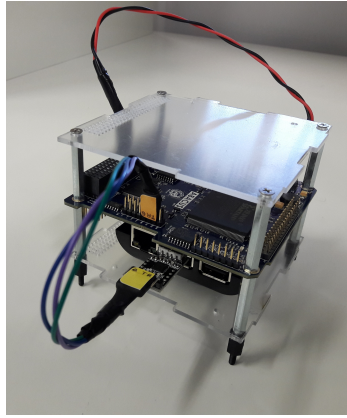


Figure 5.13: ABACUS, the CP2102 converter and the Raspberry mounted in a single structure.

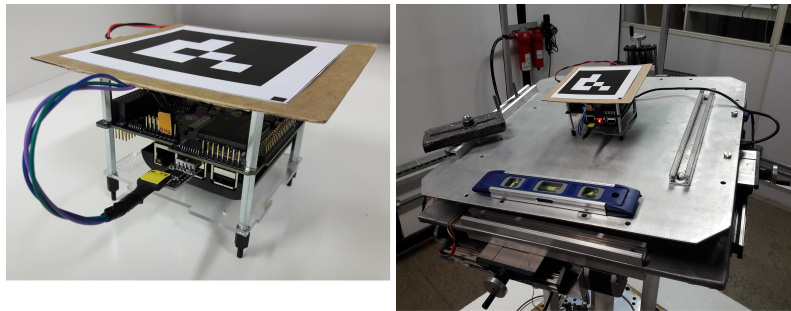


Figure 5.14: ABACUS-Raspberry structure with an ArUco marker in detail (left) and mounted in the air-bearing table (right).

After these steps, the IMU data were collected. Unfortunately, the experiments executed in laboratory showed that the interval between consecutive data acquisitions was of about 1 second. This was due to the protocol adopted to transmit the data from ABACUS to MATLAB and the limitations of the UART libraries provided by the ABACUS manufacturer. With such a high sampling interval, the attitude algorithms could not be adequately tested. Despite of the aforementioned problem, the data obtained by ABACUS IMU was received by MATLAB through the socket network and saved. Furthermore, the IMU data collected were used with TRIAD and USQUE algorithms implemented, their performances were also compared after executing them in different platforms.

Figure 5.15 shows the graph of ABACUS gyro measurements in degrees per second. Since the table was balanced, the higher values of angular velocities were measured by the gyro Z axis, that is, the rotation occurred mainly about the Z axis, as shown in the graph. Figure 5.16 shows the measurements obtained by ABACUS accelerometer in meters per second squared. The Z axis values shown in Figure 5.16 are mainly due to the gravitational acceleration. Figure 5.17 shows the measurements obtained by ABACUS magnetometer. While the table was rotating, mainly about the Z axis, the magnetometer X and Y axes measured the magnetic field produced by the cage. This is the reason why the Z axis measurements are smaller than the others.

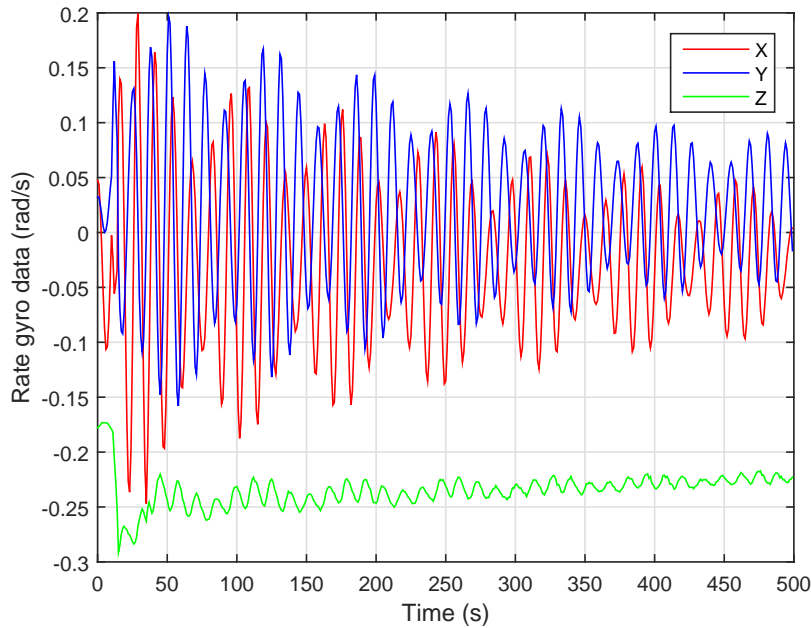


Figure 5.15: Measurements obtained by ABACUS gyro.

The data collected by the accelerometer and the magnetometer were used as inputs to TRIAD algorithm to determine the table attitude. Figure 5.18 shows the values obtained by TRIAD. At each sample time, the algorithm calculates the DCM of the table attitude. In Figure 5.18 only the three entries of the DCM main diagonal are shown.

Figure 5.19 shows the values of the q_1 component of the table quaternion of rotation. The results do not seem to contain valuable information about the attitude, probably due to the high sampling interval mentioned before. However, the convergence of the correspondent variance values, which are shown in Figure 5.20, confirms the convergence of the algorithm.

Figure 5.21 shows the values of the gyro bias components estimated by USQUE. The correspondent variance values shown in Figure 5.22 demonstrate the algorithm convergence.

The TRIAD and USQUE algorithms implemented in C were executed in the operating systems Windows 10 and Linux (Ubuntu 16.05) – in the same hardware – besides the MSP430 of ABACUS, of course. In addition, the execution performances were compared to the same algorithms implemented as MATLAB scripts. Table 5.2 compares the size of the algorithms executable for each platform and the mean execution time. This last parameter correspond to the time execution of only one sample time.

The executable for the MSP430 has the greatest size because it incorporates all the libraries included in the implementation, while in the other cases is possible to use libraries linked at execution time. Also, the MSP430 does not support floating point operations in hardware, therefore, they are performed through software thereby increasing the size of the final executable files. Furthermore, is important to mention that the execution of USQUE in MSP430 required the utilization of double floating point precision (64 bits), otherwise the algorithm fails during Cholesky factorization or other linear algebra operation.

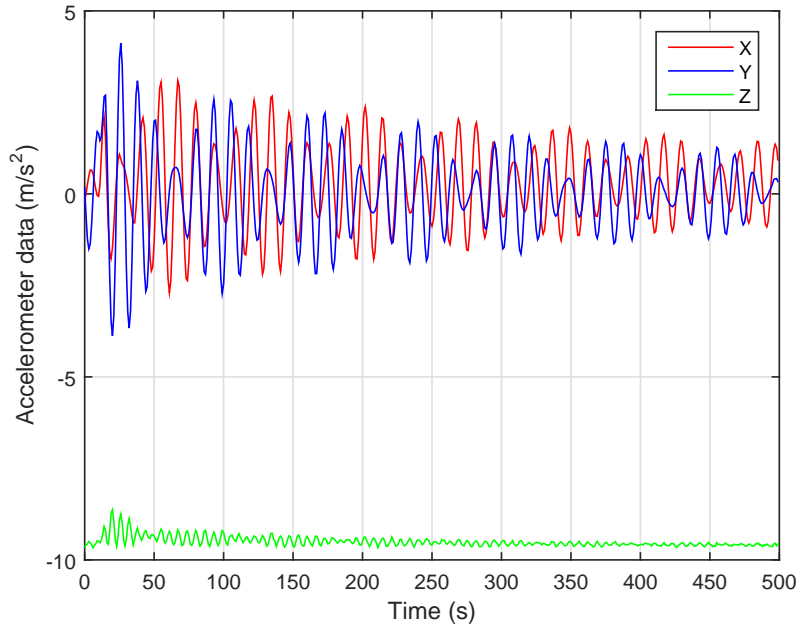


Figure 5.16: Measurements obtained by ABACUS accelerometer.

Platform	TRIAD (size)	USQUE (size)	TRIAD (exec. time)	USQUE (exec. time)
W10 VS Intel i7 2.5 GHz 8GB	58 kB	67 kB	0.004 ms	0.029 ms
Linux GCC Intel i7 2.5 GHz 8GB	99.8 kB	120.6 kB	0.0007 ms	0.019 ms
MATLAB Intel i7 2.5 GHz 8GB	-	-	0.098 ms	1.765 ms
CCS MSP430 25 MHz 16kB	387.3 kB	587.5 kB	1.674 ms	64.45 ms

Table 5.2: Algorithms performances and characteristics.

The execution times described in Table 5.2 demonstrate that the algorithms are able to be adopted in real missions. On the other hand, USQUE execution time is quite high, making it not recommendable in situations where the spacecraft rotates rapidly. However, TRIAD and USQUE could be employed together in a satellite. The faster TRIAD could be executed right after launch, when the spacecraft normally rotates with high angular velocities and an attitude determination with great precision is not critical. After stabilization – that is, the detumbling process – the satellite tends to move slowly, thus the more robust USQUE could be used to estimate the attitude with a higher precision.

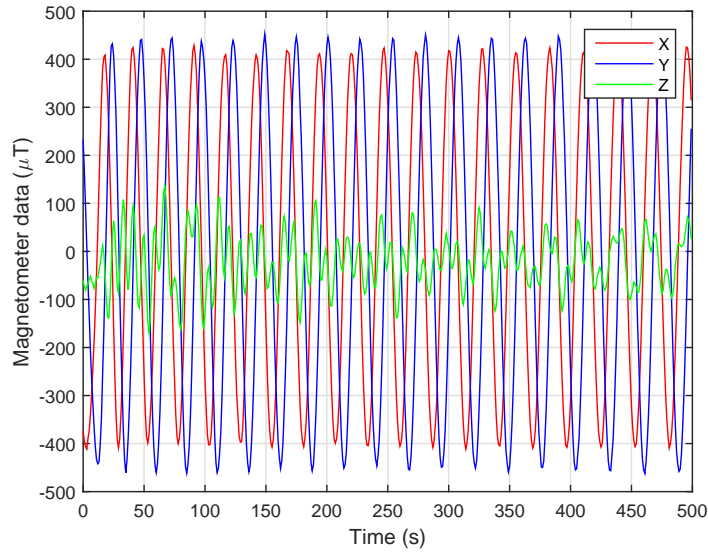


Figure 5.17: Measurements obtained by ABACUS magnetometer.

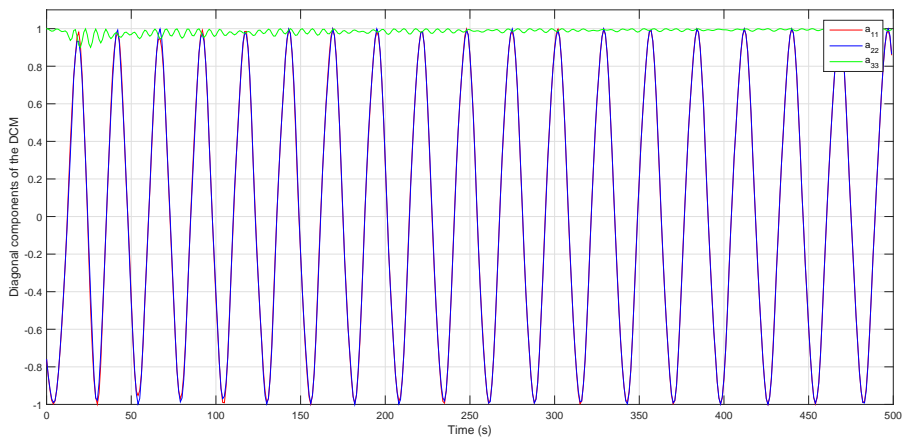


Figure 5.18: Values of the table DCM main diagonal.

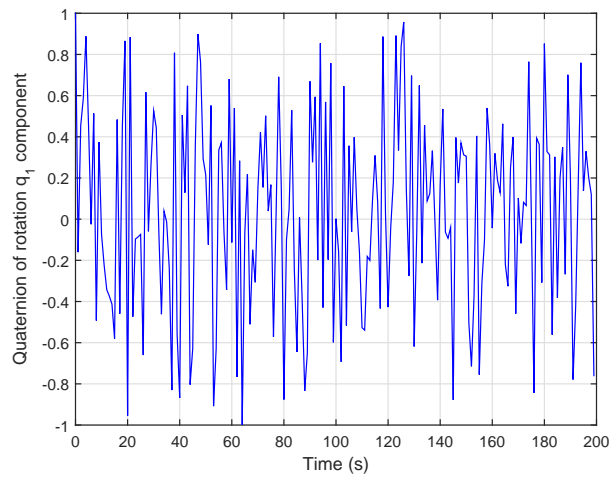


Figure 5.19: q_1 component of the table quaternion of rotation obtained by USQUE.

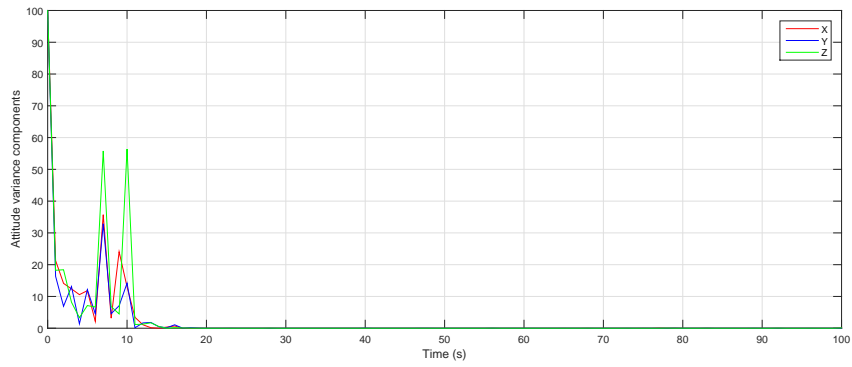


Figure 5.20: Elements of the covariance matrix P correspondent to the table quaternion of rotation.

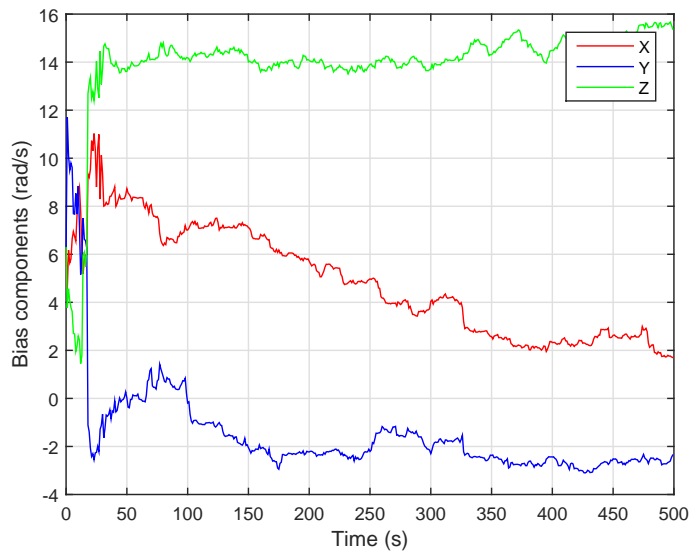


Figure 5.21: Gyro bias values estimated by USQUE.

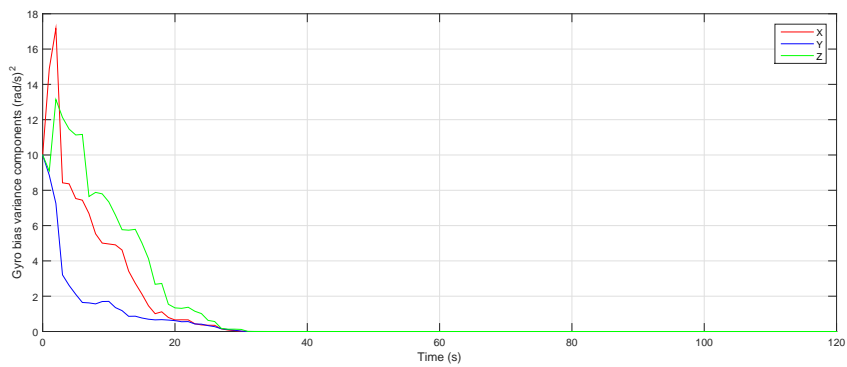


Figure 5.22: Elements of the covariance matrix P correspondent to the gyro bias.

Chapter 6

Conclusion

In this work different attitude determination techniques were implemented. The air-bearing table and the Helmholtz cage of LAICA facility were employed to perform the experiments. One of the methods implemented was the ADCV system, based on computer vision. This method is executed using a web camera and a PC completely separated from the air-bearing table and the CubeSat devices mounted on its surface. The other methods were the TRIAD and USQUE algorithms implemented in C language, making them suitable for a MCU such as the MSP430 of the CubeSat onboard computer ABACUS, which is mounted on the air-bearing table.

The camera frame of the ADCV system was adjusted with a local reference-system physically established using the gravitational acceleration and a target fixed at the laboratory, which works as the North pole for a NED system. The Euler angles were used to evaluate ADCV. This method produced accurate results with standard deviations smaller than 0.5° for all the three angles. However, the adjustment of the camera frame was difficult and not always effective. As a future work, an algorithm for an automatic adjustment of the camera frame through software could be implemented. Finally, the ADCV system has been successfully used already in several works at LAICA. Considering that different attitude algorithms employ the same IMU sensors, the ADCV system provides independent attitude measurements obtained by a completely different approach. This allows an even evaluation of several attitude determination algorithms which use body mounted sensors.

The TRIAD and USQUE algorithms implemented in C, unlike those written using MATLAB or Python, are flexible and can be executed in several processors and operating systems, such as ABACUS, a Raspberry Pi, a Linux or Windows PC, etc. The performance tests have shown that those algorithms can be utilized by CubeSats which use ABACUS as the OBC. The algorithms execution time for each sample instant establish a lower bound for the sampling interval, which in turns must be small enough in view of the spacecraft dynamics.

The TRIAD execution time of about 1.7ms demonstrate the possibility of its utilization when the satellite rotates rapidly, which usually occurs right after launch, before the detum-

bling process. The interval of 1.7ms allows a sampling rate of about 588 Hz, being adequate for a speed of about 294 rotations per second (RPS), accordingly to the Nyquist's limit. Also, during detumbling a high precision attitude determination is not critical, which makes TRIAD very suitable for this phase.

The USQUE execution time of about 64.5ms is quite high, posing an upper bound of about 15.5 Hz for the sampling rate, which is adequate for a speed of about 31 RPS. Despite of this fact, USQUE is still suitable for a spacecraft after detumbling, since a speed of 31 RPS is very high for a stabilized satellite. A high precision attitude estimation is a more critical requirement after stabilization, when the satellite initiates the regular operation for which it was designed. Thus, the more robust USQUE method would be appropriate for this phase.

The ABACUS FPGA is a device with high processing power which have not been used in any work at LAICA. Moreover, the FPGA is highly suitable to perform matrix operations, which are abundant in algorithms such as USQUE. Thus, its utilization in future works is recommended since it could significantly increase the algorithms performances, specially USQUE.

The LAICA facility is still under development, despite of the fact that several improvements have been made over the approximate three years of its existence. The published works and this masters dissertation itself prove the advances achieved, even though new features are under planning. Among others, these features include: a complete control of the magnetic field generated by the EMFS, the test of sun sensors, and the implementation of attitude control systems using the reaction wheels and other actuators.

REFERENCES

- [AEB 2018] AEB (2018). Agência Espacial Brasileira. Retrieved from <http://www.aeb.gov.br/espaco-educacao-e-tecnologia/serpens/> (May 25, 2018).
- [Aguirre 2015] Aguirre, L. A. (2015). *Introdução à identificação de sistemas: técnicas lineares e não lineares: teoria e aplicação*. Belo Horizonte: Ed. UFMG, 4th edition.
- [Bó 2007] Bó, A. P. L. (2007). Desenvolvimento de um Sistema de Localização 3D para Aplicação em Robôs Aéreos. Master's thesis, Universidade de Brasília/PPGENE.DM - 306A.
- [Bar-Shalom et al. 2001] Bar-Shalom, Y., Li, X. R., and Kirubarajan, T. (2001). *Estimation with applications to tracking and navigation*. John Wiley & Sons.
- [Batista et al. 2017] Batista, D. S., Granziera, F., Tosin, M. C., and de Melo, L. F. (2017). Three-axial helmholtz coil design and validation for aerospace applications. *IEEE Transactions on Aerospace and Electronic Systems*.
- [Black 1964] Black, H. D. (1964). A passive system for determining the attitude of a satellite. *AIAA journal*, 2(7):1350–1351.
- [Bouwmeester et al. 2008] Bouwmeester, J., Gill, E. K. A., and Verhoeven, C. J. M. (2008). Advancing Nano-satellite Platforms: The Delfi Program. In *59th International Astronautical Congress: IAC 2008, 29 September-3 October 2008, Glasgow, Scotland*.
- [Bouwmeester and Guo 2010] Bouwmeester, J. and Guo, J. (2010). Survey of worldwide pico- and nanosatellite missions, distributions and subsystem technology. *Acta Astronautica*, 67(7):854–862.
- [Bradski and Kaehler 2008] Bradski, G. and Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc."
- [Cai et al. 2011] Cai, G., Chen, B. M., and Lee, T. H. (2011). *Unmanned rotorcraft systems*. Springer Science & Business Media.
- [Cheon and Kim 2007] Cheon, Y. and Kim, J. (2007). Unscented filtering in a unit quaternion space for spacecraft attitude estimation. In *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on*, pages 66–71. IEEE.

- [Chesi et al. 2013] Chesi, S., Gong, Q., Pellegrini, V., Cristi, R., and Romano, M. (2013). Automatic mass balancing of a spacecraft three-axis simulator: Analysis and experimentation. *Journal of Guidance, Control, and Dynamics*, 37(1):197–206.
- [Crassidis 2003] Crassidis, J. L. (2003). Unscented Filtering for Spacecraft Attitude Estimation. *Journal of Guidance, Control, and Dynamics*, 26(4):536–542.
- [Crassidis et al. 2007] Crassidis, J. L., Markley, F. L., and Cheng, Y. (2007). Survey of Nonlinear Attitude Estimation Methods. *Journal of Guidance, Control, and Dynamics*, 30(1):12–28.
- [Do Carmo 2010] Do Carmo, M. P. (2010). *Geometria diferencial de curvas e superfícies*. SBM.
- [FAPESP 2016] FAPESP (2016). Fundação de Apoio à Pesquisa do Estado de São Paulo. Retrieved from <http://revistapesquisa.fapesp.br/en/2016/10/03/nanosatellite-mission-accomplished/> (May 25, 2018).
- [Forsyth and Ponce 2003] Forsyth, D. A. and Ponce, J. (2003). *Computer vision: a modern approach*. Prentice-Hall.
- [Garrido-Jurado et al. 2014] Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. J., and Marín-Jiménez, M. J. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292.
- [Guimarães et al. 2017] Guimarães, F. C., Silva, R. C., Loiola, J. V. L., Borges, G. A., Borges, R. A., Battistini, S., and Cappelletti, C. (2017). Aplicação do Filtro de Kalman para a Determinação de Atitude de Plataforma de Testes de Pequenos Satélites. *XIII Simpósio Brasileiro de Automação Inteligente - SBAI*.
- [Hajek 2015] Hajek, B. (2015). *Random processes for engineers*. Cambridge university press.
- [Heidt et al. 2000] Heidt, H., Puig-Suari, J., Moore, A. S., Nakasuka, S., and Twiggs, R. J. (2000). CubeSat: A new Generation of Picosatellite for Education and Industry Low-Cost Space Experimentation. *14th Annual/USU Conference on Small Satellites*.
- [Ivanov et al. 2018] Ivanov, D. S., Koptev, M. D., Mashtakov, Y. V., Ovchinnikov, M. Y., Proshunin, N., Tkachev, S., Fedoseev, A., and Shachkov, M. (2018). Laboratory facility for microsatellite mock-up motion simulation. *Journal of Computer and Systems Sciences International*, 57(1):115–130.
- [Julier et al. 2000] Julier, S., Uhlmann, J., and Durrant-Whyte, H. F. (2000). A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on automatic control*, 45(3):477–482.

- [Julier and Uhlmann 1994] Julier, S. J. and Uhlmann, J. K. (1994). A general method for approximating nonlinear transformations of probability distributions. Technical report, Technical report, Robotics Research Group, Department of Engineering Science, University of Oxford.
- [Julier and Uhlmann 1997] Julier, S. J. and Uhlmann, J. K. (1997). New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, pages 182–194. International Society for Optics and Photonics.
- [Julier and Uhlmann 2004] Julier, S. J. and Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422.
- [Julier et al. 1995] Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. F. (1995). A new approach for filtering nonlinear systems. In *American Control Conference, Proceedings of the 1995*, volume 3, pages 1628–1632. IEEE.
- [Kuipers 1999] Kuipers, J. B. (1999). *Quaternions and rotation sequences*, volume 66. Princeton University Press.
- [Loiola et al. 2018] Loiola, J. V. L., van der Ploeg, L. C., Silva, R. C., Guimarães, F. C., Borges, R. A., Borges, G. A., Battistini, S., and Cappelletti, C. (2018). 3 axis simulator of the Earth magnetic field. *IEEE Aerospace Conference*.
- [Markley 2003] Markley, F. L. (2003). Attitude Error Representations for Kalman Filtering. *Journal of Guidance, Control and Dynamics*, 26(2):311–317.
- [Maybeck 1979] Maybeck, P. S. (1979). *Stochastic models, estimation, and control*, volume 1. Academic press.
- [Menegaz, H. M. T. 2016] Menegaz, H. M. T. (2016). *Unscented Kalman Filtering on Euclidean and Riemannian Manifolds*. PhD thesis, Universidade de Brasília/PPGEA.TD-109/16.
- [Meyer 2000] Meyer, C. D. (2000). *Matrix analysis and applied linear algebra*, volume 71. Siam.
- [NASA 2015] NASA (2015). National Aeronautics and Space Administration. Retrieved from <https://www.nasa.gov/content/what-are-smallsats-and-cubesats> (May 22, 2018).
- [NOAA 2015a] NOAA (2015a). National Oceanic and Atmospheric Administration. Retrieved from <https://www.ngdc.noaa.gov/geomag/WMM/DoDWMM.shtml> (October 14, 2017).
- [NOAA 2015b] NOAA (2015b). National Oceanic and Atmospheric Administration. Retrieved from <https://www.ngdc.noaa.gov/geomag/WMM/limit.shtml> (October 14, 2017).

- [Oliveira et al. 2013] Oliveira, G. F., Ishihara, J. Y., Borges, R. A., Ferreira, H. C., Kula-
bukhov, A. M., Larin, V. A., and Belikov, V. V. (2013). A low-cost attitude determination
and control system for the uys-1 nanosatellite. In *Aerospace Conference, 2013 IEEE*,
pages 1–14. IEEE.
- [Oliveira et al. 2014] Oliveira, G. F., Nehme, P. H. D., and Cappelletti, C. (2014). Analysis
and simulation of attitude determination and control for the serpens nanosatellite. In
Proceedings of 2nd IAA Conference on Dynamics and Control of Space Systems, Rome,
Italy, pages 1–20.
- [Ovchinnikov et al. 2007] Ovchinnikov, M. Y., Ilyin, A. A., Kupriynova, N. V., Penkov,
V. I., and Selivanov, A. S. (2007). Attitude dynamics of the first russian nanosatellite
tns-0. *Acta Astronautica*, 61(1-6):277–285.
- [Patterson and Hennessy 2007] Patterson, D. A. and Hennessy, J. L. (2007). Computer or-
ganization and design. *Morgan Kaufmann*, pages 474–476.
- [Samuels 2014] Samuels, M. A. (2014). The Design and Testing of a Three-Degree-of-
Freedom Small Satellite Simulator Using a Linear Controller with Feedback Linearization
and Trajectory Generation. Master’s thesis, Utah State University.
- [Santilli et al. 2018] Santilli, G., Vendittozzi, C., Cappelletti, C., Battistini, S., and Gessini,
P. (2018). Cubesat constellations for disaster management in remote areas. *Acta Astro-
nautica*, 145:11–17.
- [Schaub and Junkins 1996] Schaub, H. and Junkins, J. L. (1996). Stereographic orientation
parameters for attitude dynamics: A generalization of the rodrigues parameters. *Journal*
of the Astronautical Sciences, 44(1):1–19.
- [Schutz et al. 2004] Schutz, B., Tapley, B., and Born, G. H. (2004). *Statistical orbit deter-
mination*. Academic Press.
- [Schwartz et al. 2003] Schwartz, J. L., Peck, M. A., and Hall, C. D. (2003). Historical
review of air-bearing spacecraft simulators. 26.
- [Shuster 1993] Shuster, M. D. (1993). A Survey of Attitude Representations. *The Journal*
of the Astronautical Sciences, 41(4):439–517.
- [Shuster 2004] Shuster, M. D. (2004). Deterministic Three-Axis Attitude Determination.
The Journal of the Astronautical Sciences, 52(3):405–419.
- [Shuster 2008] Shuster, M. D. (2008). The nature of the quaternion. *The Journal of the*
Astronautical Sciences, 56(3):359–373.
- [Sidi 1997] Sidi, M. J. (1997). *Spacecraft dynamics and control: a practical engineering*
approach, volume 7. Cambridge University Press.

- [Silva et al. 2018] Silva, R. C., Guimarães, F. C., Loiola, J. V. L., Borges, R. A., Battistini, S., and Cappelletti, C. (2018). Tabletop testbed for attitude determination and control of nanosatellites. *Journal of Aerospace Engineering*.
- [Silva et al. 2016] Silva, R. C., Rodrigues, U. A., Borges, R. A., Sampaio, M., Beghelli, P., da Costa, S. G. P., Popov, B. T., Battistini, S., and Cappelletti, C. (2016). A testbed for attitude determination and control of spacecrafts. *II Latin American IAA CubeSat Workshop*.
- [Silva and Rodrigues 2015] Silva, R. C. d. and Rodrigues, U. A. (2015). Simulador de sistema de determinação e controle de atitude de pequenos satélites.
- [Simon 2006] Simon, D. (2006). *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons.
- [Stuelpnagel 1964] Stuelpnagel, J. (1964). On the parametrization of the three-dimensional rotation group. *SIAM review*, 6(4):422–430.
- [UnB 2015] UnB (2015). Universidade de Brasília. Retrieved from <https://unbciencia.unb.br/exatas/39-engenharia-aeroespacial/408-unb-desenvolve-nanossatelite-para-analise-ambiental/> (Jun 17, 2018).
- [Villela et al. 2018] Villela, T., Costa, C. A., Brandão, A. M., Bueno, F. T., and Leonardi, R. (2018). Towards the thousandth cubesat: a statistical overview. *Submitted to Journal of Aerospace Engineering*.
- [Wertz 1978] Wertz, J. R., editor (1978). *Spacecraft attitude determination and control*, volume 73. Kluwer Academic Publishers.