



**COMPARISONS OF FILTERING METHODS FOR ATTITUDE
DETERMINATION OF MICROSATELLITES**

JOÃO VICTOR LOPES DE LOIOLA

**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**

**COMPARISONS OF FILTERING METHODS FOR ATTITUDE
DETERMINATION OF MICROSATELLITES**

JOÃO VICTOR LOPES DE LOIOLA

**Orientador: PROF. DR. RENATO ALVES BORGES, ENE/UNB
Coorientador: PROF. DR. JOÃO YOSHIYUKI ISHIHARA, ENE/UNB**

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA

**PUBLICAÇÃO PPGENE.DM - 700/2018
BRASÍLIA-DF, 31 DE JULHO DE 2018.**

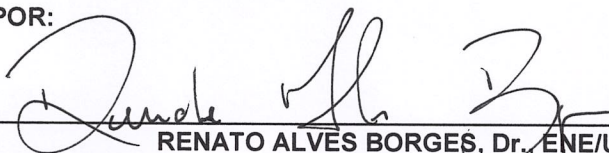
UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

COMPARISONS OF FILTERING METHODS FOR ATTITUDE
DETERMINATION OF MICROSATELLITES

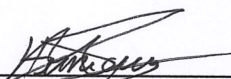
JOÃO VICTOR LOPES DE LOIOLA

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.

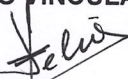
APROVADA POR:



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



HENRIQUE MARRA TAIRA MENEGAZ, Dr., FGA/UNB
(EXAMINADOR INTERNO VINCULADO AO PROGRAMA)



HÉLIO KOITI KUGA, Dr., INPE
(EXAMINADOR EXTERNO)

FICHA CATALOGRÁFICA

LOIOLA, JOÃO VICTOR LOPES DE

COMPARISONS OF FILTERING METHODS FOR ATTITUDE DETERMINATION OF MICROSATELLITES

2018xvii, 108p., 201x297 mm

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

Dissertação de Mestrado - Universidade de Brasília

Faculdade de Tecnologia - Departamento de Engenharia Elétrica

1. EKF

5. Unscented Filter

2. UKF

6. Attitude Determination

3. USQUE

7. Kalman Filter

4. Riemannian Filter

I. ENE/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

LOIOLA, J. V. L. (2018) COMPARISONS OF FILTERING METHODS FOR ATTITUDE DETERMINATION OF MICROSATELLITES. Dissertação de Mestrado em Engenharia Elétrica, Publicação PPGENE.DM-700/2018, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 108p.

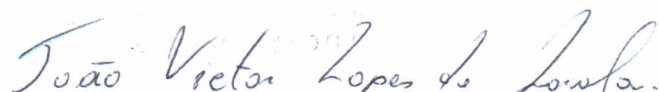
CESSÃO DE DIREITOS

AUTOR: JOÃO VICTOR LOPES DE LOIOLA

TÍTULO: COMPARISONS OF FILTERING METHODS FOR ATTITUDE DETERMINATION OF MICROSATELLITES.

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.



JOÃO VICTOR LOPES DE LOIOLA

SQN 405, Brasília.

Agradecimentos

Primeiramente gostaria de agradecer a Deus pelo dom da vida. Em segundo lugar aos meus pais e a minha namorada, Rebeca, por todo apoio e paciência durante essa jornada. Gostaria também de agradecer aos professores Renato, Simone, Chantal, João Yoshiyuki Ishihara, Henrique e Geovany por todo conhecimento transmitido e por toda a paciência e tempo dedicado. Além disso gostaria de dizer um muito obrigado a todos os meus colegas de laboratório, Rodrigo, Fernando, Lucas e Igor pelo companheirismo e por tudo que eu pude aprender com eles. Finalmente, gostaria de agradecer a todos os meu amigos do LARA pelo acolhimento assim que cheguei.

Abstract

Small satellite subsystems are in constant improvement. In order to address this issue, the Laboratory of Application and Innovation in Aerospace Science (LAICA), located at the University of Brasilia, developed a nanosatellite test platform. This platform can be divided in four main subsystems: actuation and balancing systems, Helmholtz cage and attitude determination system. The attitude determination system is responsible for determining the body orientation, and consequently it has a direct impact on most of control strategies designed for nanosatellites. In order to improve orientation data, which are normally embedded with noise, four filtering strategies were employed. The utilized filters were: the Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), Unscented Quaternion Estimator (USQUE) and Riemannian-Spheric Additive Unscented Kalman Filter (RiSAdUKF). These filters were evaluated with experimental data and the results were used to indicate the appropriated filter and attitude representation for different real situations.

Resumo

Subsistemas de pequenos satélites estão em constante desenvolvimento. Com o objetivo de contribuir com essas inovações, o Laboratório de Aplicação e Inovação em Pesquisas Aeroespaciais (LAICA), localizado na Universidade de Brasília, desenvolveu uma plataforma de testes de nanossatélites. Essa plataforma pode ser dividida em quatro subsistemas principais: sistemas de balanceamento e atuação, gaiola de Helmholtz e o sistema de determinação de orientação. O sistema de determinação de orientação é responsável por determinar a orientação do corpo e afeta diretamente em muitas das estratégias de controle empregadas em nanossatélites. Com o objetivo de melhorar os dados de orientação, que normalmente são corrompidos com ruído, quatro técnicas de filtragem foram empregadas. Os filtros utilizados foram: o Filtro de Kalman Extendido (EKF), Filtro de Kalman Unscented (UKF), Estimador de Quatérnio Unscented (USQUE) e o Filtro de Kalman Unscented Aditivo Esférico Riemanniano (RiSAdUKF). Esses filtros foram avaliados com dados experimentais e os resultados foram utilizados para indicar qual o filtro mais apropriado para diferentes condições experimentais.

SUMÁRIO

| | | |
|----------|---|-----------|
| 1 | INTRODUCTION | 1 |
| 1.1 | LAICA AIR-BEARING SPACECRAFT SIMULATOR | 2 |
| 1.1.1 | HELMHOLTZ CAGE | 3 |
| 1.1.2 | TABLETOP AIR-BEARING | 4 |
| 1.1.3 | ACTUATOR | 4 |
| 1.1.4 | ATTITUDE DETERMINATION SYSTEM | 5 |
| 1.2 | MOTIVATION | 7 |
| 1.3 | PROBLEM DEFINITION | 7 |
| 1.4 | MORE APPLICATIONS | 8 |
| 1.5 | RESULTS AND CONTRIBUTIONS | 9 |
| 1.6 | MANUSCRIPT PRESENTATION | 10 |
| 2 | ATTITUDE REPRESENTATION AND FILTERING STRATEGIES | 12 |
| 2.1 | ATTITUDE AND REFERENCE SYSTEMS | 12 |
| 2.1.1 | BODY FRAME | 12 |
| 2.1.2 | NORTH-EAST-DOWN SYSTEM | 13 |
| 2.2 | EULER ANGLES | 13 |
| 2.2.1 | ROTATION MATRIX | 13 |
| 2.2.2 | TILT-COMPENSATED eCOMPASS ALGORITHM | 14 |
| 2.2.3 | KINEMATIC EQUATION | 17 |
| 2.3 | QUATERNIONS | 18 |
| 2.3.1 | QUATERNION ALGEBRA | 18 |
| 2.3.2 | ROTATION MATRIX FROM INERTIAL TO BODY FRAME | 19 |
| 2.3.3 | KINEMATIC EQUATION | 20 |
| 2.4 | EXTENDED KALMAN FILTER (EKF) | 21 |
| 2.5 | UNSCENTED KALMAN FILTER (UKF) | 24 |
| 2.6 | PROCESS AND MEASUREMENT MODELS UTILIZED IN USQUE AND RIEMANNIAN FILTER | 26 |
| 2.6.1 | PROCESS MODEL | 26 |
| 2.6.2 | MEASUREMENT MODEL (USQUE) | 28 |
| 2.6.3 | MEASUREMENT MODEL (RISAdUKF) | 29 |
| 2.6.4 | RODRIGUES PARAMETERS | 29 |

| | | |
|----------|---|-----------|
| 2.7 | UNSCENTED QUATERNION ESTIMATOR(USQUE) | 29 |
| 2.7.1 | ATTITUDE PROPAGATION | 31 |
| 2.7.2 | PREDICTION CALCULATIONS | 31 |
| 2.7.3 | CORRECTION STEP..... | 32 |
| 2.8 | RIEMANNIAN-SPHERIC ADDITIVE UNSCENTED KALMAN FIL- TER(RISADUKF)..... | 33 |
| 3 | SIMULATOR DYNAMIC MODEL..... | 36 |
| 3.1 | ANGULAR MOMENTUM FIRST DERIVATIVE REFERRING TO THE CEN- TER OF ROTATION | 40 |
| 3.2 | DYNAMIC EQUATIONS | 41 |
| 3.3 | NUMERICAL SIMULATIONS AND EXPERIMENTAL VERIFICATION | 43 |
| 3.3.1 | SIMULATION | 44 |
| 3.3.2 | EXPERIMENTAL VERIFICATION | 46 |
| 4 | SYSTEM ARCHITECTURE..... | 48 |
| 4.1 | SENSORS | 48 |
| 4.1.1 | 9DOF IMU FROM ADAFRUIT | 48 |
| 4.1.2 | 9DOF IMU FROM YOST LABS..... | 49 |
| 4.2 | HARDWARE CONFIGURATION | 51 |
| 4.2.1 | HARDWARE CONFIGURATION FOR ADAFRUIT IMU | 51 |
| 4.2.2 | HARDWARE CONFIGURATION FOR YOST LABS IMU | 53 |
| 4.3 | CALIBRATION PROCEDURES..... | 54 |
| 4.3.1 | CALIBRATION PROCEDURES FOR ADAFRUIT IMU AND HMR2300 | 54 |
| 4.3.2 | CALIBRATION PROCEDURES FOR YOST LABS IMU | 56 |
| 5 | TESTS AND RESULTS | 57 |
| 5.1 | MAGNETIC FIELD VARIATION EFFECTS ON SENSOR DATA | 58 |
| 5.1.1 | EXPERIMENTAL SETUP FOR FILTER EVALUATION | 60 |
| 5.1.2 | COMPARISON BETWEEN THE ATTITUDE ESTIMATED USING THE UKF AND EKF | 64 |
| 5.1.3 | COVARIANCE MATRICES EQUIVALENTS FOR THE TANGENT SPACE | 74 |
| 5.1.4 | ATTITUDE ESTIMATION USING THE USQUE AND RIEMANNIAN FILTER | 75 |
| 6 | CONCLUSION | 86 |
| 6.1 | FUTURE WORK | 87 |
| | BIBLIOGRAPHY..... | 89 |
| | APPENDICES | 93 |
| A | SENSORS..... | 94 |
| B | EXPONENTIAL AND LOGARITHM MAPPING OF THE S^{n-1} APPLIED IN THE | |

| | |
|---|------------|
| RIEMANNIAN FILTER | 97 |
| C GRADIENT DESCENT ALGORITHM APPLIED IN THE RIEMANNIAN FILTER .. | 99 |
| D VECTOR PARAMETRIZATION FOR UNIT QUATERNIONS APPLIED IN THE RIEMANNIAN FILTER | 100 |
| E SIGMA SETS METHOD APPLIED IN THE RIEMANNIAN FILTER | 101 |
| F HARD-IRON AND SCALE FACTOR COMPENSATION | 102 |
| G PRACTICAL EVALUATION OF CONSISTENCY | 105 |
| H PARALLEL TRANSPORT | 106 |
| H.0.1 PARALLEL TRANSPORT OF A VECTOR..... | 106 |
| H.0.2 PARALLEL TRANSPORT OF A MATRIX | 106 |
| I TRANSFORMATIONS BETWEEN TANGENT BASIS AND CANONICAL BASIS..... | 108 |

List of Figures

| | | |
|------|---|----|
| 1.1 | Tabletop platform..... | 2 |
| 1.2 | LAICA’s test platform..... | 3 |
| 1.3 | Helmholtz cage overview; A. Helmholtz cage; B. HMR2300 magnetometer.... | 3 |
| 1.4 | Tabletop air bearing..... | 4 |
| 1.5 | Reaction Wheels..... | 4 |
| 1.6 | Attitude determination using a camera; A. Frontal view; B. System interface. . | 5 |
| 1.7 | Attitude determination using a IMU from adafruit; A. Sensor upper view; B. System interface using Matlab..... | 6 |
| 1.8 | Attitude determination using a IMU from 3-Space; A. Sensor upper view; B. System interface. | 6 |
| 2.1 | Tilt-compensated eCompass algorithm initial conditions..... | 15 |
| 3.1 | Angular momentum references frames..... | 36 |
| 3.2 | Body frame representation | 41 |
| 3.3 | Graphical representation; A. Initial position; B. Roll rotation..... | 42 |
| 3.4 | Graphical representation; A. Initial position; B. Pitch rotation..... | 42 |
| 3.5 | Graphical representation; A. Initial position; B. Yaw rotation..... | 43 |
| 3.6 | Simulation block diagram | 44 |
| 3.7 | Singularity observed in Roll and Yaw | 45 |
| 3.8 | Singularity avoidance using quaternion..... | 45 |
| 3.9 | Schematic view of the experiment | 46 |
| 3.10 | Singularity observed in Roll and Yaw | 46 |
| 3.11 | Singularity avoidance using quaternion..... | 47 |
| 4.1 | IMU location | 49 |
| 4.2 | spaceSensor | 50 |
| 4.3 | Adafruit’s IMU schematic..... | 51 |
| 4.4 | Hardware configuration; A. Platform inside view; B. Circuit upper view. | 52 |
| 4.5 | Yost Labs’ IMU schematic | 53 |
| 4.6 | Magnetometer calibration of Adafruit’s IMU | 54 |
| 4.7 | Magnetometer calibration of HMR2300..... | 55 |
| 4.8 | Sphere Calibration | 56 |

| | | |
|------|--|----|
| 5.1 | Methodology schematic view | 57 |
| 5.2 | Magnetic disturbance in quaternions | 58 |
| 5.3 | Magnetic disturbance in Euler angles | 59 |
| 5.4 | Experiment schematic view | 61 |
| 5.5 | Result schematic view | 61 |
| 5.6 | A. Initial position; B. Rotation of 20 degrees in roll. | 62 |
| 5.7 | A. Initial position; B. Rotation of 30 degrees in pitch. | 62 |
| 5.8 | Initial target setup used to make rotations around the z axis | 62 |
| 5.9 | A. Initial position; B. Rotation of 8 degrees in yaw. | 63 |
| 5.10 | Magnetic field generated by the Helmholtz cage. | 64 |
| 5.11 | Chi square test..... | 65 |
| 5.12 | Filters output..... | 66 |
| 5.13 | Rotation of 20 degrees around the x axis | 68 |
| 5.14 | Rotation of 30 degrees around the y axis | 69 |
| 5.15 | Rotation of 8 degrees around the z axis | 70 |
| 5.16 | A. UKF ($\phi - \hat{\phi}$) for a rotation of 20° around X axis.; B. EKF ($\phi - \hat{\phi}$) for a rotation of 20° around X axis. | 71 |
| 5.17 | A. UKF ($\theta - \hat{\theta}$) for a rotation of 30° around Y axis.; B. EKF ($\theta - \hat{\theta}$) for a rotation of 30° around Y axis. | 72 |
| 5.18 | A. UKF ($\psi - \hat{\psi}$) for a rotation of 8° around Z axis.; B. EKF ($\psi - \hat{\psi}$) for a rotation of 8° around Z axis. | 72 |
| 5.19 | Experimental data and UKF output for the platform moving freely | 73 |
| 5.20 | Tangent space..... | 74 |
| 5.21 | Step by Step..... | 75 |
| 5.22 | A. Chi square test for USQUE.; B. USQUE output for a rotation of 30^0 around y axis..... | 75 |
| 5.23 | Riemannian output for a rotation of 30^0 around y axis. | 76 |
| 5.24 | USQUE output for a rotation of 30^0 around y axis. | 77 |
| 5.25 | Rotation of 20^0 around x axis..... | 78 |
| 5.26 | Rotation of 30^0 around y axis..... | 79 |
| 5.27 | Rotation of 8^0 around Z axis. | 80 |
| 5.28 | A. Riemannian ($\phi - \hat{\phi}$) for a rotation of 20° around X axis.; B. USQUE ($\phi - \hat{\phi}$) for a rotation of 20° around X axis. | 81 |
| 5.29 | A. Riemannian ($\theta - \hat{\theta}$) for a rotation of 30° around Y axis.; B. USQUE ($\theta - \hat{\theta}$) for a rotation of 30° around Y axis. | 82 |
| 5.30 | A. Riemannian ($\psi - \hat{\psi}$) for a rotation of 8° around Z axis.; B. USQUE ($\psi - \hat{\psi}$) for a rotation of 8° around Z axis. | 82 |
| 5.31 | Free rotation. | 83 |
| 5.32 | Free rotation. | 84 |
| 5.33 | Free rotation. | 84 |
| 5.34 | Bias(Riemannian). | 85 |

| | |
|---|----|
| 5.35 Bias(USQUE). | 85 |
| A.1 IMU location | 94 |
| A.2 A single axis accelerometer..... | 95 |
| A.3 Theoretical conception of principles used to construct a magnetometer; A. Hall effect; B. Magneto-resistive effect. | 96 |
| A.4 Coriolis effect | 96 |
| B.1 Exponential and logarithm mapping..... | 98 |

List of Tables

| | | |
|-----|--|----|
| 4.1 | LSM303DLHC and L3GD20 settings..... | 49 |
| 4.2 | Yost Labs IMU specifications | 50 |
| 4.3 | Adafruit’s IMU calibration parameters..... | 54 |
| 4.4 | HMR2300 calibration parameters | 55 |
| 5.1 | Mean and standard deviation for the estimated measurement residuals of the UKF and EKF | 71 |
| 5.2 | Mean and standard deviation for the innovation of the UKF and EKF..... | 81 |
| 6.1 | Summarized results | 87 |
| A.1 | LSM303DLHC and L3GD20 settings..... | 95 |

Acronyms

| | |
|------------|---|
| LAICA | Laboratory of Application and Innovation in Aerospace Science |
| EKF | Extended Kalman Filter |
| UKF | Unscented Kalman Filter |
| USQUE | Unscented Quaternion Estimator |
| RiSAdUKF | Riemannian-Spheric Additive Unscented Kalman Filter |
| ADCS | Attitude Determination and Control System |
| IMU | Inertial Measurement Unit |
| R_ψ | Rotation matrix referred to Yaw axis |
| R_ϕ | Rotation matrix referred to Roll axis |
| R_θ | Rotation matrix referred to Pitch axis |
| CM | Center of mass |
| CR | Center of rotation |
| UNB | University of Brasilia |
| 9DOF | Nine degrees of freedom |
| PC | Personal Computer |
| USB | Universal Serial Bus |
| LiPo | Lithium Polymer |
| $QtoRoV$ | Quaternion to rotation vector |
| $RoVtoQ$ | Rotation vector to quaternions |

List of Symbols

| | |
|-----------------------|---|
| ψ | Yaw angle |
| ϕ | Roll angle |
| θ | Pitch angle |
| X_i | X inertial axis |
| Y_i | Y inertial axis |
| Z_i | Z inertial axis |
| \mathbf{B}_r | Magnetometer measurements referred to the inertial frame |
| \mathbf{G}_r | Accelerometer measurements referred to the inertial frame |
| \mathbf{B} | Magnetic Field Vector |
| g | Gravity acceleration |
| δ | Angle of dip |
| $\boldsymbol{\omega}$ | Angular velocity vector |
| ω_x | Angular velocity around X axis |
| ω_y | Angular velocity around Y axis |
| ω_z | Angular velocity around Z axis |
| $\dot{\phi}$ | Roll time derivative |
| $\dot{\psi}$ | Yaw time derivative |
| $\dot{\theta}$ | Pitch time derivative |
| \mathbf{q} | Quaternion |
| $Re(\mathbf{q})$ | Quaternion real part |
| $Im(\mathbf{q})$ | Quaternion imaginary part |
| R_b^i | Rotation matrix from body to inertial frame |
| \mathbf{x}_k | State vector at the instant k |
| \mathbf{y}_k | Measurement vector at the instant k |
| Q | Process noise covariance |
| R | Measurement noise covariance |
| \mathbf{u}_k | Process model input at the instant k |
| \mathbf{P}^+ | Predicted and corrected sate covariance matrix |
| \mathbf{P}^- | Predicted sate covariance matrix |
| I_3 | Identity matrix of order 3 |

| | |
|-------------------------------|---|
| $\hat{\mathbf{x}}_k^-$ | Predicted estimated state |
| $\hat{\mathbf{x}}_k^+$ | Corrected estimated state |
| \mathbf{K}_k | Kalman gain in the instant k |
| ϱ | Quaternion imaginary part |
| β_k | Gyro bias vector at the instant k |
| ΔT | Time sample |
| m_i | Infinitesimal masses placed |
| ρ_i | Distance of center of the reference system |
| \mathbf{l} | Angular momentum vector |
| \mathbf{v}_i | Linear velocity |
| $\mathbf{T}_{external}$ | Total external torque |
| T_x | X component resulting torque |
| T_y | Y component resulting torque |
| T_z | Z component resulting torque |
| \mathbf{F}_i | External forces acting on the system |
| I | Body inertia tensor |
| \mathbf{l}_o | Angular momentum referred to the center of rotation |
| $\dot{\omega}$ | Angular velocity time derivative |
| $\mathbf{T}_{actuators}$ | Torque provided by the actuators |
| $\mathbf{T}_{gravitational}$ | Gravitational torque |
| $\mathbf{T}_{aerodynamics}$ | Aerodynamic torque |
| d^2_k | Chi square values at instant k |
| S^2 | Surface sphere |
| g^{TS} | Vector g in the tangent space |
| $exp_{\mathbf{q}}^e$ | Exponential mapping in the base \mathbf{q} |
| θ^{TS} | Angle between two quaternions |
| $ln_{\mathbf{q}}(\mathbf{p})$ | Logarithm mapping in the base \mathbf{q} |
| q_{RoV}^v | Quaternion expressed in rotation vectors |
| \bar{X} | State mean vector |
| X_0 | Magnetic field offset on the X axis |
| Y_0 | Magnetic field offset on the Y axis |
| Z_0 | Magnetic field offset on the Z axis |
| M_x | X component raw magnetic measurement |
| M_y | Y component raw magnetic measurement |
| M_z | Z component raw magnetic measurement |
| SC_x | X component magnetic field scale factor |
| SC_y | Y component magnetic field scale factor |
| SC_z | Z component magnetic field scale factor |

Chapter 1

Introduction

Space missions are usually expensive and involve high risks related to subsystem complexity. Since the space mission Vanguard TV3 launched on 6 December 1957 until 2009, more than 30% of the pico- and nanosatellites were destroyed due to launch failure and only 40% of the nanosatellites successfully launched had some attitude estimation system [Bouwmeester and Guo 2010]. But due to lessons from the earlier space missions design and from recent applications demands, the number of successful missions increased and even universities are developing educational projects of small satellites ([Selva and Krejci 2012]).

Microssatellites' subsystems are in constant improvement and have a direct impact on different research areas. For example in [Giri et al. 2011], the authors used satellite data to evaluate the status and distribution of mangrove forests. In [Bellion et al. 2016], the researchers presented a new antenna, for nanosatellites, capable of covering both bands of commands and telemetry. A recent contribution for attitude determination system research is presented in [Garcia et al. 2017]. In that work, the authors developed and evaluated a fault tolerant attitude determination system for nanosatellites that will be validated in a 2U-size cubesat developed by the Brazilian Institute for Space Research (INPE).

Ground based test facilities contribute to spacecraft' subsystems improvements. These facilities are necessary to meet future operational requirements mitigating the risks of launch failure [Jablonski 2014]. In order to address these issues, the Laboratory of Application and Innovation in Aerospace Science(LAICA), located at the university of Brasilia, developed a microssatellite test platform.

In literature, this type of simulator can be found in three main configurations: tabletop, umbrella and dumbbell. More details about each air-bearing testbed style can be found in [Schwartz et al. 2003]. This work deals with a tabletop configuration represented in Fig. 1.1.

As it can be seen, this type of simulator has a semi sphere mounted directly onto a flat face whose weight is supported by pressurized air that passes through small holes in the hemispherical base. This type of platform provides a faithful representation of space conditions. Because of that, this simulator is utilized to evaluate spacecraft attitude estimation and

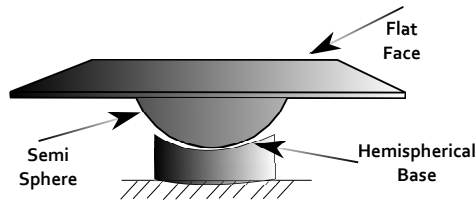


Figure 1.1: Tabletop platform

control.

In the context of attitude estimation, different filtering techniques have been used to address this issue. In literature, papers estimate the spacecraft orientation using the Extended Kalman Filter (EKF) or Unscented Kalman Filter (UKF) such as [Garcia et al. 2018], [Garcia et al. 2012] and [Pham et al. 2015]. Other papers present quaternion filters as improvement for spacecraft orientation estimation such as [Crassidis and Markley 2003] and [Menegaz 2016]. These are just some of options of algorithms used for attitude estimation, a survey of filtering algorithms can be found in [Crassidis et al. 2007]. In this work, the test-bed orientation will be estimated using four different algorithms: EKF, UKF, Unscented Quaternion Estimator (USQUE) presented in [Crassidis and Markley 2003] and Riemannian-Spheric Additive Unscented Kalman Filter (RiSAdUKF) presented in [Menegaz 2016].

This research aims evaluating filtering techniques in order to improve LAICA's platform attitude determination system and also evaluating the advantages and disadvantages of representing the platform orientation in Euler angles or quaternions. In the actual configuration, the LAICA's platform has two different inertial measurement unities (IMUs): one that provides measurements in Euler angles and the other one that provides measurements in quaternions. Using the Euler angles we will estimate the platform attitude using an EKF and UKF. Besides, using the quaternion data from the other sensor, we will estimate the platform orientation using the USQUE and RiSAdUKF, which will be first-time tested under experimental data. For future work, these quaternion filters could be implemented on ABACUS, which is a professional nanosatellite control board recently acquired by LAICA.

1.1 LAICA air-bearing spacecraft simulator

LAICA's test facility can be divided in four main parts: the Helmholtz cage, tabletop air bearing, actuators and attitude determination system. This facility communicates with two computers. One of them is used to control the magnetic field generated by the Helmholtz cage and also to determine the platform attitude using a camera. The second computer is used to run the test-bed balancing algorithm, controlling actuators such as reaction wheels and also collecting platform orientation data from two different IMUs (see Fig.1.2).

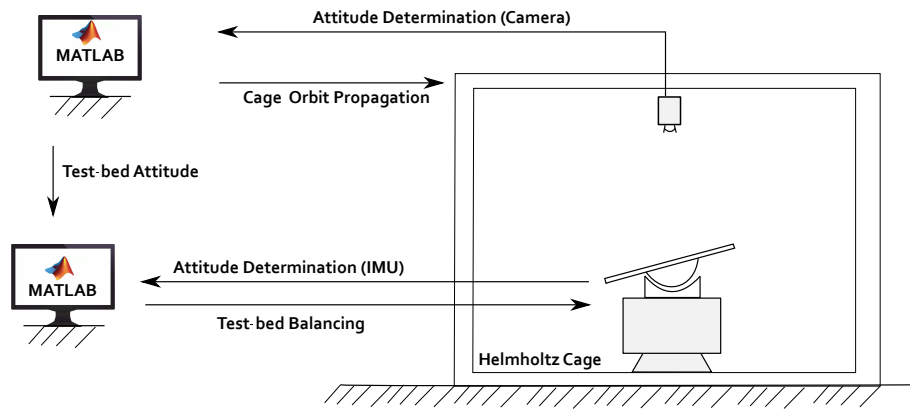
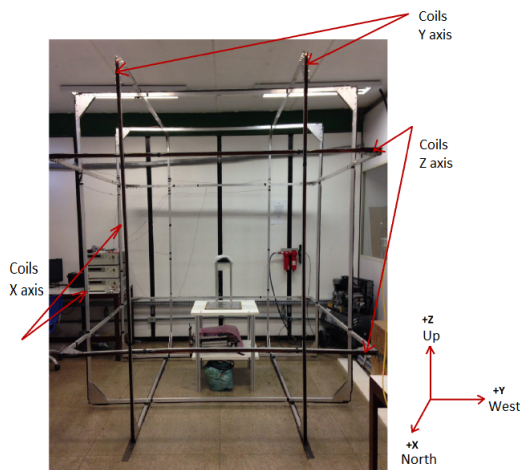


Figure 1.2: LAICA's test platform

1.1.1 Helmholtz cage

The Helmholtz cage, named after Hermann Von Helmholtz (1821-1894), presented in [Loiola et al. 2018] consists of three coil pairs placed in an orthogonal configuration, shown in Fig.1.3(A), which allows magnetic field generation in any direction. This structure is also equipped with a magnetometer from Honeywell, the HMR2300, shown in Fig.1.3(B) which is used to close the Helmholtz cage loop.

This Earth magnetic field simulator can be used in different experiments. Some of them are: orbit propagation, attitude control using magnetorquers and also as a noise source for IMUs that use magnetometers. This last application is important because the Helmholtz cage will be used to generate experimental noise in the orientation data. More details about this application will be discussed later. In this system, the author of this master's thesis gave contributions in the hardware and software implementation.



(A)



(B)

Figure 1.3: Helmholtz cage overview; **A.** Helmholtz cage; **B.** HMR2300 magnetometer.

1.1.2 Tabletop air-bearing

LAICA's table top air bearing described in [Silva and Rodrigues 2015] is composed by a semi sphere mounted directly onto a flat face whose weight is supported by pressurized air provided by a compressor and filtered by compressed air filters (see Fig.1.4). Although it is possible reach many orientations, this type of platform has some structural limitations. This simulator allows rotations of 360° around z axis, but has limitations of 45° around x and y axes. In this structure, the author of this master's thesis did improvements in internal cable connections also made the P64 standard on the platform in order to fixate nanosatellites. In addition, it is important mentioning that in Brazil there are other types of simulators as for instance the INPE air bearing table presented in [Oliveira et al. 2015].

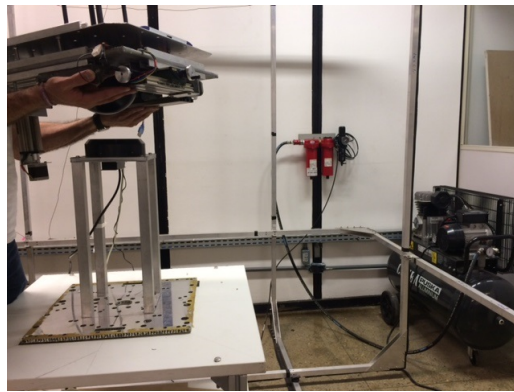


Figure 1.4: Tabletop air bearing

1.1.3 Actuator

This facility is also equipped with the reaction wheels presented in Fig.1.5. The actuation system, presented in [Loiola et al. 2017], is composed by three brushless motors with masses attached. These reaction wheels are mounted along each axis direction and by conservation of angular momentum they can change the test-bed orientation. The author of this master's thesis gave contributions in the electrical circuit design and software implementation of this actuation system.

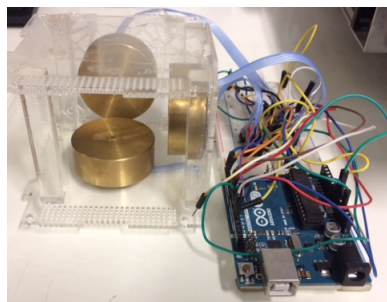


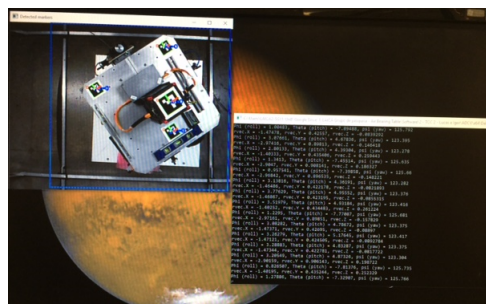
Figure 1.5: Reaction Wheels

1.1.4 Attitude determination system

LAICA's attitude determination system provides the test-bed orientation which is obtained from three different sensors: a camera and two different IMUs. The attitude determination using the camera requires a ARUCO module placed on the platform surface. In this system, the orientation is calculated based on distortions on the matrix that represents ARUCO's image. More details about the camera location and system interface are presented in Fig.1.6(A) and Fig.1.6(B).



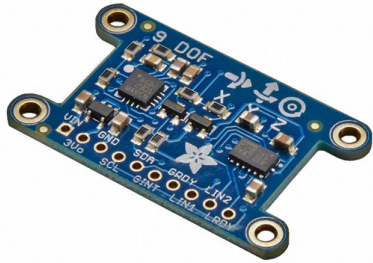
(A)



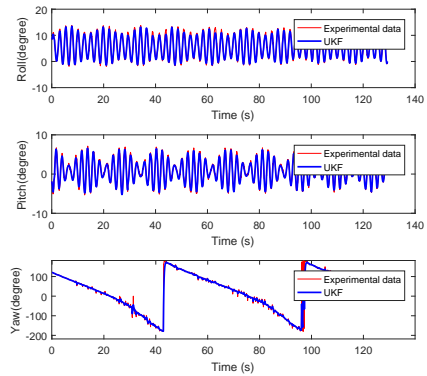
(B)

Figure 1.6: Attitude determination using a camera; **A**. Frontal view; **B**. System interface.

The other two attitude determination algorithms utilize measurements of three axis accelerometer, magnetometer and gyroscope from each IMU. The system using the Adafruit 9-DOF IMU utilizes the tilt-compensated eCompass algorithm, which will be explained later, to convert sensor raw measurements into Euler angles. While the second system uses an IMU from Yost Labs which provides a closed software to determine the orientation in quaternions from raw measurements. A general view of each sensor and also each system interface are presented in Fig.1.7 and Fig.1.8. In the attitude determination system, the author of this master's thesis got directly involved in the attitude determination using the tilt-compensated eCompass algorithm and also operated the attitude determination software developed by Yost Labs.



(A)

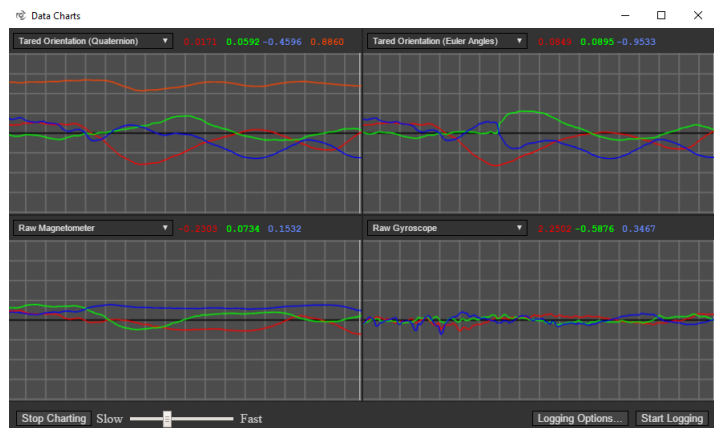


(B)

Figure 1.7: Attitude determination using a IMU from adafruit; **A.** Sensor upper view; **B.** System interface using Matlab.



(A)



(B)

Figure 1.8: Attitude determination using a IMU from 3-Space; **A.** Sensor upper view; **B.** System interface.

1.2 Motivation

Attitude estimation is a fundamental and important task for many spacecraft applications [Cheon and Kim 2007]. Pointing control of a spacecraft antenna and Earth observation are examples of those applications that also have a direct impact on society. The first procedure, largely used in telecommunication, utilizes attitude estimation and control techniques to perform maneuvers and suppress antenna vibrations. While the second application uses the estimated orientation to improve the spacecraft attitude control during camera shots which are used to monitor events on Earth such as fire activity.

This research deals with attitude estimation of a spacecraft. In the literature, there are two main approaches to represent orientation. One of them is through Euler angles. This solution presents a singularity when the pitch angle reaches 90^0 . The second approach mitigates this singularity using quaternions. That is why quaternions are largely utilized in attitude determination especially for aerospace applications.

Although quaternions are not affected by singularity, they are affected by noise. In order to obtain better attitude measurements, the signal, embedded with noise, must be treated. Filtering literature presents different options to deal with attitude measurements corrupted by noise. One of these options is using filters capable of deal directly with quaternions. Among the filters designed to receive quaternions as input, there is one based on Riemannian manifolds which is capable of preserving distance and angles, robust to miss-defined operations and also has probability concepts and statistic theory well defined. For these reasons, this new algorithm was chosen to be evaluated using data coming from LAICA's nanosatellite test platform.

1.3 Problem definition

Attitude determination is the process of determining the orientation of a spacecraft referring to an inertial frame [Wertz 1990]. This research will utilize a three-axis magnetometer, accelerometer and gyroscope. But it is important to mention that other sensors can also be used such as sun sensors or cameras.

These sensors are classified either as proprioceptive or exteroceptive sensors. A proprioceptive sensor measures values internal to the system, one example of this is the gyroscope. An exteroceptive sensor obtains information from external references, examples of this are the magnetometer, accelerometer and sun sensors.

Both of these sensor types have advantages and disadvantages. Exteroceptive sensors tends to be more precise when the external reference does not changes. But if this external reference changes or is affected by another external source, this type of sensor will provide wrong measurements. On the other hand, proprioceptive sensors are not affected by external sources as the exteroceptive ones, but this second sensor type accumulates errors during the

time.

In order to take advantage of each sensor type, four different filtering algorithms, EKF, UKF, USQUE and RiSAdUKF were implemented to estimating the orientation of a microsatellite testbed. These filters were used to perform a sensor fusion which consists of using exteroceptive sensors, three axis magnetometer and accelerometer to correct the orientation obtained from proprioceptive sensors which in this case was the three axis gyroscope.

These exteroceptive and proprioceptive sensor data were obtained from two different IMUs. For one of them, it was implemented the tilt-compensated eCompass algorithm, which will be explained in the next chapter, to obtain the orientation in Euler angles from the three axis magnetometer and accelerometer sensors. Using this orientation and the attitude provided by integration of the Kinematic equation for Euler angles, also presented in the next chapter, the EKF and UKF were used to estimate the testbed orientation.

For the second IMU, a software provided by Yost Labs was used to obtain the testbed orientation in quaternions based on three axis magnetometer and accelerometer measurements. Based on these unity quaternions, which will be explained in the next chapter, and in the orientation obtained using the angular velocity vector, the USQUE and RiSAdUKF were implemented to estimate the testbed orientation.

In order to evaluate these estimated orientations, three main experimental strategies, described in Chapter 5, were utilized.

1.4 More applications

The results and methodology presented in this work are useful to a variety of applications. In general, the filters techniques presented in this research can be applied for applications that have to deal with the body orientation. In this context, some related applications for these filters are:

Drones that operate in water, on land or air : These devices normally are equipped with sensors to measure its attitude. In some operations the orientation measurement must be precise, such as when the drone rotates a specific angle or even keeps its orientation on a specific height.

Virtual reality : This type of application utilizes attitude information to determine what region of the simulated scenario the user is looking.

Animation : Some of these systems utilize the attitude of a actor body junction to determine the animation orientation.

Health applications : The using of orientation data is increasing in the health applications such as treatments that using attitude data to evaluate patient's equilibrium recover after a specific treatment. Another example are systems that have robots in surgery con-

trolled by doctor located in another country and these robots use attitude estimation to avoid undesirable vibrations created by doctors.

Vehicle navigation : Modern cars especially the autonomous ones, utilize attitude data combine with GPS information to improve location estimation in tunnels.

Meteor observations and tracking : This application uses attitude sensor data and computer vision to control the telescope orientation and keeping tracking of a specific meteor.

Agriculture of precision : For this application, the attitude data is used to improve for example the trajectory of trucks responsible for planting seeds.

Mining : Some companies in the world already have fully automated trucks used in operations inside mines where the GPS signal is not always available.

Military applications : In the context of military applications attitude estimation is vastly used such missile trajectory correction, air plane stability control, autonomous guns and so on.

Industrial instrumentation : In industry, attitude data is used to manipulating robots to execute specific tasks for example manipulators used in the automotive industry to fix screws.

3D reconstruction : In this type of application the differences in orientation provided by different cameras are used to reconstruct a scene.

1.5 Results and Contributions

This research evaluated different filtering algorithms: EKF, UKF, USQUE and RiSAdUKF used to estimate the orientation of a spacecraft testbed. This investigation led to some important results obtained from experimental data:

- The implemented UKF was faster than the EKF and for the proposed experiments, both filters presented a similar answer.
- The RiSAdUKF was capable of effectively estimating the testbed attitude for all tests. But it was slower than USQUE which had some problems to estimate the orientation in peak regions.
- The gyro bias estimated by the RiSAdUKF was more correct than the gyro bias estimated by the USQUE.

This investigation also led to some other contributions:

- Experimental methodologies were developed to test each filter using an experimental ground truth.

- The values of Q and R for the RiSAdUKF were chosen using a method designed to find the covariance matrices equivalents for the tangent space which is the space used by this filter.

Previous results already published provided the basic knowledge to achieve these final results and contributions. The chronological order of publications is:

- 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]. This work used an EKF to estimate LAICA's testbed orientation through the data fusion of exteroceptive and proprioceptive sensors.
- Loiola, J. V. L., Silva, L. M. B., Battistini, S., Borges, R. A., and Cappelletti, C. (2017). *Development of a hardware-in-the-loop test platform for nanosatellites adcs integrated with an ukf*. 4th IAA Conference on University Satellite Missions and Cubesats Workshop [Loiola et al. 2017]. This work presented the development of a hardware-in-the-loop system. This system used an orbit propagator to simulate the angular variations experienced by a spacecraft on a specific orbit. These simulated angular variations were passed as reference to a discrete-time PD controller which had the objective of controlling the testbed attitude around the z axis using as feedback the orientation filtered by an UKF.
- Loiola, J. V. L., Ploeg, L. C. V., Silva, R. C., aes, F. C. G., Borges, R. A., Borges, G. A., Battistini, S., and Cappelletti, C. (2018). *3 axis simulator of the Earth magnetic field*. IEEE Aerospace Conference [Loiola et al. 2018]. This work presented the development and implementation of an Earth magnetic field simulator used to emulate the Earth magnetic field vector for different orbit points. This research also presented an experimental strategy to measure the volume of field uniformity inside the cage.
- 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]. This work presented the design and realization of a testbed used to reproduce nanosatellite attitude motion. In addition, an efficient automated balancing algorithm based on the Least Square Method (LSM) was proposed and validated experimentally.

1.6 Manuscript presentation

This manuscript is divided in six chapters. The first three chapters are dedicated to provide a general view of nanosatellite test platforms constructed around the world and present-

ing the LAICA's platform. In addition, those chapters were designed aiming to provide a fundamental knowledge required to understand all the concepts involved. The other three chapters will focus on experimental procedures, results and the conclusions taken from those analysis.

The chapter 1 starts presenting the context of nanosatellite test platforms utilized to evaluate nanosatellite subsystems. After that the problem in study is defined and finally the research contributions are presented.

The chapter 2 presents the fundamental knowledge required to understand the method used to obtain Euler angles from raw measurements of three axis accelerometer, magnetometer and gyroscope and also to understand all the attitude estimation methods utilized.

The chapter 3 presents the derivation of the system dynamic model utilized to conduct simulations. This chapter presents the singularity observed in theory, using these simulations, and in experimental data.

The chapter 4 presents a detailed description of the sensors and instrumental apparatus utilized on the experiments. In addition this chapter presents some of the experimental challenges solved during the process.

The chapter 5 presents the summary of the results obtained together with the evaluation of them.

The chapter 6 presents the conclusions and some proposals for future works.

Chapter 2

Attitude Representation and Filtering Strategies

As previously discussed, the spacecraft attitude can be represented using Euler angles or quaternions. But in either representation has noise embedded in the measurements. This chapter will present more detail about each representation and present some filtering techniques used to mitigate this noise effect.

2.1 Attitude and reference systems

Attitude refers to the body orientation in space which is determined using appropriated reference systems [Wertz 1990]. In this work, we used for space frame the North-East-Down coordinate system.

2.1.1 Body frame

In most of attitude determination applications, the sensors' axes are made to coincide with the axes of the moving platform. This axes are the body frame [Noureldin et al. 2012] which in this work will be referred by b . Some definitions, used in this work, about the body frame are given by

- The origin coincides with the test-bed center of rotation.
- The rotation angle around the x axis is called as roll angle defined around this axis using the right-hand rule.
- The rotation angle around the y axis is called as pitch angle defined around this axis using the right-hand rule.

- The z-axis is orthogonal to the other axes and completing the right-handed coordinate system. The rotation angle around the z axis is called as yaw angle defined around this axis using the right-hand rule.

2.1.2 North-East-Down system

A local-level frame (LLF), also known as navigation frame, is utilized to represent the orientation of a body when it is located near the surface of the Earth [Noureldin et al. 2012]. In this work, the navigation frame will have the same properties and origin of the body frame and axes fixed referring to Earth. Some definitions, used in this work, about this local-level frame are given by

- The X axis points to geodetic north.
- The Y axis points to geodetic east.
- The Z axis is orthogonal to the other axes and completing the right-handed coordinate system.

Because of these characteristics, according to [Cai et al. 2011], this navigation frame is also called North-East-Down (NED) system.

2.2 Euler angles

The test-bed attitude can be represented by Euler angles which provide a clear physical interpretation [Wertz 1990]. Initially it will be defined the concept of rotation matrix and after that it will be described an algorithm to obtain Euler angles from raw measurements.

2.2.1 Rotation matrix

According to [Titterton and Weston 1997], three different rotations of a body can be mathematically expressed as three separated direction cosine matrices

$$R_\psi = \begin{bmatrix} C\psi & S\psi & 0 \\ -S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ where } \psi \text{ represents a rotation about } Z \text{ axis,} \quad (2.1)$$

$$R_\theta = \begin{bmatrix} C\theta & 0 & -S\theta \\ 0 & 1 & 0 \\ S\theta & 0 & C\theta \end{bmatrix} \text{ where } \theta \text{ represents a rotation about } Y \text{ axis,} \quad (2.2)$$

$$R_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\phi & S\phi \\ 0 & -S\phi & C\phi \end{bmatrix} \text{ where } \phi \text{ represents a rotation about } X \text{ axis.} \quad (2.3)$$

Thus a transformation from the inertial frame, represented by the letter "i", to body frame, represented by the letter "b", following the sequence ZYX, is given by

$$C_i^b = R_\phi R_\theta R_\psi \quad (2.4)$$

$$C_i^b = \begin{bmatrix} C\theta C\psi & C\theta S\psi & -S\theta \\ -C\phi S\psi + S\phi S\theta C\psi & C\phi C\psi + S\psi S\theta S\psi & S\phi C\theta \\ S\phi S\psi + C\phi S\theta C\psi & -S\phi C\psi + C\phi S\theta S\psi & C\phi C\theta \end{bmatrix}. \quad (2.5)$$

Similarly, the transformation from body frame to inertial frame is given by

$$C_b^i = (C_i^b)^T = R_\psi^T R_\theta^T R_\phi^T \quad (2.6)$$

$$C_b^i = \begin{bmatrix} C\psi & -S\psi & 0 \\ S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\theta & 0 & S\theta \\ 0 & 1 & 0 \\ -S\theta & 0 & C\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\phi & -S\phi \\ 0 & S\phi & C\phi \end{bmatrix} \quad (2.7)$$

$$C_b^i = \begin{bmatrix} C\theta C\psi & -C\phi S\psi + S\phi S\theta C\psi & S\phi S\psi + C\phi S\theta C\psi \\ C\theta S\psi & C\phi C\psi + S\psi S\theta S\psi & -S\phi C\psi + C\phi S\theta S\psi \\ -S\theta & S\phi C\theta & C\phi C\theta \end{bmatrix}. \quad (2.8)$$

This result is known as rotation matrix and is used for representing a vector from the body frame in the inertial frame.

2.2.2 Tilt-compensated eCompass algorithm

This section will present a method used to obtain the roll(ϕ),pitch(θ) and yaw(ψ) angles using accelerometer and magnetometer measurements. This technique is known as tilt-compensated eCompass algorithm and was obtained from [Ozyagcilar 2012], more information can also be found in [Bar-Itzhack and Harman 1997].

The derivation of the tilt-compensated eCompass algorithm starts with three assumptions. Initially the accelerometer and magnetometer axes are aligned. Second, the gravitational acceleration points to Z-axis direction. And finally, the X axis points to the direction of greater magnetic field intensity.

These initial conditions are represented in Fig.2.1, where X_i , Y_i and Z_i are the inertial

axes, \mathbf{B} is the magnetic field vector, \mathbf{g} is gravity vector and δ , known as angle of dip, which is the angle between the magnetic field vector and a line parallel to the ground. In addition, it is important to mention that although δ changes for different earth points, this will not be a problem because it will cancel out during the derivation.

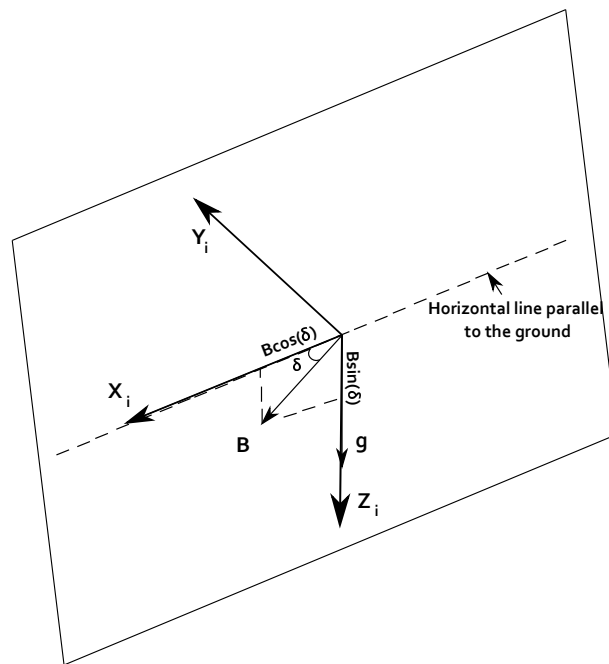


Figure 2.1: Tilt-compensated eCompass algorithm initial conditions

In these conditions the accelerometer and magnetometer measurement referring to the inertial frame, \mathbf{B}_i and \mathbf{G}_i respectively are given by:

$$\mathbf{B}_i = B \begin{bmatrix} \cos(\delta) \\ 0 \\ \sin(\delta) \end{bmatrix}, \quad (2.9)$$

$$\mathbf{G}_i = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}. \quad (2.10)$$

These two vectors can also be represented in the body frame using the rotation matrix presented in Eq. (2.5)

$$\mathbf{B}_b = \mathbf{C}_i^b \mathbf{B}_i = R_\phi R_\theta R_\psi \mathbf{B}_i, \quad (2.11)$$

$$\mathbf{G}_b = \mathbf{C}_i^b \mathbf{G}_i = R_\phi R_\theta R_\psi \mathbf{G}_i, \quad (2.12)$$

where R_ψ , R_θ , R_ϕ , are rotation matrices referring to axes Z, Y and X respectively. The next step in the method consist of obtain roll and pitch angle though Eq. (2.12). Multiplying

both sides of this equation by inverse matrices $R_{-\theta}R_{-\phi}$, we have that:

$$R_{-\theta}R_{-\phi}\mathbf{G}_b = R_{-\theta}R_{-\phi}R_\phi R_\theta R_\psi \mathbf{G}_i \quad (2.13)$$

$$R_{-\theta}R_{-\phi}\mathbf{G}_b = R_\psi \mathbf{G}_i \quad (2.14)$$

$$\begin{bmatrix} C\theta & 0 & S\theta \\ 0 & 1 & 0 \\ -S\theta & 0 & C\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\phi & -S\phi \\ 0 & S\phi & C\phi \end{bmatrix} \begin{bmatrix} G_{bx} \\ G_{by} \\ G_{bz} \end{bmatrix} = \begin{bmatrix} C\psi & S\psi & 0 \\ -S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}, \quad (2.15)$$

where G_{bx}, G_{by}, G_{bz} are components of \mathbf{G}_b . From Eq. (2.15) we have that

$$G_{by}C\phi - G_{bz}S\phi = 0 \quad (2.16)$$

$$\implies \tan(\phi) = \frac{G_{by}}{G_{bz}}. \quad (2.17)$$

And also

$$G_{bx}C\theta + G_{by}S\theta S\phi + G_{bz}S\theta C\phi = 0 \quad (2.18)$$

$$\implies \tan(\theta) = \frac{-G_{bx}}{G_{by}S\phi + G_{bz}C\phi}. \quad (2.19)$$

Until now the accelerometer measurements were utilized to obtain the angles ϕ and θ provided by Eq. (2.17) and Eq. (2.19). In order to represent a body attitude, it is also necessary determining the yaw angle (ψ). This angle can be determined using magnetometer measurements. Multiplying Eq. (2.11) by $R_{-\theta}R_{-\phi}$ in both sides we have that:

$$R_{-\theta}R_{-\phi}\mathbf{B}_b = R_{-\theta}R_{-\phi}R_\phi R_\theta R_\psi \mathbf{B}_i \quad (2.20)$$

$$\begin{bmatrix} C\theta & 0 & S\theta \\ 0 & 1 & 0 \\ -S\theta & 0 & C\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\phi & -S\phi \\ 0 & S\phi & C\phi \end{bmatrix} \begin{bmatrix} B_{bx} \\ B_{by} \\ B_{bz} \end{bmatrix} = \begin{bmatrix} C\psi & S\psi & 0 \\ -S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} BC_\delta \\ 0 \\ BS_\delta \end{bmatrix} \quad (2.21)$$

$$\begin{bmatrix} C\theta & S\theta S\phi & S\theta C\phi \\ 0 & C\phi & -S\phi \\ -S\theta & C\theta S\phi & C\theta C\phi \end{bmatrix} \begin{bmatrix} B_{bx} \\ B_{by} \\ B_{bz} \end{bmatrix} = \begin{bmatrix} C\psi & S\psi & 0 \\ -S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} BC_\delta \\ 0 \\ BS_\delta \end{bmatrix} \quad (2.22)$$

$$\begin{bmatrix} C\theta B_{bx} + S\theta S\phi B_{by} + S\theta C\phi B_{bz} \\ C\phi B_{by} - S\phi B_{bz} \\ -S\theta B_{bx} + C\theta S\phi B_{by} + C\theta C\phi B_{bz} \end{bmatrix} = \begin{bmatrix} C\psi BC_\delta \\ -S\psi BC_\delta \\ BS_\delta \end{bmatrix} \quad (2.23)$$

Considering $B_{fx} := C\psi BC_\delta$, $B_{fy} := -S\psi BC_\delta$ and $B_{fz} := BS_\delta$ we have that

$$\begin{bmatrix} C\theta B_{bx} + S\theta S\phi B_{by} + S\theta C\phi B_{bz} \\ C\phi B_{by} - S\phi B_{bz} \\ -S\theta B_{bx} + C\theta S\phi B_{by} + C\theta C\phi B_{bz} \end{bmatrix} = \begin{bmatrix} B_{fx} \\ B_{fy} \\ B_{fz} \end{bmatrix}, \quad (2.24)$$

From the right side of Eq. (2.23) and Eq. (2.24), the yaw angle(ψ) is given by a relation between the magnetic field component in x and y direction

$$C\psi BC_\delta = B_{fx} \quad \text{and} \quad -S\psi BC_\delta = B_{fy}, \quad (2.25)$$

Reorganizing these two equations we have that:

$$\implies \tan(\psi) = \frac{-B_{fy}}{B_{fx}} = \frac{S\phi B_{bz} - C\phi B_{by}}{C\theta B_{bx} + S\theta S\phi B_{by} + S\theta C\phi B_{bz}}. \quad (2.26)$$

This final equation provides the value of ψ from magnetometer data and also verifies that there is no need of knowing the δ angle because it is canceled out in Eq. (2.26).

2.2.3 Kinematic equation

The orientation of a body can be propagated in time using the angular velocity vector. According to [Titterton and Weston 1997] and [Thomson 1961], it is possible express a angular velocity vector

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \quad (2.27)$$

in terms of $\dot{\phi}$, $\dot{\theta}$ and $\dot{\psi}$. This relation can be obtained by the sum of three main parts. First, it is considered the derivative of the rotation around the x axis. After that, it is considered the derivative of the rotation around the y, but before adding this new term, this vector should be multiplied by R_ϕ in order to represent this vector in the new frame generated by the fist

rotation. And finally, the derivative of the rotation around the Z should be multiplied by $R_\phi R_\theta$ to take into account the second and the first rotation. The general equation is given by

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R_\phi \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_\phi R_\theta \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}. \quad (2.28)$$

This equation can be reorganized and expressed as

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \quad (2.29)$$

which is the kinematic equation for Euler angles.

2.3 Quaternions

The orientation can also be represented by quaternions. According to [Kuipers 1999], this parametrization is a good choice to represent rotations. Comparing it with other representations such as Euler angles, quaternions are not affected by singularity.

2.3.1 Quaternion Algebra

Quaternions, originally presented in [Hamilton 1848], represent an extension of complex numbers. According to [Cohn 2003], the quaternion algebra (\mathbb{H}) is generated by its basis elements $1, \vec{i}, \vec{j}$ and \vec{k} .

These elements are related by the follow rule:

$$\hat{i}^2 = \hat{j}^2 = \hat{k}^2 = \hat{i}\hat{j}\hat{k} = -1. \quad (2.30)$$

And every element $\mathbf{q} \in \mathbb{H}$ is represented as

$$\mathbf{q} := q_0 + q_1 \hat{i} + q_2 \hat{j} + q_3 \hat{k}, \quad (2.31)$$

where $q_0, q_1, q_2, q_3 \in \mathbb{R}$ are called Euler symmetric parameters. Analogously to complex numbers, quaternions have a real part $Re(\mathbf{q}) := q_0$ and an imaginary part $Im(\mathbf{q}) := [q_1 \ q_2 \ q_3]^T$. This is the quaternion notation that will be used in this work. But the reader should note that quaternion notations may change in other works such as [Crassidis and Markley 2003] and [Leeghim et al. 2010]. Using the notation defined in this

section , the sum of two quaternions $\mathbf{c} := c_0 + c_1\hat{i} + c_2\hat{j} + c_3\hat{k}$ and $\mathbf{d} := d_0 + d_1\hat{i} + d_2\hat{j} + d_3\hat{k}$ is given by:

$$\mathbf{c} \pm \mathbf{d} := c_0 \pm d_0 + (c_1 \pm d_1)\hat{i} + (c_2 \pm d_2)\hat{j} + (c_3 \pm d_3)\hat{k} \quad (2.32)$$

and the multiplication is defined as

$$\begin{aligned} \mathbf{c} \otimes \mathbf{d} := & (c_0d_0 - c_1d_1 - c_2d_2 - c_3d_3) + (c_0d_1 + d_0c_1 + c_2d_3 - c_3d_2)\hat{i} \\ & + (c_0d_2 - c_1d_3 + c_2d_0 + c_3d_1)\hat{j} + (c_0d_3 + c_1d_2 - c_2d_1 + c_3d_0)\hat{k}. \end{aligned} \quad (2.33)$$

In addition, considering that $\mathbf{q}, \mathbf{q}^{-1} \in \mathbb{H}$, the \mathbf{q}^{-1} is inverse if and only if $\mathbf{q} \otimes \mathbf{q}^{-1} = \mathbf{q}^{-1} \otimes \mathbf{q} = 1$. The equation of the inverse is given by:

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{\|\mathbf{q}\|^2}, \quad \text{for } \|\mathbf{q}\| \neq 0, \quad (2.34)$$

where the conjugate is

$$\mathbf{q}^* := Re(\mathbf{q}) - \iota_m Im(\mathbf{q}). \quad (2.35)$$

And the norm is defined as

$$\|\mathbf{q}\| := \sqrt{Re(\mathbf{q})^2 - Im(\mathbf{q})^T Im(\mathbf{q})}. \quad (2.36)$$

Only a subgroup of quaternions are used to represent rotations. According to [Menegaz 2016], if $\|\mathbf{q}\| = 1$, we call \mathbf{q} as quaternion of rotation or unity quaternion. But although unit quaternions form a group under quaternion multiplication, it does not form a group under the sum nor the scalar multiplication [Altmann 2005]. Considering a rotation of an angle ϑ around a unit vector n^* , the quaternion of rotation is defined by

$$\mathbf{q} = \cos\left(\frac{\vartheta}{2}\right) + \iota_m n^* \sin\left(\frac{\vartheta}{2}\right). \quad (2.37)$$

2.3.2 Rotation matrix from inertial to body frame

Unity quaternions are used to represent the quaternion rotation operator $L_q(v)$. According to [Kuipers 1999], this operator is defined as the product between the conjugate unity quaternion \mathbf{q}^* , a vector $\mathbf{v} \in \mathbb{R}^3$ and the unity quaternion \mathbf{q} .

$$L_q(v) = \mathbf{q}^* \mathbf{v} \mathbf{q}. \quad (2.38)$$

This rotation is interpreted as a frame rotation of an angle ϑ around \mathbf{q} axis. Which means that if we apply this rotation operator to a vector \mathbf{v} defined in the inertial frame, this vector can be expressed in the body frame as a vector \mathbf{W} . Therefore, the Eq. (2.38) can be rewritten as

$$\mathbf{W} = L_{\mathbf{q}}(\mathbf{v}) = (2q_0^2 - 1)\mathbf{v} + 2(\mathbf{v} \cdot \text{Im}(\mathbf{q}))\text{Im}(\mathbf{q}) + 2q_0(\mathbf{v} \times \text{Im}(\mathbf{q})), \quad (2.39)$$

where $\text{Im}(\mathbf{q}) = +q_1\hat{i} + q_2\hat{j} + q_3\hat{k}$ and

$$(2q_0^2 - 1)\mathbf{v} = \begin{bmatrix} 2(q_0^2 - 1) & 0 & 0 \\ 0 & 2(q_0^2 - 1) & 0 \\ 0 & 0 & 2(q_0^2 - 1) \end{bmatrix} \begin{bmatrix} v_X \\ v_Y \\ v_Z \end{bmatrix} \quad (2.40)$$

$$2(\mathbf{v} \cdot \text{Im}(\mathbf{q}))\text{Im}(\mathbf{q}) = \begin{bmatrix} 2q_1^2 & 2q_1q_2 & 2q_1q_3 \\ 2q_1q_2 & 2q_2^2 & 2q_2q_3 \\ 2q_1q_3 & 2q_2q_3 & 2q_3^2 \end{bmatrix} \begin{bmatrix} v_X \\ v_Y \\ v_Z \end{bmatrix} \quad (2.41)$$

$$2q_0(\mathbf{v} \times \text{Im}(\mathbf{q})) = \begin{bmatrix} 0 & 2q_0q_3 & -2q_0q_2 \\ -2q_0q_3 & 0 & 2q_0q_1 \\ 2q_0q_2 & -2q_0q_1 & 0 \end{bmatrix} \begin{bmatrix} v_X \\ v_Y \\ v_Z \end{bmatrix} \quad (2.42)$$

Using these equations, \mathbf{W} can be rewritten as

$$\mathbf{W} = R_i^b \begin{bmatrix} v_X \\ v_Y \\ v_Z \end{bmatrix} \quad (2.43)$$

$$\mathbf{W} = \begin{bmatrix} 2q_0^2 - 1 + 2q_1^2 & 2q_1q_2 + 2q_0q_3 & 2q_1q_3 - 2q_0q_2 \\ 2q_1q_2 - 2q_0q_3 & 2q_0^2 - 1 + 2q_2^2 & 2q_2q_3 + 2q_0q_1 \\ 2q_1q_3 + 2q_0q_2 & 2q_2q_3 - 2q_0q_1 & 2q_0^2 - 1 + 2q_3^2 \end{bmatrix} \begin{bmatrix} v_X \\ v_Y \\ v_Z \end{bmatrix}, \quad (2.44)$$

where R_i^b is the quaternion rotation matrix which transfer a vector from inertial to body frame

$$R_i^b = \begin{bmatrix} 2q_0^2 - 1 + 2q_1^2 & 2q_1q_2 + 2q_0q_3 & 2q_1q_3 - 2q_0q_2 \\ 2q_1q_2 - 2q_0q_3 & 2q_0^2 - 1 + 2q_2^2 & 2q_2q_3 + 2q_0q_1 \\ 2q_1q_3 + 2q_0q_2 & 2q_2q_3 - 2q_0q_1 & 2q_0^2 - 1 + 2q_3^2 \end{bmatrix}. \quad (2.45)$$

2.3.3 Kinematic equation

Quaternions can also be propagated in time. According to [Kuipers 1999] unity quaternions can be propagated in time using the angular velocity vector and the previous unity

quaternion.

$$\frac{dq}{dt} = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}, \quad (2.46)$$

or

$$\frac{dq}{dt} = \frac{1}{2} \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix}, \quad (2.47)$$

Where ω_n means angular velocity referring to axis n and q_0, q_1, q_2, q_3 are the quaternions terms from the previous instant.

2.4 Extended Kalman Filter (EKF)

The version of the EKF implemented was obtained from [Simon 2006]. This version is described as follows. First, consider the prediction and measurement models given by:

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, u_{k-1}, w_{k-1}) \quad (2.48)$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, v_k) \quad (2.49)$$

where \mathbf{x}_{k-1} is the 3×1 state vector and \mathbf{y}_k is the 3×1 measurement vector. we also assume that

$$w_k \sim (0, Q_k) \quad (2.50)$$

$$v_k \sim (0, R_k) \quad (2.51)$$

where Q and R are process and measurement noise covariance. In this filter, the state vector at time k is given by the Euler angles

$$\mathbf{x}_{k-1} = [\phi_{k-1} \quad \theta_{k-1} \quad \psi_{k-1}]^T. \quad (2.52)$$

And, the process model receives as input the angular velocity vector u_k given by:

$$u_{k-1} = \begin{bmatrix} \omega_{x_{k-1}} \\ \omega_{y_{k-1}} \\ \omega_{z_{k-1}} \end{bmatrix}. \quad (2.53)$$

Using an approximation of the process model for small values of time sample (ΔT) presented in [Titterton and Weston 1997], we can write Eq. (2.48) as:

$$\begin{bmatrix} \phi_k \\ \theta_k \\ \psi_k \end{bmatrix} = \begin{bmatrix} \phi_{k-1} \\ \theta_{k-1} \\ \psi_{k-1} \end{bmatrix} + \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} \omega_{x_{k-1}} \\ \omega_{y_{k-1}} \\ \omega_{z_{k-1}} \end{bmatrix} \Delta T + w_{k-1}. \quad (2.54)$$

Observe that this equation is a simplified version of the process model because it does not take into account the gyro bias. A more robust process model is presented in section 2.6.1. In addition, this filter also assumes that $\mathbf{y}_k = [\phi_k \quad \theta_k \quad \psi_k]^T$. This means that $\mathbf{y}_k = \mathbf{x}_k + v_k$.

The filter is initialized with:

$$\hat{\mathbf{x}}_0^+ = E(\mathbf{x}_0) \quad (2.55)$$

$$\mathbf{P}_0^+ = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0^+)(\mathbf{x}_0 - \hat{\mathbf{x}}_0^+)^T] \quad (2.56)$$

After that for $k=1,2,\dots$ the follow steps should be performed:

The follow partial derivatives are calculated,

$$\mathbf{F}_{k-1} = \frac{\partial \mathbf{f}_{k-1}}{\partial \mathbf{x}} \Big|_{\hat{\mathbf{x}}_{k-1}^+}, \quad (2.57)$$

$$\mathbf{L}_{k-1} = \frac{\partial \mathbf{f}_{k-1}}{\partial w} \Big|_{\hat{\mathbf{x}}_{k-1}^+}. \quad (2.58)$$

Considering $\mathbf{F}_{k-1_{ij}}$ the element of line i and column j of the jacobian matrix, \mathbf{F}_{k-1} is given by:

$$\mathbf{F}_{k-1_{11}} = 1 + \Delta t \cdot \tan \theta (\omega_y \cos \phi - \omega_z \sin \phi) \quad (2.59)$$

$$\mathbf{F}_{k-1_{12}} = \Delta t \cdot \sec^2 \theta (\omega_y \sin \phi + \omega_z \cos \phi) \quad (2.60)$$

$$\mathbf{F}_{k-121} = -\Delta t(\omega_y \sin \phi + \omega_z \cos \phi) \quad (2.61)$$

$$\mathbf{F}_{k-131} = \Delta t \cdot \sec \theta (\omega_y \cos \phi - \omega_z \sin \phi) \quad (2.62)$$

$$\mathbf{F}_{k-132} = \Delta t \cdot \sec \theta \cdot \tan \theta (\omega_y \sin \phi + \omega_z \cos \phi) \quad (2.63)$$

$$\mathbf{F}_{k-122} = D_{\mathbf{x}} f_{33} = 1 \quad (2.64)$$

$$\mathbf{F}_{k-113} = D_{\mathbf{x}} f_{23} = 0 \quad (2.65)$$

In addition is easy to see that $\mathbf{L}_{k-1} = I_3$, where the notation I_3 means: the identity matrix of order 3.

The next step is updating the predicted state estimation and estimation-error covariance matrix

$$\mathbf{P}_k^- = \mathbf{F}_{k-1} \mathbf{P}_{k-1}^+ \mathbf{F}_{k-1}^T + \mathbf{L}_{k-1} \mathbf{Q}_{k-1} \mathbf{L}_{k-1}^T \quad (2.66)$$

$$\hat{\mathbf{x}}_k^- = \mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1}^+, u_{k-1}, 0) \quad (2.67)$$

After, because the measurement model is linear, the optimum values of the Kalman gain, measurement update of the state estimate and the error covariance are given by the classic Kalman filter:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + R_k)^{-1}, \quad (2.68)$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k [\mathbf{y}_k - H_k \hat{\mathbf{x}}_k^-], \quad (2.69)$$

$$\mathbf{P}_k^+ = (I - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-. \quad (2.70)$$

2.5 Unscented Kalman Filter (UKF)

The version of the UKF implemented was obtained from [Simon 2006]. This version is described as follows. First, consider the system model given by:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, u_k, t_k) + w_k \quad (2.71)$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, t_k) + v_k \quad (2.72)$$

where \mathbf{x}_k is the 3×1 state vector and \mathbf{y}_k is the 3×1 measurement vector. we also assume that

$$w_k \sim N(0, Q_k) \quad (2.73)$$

$$v_k \sim N(0, R_k) \quad (2.74)$$

where Q and R are process and measurement noise covariance. In this filter the state vector at time k is given by the Euler angles

$$\mathbf{x}_k = [\phi_k \quad \theta_k \quad \psi_k]^T. \quad (2.75)$$

In addition, the process model receives as input the angular velocity vector u_k given by:

$$u_k = \begin{bmatrix} \omega_{x_k} \\ \omega_{y_k} \\ \omega_{z_k} \end{bmatrix}. \quad (2.76)$$

Using the process model proposed by [Titterton and Weston 1997], we can write Eq. (2.71) as:

$$\begin{bmatrix} \phi_{k+1} \\ \theta_{k+1} \\ \psi_{k+1} \end{bmatrix} = \begin{bmatrix} \phi_k \\ \theta_k \\ \psi_k \end{bmatrix} + \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} \omega_{x_k} \\ \omega_{y_k} \\ \omega_{z_k} \end{bmatrix} \Delta T + w_k. \quad (2.77)$$

In addition, this filter also assumes that $\mathbf{y}_k = [\phi_k \quad \theta_k \quad \psi_k]^T$. This means that $\mathbf{y}_k = \mathbf{x}_k + v_k$.

The filter initialization is done as follows:

$$\hat{\mathbf{x}}_0^+ = E(\mathbf{x}_0) \quad (2.78)$$

$$\mathbf{P}_0^+ = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0^+)(\mathbf{x}_0 - \hat{\mathbf{x}}_0^+)^T] \quad (2.79)$$

In order to propagate the state estimate and covariance, the following time update equations are used:

First the sigma points should be chosen using the best guess of mean $\hat{\mathbf{x}}_{k-1}^+$ and covariance \mathbf{P}_{k-1}^+ :

$$\hat{\mathbf{x}}_{k-1}^{(i)} = \hat{\mathbf{x}}_{k-1}^+ + \tilde{\mathbf{x}}^{(i)} \quad i = 1, \dots, 2n \quad (2.80)$$

$$\tilde{\mathbf{x}}^{(i)} = (\sqrt{n\mathbf{P}_{k-1}^+})^T \quad i = 1, \dots, n \quad (2.81)$$

$$\tilde{\mathbf{x}}^{(n+i)} = -(\sqrt{n\mathbf{P}_{k-1}^+})^T \quad i = 1, \dots, n \quad (2.82)$$

where these sigma points are also known as Cubature point set [Arasaratnam and Haykin 2009]. But this is just one of the variations of the UKF proposed by [Julier and Uhlmann 2004], more information about other variations can be found in [Menegaz 2016].

Now the sigma points can be used to propagating the estimated state $\hat{\mathbf{x}}_k^{(i)}$ using the non-linear system equation $\mathbf{f}(\cdot)$:

$$\hat{\mathbf{x}}_k^{(i)} = \mathbf{f}(\hat{\mathbf{x}}_{k-1}^{(i)}, u_k, t_k) \quad (2.83)$$

From 2.83, the a priori state can be obtained by:

$$\hat{\mathbf{x}}_k^- = \frac{1}{2n} \sum_{n=1}^{2n} \hat{\mathbf{x}}_k^{(i)} \quad (2.84)$$

And the a priori error covariance is computed doing

$$\hat{\mathbf{P}}_k^- = \frac{1}{2n} \sum_{n=1}^{2n} (\hat{\mathbf{x}}_k^{(i)} - \hat{\mathbf{x}}_k^-)(\hat{\mathbf{x}}_k^{(i)} - \hat{\mathbf{x}}_k^-)^T + Q_{k-1} \quad (2.85)$$

From this point will be implemented the measurement update equations. But, because the measurement model is linear, the optimum values of the Kalman gain, measurement update

of the state estimate and the error covariance are given by the classic Kalman filter:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + R_k)^{-1}, \quad (2.86)$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k [\mathbf{y}_k - H_k \hat{\mathbf{x}}_k^-], \quad (2.87)$$

$$\mathbf{P}_k^+ = (I - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-. \quad (2.88)$$

2.6 Process and measurement models utilized in USQUE and Riemannian filter

This section will present the process and measurement models utilized in USQUE and Riemannian filters. After that, it will present a generalization of the Rodrigues parameters which will be used in USQUE. Initially, doing some adaptations in Crassidis notation, a quaternion can be given as $\mathbf{q} \equiv [q_0 \ \varrho^T]^T$ with $\varrho \equiv [q_1 \ q_2 \ q_3]^T = \hat{\mathbf{e}} \sin(v/2)$ and $q_0 = \cos(v/2)$, where $\hat{\mathbf{e}}$ is the rotation axis and v is the rotation angle.

2.6.1 Process model

In order to obtain the process model, we started with the quaternion Kinematics equation for continuous time

$$\dot{\mathbf{q}}(t) = \frac{1}{2} \Xi[\mathbf{q}(t)] \omega(t), \quad (2.89)$$

where ω is a 3X1 angular velocity vector and

$$\Xi(\mathbf{q}) \equiv \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix}. \quad (2.90)$$

After performing a trapezoidal discretization of Eq. (2.89), the quaternion prediction equation in terms of $\hat{\omega}_k^+$ and \hat{q}_k^+ is given by

$$\hat{q}_{k+1}^- = \Omega(\hat{\omega}_k^+) \hat{q}_k^+, \quad (2.91)$$

with,

$$\Omega(\hat{\omega}_k^+) = \begin{bmatrix} \hat{\psi}_k^+ & \cos(0.5\|\hat{\omega}_k^+\|\Delta t)I_{3 \times 3} - [\hat{\psi}_k^+ \times] \\ \cos(0.5\|\hat{\omega}_k^+\|\Delta t) & -\hat{\psi}_k^{+T} \end{bmatrix} \quad (2.92)$$

where $\hat{\psi}_k^+ \equiv \sin(0.5\|\hat{\omega}_k^+\|\Delta t)\hat{\omega}_k^+/\|\hat{\omega}_k^+\|$ and Δt is the sampling interval.

Until now, we do have the model to estimate quaternions given by Eq. (2.91), but we also aim estimating the gyro bias. In order to do that, we used a gyroscope model widely used in the literature given by:

$$\tilde{\omega}(t) = \omega(t) + \beta(t) + \eta_v(t) \quad (2.93)$$

$$\dot{\beta}(t) = \eta_u(t) \quad (2.94)$$

where $\tilde{\omega}(t)$ is the continuous-time angular velocity and $\eta_v(t)$ and $\eta_u(t)$ are independent white-noise processes with:

$$E\{\eta_v(t)\eta_v^T(\tau)\} = I_{3 \times 3}\sigma_v^2\delta(t - \tau) \quad (2.95)$$

$$E\{\eta_u(t)\eta_u^T(\tau)\} = I_{3 \times 3}\sigma_u^2\delta(t - \tau) \quad (2.96)$$

where $\delta(t - \tau)$ is the Dirac delta function. Based on the postupdate $\hat{\beta}_k^+$, the postupdate angular velocity and propagated gyro bias are given by:

$$\hat{\omega}_k^+ = \tilde{\omega}_k - \hat{\beta}_k^+ \quad (2.97)$$

$$\hat{\beta}_{k+1}^- = \hat{\beta}_k^+ \quad (2.98)$$

Using Eq. (2.98) and Eq. (2.91), the complete process model is given by:

$$\begin{bmatrix} \mathbf{q}_k \\ \beta_k \end{bmatrix} = \begin{bmatrix} \Omega(\hat{\omega}_{k-1}^+) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{q}_{k-1} \\ \beta_{k-1} \end{bmatrix} + \varpi_k \quad (2.99)$$

where $\varpi_k \sim N([0]_{6 \times 1}, Q_k)$ is the process noise with

$$Q_k = \frac{\Delta t}{2} \begin{bmatrix} (\sigma_v^2 - \frac{1}{6}\sigma_u^2\Delta t^2)I_3 & 0_{3 \times 3} \\ 0_{3 \times 3} & \sigma_u^2 I_3 \end{bmatrix}. \quad (2.100)$$

where Δt is the time sample.

2.6.2 Measurement model (USQUE)

According to [Crassidis and Markley 2003], the discrete-time attitude observations for N measurements are given by:

$$\tilde{y}_K = \begin{bmatrix} A(q)r_1 \\ A(q)r_2 \\ \cdot \\ \cdot \\ A(q)r_N \end{bmatrix} + \begin{bmatrix} \nu_1 \\ \nu_2 \\ \cdot \\ \cdot \\ \nu_N \end{bmatrix} \quad (2.101)$$

where r_i is the i th 3×1 reference vector and ν_i is the i th sensor gaussian error-vector and

$$A(\mathbf{q}) = \begin{bmatrix} 2q_0^2 - 1 + 2q_1^2 & 2q_1q_2 + 2q_0q_3 & 2q_1q_3 - 2q_0q_2 \\ 2q_1q_2 - 2q_0q_3 & 2q_0^2 - 1 + 2q_2^2 & 2q_2q_3 + 2q_0q_1 \\ 2q_1q_3 + 2q_0q_2 & 2q_2q_3 - 2q_0q_1 & 2q_0^2 - 1 + 2q_3^2 \end{bmatrix}. \quad (2.102)$$

In addition, the reference vectors r_1 , r_2 and r_3 are given by:

$$r_1 = [1 \quad 0 \quad 0]^T \quad (2.103)$$

$$r_2 = [0 \quad 1 \quad 0]^T \quad (2.104)$$

$$r_3 = [0 \quad 0 \quad 1]^T \quad (2.105)$$

Therefore using Eq. (2.101) and Eq. (2.103)- Eq. (2.105), the final measurement model is given by:

$$\tilde{y}_K = \begin{bmatrix} A(q)r_1 \\ A(q)r_2 \\ A(q)r_3 \end{bmatrix} + \begin{bmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{bmatrix} \quad (2.106)$$

$$R_k = \text{diag}[\sigma_1^2 \quad \sigma_2^2 \quad \dots \quad \sigma_N^2] \quad (2.107)$$

2.6.3 Measurement model (RiSAdUKF)

Using an similar approach presented in [Menegaz 2016], the measurement model was defines as:

$$\tilde{y}_K = \begin{bmatrix} I_{4 \times 4} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} + \begin{bmatrix} \mathbf{q}_k \\ \beta_k \end{bmatrix} + w_k, \quad (2.108)$$

where $I_{4 \times 4}$ is the identity matrix, \mathbf{q}_k and β_k are the quaternion and the bias vector at time k and w_k is a vector composed by gaussian variables.

2.6.4 Rodrigues parameters

In addition, the local error-quaternion $\delta q \equiv [\delta q_0 \quad \delta \varrho^T]^T$ is obtained using Rodrigues parameters:

$$\delta p \equiv f[\delta \varrho / (a + \delta q_0)]. \quad (2.109)$$

The inverse transformation is given by:

$$\delta q_0 = \frac{-a \|\delta p\|^2 + f \sqrt{f^2 + (1 - a^2) \|\delta p\|^2}}{f^2 + \|\delta p\|^2} \quad (2.110)$$

$$\delta \varrho = f^{-1}(a + \delta q_0) \delta p \quad (2.111)$$

2.7 Unscented Quaternion Estimator(USQUE)

The version of the USQUE implemented was proposed by [Crassidis and Markley 2003]. Initially consider the discrete-time nonlinear models given by:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, k) + \mathbf{w}_k, \quad (2.112)$$

$$\tilde{\mathbf{y}}_k = \mathbf{h}(\mathbf{x}_k, k) + \mathbf{v}_k, \quad (2.113)$$

where $\mathbf{x}_k \in \mathbb{R}^{n \times 1}$ and $\tilde{\mathbf{y}}_k \in \mathbb{R}^{m \times 1}$ and \mathbf{f} and \mathbf{h} are nonlinear models of prediction and correction respectively. In addition, \mathbf{w}_k and \mathbf{v}_k are considered zero mean Gaussian noise with co-variance matrices respectively given by Q_k and R_k . In this work the state vector is

defined as

$$\mathbf{x} \triangleq \begin{bmatrix} \delta \mathbf{p} \\ \beta \end{bmatrix}, \quad (2.114)$$

where $\delta \mathbf{p} \in \mathbb{R}^{3 \times 1}$ is the error in generalized Rodrigues parameters, and $\beta \in \mathbb{R}^{3 \times 1}$ is the gyro-bias. The relation between the error in generalized Rodrigues parameters $\delta \mathbf{p}$ and the quaternion error $\delta \mathbf{q} = [\delta q_0 \quad \delta \varrho^T]$ is given by:

$$\delta q_0 = \frac{-a \|\delta \mathbf{p}\|^2 + f \sqrt{f^2 + (1 - a^2) \|\delta \mathbf{p}\|^2}}{f^2 + \|\delta \mathbf{p}\|^2}, \quad (2.115)$$

$$\delta \varrho = f^{-1}(a + \delta q_0) \delta \mathbf{p}, \quad (2.116)$$

where a is a parameter from 0 to 1 and f is a scale factor. The sigma points in terms of state vector are defined as

$$\chi_k(i) \triangleq \begin{bmatrix} \chi_k^{\delta p}(i) \\ \chi_k^{\beta}(i) \end{bmatrix}, \quad (2.117)$$

where $\chi_k^{\delta p}(i)$ refers to the attitude error part and $\chi_k^{\beta}(i)$ refers to the gyro-bias part. Given the state estimated $\hat{\mathbf{x}}_k^+$ and the matrices P_k^+ and

$$\bar{Q}_k = \frac{\Delta t}{2} \begin{bmatrix} (\sigma_v^2 - \frac{1}{6} \sigma_u^2 \Delta t^2) I_3 & 0_{3 \times 3} \\ 0_{3 \times 3} & \sigma_u^2 I_3 \end{bmatrix}, \quad (2.118)$$

the computation of sigma points is done as follows

$$\sigma_k \leftarrow 2n \text{ columns from } \pm \sqrt{(n + \lambda)[P_k^+ + \bar{Q}_k]}, \quad (2.119)$$

considering

$$\chi_k(0) = \hat{\mathbf{x}}_k^+, \quad (2.120)$$

the states will be given by:

$$\chi_k(i) = \sigma_k(i) + \hat{\mathbf{x}}_k^+, \quad (2.121)$$

where λ is a parameter that determines how spread are the sigma points.

2.7.1 Attitude propagation

First, the sigma points $\chi_k^{\delta p}(i)$ are converted into quaternions

$$\delta q_{0_k}^+(i) = \frac{-a \|\chi_k^{\delta p}(i)\|^2 + f \sqrt{f^2 + (1 - a^2) \|\chi_k^{\delta p}(i)\|^2}}{f^2 + \|\chi_k^{\delta p}(i)\|^2}, \quad (2.122)$$

$$\delta \varrho_k^+(i) = f^{-1}(a + \delta q_{0_k}^+(i)) \chi_k^{\delta p}(i), \quad (2.123)$$

and

$$\delta q_k^+(i) \equiv [\delta q_{0_k}^+(i) \quad \delta \varrho_k^+(i)]^T. \quad (2.124)$$

Assuming that

$$\hat{q}_k^+(0) = \hat{q}_k^+, \quad (2.125)$$

the new quaternion can be obtained by multiplying the error quaternion by the current estimate.

$$\hat{q}_k^+(i) = \delta q_k^+(i) \otimes \hat{q}_k^+. \quad (2.126)$$

After that, the quaternion can be propagated in time using the angular velocity vector:

$$\hat{q}_{k+1}^-(i) = \Omega[\hat{\omega}_k^+(i)] \hat{q}_k^+(i). \quad (2.127)$$

Using this result the quaternion error can be propagated by:

$$\delta q_{k+1}^-(i) = \hat{q}_{k+1}^-(i) \otimes [\hat{q}_{k+1}^-(0)]^{-1} \quad (2.128)$$

Based on Eq. (2.128) the quaternion error can be mapped into Rodrigues parameters using

$$\chi_{k+1}^{\delta p}(i) = f \frac{\delta \varrho_{k+1}^-(i)}{a + \delta q_{0_{k+1}}^-(i)} \quad (2.129)$$

$$\chi_{k+1}^\beta(i) = \chi_k^\beta(i). \quad (2.130)$$

2.7.2 Prediction calculations

Based on χ_{k+1} it is possible to obtain the state vector predicted $\hat{\mathbf{x}}_{k+1}^-$ by

$$\hat{\mathbf{x}}_{k+1}^- = \frac{1}{n + \lambda} \left\{ \lambda \chi_{k+1}(0) + \frac{1}{2} \sum_{i=1}^{2n} \chi_{k+1}(i) \right\}, \quad (2.131)$$

and the predicted co-variance matrix of the propagated states is given by:

$$\mathbf{P}_{k+1}^- = \frac{1}{n + \lambda} \{ \lambda [\chi_{k+1}(0) - \hat{\mathbf{x}}_{k+1}^-] [\chi_{k+1}(0) - \hat{\mathbf{x}}_{k+1}^-]^T + \frac{1}{2} \sum_{i=1}^{2n} [\chi_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^-] [\chi_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^-]^T \} + \bar{\mathbf{Q}}_k. \quad (2.132)$$

After that the predicted measurements are obtained by:

$$\hat{\mathbf{y}}_{k+1}^- = \frac{1}{n + \lambda} \{ \lambda \gamma_{k+1}(0) + \frac{1}{2} \sum_{i=1}^{2n} \gamma_{k+1}(i) \}, \quad (2.133)$$

where $\gamma_{k+1}(i) = \mathbf{h}(\chi_{k+1}(i), k)$.

The co-variance matrix associated with the predicted measurements is given by:

$$\mathbf{P}_{k+1}^{yy} = \frac{1}{n + \lambda} \{ \lambda [\gamma_{k+1}(0) - \hat{\mathbf{y}}_{k+1}^-] [\gamma_{k+1}(0) - \hat{\mathbf{y}}_{k+1}^-]^T + \frac{1}{2} \sum_{i=1}^{2n} [\gamma_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^-] [\gamma_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^-]^T \} \quad (2.134)$$

$$\mathbf{P}_{k+1}^{xy} = \frac{1}{n + \lambda} \{ \lambda [\chi_{k+1}(0) - \hat{\mathbf{x}}_{k+1}^-] [\gamma_{k+1}(0) - \hat{\mathbf{y}}_{k+1}^-]^T + \frac{1}{2} \sum_{i=1}^{2n} [\chi_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^-] [\gamma_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^-]^T \} \quad (2.135)$$

$$\mathbf{P}_{k+1}^{vv} = \mathbf{P}_{k+1}^{yy} + R_{k+1} \quad (2.136)$$

2.7.3 Correction step

The Kalman gain is calculated

$$\mathbf{K}_k = \mathbf{P}_k^{xy} (\mathbf{P}_k^{yy})^{-1} \quad (2.137)$$

$$\mathbf{x}_k^+ = \mathbf{x}_k^- + \mathbf{K}_k (\tilde{\mathbf{y}}_k - \hat{\mathbf{y}}_k^-) \quad (2.138)$$

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_k^{vv} \mathbf{K}_k^T \quad (2.139)$$

2.8 Riemannian-Spheric Additive Unscented Kalman Filter(RiSAdUKF)

Recent achievements on Unscented Kalman filtering theory for Riemannian manifolds presented in [Menegaz 2016], had led to the RiSAdUKF. This filter was proposed under a consistent theory and also presented a better performance when compared with other quaternion filters. This Riemannian filter is initialized with the following states $\hat{\mathbf{x}}_{0|0} := \bar{\mathbf{x}}_0$ and $\hat{\mathbf{P}}_{xx}^{0|0} := \mathbf{P}_{xx}^0$. After that, for $k = 1, 2, \dots$ the following steps should be performed:

Initially it is obtained the sigma representation for the states represented on the tangent space(TM) by:

$$\{\chi_{i,k-1|k-1}^{TM}, w_i^{1,m}, w_i^{1,c}, \bullet\}_{i=1}^{N_1} = \sigma R_1([0]_{n_x \times 1}, \hat{\mathbf{P}}_{xx}^{k-1|k-1}) \quad (2.140)$$

where, σ refers to the sigma method presented in appendix E .

After obtaining these sigma points on tangent space, they are represented on canonical space using the exponential mapping given by:

$$\chi_i^{k-1|k-1} := exp_{\hat{\mathbf{x}}_{k-1|k-1}}^{e_x}(TBtoCB(\chi_{i,k-1|k-1}^{TM})), \quad i = 1, \dots, N_1. \quad (2.141)$$

where, $TBtoCB$ refers to the transformation from tangent basis to canonical basis presented in appendix I and $exp_{\hat{\mathbf{x}}_{k-1|k-1}}^{e_x}$ refers to the exponential mapping presented in appendix B.

Based on this in canonical basis representation, we can calculate the state's predicted sigma points by:

$$\chi_{i,*}^{k|k-1} := f_k(\chi_i^{k-1|k-1}), \quad i = 1, \dots, N_1. \quad (2.142)$$

Until now, we do have predicted quaternions given by $\chi_{i,*}^{k|k-1}$. The next step consists of finding the mean of these quaternions using a optimization algorithm to find the quaternion which has the smallest distance from the other quaternions. This can be done doing

$$\hat{\mathbf{x}}_{k|k-1} := arg \min_{a \in M_x} \sum_{i=1}^{N_1} w_i^{1,m} dist^2(\chi_{i,*}^{k|k-1}, a). \quad (2.143)$$

where the previous equation was implemented using the gradient descent algorithm presented in appendix C.

After that, we have to find the predicted covariance estimate represented on the tangent space by:

$$\hat{\mathbf{P}}_{xx}^{k|k-1} := \sum_{i=1}^{N_1} w_i^{1,c} (CBtoTB(\log_{\hat{\mathbf{x}}_{k|k-1}}^{e_x}(\chi_{i,*}^{k|k-1}))) (\diamond)^T + Q_k. \quad (2.144)$$

where \log refers to the logarithm mapping presented in appendix B.

Based on this predicted covariance, the regenerated predicted states represented on the tangent space can be obtained by:

$$\{\chi_{i,k|k-1}^{TM}, w_i^{2,m}, w_i^{2,c}, w_i^{2,cc}\}_{i=1}^{N_2} := \sigma R_2([0]_{n_x \times 1}, \hat{\mathbf{P}}_{xx}^{k|k-1}) \quad i = 1, \dots, N_2. \quad (2.145)$$

After that, we do using the exponential mapping to convert from the tangent space to canonical space by:

$$\chi_i^{k|k-1} := \exp_{\hat{\mathbf{x}}_{k|k-1}}^{e_x}(TBtoCB(\chi_{i,k|k-1}^{TM})), \quad i = 1, \dots, N_2. \quad (2.146)$$

Now, we can calculate the measurement's predicted sigma points through

$$\gamma_i^{k|k-1} := h_k(\chi_i^{k|k-1}), \quad i = 1, \dots, N_2. \quad (2.147)$$

where h_k , for this work, assume the form presented in 2.108.

And based on these measurement's predicted sigma points, we can calculate the measurement's predicted estimate by:

$$\hat{\mathbf{y}}_{k|k-1} := \arg \min_{b \in M_y} \sum_{i=1}^{N_2} w_i^{2,m} \text{dist}^2(\gamma_i^{k|k-1}, b). \quad (2.148)$$

After that, the measurement's estimated predicted covariance and the estimated predicted cross-covariance are given by:

$$\hat{\mathbf{P}}_{yy}^{k|k-1} := \sum_{i=1}^{N_2} w_i^{2,c} (CBtoTB(\log_{\hat{\mathbf{y}}_{k|k-1}}^{e_y}(\gamma_i^{k|k-1}))) (\diamond)^T + R_k \quad (2.149)$$

and

$$\hat{\mathbf{P}}_{xy}^{k|k-1} := \sum_{i=1}^{N_2} w_i^{2,cc} (CBtoTB(\log_{\hat{\mathbf{x}}_{k|k-1}}^{e_x}(\chi_i^{k|k-1}))) (CBtoTB(\log_{\hat{\mathbf{y}}_{k|k-1}}^{e_y}(\gamma_i^{k|k-1})))^T. \quad (2.150)$$

Based on these two matrices we can calculating the Kalman Gain by:

$$G_k := (\hat{\mathbf{P}}_{xy}^{k|k-1})(\hat{\mathbf{P}}_{yy}^{k|k-1})^{-1}. \quad (2.151)$$

After that, the Kalman Gain is used to correct state estimated on the tangent space by:

$$\hat{\boldsymbol{x}}_{k|k}^{TM} := \hat{\boldsymbol{x}}_{k|k-1}^{TM} + G_k(CBtoTB(\log_{\hat{\boldsymbol{y}}_{k|k-1}}^{e_y}(\boldsymbol{y}_k))). \quad (2.152)$$

And the corrected covariance of the estimate at $\hat{\boldsymbol{x}}_{k|k-1}$ is given by:

$$\hat{\boldsymbol{P}}_{\boldsymbol{x}\boldsymbol{x}}^{k|k, \hat{\boldsymbol{x}}_{k|k-1}} := \hat{\boldsymbol{P}}_{\boldsymbol{x}\boldsymbol{x}}^{k|k-1} - (G_k) \hat{\boldsymbol{P}}_{\boldsymbol{y}\boldsymbol{y}}^{k|k-1} (G_k)^T. \quad (2.153)$$

After that, the corrected state estimation is given by:

$$\hat{\boldsymbol{x}}_{k|k} := \exp_{\hat{\boldsymbol{x}}_{k|k-1}}^{e_x}(TBtoCB(\hat{\boldsymbol{x}}_{k|k}^{TM})) \quad (2.154)$$

And finally the state's corrected covariance estimate at $\hat{\boldsymbol{x}}_{k|k}$ is given by:

$$\hat{\boldsymbol{P}}_{\boldsymbol{x}\boldsymbol{x}}^{k|k} := PT(\hat{\boldsymbol{P}}_{\boldsymbol{x}\boldsymbol{x}}^{k|k, \hat{\boldsymbol{x}}_{k|k-1}}, \hat{\boldsymbol{x}}_{k|k-1}, \hat{\boldsymbol{x}}_{k|k}). \quad (2.155)$$

where PT refers to the parallel transport of a matrix presented in appendix H.

Chapter 3

Simulator dynamic model

In order to simulate each attitude representation advantages and disadvantages, this section will present the mathematical derivation of a nanosatellite test platform dynamic model. Initially, there will be defined basic terms and equations. These equations will be presented as propositions followed by their proofs. After that, those definitions and propositions will be used to describe the model.

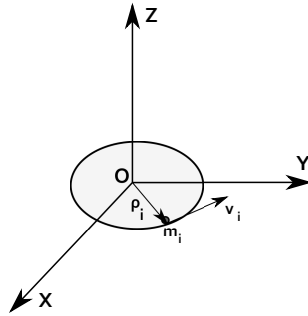


Figure 3.1: Angular momentum references frames

Fig.3.1 illustrates the rotation of a body composed by m_i infinitesimal masses placed in distances ρ_i from O placed at the center of a xyz reference frame. According to [Nussenzveig 2002], the angular momentum can be understood as a linear momentum applied to a mass point located at a certain distance from the rotation axis. Mathematically, the angular momentum, which in this work is represented by the vector l , is defined as the cross product between a distance vector ρ_i and a linear velocity vector v_i experienced by a mass m_i times this mass

$$l_i = \rho_i \times (m_i v_i). \quad (3.1)$$

This is the main definition that will be used from now. In order to derive the table dynamic model, some propositions must be addressed.

Proposition 1: The time rate of change of the angular momentum vector l about a fixed point O is equal to the total external torque T , about O , generated by external forces acting

on the system

$$\frac{d\mathbf{l}}{dt} = \mathbf{T}_{external}. \quad (3.2)$$

Proof:

Based on [Greenwood 1965] and using Eq. (3.1), the angular momentum can be expressed as a cross product between a distance vector $\boldsymbol{\rho}_i$ and a linear velocity vector \mathbf{v}_i times a mass m_i

$$\mathbf{l}_i = \boldsymbol{\rho}_i \times m_i \dot{\boldsymbol{\rho}}_i, \quad (3.3)$$

where $\dot{\boldsymbol{\rho}}_i$ is used to denote the first derivative of $\boldsymbol{\rho}_i$, which is equal to \mathbf{v}_i . Once you know the angular momentum of an infinitesimal particle, the total angular momentum can be obtained by the contribution of all body particles

$$\mathbf{l}_{total} = \sum_{i=1}^n \mathbf{l}_i. \quad (3.4)$$

From Eq. (3.3) and Eq. (3.4), the first derivative of the total angular momentum is expressed as the first derivative of the sum of all \mathbf{l}_i

$$\dot{\mathbf{l}}_{total} = \frac{d(\sum_{i=1}^n \boldsymbol{\rho}_i \times m_i \dot{\boldsymbol{\rho}}_i)}{dt} \quad (3.5)$$

$$\dot{\mathbf{l}}_{total} = \sum_{i=1}^n \dot{\boldsymbol{\rho}}_i \times m_i \dot{\boldsymbol{\rho}}_i + \sum_{i=1}^n \boldsymbol{\rho}_i \times m_i \ddot{\boldsymbol{\rho}}_i. \quad (3.6)$$

By definition, the cross product between two identical vectors is zero, so $\dot{\boldsymbol{\rho}}_i \times \dot{\boldsymbol{\rho}}_i = 0$ and because of this

$$\dot{\mathbf{l}}_{total} = \sum_{i=1}^n \boldsymbol{\rho}_i \times m_i \ddot{\boldsymbol{\rho}}_i. \quad (3.7)$$

Assuming that internal forces sum up to zero

$$\dot{\mathbf{l}}_{total} = \sum_{i=1}^n \boldsymbol{\rho}_i \times \mathbf{F}_i, \quad (3.8)$$

where \mathbf{F}_i represents external forces acting on the system. From Eq. (3.8), the first derivative of the total angular moment, in this work also represented by \mathbf{l} , is equal to the sum of all

external torques acting on the body.

$$\dot{\mathbf{l}}_{total} = \dot{\mathbf{l}} = \mathbf{T}_{external}. \quad (3.9)$$

This result is important and later it will be used to connect the body angular momentum equation with the external torques considered in the model.

Proposition 2: The angular momentum of a body is given by the product between the body tensor of inertia I and the angular velocity vector $\boldsymbol{\omega}$.

$$\mathbf{l} = I\boldsymbol{\omega}. \quad (3.10)$$

Proof:

From [Greenwood 1965] we have that the angular momentum is given by:

$$\mathbf{l} = \int_m \boldsymbol{\rho} \times (\boldsymbol{\omega} \times \boldsymbol{\rho}) dm. \quad (3.11)$$

In order to provide an explicit description of inertia tensor terms, the Eq. (3.11) will be expanded. Let $\boldsymbol{\rho}$ and $\boldsymbol{\omega}$ be the follow vectors:

$$\boldsymbol{\rho} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}, \quad (3.12)$$

$$\boldsymbol{\omega} = \omega_x\mathbf{i} + \omega_y\mathbf{j} + \omega_z\mathbf{k}. \quad (3.13)$$

So $\boldsymbol{\omega} \times \boldsymbol{\rho}$ is given by:

$$\boldsymbol{\omega} \times \boldsymbol{\rho} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \omega_x & \omega_y & \omega_z \\ x & y & z \end{vmatrix} = (z\omega_y - y\omega_z)\mathbf{i} + (x\omega_z - z\omega_x)\mathbf{j} + (y\omega_x - x\omega_y)\mathbf{k} \quad (3.14)$$

Following the same process $\boldsymbol{\rho} \times (\boldsymbol{\omega} \times \boldsymbol{\rho})$ can be written as

$$\boldsymbol{\rho} \times (\boldsymbol{\omega} \times \boldsymbol{\rho}) = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x & y & z \\ (z\omega_y - y\omega_z) & (x\omega_z - z\omega_x) & (y\omega_x - x\omega_y) \end{vmatrix} \quad (3.15)$$

$$\begin{aligned}
\boldsymbol{\rho} \times (\boldsymbol{\omega} \times \boldsymbol{\rho}) &= [(y^2 + z^2)\omega_x - xy\omega_y - xz\omega_z]\mathbf{i} \\
&+ [-yx\omega_x + (x^2 + z^2)\omega_y - yz\omega_z]\mathbf{j} \\
&+ [-zx\omega_x - zy\omega_y + (x^2 + y^2)\omega_z]\mathbf{k}.
\end{aligned} \tag{3.16}$$

Applying Eq. (3.16) in Eq. (3.11), we have that:

$$\begin{aligned}
\int_m \boldsymbol{\rho} \times (\boldsymbol{\omega} \times \boldsymbol{\rho}) dm &= \left[\int_m (y^2 + z^2)\omega_x dm - \int_m xy\omega_y dm - \int_m xz\omega_z dm \right] \mathbf{i} \\
&+ \left[- \int_m yx\omega_x dm + \int_m (x^2 + z^2)\omega_y dm - \int_m yz\omega_z dm \right] \mathbf{j} \\
&+ \left[- \int_m zx\omega_x dm - \int_m zy\omega_y dm + \int_m (x^2 + y^2)\omega_z dm \right] \mathbf{k}.
\end{aligned} \tag{3.17}$$

At this point we can define the products and moments of inertia. The moments of inertia are defined as follows:

$$I_{xx} = \int_m (y^2 + z^2)\omega_x dm, \tag{3.18}$$

$$I_{yy} = \int_m (x^2 + z^2)\omega_y dm, \tag{3.19}$$

$$I_{zz} = \int_m (x^2 + y^2)\omega_z dm. \tag{3.20}$$

The products of inertia are given by:

$$I_{xy} = I_{yx} = - \int_m xy dm, \tag{3.21}$$

$$I_{xz} = I_{zx} = - \int_m xz dm, \tag{3.22}$$

$$I_{yz} = I_{zy} = - \int_m yz dm. \tag{3.23}$$

From those definitions, the Eq. (3.17) can be written as:

$$\begin{aligned} \int_m \boldsymbol{\rho} \times (\boldsymbol{\omega} \times \boldsymbol{\rho}) dm &= [I_{xx}\omega_x + I_{xy}\omega_y + I_{xz}\omega_z] \mathbf{i} \\ &+ [I_{yx}\omega_x + I_{yy}\omega_y + I_{yz}\omega_z] \mathbf{j} \\ &+ [I_{zx}\omega_x + I_{zy}\omega_y + I_{zz}\omega_z] \mathbf{k}. \end{aligned} \quad (3.24)$$

Now, defining the tensor of inertia as:

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}_{3 \times 3}. \quad (3.25)$$

The Eq. (3.24) can be written as:

$$\mathbf{l} = I\boldsymbol{\omega}. \quad (3.26)$$

3.1 Angular momentum first derivative referring to the center of rotation

From Eq. (3.26), the angular momentum equation referring to the center of rotation is given by:

$$\mathbf{l}_o = I\boldsymbol{\omega}. \quad (3.27)$$

The first time derivative of Eq. (3.27) is given by:

$$\frac{d\mathbf{l}_o}{dt} = (\dot{\mathbf{l}}_o)_{xyz} + \boldsymbol{\omega} \times \mathbf{l}_o, \quad (3.28)$$

where $(\dot{\mathbf{l}}_o)_{xyz}$ refers to the body frame. From Eq. (3.27) we can express the first angular momentum derivative as:

$$\frac{d\mathbf{l}_o}{dt} = I\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times I\boldsymbol{\omega}. \quad (3.29)$$

3.2 Dynamic equations

The propositions presented until now shown two main facts. First, the angular momentum time derivative is equal to external torques shown in Eq. (3.2). After that, the second proposition shows that the angular momentum could be expressed as the product between the tensor of inertia times the angular velocity vector shown in Eq. (3.26). Furthermore, as shown in Eq. (3.29) the derivative of the angular momentum referring to the center of rotation was obtained in terms of body tensor of inertia, angular velocity vector and it's first derivative. This subsection has the objective of put together all those conclusions, in order to generate a model that relates external torques with the body tensor of inertia.

But before that, some considerations must be made. First, this platform can be modeled as a rigid body with center of mass (CM) located at a certain distance from the center of rotation (CR). In addition, this model have two reference systems. One of them XYZ is fixed referring to earth, also called inertial frame. And the second one xyz is fixed referring to the table and presented in Fig.3.2. In another words, this second frame rotates with the table. These rotations are presented in Fig.3.5, Fig.3.3 and Fig.3.4 which represent successive rotations of roll, pitch and yaw respectively.

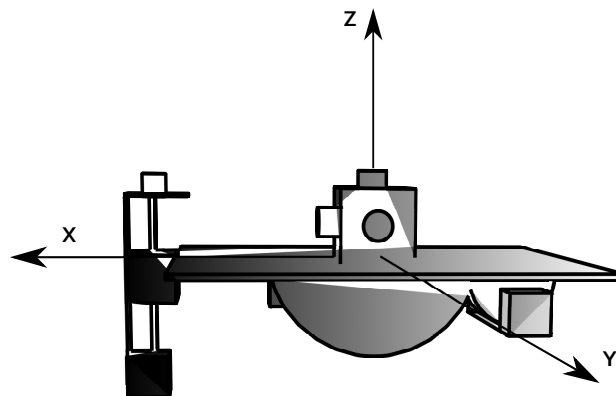


Figure 3.2: Body frame representation

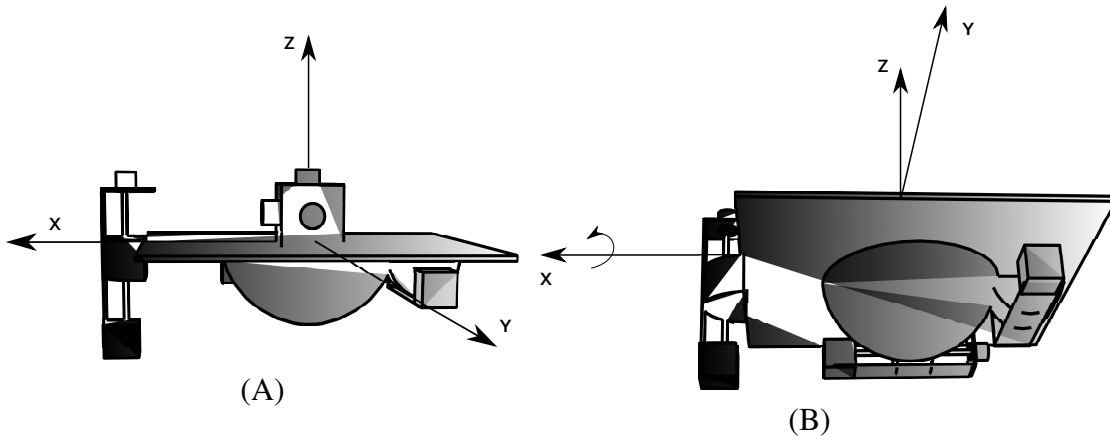


Figure 3.3: Graphical representation; **A.** Initial position; **B.** Roll rotation.

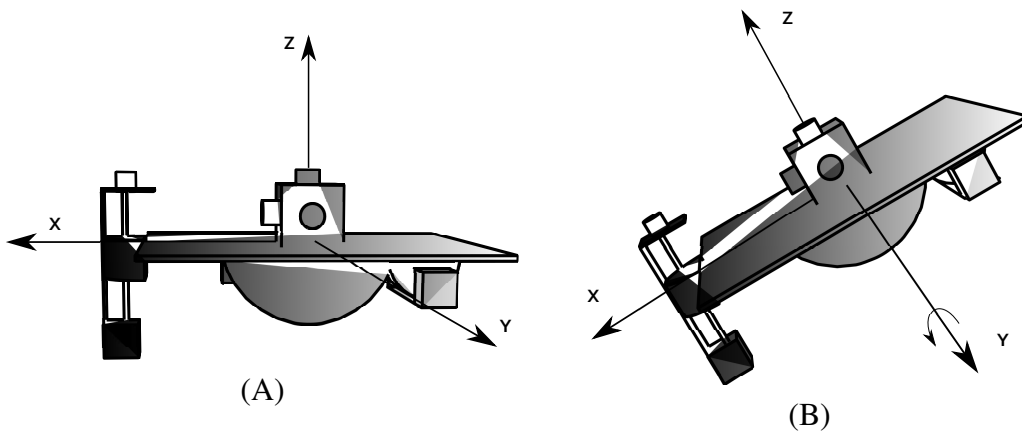


Figure 3.4: Graphical representation; **A.** Initial position; **B.** Pitch rotation.

From now, the inertial reference frame will be referred by the letter "i" and the body frame by the letter "b".

Continuing the derivation of the table dynamic equation, we can rewrite Eq. (3.29):

$$\frac{dl_o}{dt} = I\dot{\omega} + \omega \times I\omega. \quad (3.30)$$

From proposition one and second Newton's law, external torques are related with the angular momentum by

$$\mathbf{T} = \frac{d\mathbf{l}}{dt}. \quad (3.31)$$

The resulting external torque \mathbf{T} presented in Eq. (3.31) can be written as the sum of three main external torques

$$\mathbf{T} = \mathbf{T}_{actuators} + \mathbf{T}_{gravitational} + \mathbf{T}_{aerodynamic}, \quad (3.32)$$

where $\mathbf{T}_{actuators}$ refers to torques generated by components connected to the table, for

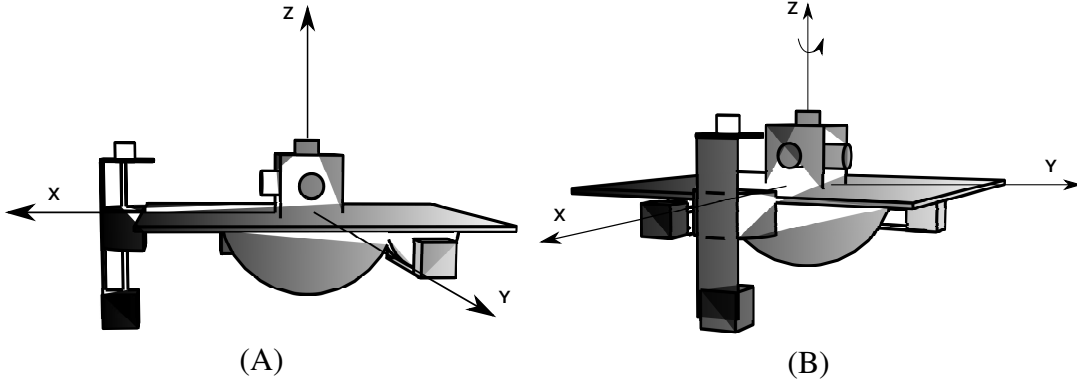


Figure 3.5: Graphical representation; **A.** Initial position; **B.** Yaw rotation.

example actuator such as reaction wheels, $T_{gravitational}$ represents the gravitational torque and $T_{aerodynamicdrift}$ represents the aerodynamic torque.

From Eq. (3.30), Eq. (3.31) and Eq. (3.32) we have that

$$\frac{d\mathbf{l}_o}{dt} = I\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times I\boldsymbol{\omega} = \mathbf{T}. \quad (3.33)$$

where

$$I\boldsymbol{\omega} = \begin{bmatrix} I_{xx}\omega_x + I_{xy}\omega_y + I_{xz}\omega_z \\ I_{yx}\omega_x + I_{yy}\omega_y + I_{yz}\omega_z \\ I_{zx}\omega_x + I_{zy}\omega_y + I_{zz}\omega_z \end{bmatrix}, \quad (3.34)$$

and

$$\mathbf{T} = [T_x \quad T_y \quad T_z]^T. \quad (3.35)$$

Reorganizing Eq. (3.33) we can obtain the angular momentum first derivative in terms of the inertia tensor, external torque vector and angular velocity vector:

$$\frac{d\boldsymbol{\omega}}{dt} = I^{-1}(\mathbf{T} - \boldsymbol{\omega} \times I\boldsymbol{\omega}). \quad (3.36)$$

This format of the platform dynamic equation is a important result because it will be used in the next section to run numerical simulations.

3.3 Numerical simulations and experimental verification

According to [Tewari 2007], considering the rotation sequence 3-2-1, the Euler angles reach the singularity region when $\theta = \pm 90^\circ$. This section has the objective of showing the singularity in Euler angles and also to show that quaternions can avoid it. In order to

address this issue, the singularity effect was simulated (see subsection 3.3.1) and after that it was designed an experiment to observe this effect in experimental data collected from the platform (see subsection 3.3.2).

3.3.1 Simulation

The simulation block diagram is presented in Fig.3.6. As it can be seen, This Theoretical model block receives as input the stop time, initial angles and initial angular velocities.

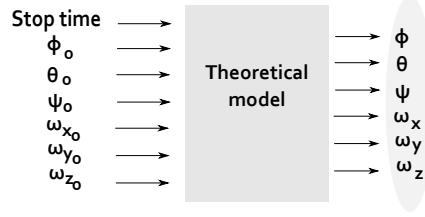


Figure 3.6: Simulation block diagram

Based on these inputs, the program integrates the first derivative of the angular velocity vector and orientation using Eq. (3.36), (3.37) and (3.38). The result of this integration will be the new orientation and angular velocity vector used in the next iteration.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}. \quad (3.37)$$

$$\frac{dq}{dt} = \frac{1}{2} \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix}, \quad (3.38)$$

Initially, we evaluated the attitude propagation using the kinematics model based on Euler angles. Even before simulating, just by matrix terms inspection, we observed that the matrix terms $\sin \phi \tan \theta$ and $\sin \phi \sec \theta$ would increase indefinitely when θ approaches 90° .

As we expected, this singularity was observed during simulations presented in Fig.(3.7). This figure has a singularity flag activated when $|\cos(\theta)| < 0.07$. As it can be seen, when the $\cos \theta$ approaches zero, the angle values of Yaw and Roll diverge to almost 200 degrees. In addition, it should be point out that this value does not go to infinity just because Matlab approximates $\cos(90^\circ)$ by 6.1232×10^{-17} .

The second evaluation was done using the propagation matrix based in quaternions. The simulation was designed in order to lead the system to the singularity region observed previously. The simulation result, presented in Fig.3.8 shows that when the pitch angle reaches absolute values closer to 90° , the values of roll and yaw are not affected by the singularity.

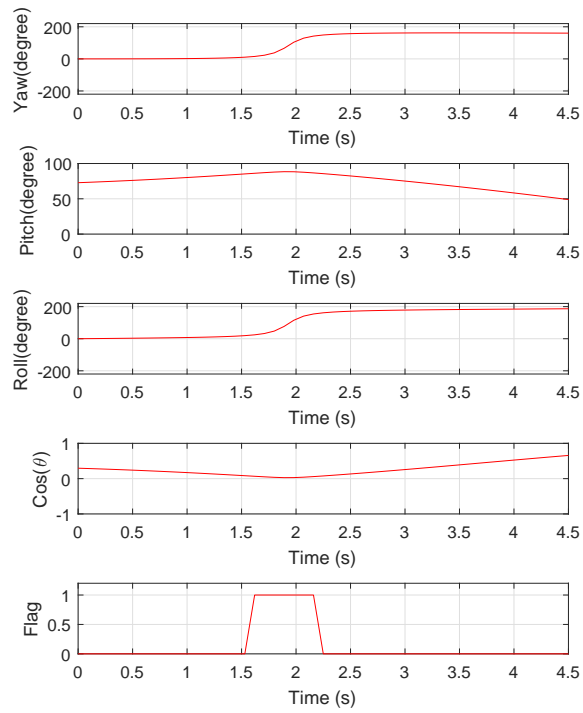


Figure 3.7: Singularity observed in Roll and Yaw

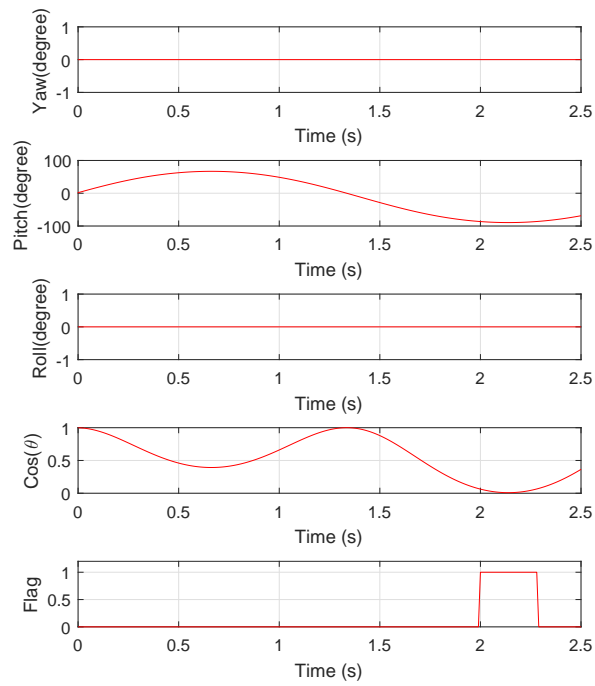


Figure 3.8: Singularity avoidance using quaternion

3.3.2 Experimental verification

In order to observe the singularity effect on experimental data, the platform was initially placed on a horizontal position and the IMUs were placed with an initial angle, referring to the surface of the platform, such that it was possible reach values of θ closer to 90° (see Fig.3.9).

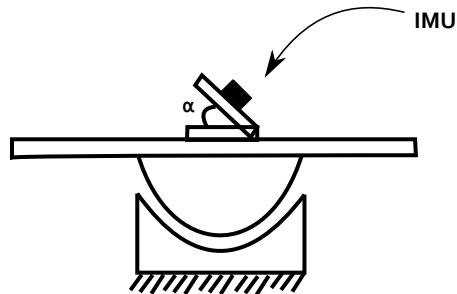


Figure 3.9: Schematic view of the experiment

The experiment consisted of rotating the platform around the Y axis in order to reach pitch angles closer to 90° . From theory, the singularity happens in this specific orientation. As it can be seen in Fig.3.10, when the absolute value of pitch reaches angles closer to 90° the singularity happens and affects roll and yaw measurements as it was expected.

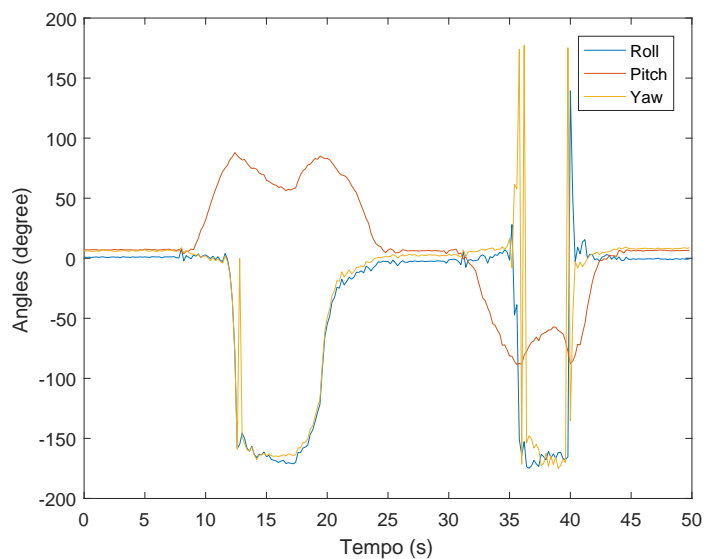


Figure 3.10: Singularity observed in Roll and Yaw

But the same does not happens when the orientation was parameterized with quaternions. As it can be seen in Fig.3.11, when θ reaches values closer to 90° the singularity was avoided.

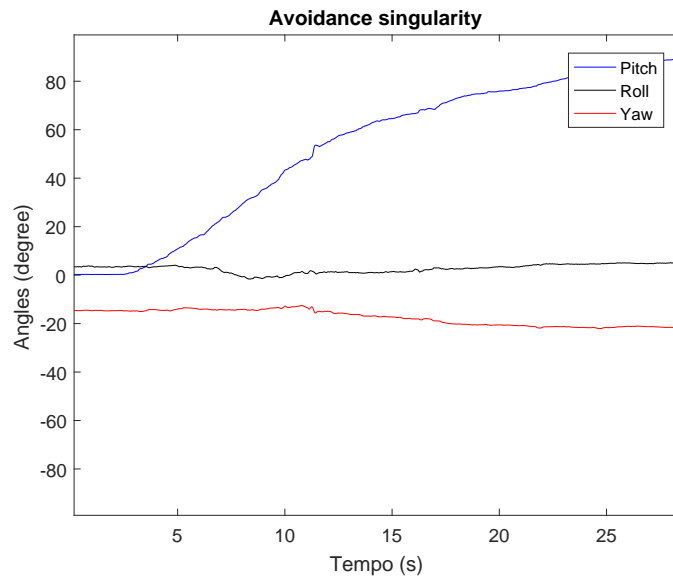


Figure 3.11: Singularity avoidance using quaternion

Chapter 4

System Architecture

This section is dedicated to the subsystems utilized in this research. This chapter will provide a detailed description of the two attitude determination systems embedded on LAICA's platform. One of them was developed in the University of Brasilia (UNB) using a 9DOF IMU from Adafruit. The second one was obtained from Yost Labs.

Initially, it will be described the sensors utilized for each attitude determination system. After that, it will presented hardware project for each system followed by sensor calibration procedures. Finally, it will be discussed some experimental issues and the solutions applied.

4.1 Sensors

4.1.1 9DOF IMU from Adafruit

This 9DOF IMU from Adafruit integrates one of the attitude determination systems in LAICA's facility. The location of this sensor is illustrated in Fig.4.1. As it can be seen, this sensor is located closer to the center of rotation and is utilized to measure the platform orientation. This orientation is calculated based on three axis measurements from accelerometers, gyroscopes and magnetometers.

These sensors are encapsulated in two electronic devices, which are a LSM303DLHC and a L3GD20. The specifications of those components are presented in table 4.1. In addition, it is important to mention that those specifications were set in order to meet the platform's expected operation range, but they can also be adjusted to different ranges through register settings.

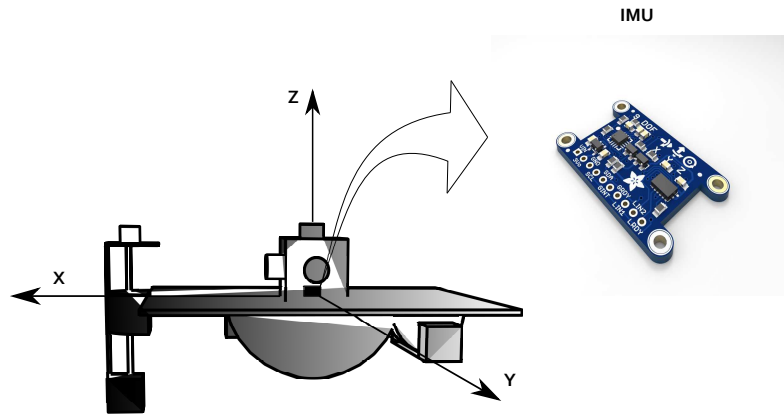


Figure 4.1: IMU location

Table 4.1: LSM303DLHC and L3GD20 settings

| System parameters | specifications |
|---------------------|---|
| Sample time | 0.1 s |
| Temperature range | -40°C to 85°C |
| Accelerometer scale | $\pm 2g$ |
| Gyro scale | $\pm 250^{\circ}/\text{sec}$ |
| Gyro bias | not specified |
| Compass scale | ± 1.3 Gauss |
| Magnetic resolution | 2 mGauss |
| Orientation range | 360° about all axes |
| Supply voltage | 3 V |

4.1.2 9DOF IMU from Yost Labs

The second inertial measurement unity used to measure the platform orientation was produced by Yost Labs. The location and a upper view of this unity are presented in Fig.4.2.

This IMU is a closed solution certificated under Industry Canada(IC)Radio Standards Specification(RSS). And some of the important specifications are presented in table 4.2.

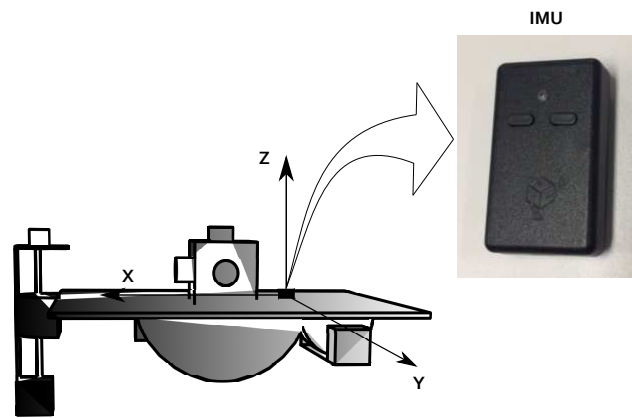


Figure 4.2: spaceSensor

Table 4.2: Yost Labs IMU specifications

| System parameters | specifications |
|---------------------------|---|
| Sample time | 0.01 s |
| Temperature range | -40°C to 85°C |
| Orientation accuracy | $\pm 1^{\circ}$ |
| Orientation resolution | $< 0.08^{\circ}$ |
| Orientation repeatability | $< 0.085^{\circ}$ for all directions |
| Accelerometer scale | $\pm 2g$ |
| Gyro scale | $\pm 250^{\circ}/\text{sec}$ |
| Gyro bias | $2.5^{\circ}/\text{hr}$ average for all axes |
| Compass scale | ± 1.3 Gauss |
| Orientation range | 360° about all axes |
| Supply voltage | 5 V |

4.2 Hardware configuration

The hardware configuration used by each inertial measurement unity is different. In order to provide a better description of each hardware solution, the following subsections will focus on each configuration separately.

4.2.1 Hardware configuration for Adafruit IMU

The schematic view of the hardware configuration used for Adafruit’s IMU is presented in Fig.4.3. This picture shows to big blocks that communicates through XBees. The PC block represents the client side. In this side, the operator receives the orientation data from the XBee, processes this information on Matlab and sends or not a command to the platform. On the platform side, we do have an arduino, powered by a 12V DC battery, responsible for collecting data from IMU’s gyroscopes, magnetometers and accelerometers.

After collecting this data, this microcontroller uses the tilt-compensated eCompass algorithm to converts this measurements into Euler angles. After this process, the arduino sends the orientation, represented in Euler angles, to the client side. In addition, this system is also used in the balancing system. for this use, the microcontroller utilizes motor drivers, connected to moving masses, to displace the center of mass. More details about the real circuit are presented in Fig.4.4.

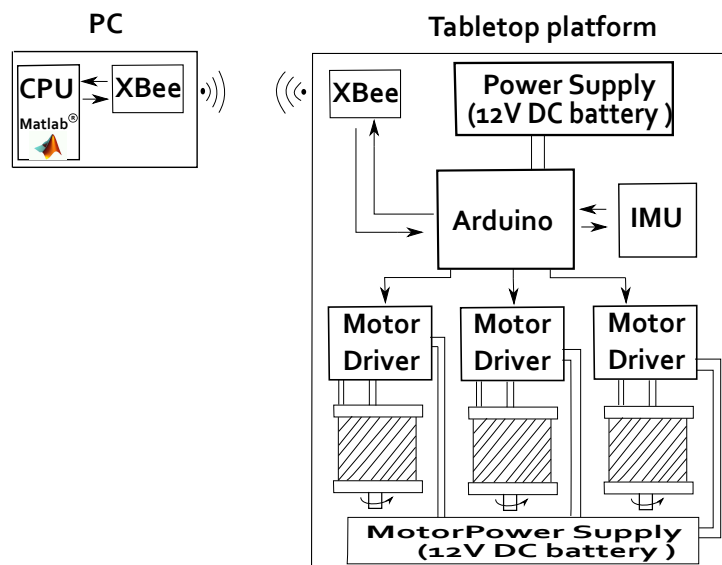
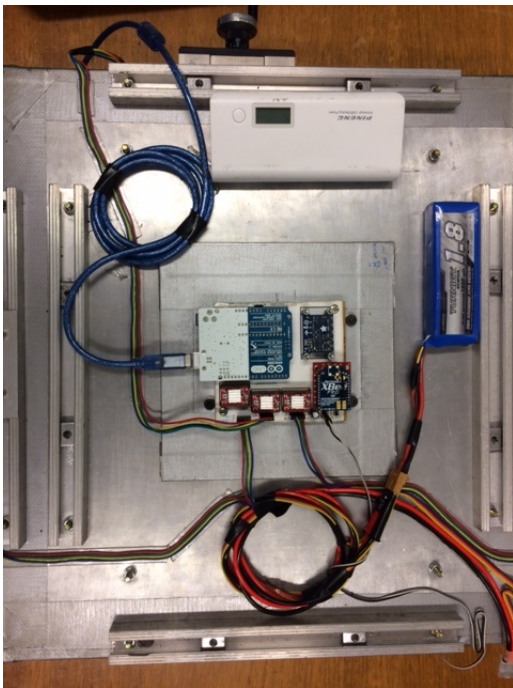
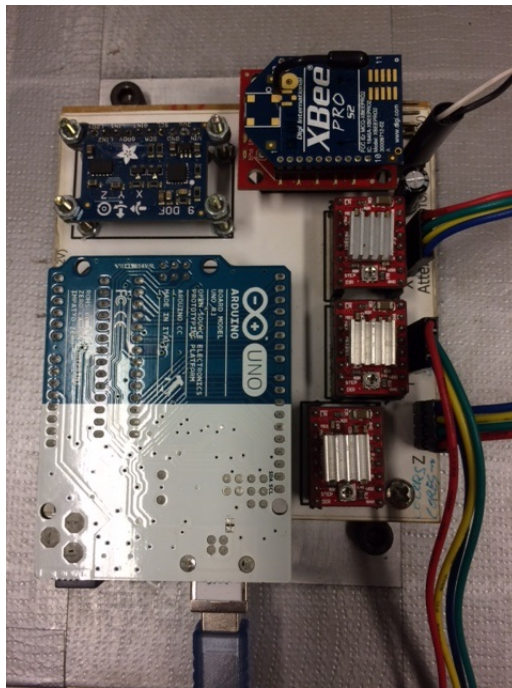


Figure 4.3: Adafruit’s IMU schematic



(A)



(B)

Figure 4.4: Hardware configuration; **A.** Platform inside view; **B.** Circuit upper view.

4.2.2 Hardware configuration for Yost Labs IMU

The schematic view of Yost Labs' IMU is presented in Fig.4.5. This picture shows the data flux inside the inertial measurement unity. First, raw measurements are collected from 3- axis magnetometer, gyro and compass. After that, this data is corrected with the initial calibration settings. The result of this operation is sent to a filter to generate the final orientation.

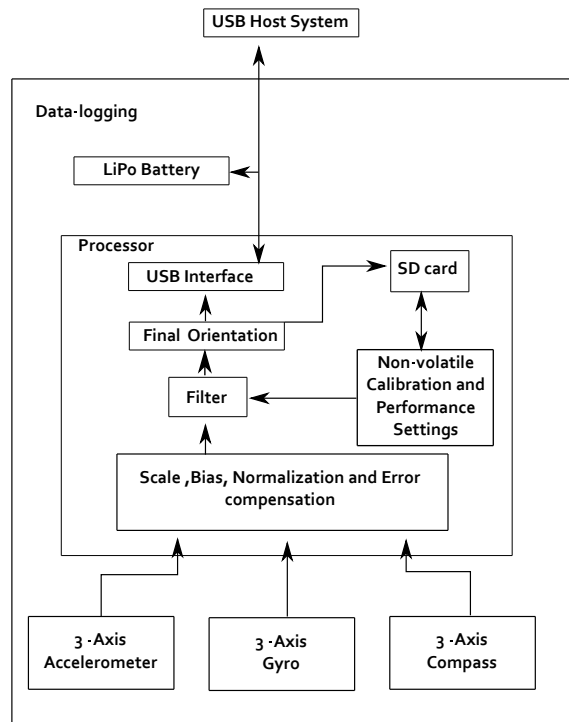


Figure 4.5: Yost Labs' IMU schematic

4.3 Calibration procedures

4.3.1 Calibration procedures for Adafruit IMU and HMR2300

The calibration process applied to the Adafruit IMU and HMR2300 consisted of three main parts. First, each sensor was rotated in all directions until the magnetometer measurements formed an almost complete ellipsoid (see Fig.4.6 and Fig.4.7). After, based on this data, the offset parameters and scale factors, presented in Table(4.3) and Table(4.4), are calculated using the Hard-iron and scale factor compensation presented in appendix F. Based on these parameters and on the laboratory magnetic field intensity, which is about $26\mu T$, the corrected magnetic field vector was obtained. About the laboratory magnetic field intensity, it was measured using another HMR2300 magnetometer calibrated by Honeywell. In addition, it was verified that this measurement was also coherent with the magnetic field intensity provided by the WMM (Wold Magnetic Model) from the National Oceanic and Atmospheric Administration (NOAA) [NOAA 2018].

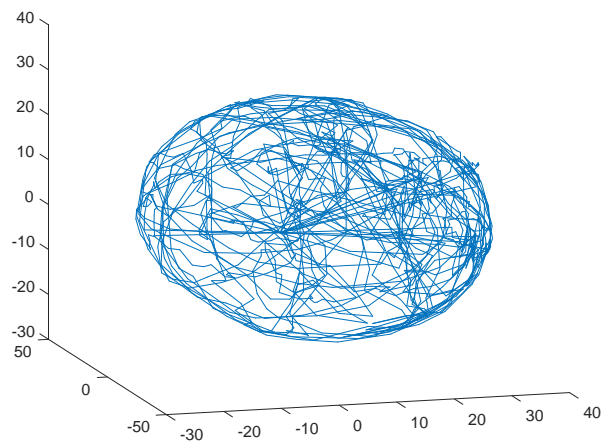


Figure 4.6: Magnetometer calibration of Adafruit's IMU

Table 4.3: Adafruit's IMU calibration parameters

| Parameters | Values |
|------------|------------------|
| $Offset_x$ | $3.477 \mu T$ |
| $Offset_y$ | $-11.8123 \mu T$ |
| $Offset_z$ | $4.9395 \mu T$ |
| $Scale_x$ | 29.9352 |
| $Scale_y$ | 29.3514 |
| $Scale_z$ | 27.4197 |

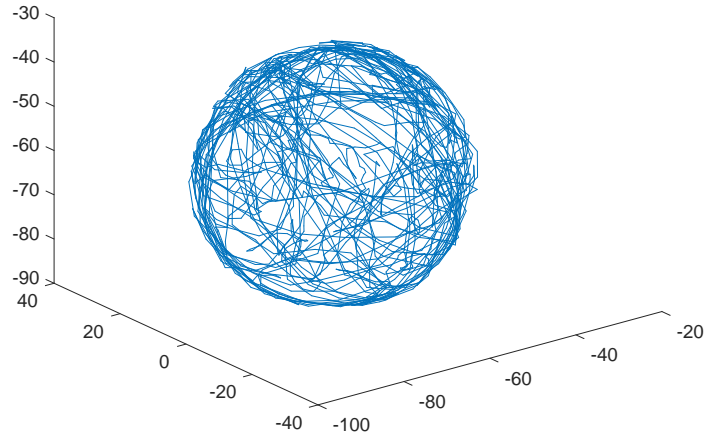


Figure 4.7: Magnetometer calibration of HMR2300

Table 4.4: HMR2300 calibration parameters

| Parameters | Values |
|------------|----------------|
| $Offset_x$ | $-65 \mu T$ |
| $Offset_y$ | $0.6589 \mu T$ |
| $Offset_z$ | $-61.61 \mu T$ |
| $Scale_x$ | 25.41 |
| $Scale_y$ | 26.73 |
| $Scale_z$ | 27.26 |

4.3.2 Calibration procedures for Yost Labs IMU

The calibration of the Yost Labs IMU consisted of running the sphere calibration wizard, which is a closed software, presented in Fig.4.8. This picture shows the result of magnetometer calibration which consisted of rotating the IMU in all directions until the system drawn a complete sphere of green arrows.

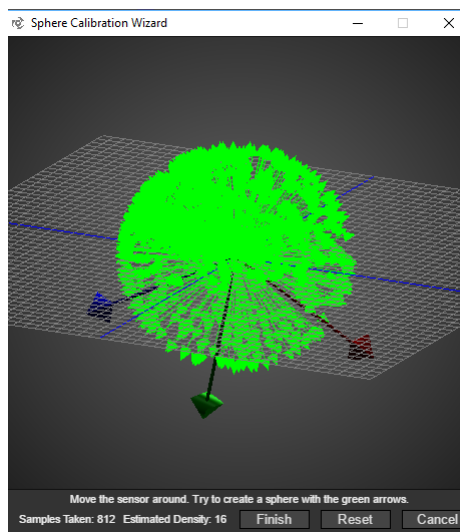


Figure 4.8: Sphere Calibration

Chapter 5

Tests and results

This section evaluates and compares the estimated orientation obtained using the EKF, UKF, USQUE and RiSAdUKF, which in this chapter will be referred as Riemannian filter. As it can be seen in Fig.5.1, the data used by these filters were obtained from two sensors. The Yost Labs sensor, in the left, provides the orientation in quaternions, while the Adafruit sensor, in the right, provides the orientation in Euler angles.

In addition, in order to emulate magnetic disturbances experienced by a nanosatellite, we induced random magnetic field variations using a Helmholtz cage. As will be shown in the next section, this magnetic field variations generate disturbances in the orientation data which were sent to the filters.

The orientation data represented in Euler angles were filtered using the UKF and EKF filters. While the orientation data represented in quaternions were filtered using the USQUE and Riemannian filters. In order to evaluate these estimated angles, three tests were proposed, as will be explained in the following sections.

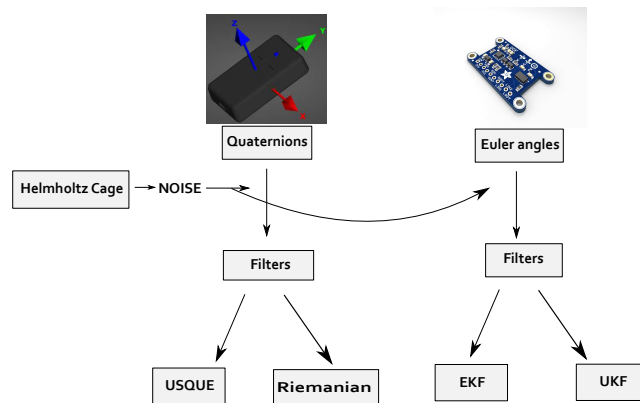


Figure 5.1: Methodology schematic view

5.1 Magnetic field variation effects on sensor data

The experiment presented in this section consisted in generating magnetic field variations using the Helmholtz cage while the sensor was kept on a fixed orientation. The initial objective was observing the effects of magnetic field variations and depending of this result, we could use magnetic field to generate magnetic field disturbances experienced by a microsatellite under real conditions.

As it can be seen in Fig.5.2 and Fig.5.3, initially the sensor provided a fixed initial orientation. After about twenty seconds, the magnetic field was constantly increased. Although the sensor was kept on a fixed orientation during the experiment, the orientation data indicated movements in x and y axes.

This wrong angle variation was observed until the Helmholtz cage was turned off, that happened at about 109 seconds. After that, the sensor started to measure it's initial orientation. This experiment evidenced that the sensors are directed affected by magnetic field variations and also showed that these variations could be used to emulate magnetic disturbances observed in space. Although these magnetic variations are not big during an orbit trajectory, it can be significant when the microsatellite has to use magnetorquers in the actuation system.

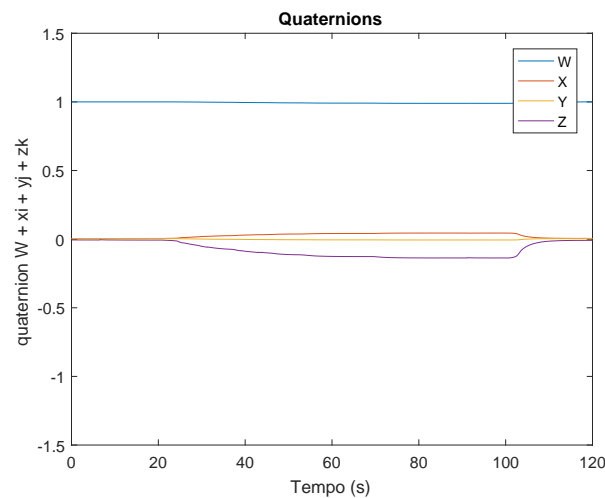


Figure 5.2: Magnetic disturbance in quaternions

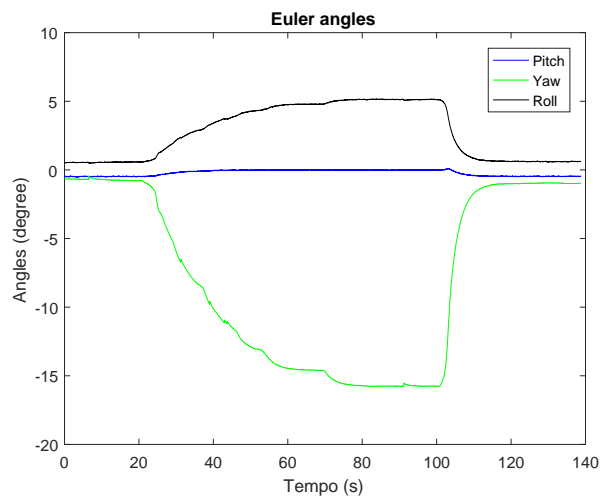


Figure 5.3: Magnetic disturbance in Euler angles

5.1.1 Experimental setup for filter evaluation

In order to evaluate each filter performance, three different experiments were conducted for each algorithm:

Experiment 1:

Initially the testbed is kept in the horizontal position. After that, the Helmholtz cage is used to generate noise in the orientation data. Based on the data collected, the filter consistency is evaluated using the normalized state estimation error squared (NEES), presented in appendix G and referred in this work as (d2) or Chi values. The objective of this experiment is finding the values of Q and R matrices that provides a consistent filter answer.

Experiment 2: For this experiment, Q and R must be assumed constant and equal to those found in Experiment 1. Initially, the testbed is kept in the horizontal position with a protractor fixed in the platform. One example of this configuration can be seen in see Fig.5.6(A). After that, using the same magnetic field described in Experiment 1, the testbed should be rotated by a known angle, each axis at a time. A schematic view of the experiment and a graphical view of the sensor and filter output are presented in Fig.5.4 and Fig.5.5. Observe that the angular variation generated by the movement will be measured using a protractor and a IMU. The variation measured by the protractor is given by the difference between the initial and final angle value measured by this instrument $\theta_{Protractor(t_f)} - \theta_{Protractor(t_0)} = \Delta\theta_{Protractor}$. In addition, the same angular variation must be measured by an IMU. This sensor data is passed through a filter to estimate the angular variation, which is given in terms of it's initial and final corrected state estimated values $\hat{\theta}_{k|k} - \hat{\theta}_{0|-1} = \Delta\hat{\theta}_{Filter}$.

The objective of this experiment is evaluating the angular variation obtained from each filter and comparing with the angular variation measured with the protractor for the follow rotations:

- A rotation of 20 degrees around the x axis (see Fig.5.6).
- A rotation of 30 degrees around the y axis (see Fig.5.7).
- A rotation of 8 degrees around the z axis (see Fig.5.9).

In order to do the last angular variation, around z axis, some setups must be done. In order to guaranty a variation of 8° around the z axis, an initial target setup presented in Fig.5.8 should be done. This procedure consisted of placing two targets in positions specified by a variation of 8° measured using a protractor in the center of the platform.

After placing the targets, a laser point should be used to drive the platform from one target to the other as shown in Fig.5.9. Therefore when the laser reach the second target, the platform rotated 8° around the z axis.

Experiment 3: The next experiment consists of keeping the same magnetic field interference and allowing the testbed to have free movements. The objective of this experiment is

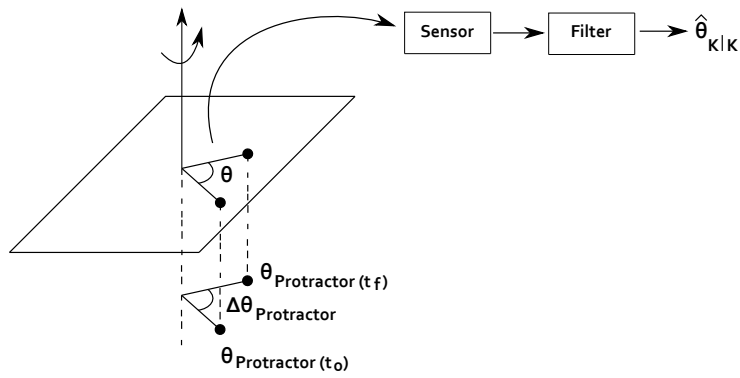


Figure 5.4: Experiment schematic view

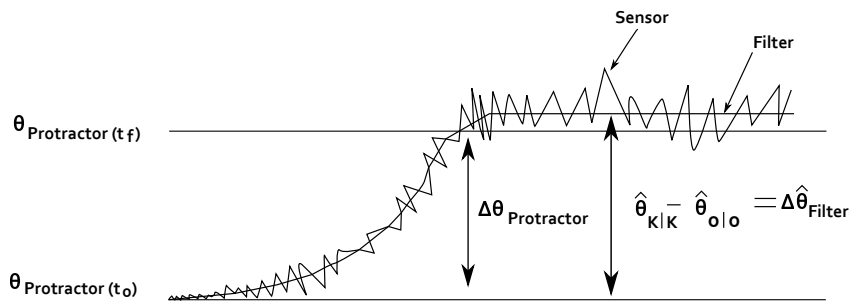
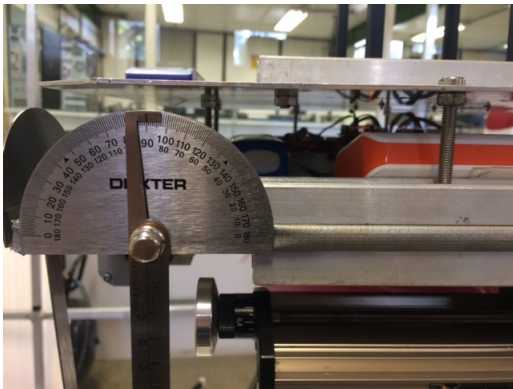


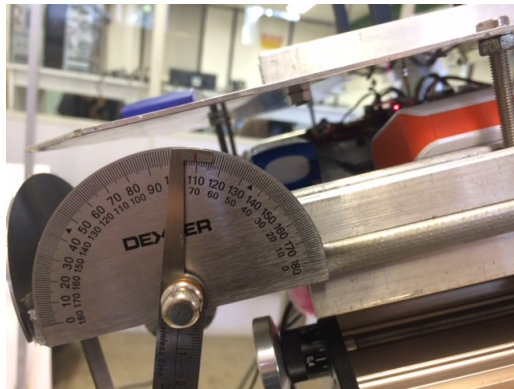
Figure 5.5: Result schematic view

evaluating each filter behavior under a more realistic situation.

In the next sections, these three experimental setups will be used to compare the UKF with the EKF and in other set of experiments it will be compared the USQUE and the Riemannian filter.



(A)

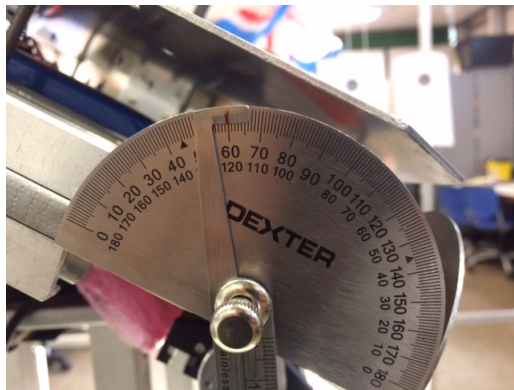


(B)

Figure 5.6: **A.** Initial position; **B.** Rotation of 20 degrees in roll.



(A)



(B)

Figure 5.7: **A.** Initial position; **B.** Rotation of 30 degrees in pitch.

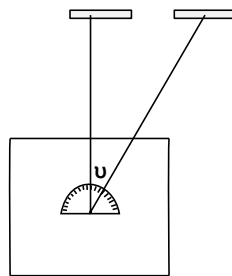
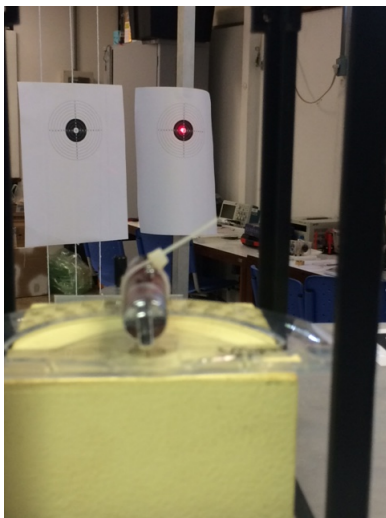
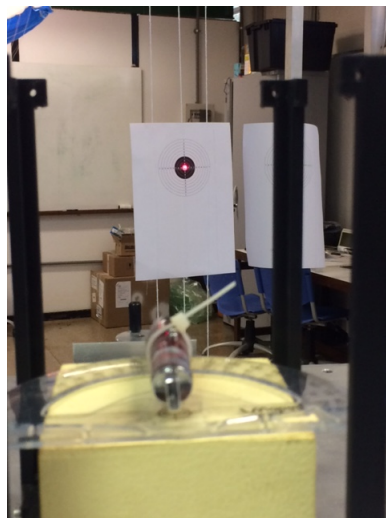


Figure 5.8: Initial target setup used to make rotations around the z axis



(A)



(B)

Figure 5.9: **A.** Initial position; **B.** Rotation of 8 degrees in yaw.

5.1.2 Comparison between the attitude estimated using the UKF and EKF

Experiment 1: Initially, the testbed was fixed in the horizontal position. In order to create experimental noise, we generated random magnetic pulses using two power supplies that provided a voltage output with a distribution $N(2.5,3)$. In addition, we also measured the generated magnetic field using the magnetometer (HMR2300) (see Fig.5.10).

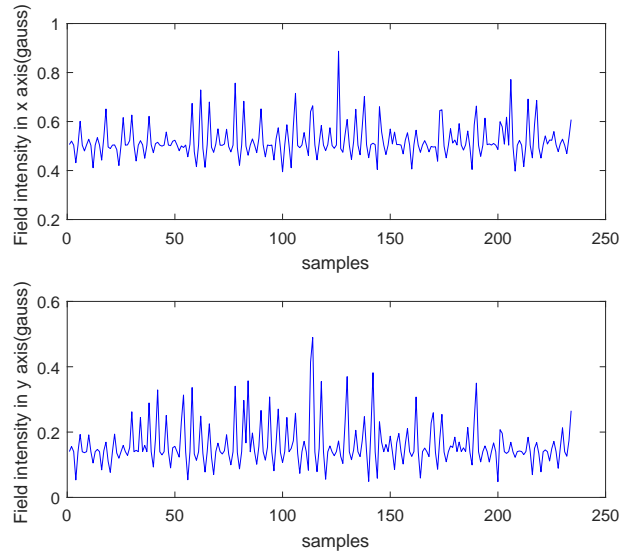


Figure 5.10: Magnetic field generated by the Helmholtz cage.

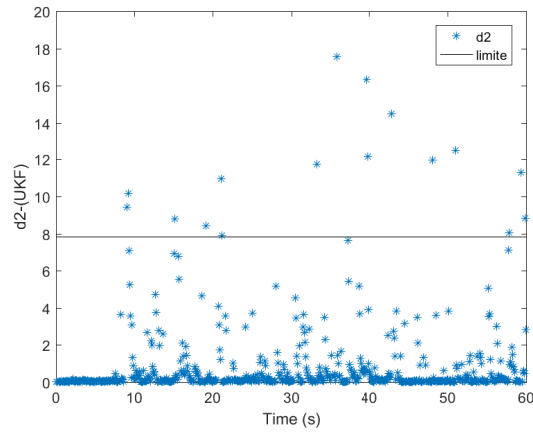
After collecting the orientation data under these circumstances, we tried different values of Q and R matrices in order to keep about 95% of the Chi values under the probabilistic limit. As it can be seen in Fig.5.11(A) and Fig.5.11(B) either Chi values are mostly spread and located under the three degree of freedom limit. To be more specific 96.7% of the UKF Chi values and 96.4% of the EKF Chi values are located below the limit.

This procedure lead us to the values of R and Q given by:

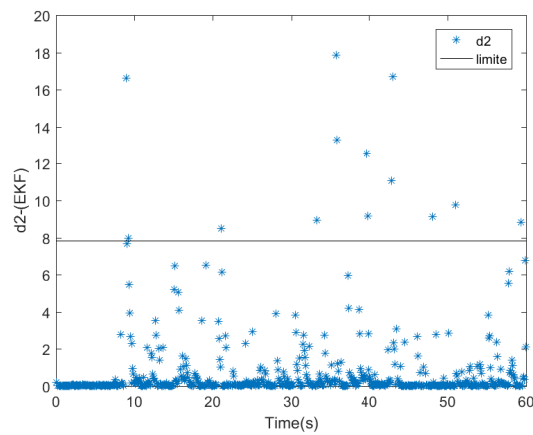
$$R_{(UKF)} = \begin{bmatrix} 8.91 & 0 & 0 \\ 0 & 8.01 & 0 \\ 0 & 0 & 9 \end{bmatrix} \quad R_{(EKF)} = \begin{bmatrix} 8.73 & 0 & 0 \\ 0 & 8.01 & 0 \\ 0 & 0 & 8.1 \end{bmatrix}. \quad (5.1)$$

and

$$Q_{(UKF)} = \begin{bmatrix} 0.23 & 0 & 0 \\ 0 & 0.37 & 0 \\ 0 & 0 & 0.27 \end{bmatrix} \quad Q_{(EKF)} = \begin{bmatrix} 0.225 & 0 & 0 \\ 0 & 0.3650 & 0 \\ 0 & 0 & 0.2646 \end{bmatrix}. \quad (5.2)$$



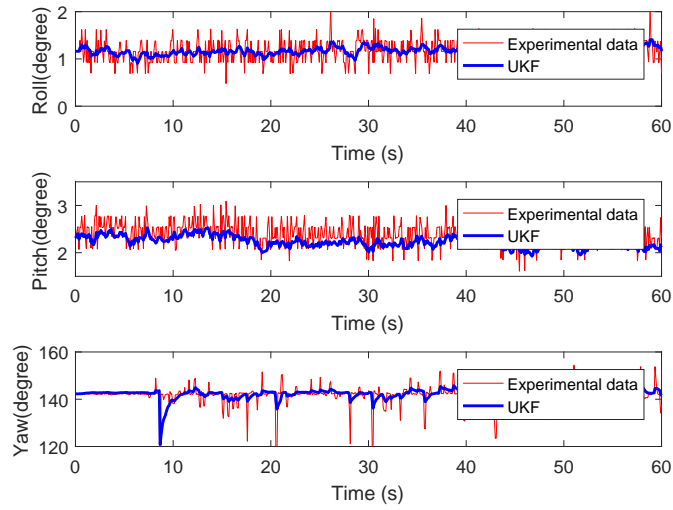
(a) UKF chi square values



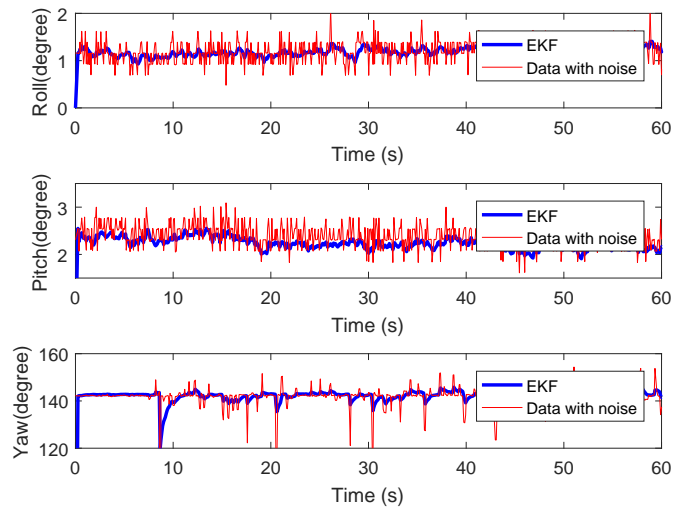
(b) EKF chi square values

Figure 5.11: Chi square test

After determining the values of Q and R , we compared the orientation estimated by each filter for the first experiment. As it can be seen in Fig.5.12, there were some spikes in the sensor signal. Some of these spikes were caused by problems in the magnetometer, which sometimes sends wrong measurement values equal or close to zero for all axes resulting in wrong values of $\psi = 0$, as happened in $T=8.67s$. In addition, besides the noise coming from the other sensors or even by external magnetic interference, this signal was also affected by the magnetic noise generated by the Helmholtz cage. Furthermore, this picture also shows that for these experimental conditions, both filters presented a similar and smoother answer than the original sensor signal for the angles of roll, pitch and yaw.



(a) UKF output for Experiment 1.



(b) EKF output for Experiment 1.

Figure 5.12: Filters output

Experiment 2:

As it was explained in section 5.1.1, the experiment 2 started with a rotation around the x axis. Initially, the platform was kept on a horizontal position. In this orientation, the protractor measured $\phi_{Protractor(t_0)} = 90^0$ and after the rotation the protractor measured $\phi_{Protractor(t_f)} = 110^0$, so the angular variation measured was $\Delta\phi_{Protractor} = 20^0$, which is the angular variation that we expect observing using the sensor.

The sensor data and the filters' outputs are presented in Fig.5.13. In this picture, the UKF estimated value started in about $\hat{\phi}_{0|-1} = 1.333^0$, which was the first sensor measurement, and after the movement, the estimated value reached $\hat{\phi}_{57|57} = 22.62^0$, resulting in an angular variation of $\Delta\phi_{UKF} = 21.28^0$ (see Fig.5.13(a)). In addition, this picture also shows that the EKF estimated value started in about $\hat{\phi}_{0|-1} = 1.333^0$ and reached $\hat{\phi}_{57|57} = 22.65^0$, resulting in an angular variation of $\Delta\phi_{EKF} = 21.31^0$ (see Fig.5.13(b)). Observe that both filters provided a much smooth answer and the variation measured by the filters was more precise than the variation given by the sensor. Although there was a difference between the measured filters variation and the protractor variation $\Delta\phi_{UKF} - \Delta\phi_{Protractor} = 1.28^0$ and $\Delta\phi_{EKF} - \Delta\phi_{Protractor} = 1.31^0$, part of this difference was probably caused by small errors coming from the sensor and protractor.

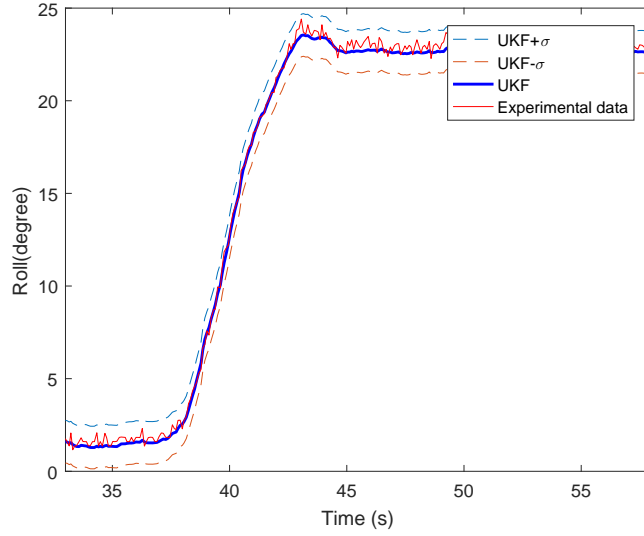
Furthermore, the filter performance was also evaluated using the $\pm\sigma$ bound as shown in [Mirzaei and Roumeliotis 2008] and [Guimarães et al. 2017], where the values of σ were obtained from the diagonal terms of the state covariance matrix $P_{k|k}$. As it can be seen in Fig.5.13, both filters were capable of keeping their estimated values under the $\pm\sigma$ bound which indicates that both filters were tuned with appropriate values of Q and R.

The second rotation was done around the y axis. First, the platform started in the horizontal position. In this initial orientation, the protractor measured an initial value $\theta_{Protractor(t_0)} = 90^0$. After the rotation, the protractor measurement reached $\theta_{Protractor(t_f)} = 60^0$. So, the angular variation measured by the protractor was $\Delta\theta_{Protractor} = 30^0$, which is the angular variation that we expect observing using the sensor.

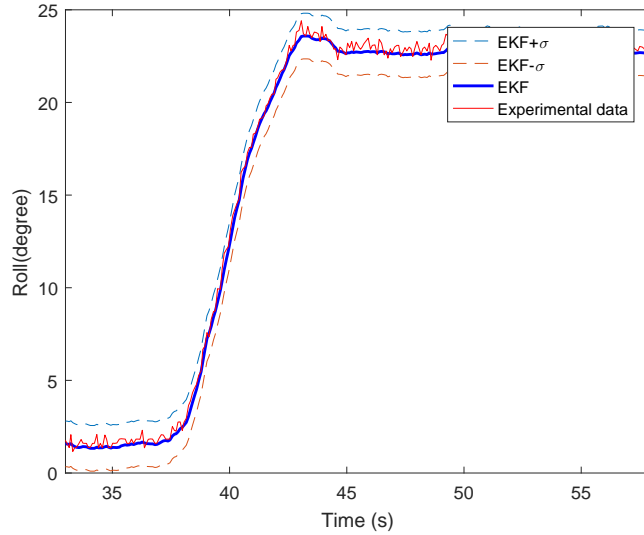
As it can be seen in Fig.5.14, the UKF estimated value started in about $\hat{\theta}_{0|-1} = 3.599^0$, which was the first sensor measurement, and after the movement, the estimated value reached $\hat{\theta}_{22|22} = 34.47^0$, resulting in an angular variation of $\Delta\theta_{UKF} = 30.87^0$ (see Fig.5.13(a)). In addition, this picture also shows that the EKF estimated value started in about $\hat{\theta}_{0|-1} = 3.599^0$ and reached $\hat{\theta}_{57|57} = 34.44^0$, resulting in an angular variation of $\Delta\theta_{EKF} = 30.84^0$ (see Fig.5.13(b)). Although there was a difference between the measured filters variation and the protractor variation $\Delta\theta_{UKF} - \Delta\theta_{Protractor} = 0.87^0$ and $\Delta\theta_{EKF} - \Delta\theta_{Protractor} = 0.84^0$, part of this difference was probably caused by small errors coming from the sensor and protractor.

In addition, these pictures also show that both filters had a similar behavior and also their estimated values were always kept inside the $\pm\sigma$ bound, which indicates that both were well conditioned.

After that, the third rotation was done around the z axis. The experiment started with the



(a) UKF output.

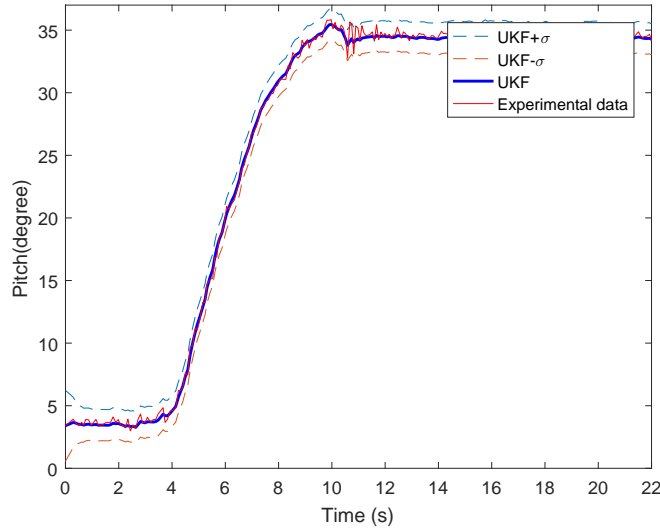


(b) EKF output.

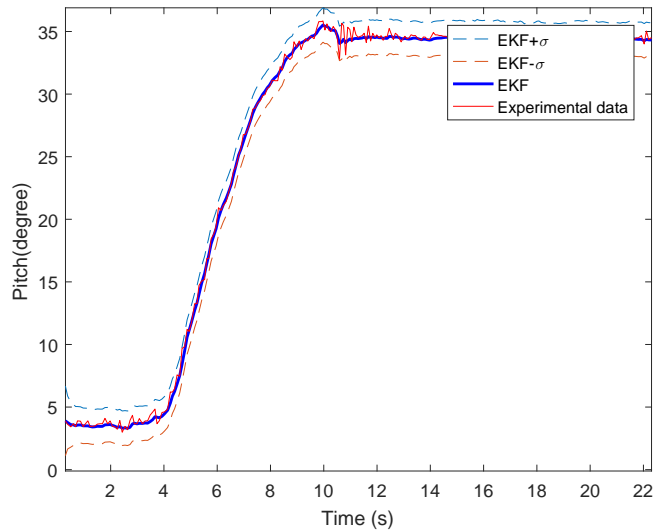
Figure 5.13: Rotation of 20 degrees around the x axis

platform in the horizontal position with a laser pointing to the first target. In this orientation the protractor started measuring $\psi_{Protractor(t_0)} = 120^0$. After that, the laser pointing was used to guide the platform from one target to the other. In this new configuration the protractor measured $\psi_{Protractor(t_f)} = 112^0$, resulting in an angular variation of $\Delta\psi_{Protractor} = 8^0$, which is the angular variation that we expect observing using the sensor.

The data collected from the sensor and the filters output are presented in Fig.5.15. As it can be seen, the UKF estimated value started in about $\hat{\psi}_{0|1} = 7.79^0$, which was the first sensor measurement, and after the movement, the estimated value reached $\hat{\psi}_{36|36} = 1.182^0$, resulting in an angular variation of $\Delta\psi_{UKF} = 6.6^0$ (see Fig.5.13(a)). Furthermore, this picture also shows that the EKF estimated value started in about $\hat{\psi}_{0|1} = 7.79^0$ and reached $\hat{\psi}_{36|36} = 1.078^0$, resulting in an angular variation of $\Delta\psi_{EKF} = 6.71^0$ (see Fig.5.13(b)).



(a) UKF output.

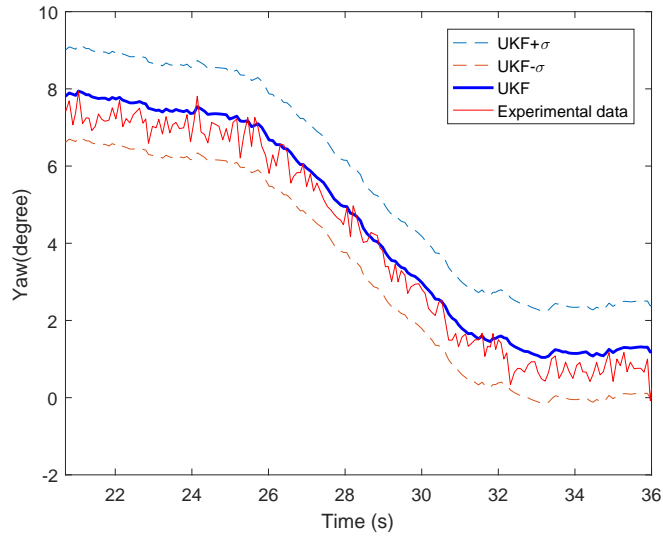


(b) EKF output.

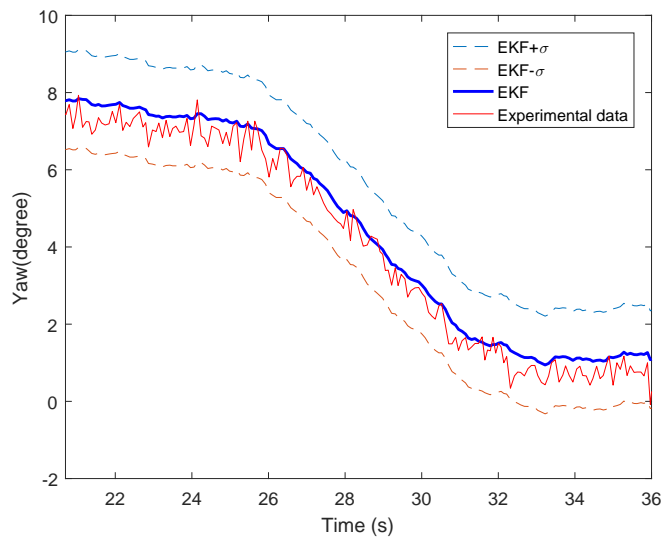
Figure 5.14: Rotation of 30 degrees around the y axis

Although there was a difference between the filters' variation and the protractor variation $\Delta\psi_{UKF} - \Delta\psi_{Protractor} = 1.4^{\circ}$ and $\Delta\psi_{EKF} - \Delta\psi_{Protractor} = 1.29^{\circ}$, part of this difference was probably caused by small errors coming from the sensor and protractor.

Furthermore, these pictures also show that both filters had a similar behavior and also their estimated values were always kept inside the $\pm\sigma$ bound, which indicates that both were well conditioned.



(a) UKF output.



(b) EKF output.

Figure 5.15: Rotation of 8 degrees around the z axis

After, it was plotted the estimated measurement residuals for each rotation (see Fig.5.16, Fig.5.17 and Fig.5.18). As it can be seen in these pictures, both filters presented similar values of mean and standard deviation also presented in Table 5.1. From this table, we see that although the estimated measurement residuals mean, for both filters, is close to zero, it is not zero, which indicates that both estimators present a small tendency which could be caused by the simplicity of the adopted model since both filters present almost the same mean.

Table 5.1: Mean and standard deviation for the estimated measurement residuals of the UKF and EKF

| Rotation | Filter | Parameter | value |
|--|--------|-----------|----------------|
| $(\phi - \hat{\phi})$ for a rotation of 20° around the X axis | UKF | mean | 0.276° |
| | | σ | 0.3090° |
| | EKF | mean | 0.246° |
| | | σ | 0.3086° |
| $(\theta - \hat{\theta})$ for a rotation of 30° around the Y axis | UKF | mean | 0.219° |
| | | σ | 0.409° |
| | EKF | mean | 0.204° |
| | | σ | 0.429° |
| $(\psi - \hat{\psi})$ for a rotation of 8° around the Z axis | UKF | mean | -0.551° |
| | | σ | 0.334° |
| | EKF | mean | -0.471° |
| | | σ | 0.324° |

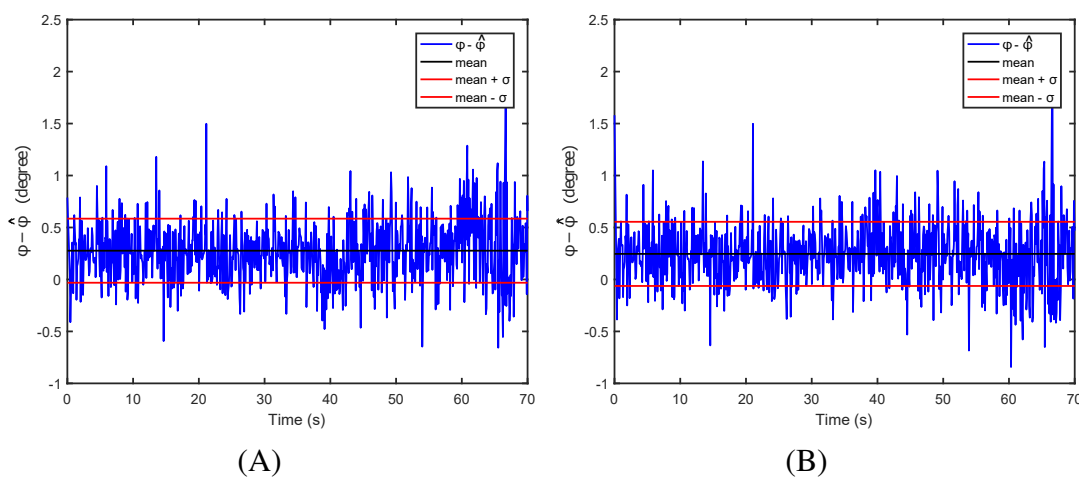
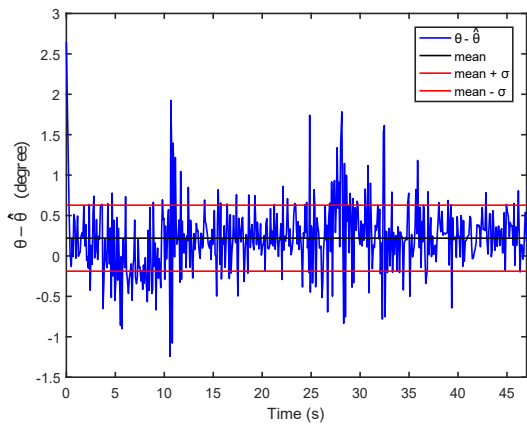
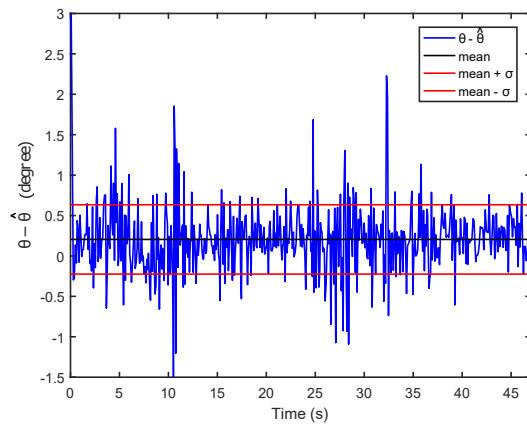


Figure 5.16: **A.** UKF $(\phi - \hat{\phi})$ for a rotation of 20° around X axis.; **B.** EKF $(\phi - \hat{\phi})$ for a rotation of 20° around X axis.

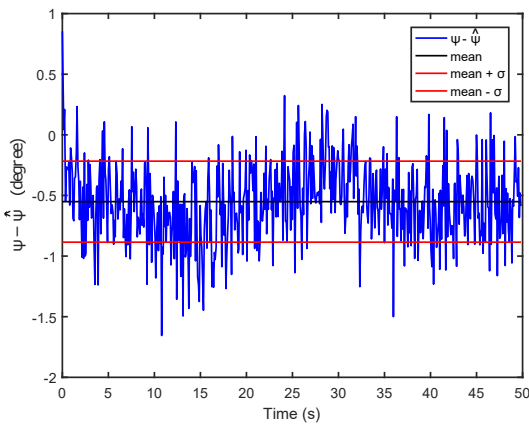


(A)

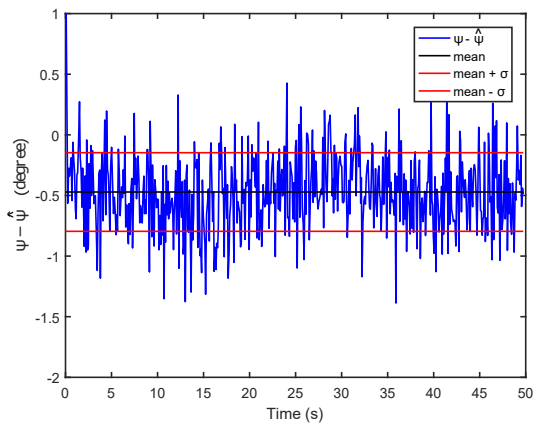


(B)

Figure 5.17: **A.** UKF ($\theta - \hat{\theta}$) for a rotation of 30° around Y axis.; **B.** EKF ($\theta - \hat{\theta}$) for a rotation of 30° around Y axis.



(A)



(B)

Figure 5.18: **A.** UKF ($\psi - \hat{\psi}$) for a rotation of 8° around Z axis.; **B.** EKF ($\psi - \hat{\psi}$) for a rotation of 8° around Z axis.

Experiment 3:

After conducting these controllable tests, we allowed the platform to have free movements. The results of this experiment are shown in Fig.5.19.

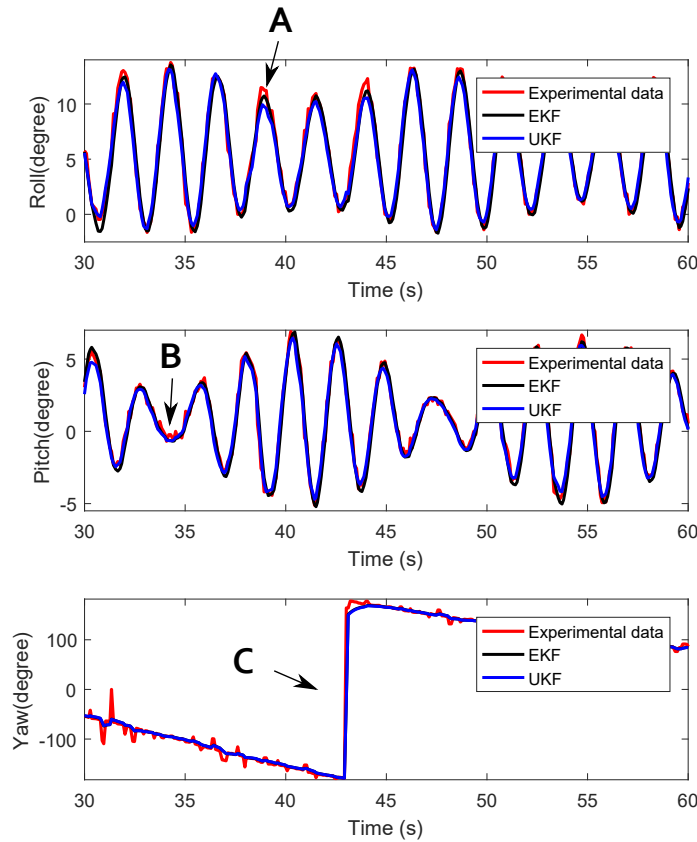


Figure 5.19: Experimental data and UKF output for the platform moving freely

In this picture, some important observations are evidenced by arrows A, B, C. First, from arrow A, we observed that the estimated orientations presented a slight difference but still reasonable when peak regions are affected by noise of high amplitude. But, when the signal is affected by noise of low amplitude, both filters are capable of providing a smoother and similar answer as shown by arrow B. In addition, arrow C shows that both filters provided a faster and similar answer when the orientation provided by the sensor changed from -180° to 180° . About this last region, it is important calculating the smallest value of $\psi - \hat{\psi}$ because the orientation data provided by the sensor has a discontinuity, it jumps from -180° to 180° . At this point, the reader should notice that this discontinuity region is caused by trigonometric functions whose results are in the $] -180^\circ, 180^\circ]$ interval. It is not verified in the experimental platform. Furthermore, we also observed the time processing of these two algorithms. The UKF spent about 0.18s to process input data vectors of size 651×1 while the EKF spent about 0.2296s to process the same amount of data.

5.1.3 Covariance matrices equivalents for the tangent space

This subsection will present a procedure to obtain approximate values of Q and R matrices in the tangent space, used in the Riemannian filter. This method facilitates the choice of Q and R values for this filter because it provides a way to transfer the notion that we have of degrees angles into the tangent space.

This procedure consists of representing a quaternion from the sphere in S^2 into the tangent space, as it can be seen in Fig.5.20. This picture shows that there is a function that takes g and maps it into the tangent space as g^{TS} based on a reference quaternion q .

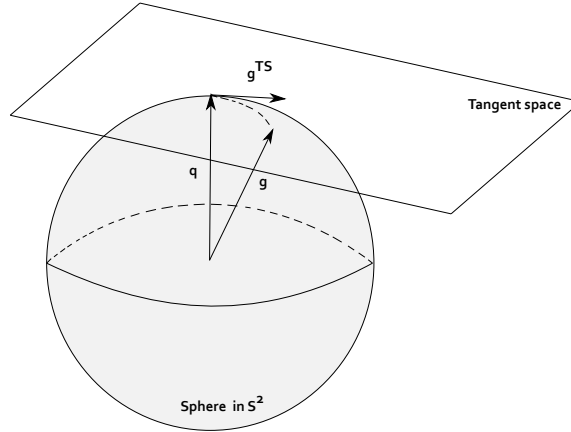


Figure 5.20: Tangent space

This function, obtained from [Menegaz 2016], is given by:

$$g^{TS} = \ln_q(g) \triangleq \frac{\theta}{\sin(\theta)}g - \frac{\theta \cos(\theta)}{\sin(\theta)}q. \quad (5.3)$$

where

$$\theta \triangleq \arccos(\langle q, g \rangle). \quad (5.4)$$

The procedure to find the approximate values of Q and R for tangent space is divided in five different steps as it can be seen in Fig.5.21. The first step consisting of building a 3×3 diagonal matrix, where each diagonal term refers to a state variance. After that it is created a vector composed by the square root of the diagonal terms. The third step consisting of changing the vector representation from Euler angles to quaternion. After that, this quaternion is represented in the tangent space through the log function presented in Eq. (5.3) using the base $[1 \ 0 \ 0 \ 0]^T$. And finally, the generated vector is represented in the tangent basis using the transformation matrix $T_{L|C}^{-1}$ presented in appendix I. This last operation will provide g_L^{TS} represented in the tangent space. Based on this vector, the R_L represented in the tangent space will be given by a diagonal matrix formed by the square of each term of g_L^{TS} as can be seen in the step 5 of Fig.5.21.

As it can be seen in the Fig.5.21, the methodology was represented using the matrix R, but the same process can also be done to the matrix Q.

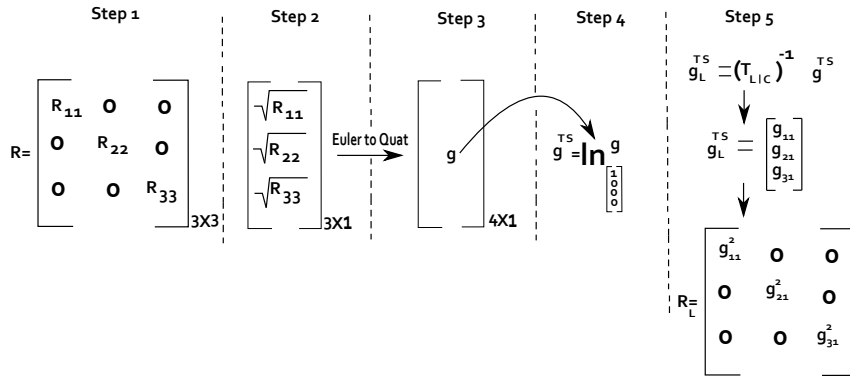


Figure 5.21: Step by Step

5.1.4 Attitude estimation using the USQUE and Riemannian filter

This section will present the platform attitude estimated using the USQUE and Riemannian filters. These filters received quaternions as input and provided the output in quaternions. But, in order to facilitate the comprehension, we changed the output representation from quaternions to Euler angles.

Experiment 1:

Initially, following the experiment 1 setup described in section 5.1.1, we analyzed the filters' performance using the consistence method presented in appendix G obtained from [Bar-Shalom et al. 2004]. Using this method, we found the values of Q and R to keep 95% of the Chi values under the probabilistic limit (see Fig.5.22(A)). But after testing the filter performance for a rotation of 30° around the y axis, we observed, for example, that the USQUE could not follow the system dynamic (see Fig.5.22(B)).

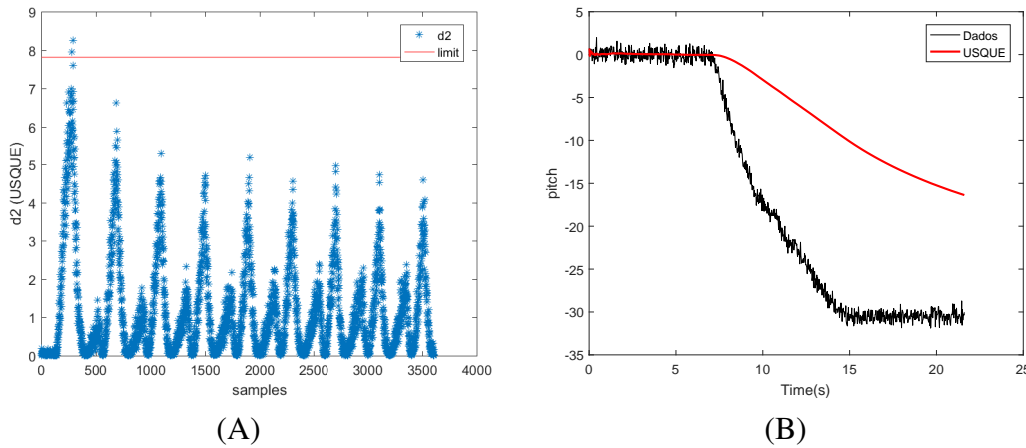


Figure 5.22: **A.** Chi square test for USQUE.; **B.** USQUE output for a rotation of 30° around y axis.

Since the filters could not follow the sensor signal dynamic, we had to choose different values of Q and R in order to improve each filter performance.

- Q and R matrices for the Riemannian filter: Initially, R was considered to be a diag-

onal matrix composed by the squared variance of each sensor axis, which is almost 1 degree for all axes, obtained from the sensor datasheet. After that, we used the covariance matrices equivalents for the tangent space described in section 5.1.3 to map the matrices Q and R in the tangent space. The best obtained values of Q and R expressed in the tangent space are given by:

$$R_{(Riemannian)} = \begin{bmatrix} 5.4e^{-6} & 0 & 0 \\ 0 & 5.5e^{-6} & 0 \\ 0 & 0 & 5.4e^{-6} \end{bmatrix} \quad (5.5)$$

$$Q_{(Riemannian)} = \begin{bmatrix} 1.08e^{-7} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.09e^{-7} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.08e^{-7} & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.24e^{-21} & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.24e^{-21} & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.24e^{-21} \end{bmatrix}. \quad (5.6)$$

Observe in Fig.5.23, that for these matrices, the Riemannian filter could effectively follow the system dynamic and also reduce the sensor noise.

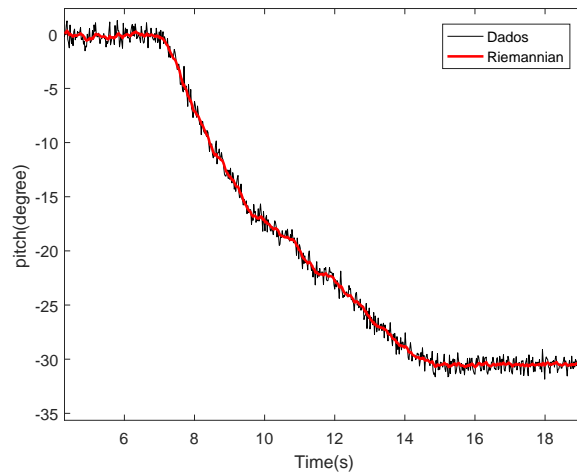


Figure 5.23: Riemannian output for a rotation of 30^0 around y axis.

- Q and R matrices for the USQUE: Initially, we used the Q matrix format presented in [Crassidis and Markley 2003].

$$\bar{Q}_k = \frac{\Delta t}{2} \begin{bmatrix} (\sigma_v^2 - \frac{1}{6}\sigma_u^2\Delta t^2)I_3 & 0_{3 \times 3} \\ 0_{3 \times 3} & \sigma_u^2 I_3 \end{bmatrix}, \quad (5.7)$$

where σ_v was tested considering values between $(0.31623 \times 10^{-5}$ and $0.31623 \times 10^{-7})$, while σ_u was tested considering values between $(3.1623 \times 10^{-3}$ and $3.1623 \times 10^{-10})$. Considering this, after testing different values of Q and R, the USQUE presented the best answer using the following matrices:

$$R_{(USQUE)} = \begin{bmatrix} 8.7e^{-13} & 0 & 0 \\ 0 & 6.9e^{-13} & 0 \\ 0 & 0 & 1.7e^{-13} \end{bmatrix}. \quad (5.8)$$

$$Q_{(USQUE)} = \begin{bmatrix} 1.1e^{-13} & 0 & 0 & -1.2e^{-21} & 0 & 0 \\ 0 & 1.1e^{-13} & 0 & 0 & -1.2e^{-21} & 0 \\ 0 & 0 & 1.1e^{-13} & 0 & 0 & -1.2e^{-21} \\ -1.2e^{-21} & 0 & 0 & 1.1e^{-19} & 0 & 0 \\ 0 & -1.2e^{-21} & 0 & 0 & 1.1e^{-19} & 0 \\ 0 & 0 & -1.2e^{-21} & 0 & 0 & 1.1e^{-19} \end{bmatrix}. \quad (5.9)$$

Observe in Fig.5.24, that for these matrices, the USQUE could effectively follow the system dynamic and also reducing the sensor noise.

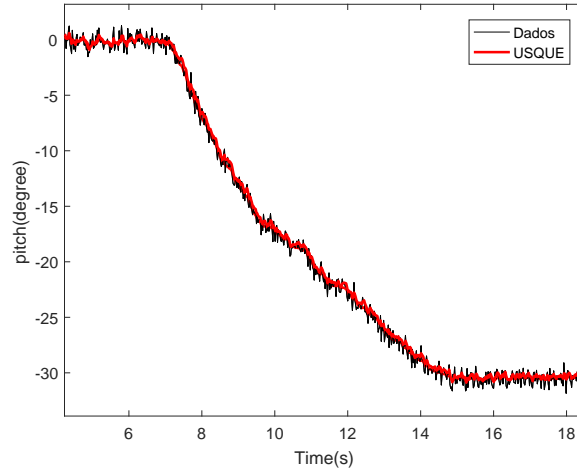


Figure 5.24: USQUE output for a rotation of 30^0 around y axis.

Experiment 2:

Initially, as we did for the UKF and EKF, the experiment 2 started with a rotation around the x axis. First, the platform was kept on a horizontal position. In this orientation, the protractor measured $\phi_{Protractor}(t_0) = 90^0$ and after the rotation, the protractor measured $\phi_{Protractor}(t_f) = 110^0$, so the angular variation measured was $\Delta\phi_{Protractor} = 20^0$, which is the angular variation that we expect observing using the sensor.

The sensor data and the filters output are presented in Fig5.25. As it can be seen, the USQUE estimated value started in about $\hat{\phi}_{0|-1} = 0.091^0$, which was the first sensor measurement, and after the movement, the estimated value reached $\hat{\phi}_{11|11} = 20.37^0$, resulting in an angular variation of $\Delta\phi_{USQUE} = 20.279^0$. While the Riemannian filter estimated value started in about $\hat{\phi}_{0|-1} = 0.091^0$ and reached $\hat{\phi}_{11|11} = 20.48^0$, resulting in an angular variation of $\Delta\phi_{Riemanian} = 20.38^0$. Although there was a difference between the filters' variation and the protractor variation $\Delta\phi_{USQUE} - \Delta\phi_{Protractor} = 0.279^0$ and $\Delta\phi_{Riemanian} - \Delta\phi_{Protractor} = 0.38^0$, part of this difference was probably caused by small errors coming from the sensor and protractor. In addition, this picture also shows the Riemannian filter provides a smoother answer than the USQUE.

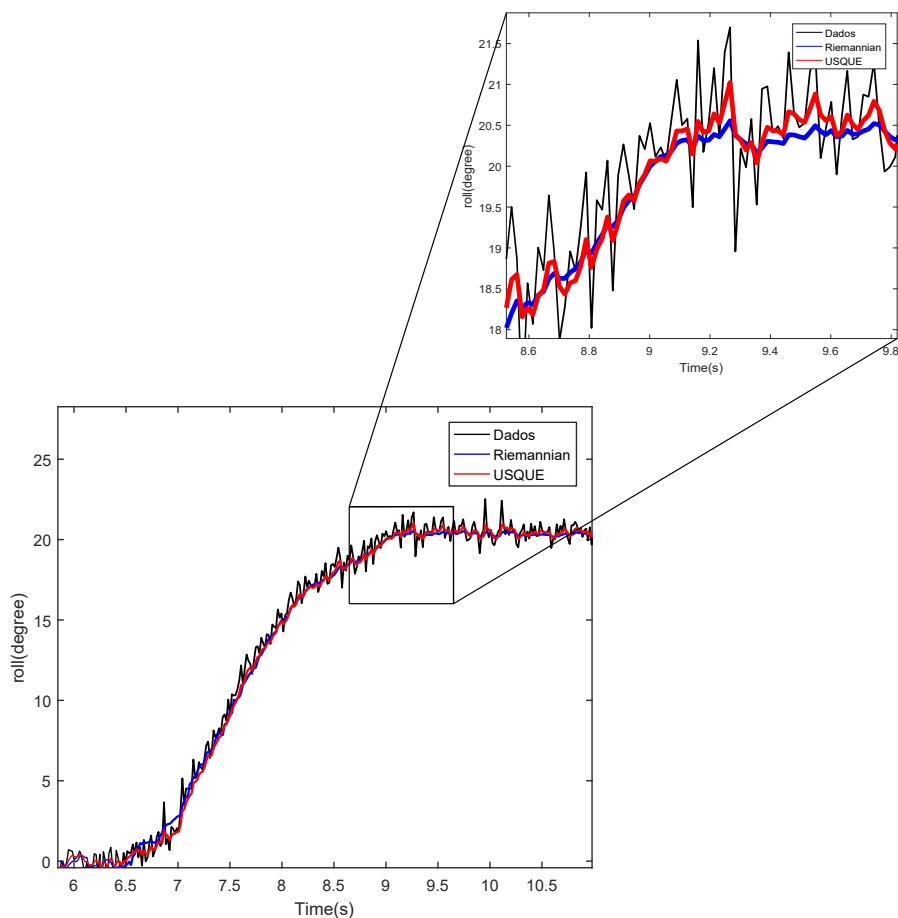


Figure 5.25: Rotation of 20^0 around x axis.

The second rotation was around the y axis. Initially, the platform was placed in a horizontal position. Using the protractor we measured an initial angle value $\theta_{Protractor(t_0)} = 90^0$. After the rotation, the protractor measurement reached $\theta_{Protractor(t_f)} = 60^0$. So, the angular variation measured by the protractor was $\Delta\theta_{Protractor} = 30^0$, which is the angular variation that we expect observing using the sensor.

The results of this experiment are presented in Fig.5.26. This picture shows that the USQUE estimated value started in about $\hat{\theta}_{0|-1} = 0.71^0$, which was the first sensor measurement, and after the movement, the estimated value reached $\hat{\theta}_{17|17} = -30.28^0$, result-

ing in an angular variation of $\Delta\theta_{USQUE} = 30.99^0$. While the Riemannian filter estimated value started in about $\hat{\theta}_{0|-1} = 0.71^0$ and reached $\hat{\theta}_{17|17} = -30.43^0$, resulting in an angular variation of $\Delta\theta_{Riemannian} = 31.14^0$. Although there was a difference between the filters' variation and the protractor variation $\Delta\theta_{USQUE} - \Delta\theta_{Protractor} = 0.99^0$ and $\Delta\theta_{Riemannian} - \Delta\theta_{Protractor} = 1.14^0$, part of this difference was probably caused by small errors coming from the sensor and protractor. In addition, in the zoom, this picture shows the Riemannian filter provides a smoother answer than the USQUE.

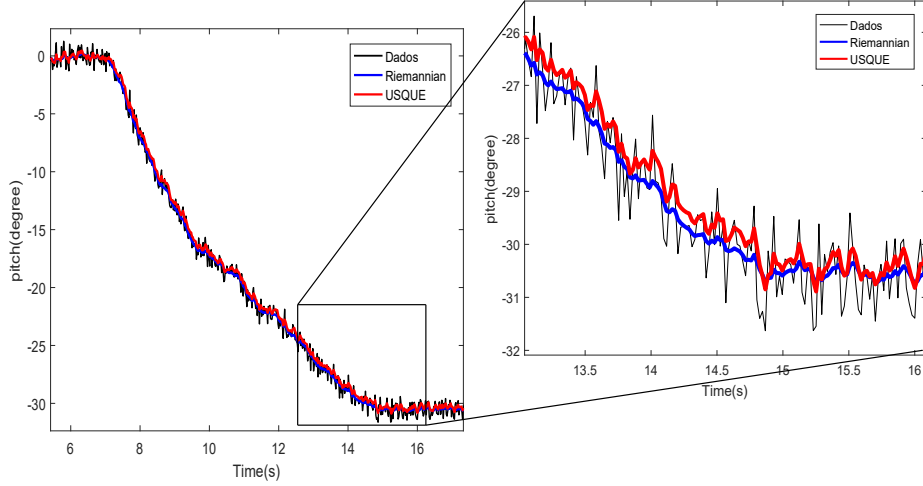


Figure 5.26: Rotation of 30^0 around y axis.

After that, the third rotation was done around the z axis. For this experiment, the testbed was initially kept in a horizontal position with the laser pointing to the first target. In this orientation, the protractor started measuring $\psi_{Protractor}(t_0) = 120^0$. After that, the laser pointing was used to guide the platform from one target to the other. In this new configuration the protractor measured $\psi_{Protractor}(t_f) = 112^0$, resulting in an angular variation of $\Delta\psi_{Protractor} = 8^0$, which is the angular variation that we expect observing using the sensor.

As it can be seen in Fig.5.27, the USQUE estimated value started in about $\hat{\psi}_{0|-1} = -0.3^0$, which was the first sensor measurement, and after the movement, the estimated value reached $\hat{\psi}_{23|23} = 7.83^0$, resulting in an angular variation of $\Delta\psi_{USQUE} = 8.13^0$. While the Riemannian filter estimated value started in about $\hat{\psi}_{0|-1} = -0.3^0$ and reached $\hat{\psi}_{23|23} = 7.95^0$, resulting in an angular variation of $\Delta\psi_{Riemannian} = 8.25^0$. Although there was a difference between the filters' variation and the protractor variation $\Delta\psi_{USQUE} - \Delta\psi_{Protractor} = 0.13^0$ and $\Delta\psi_{Riemannian} - \Delta\psi_{Protractor} = 0.25^0$, part of this difference was probably caused by small errors coming from the sensor and protractor.

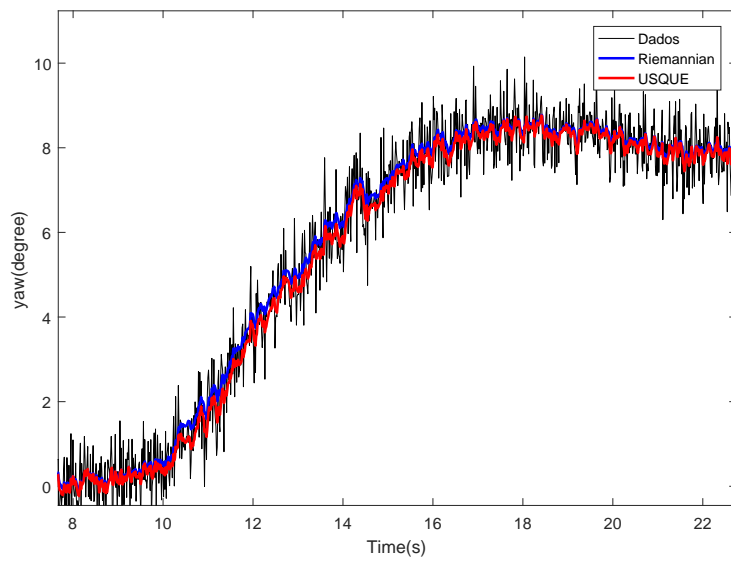


Figure 5.27: Rotation of 8^0 around Z axis.

After, it was plotted the estimated measurement residuals for each rotation (see Fig.5.28, Fig.5.29 and Fig.5.30). As it can be seen in these pictures, both filters presented similar values of mean and standard deviation also presented in Table 5.2. From this table, we see that although the estimated measurement residuals mean, for both filters, is close to zero, it is not zero, which indicates that both estimators have a small tendency.

Table 5.2: Mean and standard deviation for the innovation of the UKF and EKF

| Rotation | Filter | Parameter | value |
|--|----------|-----------|----------------|
| $(\phi - \hat{\phi})$ for a rotation of 20° around the X axis | RiSAdUKF | mean | 0.0017° |
| | | σ | 0.5293° |
| | USQUE | mean | 0.01° |
| | | σ | 0.4980° |
| $(\theta - \hat{\theta})$ for a rotation of 30° around the Y axis | RiSAdUKF | mean | -0.01° |
| | | σ | 0.51° |
| | USQUE | mean | -0.188° |
| | | σ | 0.454° |
| $(\psi - \hat{\psi})$ for a rotation of 8° around the Z axis | RiSAdUKF | mean | -0.034° |
| | | σ | 0.5310° |
| | USQUE | mean | 0.045° |
| | | σ | 0.503° |

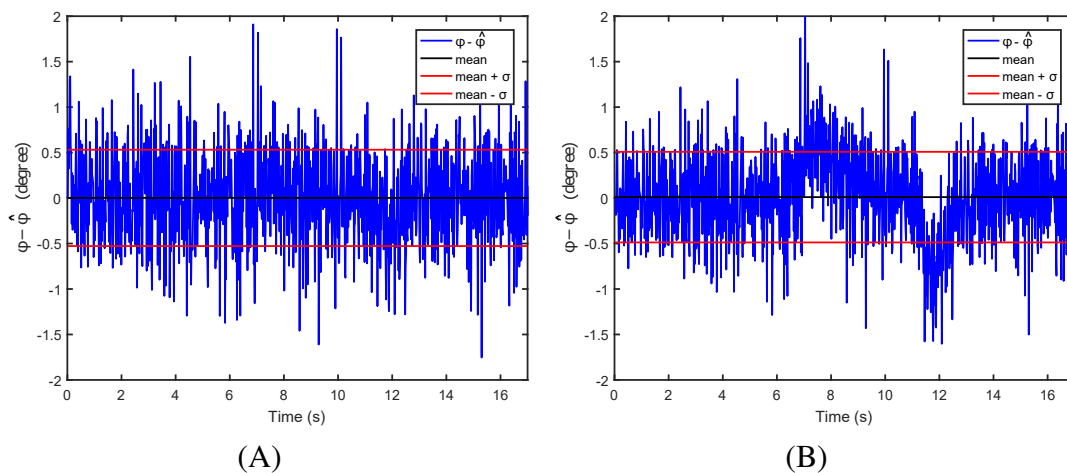
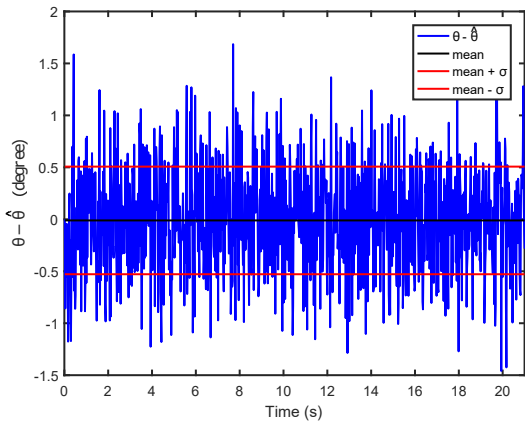
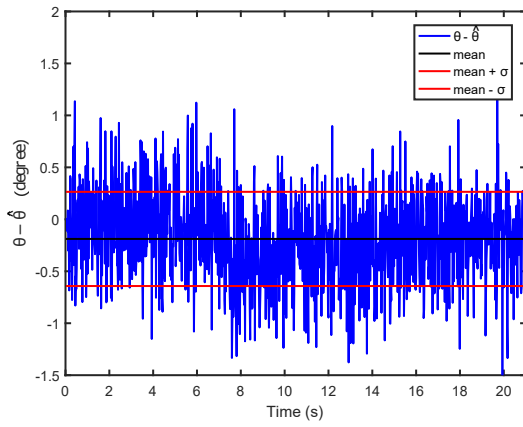


Figure 5.28: **A.** Riemannian $(\phi - \hat{\phi})$ for a rotation of 20° around X axis.; **B.** USQUE $(\phi - \hat{\phi})$ for a rotation of 20° around X axis.

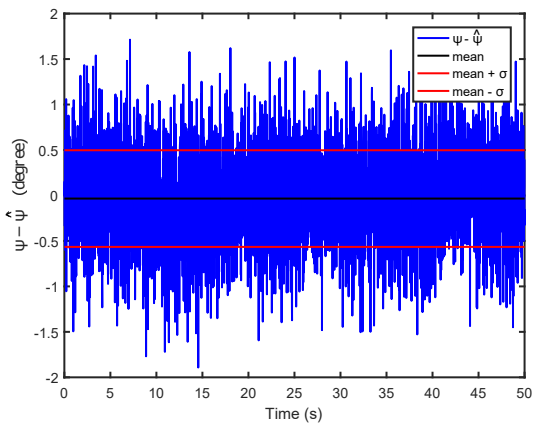


(A)

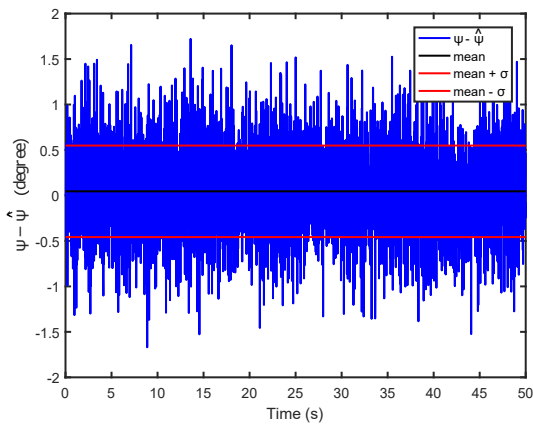


(B)

Figure 5.29: **A.** Riemannian $(\theta - \hat{\theta})$ for a rotation of 30° around Y axis.; **B.** USQUE $(\theta - \hat{\theta})$ for a rotation of 30° around Y axis.



(A)



(B)

Figure 5.30: **A.** Riemannian $(\psi - \hat{\psi})$ for a rotation of 8° around Z axis.; **B.** USQUE $(\psi - \hat{\psi})$ for a rotation of 8° around Z axis.

Experiment 3:

The next experiment consisted in allowing the platform to move freely. For the roll angles, both filters presented a very similar answer. But in the peak that happened in about 1s, the USQUE got a bit lost but quickly recovered (see Fig.5.31). In addition, in terms of pitch angles, the difference was more significant. For this axis, the Riemannian filter presented a much smooth answer. Furthermore, the USQUE presented some problems during the estimation of peak values (see Fig.5.32). And finally, considering yaw angles shown in Fig.5.33, both filters presented a similar answer with a difference of about 1 degree.

This experiment showed that both filters were capable of following the signal dynamic and providing a smooth answer on emulated real conditions. Furthermore, it is worth mentioning that these filters presented different time processing. The Riemannian filter spent 73.57s to process input data vectors of size 563×1 while the USQUE spent 10.989s to process the same amount of data.

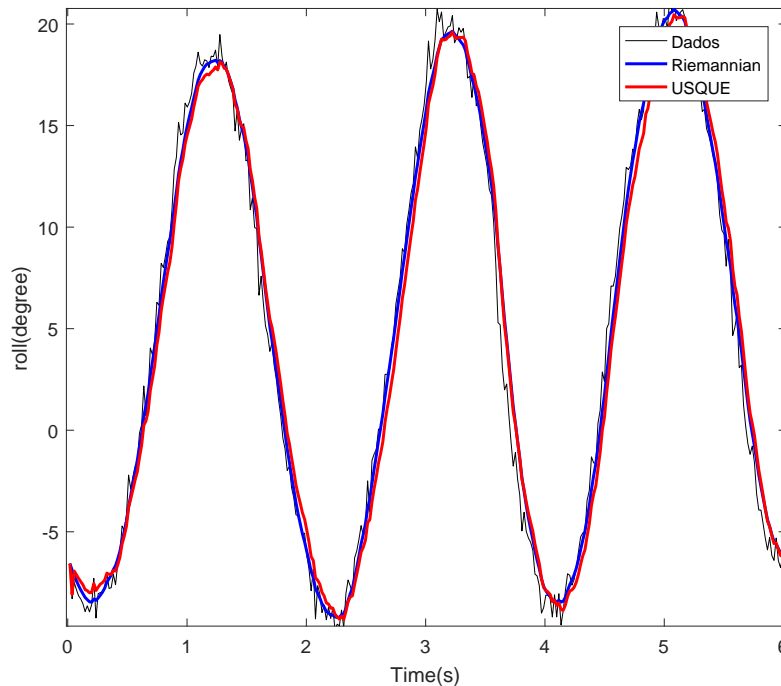


Figure 5.31: Free rotation.

In addition, these filters also provided the estimated gyro bias shown in Fig.5.34 and Fig.5.35. According to the datasheet, this sensor has an average bias of $2.5^0/hr$ for all axes, which can also be expressed in (radian/second) doing $\frac{0.0436}{3600}(rad/s) = 1.2 \times 10^{-5}(rad/s)$. Considering the datasheet information, the Riemannian filter presented a better estimation of the gyro bias reaching values of order $10^{-5}(rad/s)$ while USQUE reached values of order $10^{-1}(rad/s)$ (see Fig.5.34 and Fig.5.35).

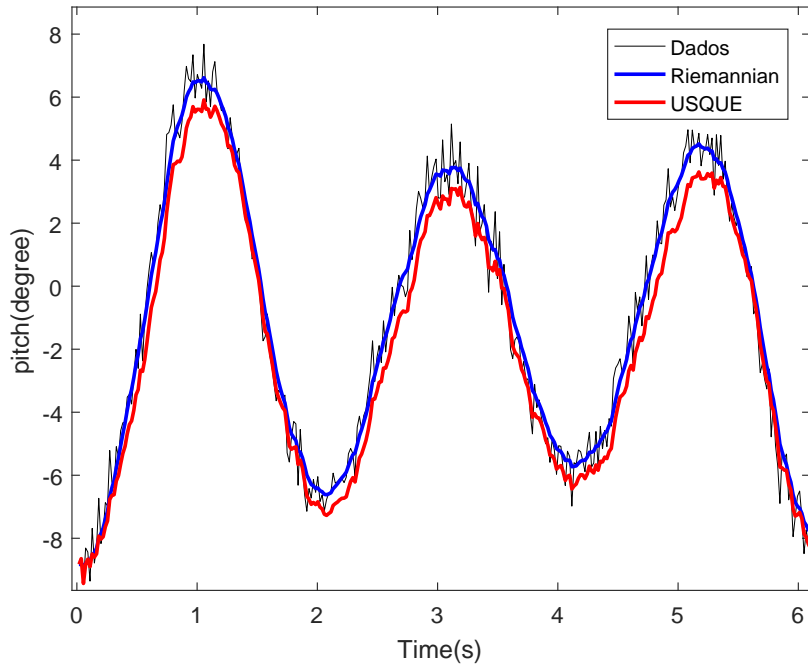


Figure 5.32: Free rotation.

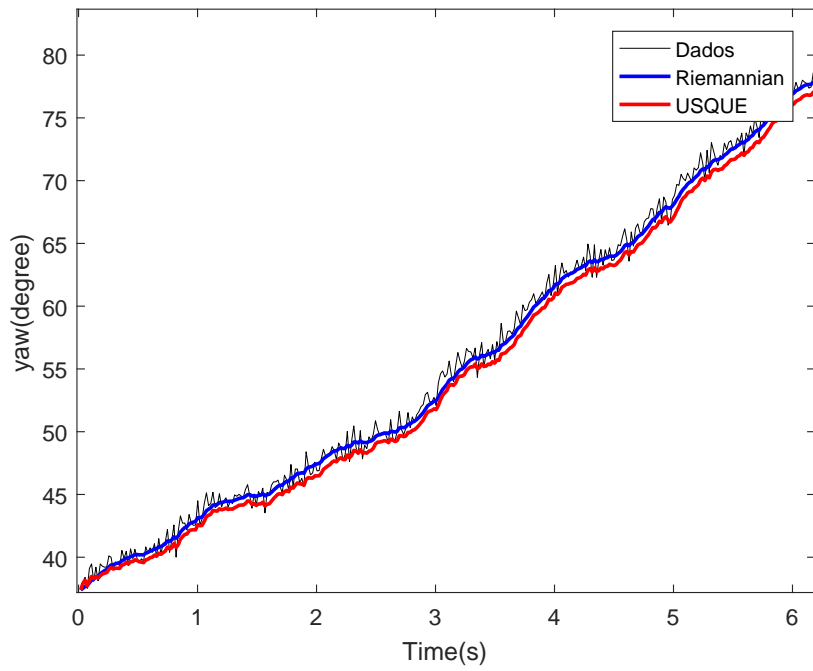


Figure 5.33: Free rotation.

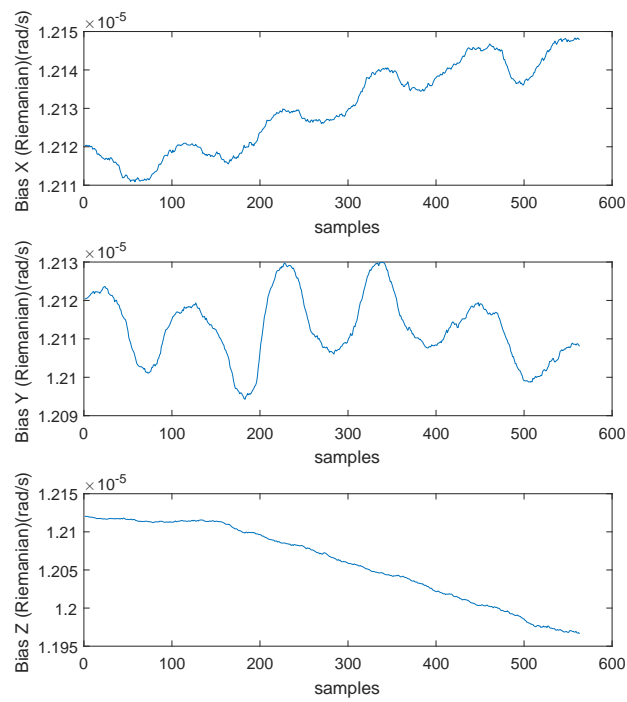


Figure 5.34: Bias(Riemannian).

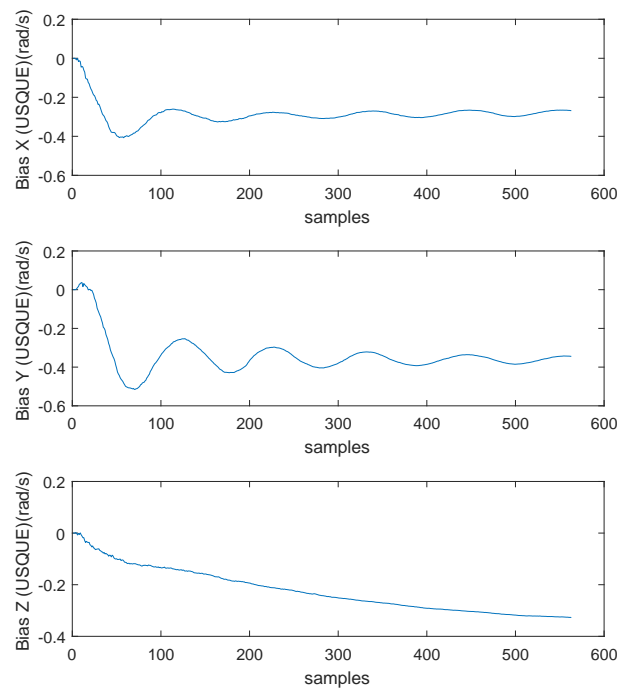


Figure 5.35: Bias(USQUE).

Chapter 6

Conclusion

In chapter 3, after deriving the testbed dynamic model, we showed in simulations and experiments, the singularity effect. These observations lead to some well know conclusions:

- Considering the rotation sequence 3-2-1, the Euler angles are affected by singularity when the pitch angle reaches $\pm 90^0$.
- Quaternions are not affected by singularity.

But the interesting point was observing that in simulation, the singularity was not as abrupt as it was in reality. This happens because Matlab approximates $\cos(90^0)$ by 6.1232×10^{-17} . While in the tilt-compensated eCompass algorithm, the $\tan(\phi)$ was implemented using the atan2 function, which generates the abrupt variations observed in experimental data.

Up to this point, we concluded that Euler angles provide a clear physical interpretation of the testbed orientation and that naturally the platform will not reach the singularity due to it's limitation of $\pm 45^0$ in x and y axes. But if we decide representing the orientation of a small satellite which has an initial angle, referring to the surface of the platform, such that it was possible reach values of θ closer to 90^0 , we can easily reach the singularity. In addition, in the context of small satellites that will operate in space, the singularity region is easily reachable. Therefore, as possible, quaternions could be used to avoid the singularity.

After observing the advantages and disadvantages of each parametrization of the attitude, in Chapter 5 we evaluated four filtering algorithms. EKF and UKF were the first two. From the results, we concluded that:

- The consistency method presented in Appendix G provided an easier way to choose the values of Q and R matrices.
- Both filters presented coherent estimated angular variations $\Delta\theta_{Filter}$ compared with the angular variations obtained from the protractor $\Delta\theta_{Protractor}$ when the platform was rotated each axis at the time.

- The implemented UKF was faster than the EKF and for the Experiment 3 both filters presented an almost equal answer.

After that, the quaternion filters were evaluated and the results led to the following conclusions:

- The method to determine approximate values of Q and R for the tangent space presented in section 5.1.3 provided an easy way to determine the values of Q and R matrices for the Riemannian filter.
- Both filters also presented coherent estimated angular variations $\Delta\theta_{Filter}$ compared with the angular variations obtained from the protractor $\Delta\theta_{Protractor}$ when the platform was rotated each axis at the time.
- The RiSAdUKF was capable of effectively estimating the testbed attitude for all tests. But it was slower than USQUE which had some problems to estimate the orientation in peak regions when the testbed preformed free movements.
- The gyro bias estimated by the RiSAdUKF was more correct than the gyro bias estimated by the USQUE.

In addition, it is worthy mentioning that all the filters were capable of providing a smoother and more precise answer compared with the sensors' output. This suggests that in order to obtain the same orientation result, the user either can save money in sensor and spend in processing or spend money in sensor and save in processing.

These conclusions are summarized in the following table:

Table 6.1: Summarized results

| Input | Algorithms | If the user has limited processing | If the user reaches $\theta = 90^0$ | If the user has high processing |
|-------------|------------|------------------------------------|-------------------------------------|---------------------------------|
| Euler angle | EKF | X | | |
| | UKF | X | | |
| Quaternions | USQUE | X | X | |
| | RiSAdUKF | | X | X |

6.1 Future work

For future work, this research can be extended by the following evaluations:

- The USQUE and RiSAdUKF filters could be implemented on ABACUS which is a professional nanosatellite control board. This implementation would improve the comprehension of each filter behaviour under limited processing conditions.

- In RiSAdUKF, the gradient descent algorithm could be substituted by another faster optimization algorithm. This would keep the advantages of this filter and also improve the filter behavior under limited processing conditions.
- Different control techniques could be implemented on ABACUS using the estimated orientation from these filtering algorithms.

Bibliography

- [Altmann 2005] Altmann, S. L. (2005). *Rotations, quaternions, and double groups*. Courier Corporation.
- [Arasaratnam and Haykin 2009] Arasaratnam, I. and Haykin, S. (2009). Cubature kalman filters. *IEEE Transactions on automatic control*, 54(6):1254–1269.
- [Bar-Itzhack and Harman 1997] Bar-Itzhack, I. Y. and Harman, R. R. (1997). Optimized triad algorithm for attitude determination. *Journal of guidance, control, and dynamics*, 20(1):208–211.
- [Bar-Shalom et al. 2004] Bar-Shalom, Y., Li, X. R., and Kirubarajan, T. (2004). *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons.
- [Bellion et al. 2016] Bellion, A., Elis, K., and Gaetano, S. D. (2016). *New compact S-band antenna for Nanosatellite TeleMetry and TeleCommand applications-EyeSat program*. 10th European Conference on Antennas and Propagation (EuCAP)-IEEE.
- [Bouwmeester and Guo 2010] Bouwmeester, J. and Guo, J. (2010). Survey of worldwide pico- and nanosatellite missions, distributions and subsystem technology. *Acta Astronautica*, 67(7-8):854–862.
- [Cai et al. 2011] Cai, G., Chen, B. M., and Lee, T. H. (2011). *Unmanned Rotorcraft Systems*. Springer.
- [Cheon and Kim 2007] Cheon, Y. J. and Kim, J. H. (2007). Unscented filtering in a unit quaternion space for spacecraft attitude estimation. *2007 IEEE International Symposium on Industrial Electronics*, pages 66–71.
- [Cohn 2003] Cohn, M. P. (2003). *Further algebra and applications*. UK:Springer-Verlag, London.
- [Crassidis and Markley 2003] Crassidis, J. L. and Markley, F. L. (2003). Unscented filtering for spacecraft attitude estimation. *Journal of guidance, control, and dynamics*, 26(4):401–422.

- [Crassidis et al. 2007] Crassidis, J. L., Markley, F. L., and Cheng, Y. (2007). Survey of non-linear attitude estimation methods. *Journal of guidance, control, and dynamics*, 30(1):12–28.
- [Frautschi et al. 2007] Frautschi, S. C., Olenick, R. P., and Apostol, T. M. (2007). *The mechanical universe: Mechanics and heat*. Cambridge University Press.
- [Garcia et al. 2017] Garcia, C. B. A., Vale, S. R. C., Martins-Filho, L. S., Duarte, R. O., Kuga, H. K., and Carrara, V. (2017). Validation tests of attitude determination software for nanosatellite embedded systems. *Measurement (ELSEVIER)*, 116:391–401.
- [Garcia et al. 2018] Garcia, R. V., Kuga, H. K., Silva, W. R., and Zanardi, C. (2018). Unscented kalman filter and smoothing applied to attitude estimation of artificial satellites. *Computational and Applied Mathematics(Springer)*, 116:1–10.
- [Garcia et al. 2012] Garcia, R. V., Kuga, H. K., and Zanardi, M. C. F. P. S. (2012). Unscented kalman filter applied to the spacecraft attitude estimation with euler angles. *Mathematical Problems in Engineering*, 2012.
- [Giri et al. 2011] Giri, C., Ochieng, E., Tieszen, L. L., Zhu, Z., Singh, A., Loveland, T., Masek, J., and Duke, N. (2011). Status and distribution of mangrove forests of the world using earth observation satellite data. *Global Ecology and Biogeography*, 20(1):154–159.
- [Greenwood 1965] Greenwood, D. T. (1965). *Principles of dynamics*. Prentice Hall, Englewood Cliffs, New Jersey.
- [Guimarães et al. 2017] Guimarães, F. C., Silva, R. C., Loiola, J. V. L., Borges, G. A., R. A. B., 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*, pages 1784–1789.
- [Hamilton 1848] Hamilton, W. R. (1848). Xi. on quaternions; or on a new system of imaginaries in algebra. *Philosophical Magazine Series 3*, 33(219):58–60.
- [Jablonski 2014] Jablonski, A. M. (2014). *Recent Advances in Assembly, Integration, and Testing (AIT) at the David Florida Laboratory*. AIAA SPACE 2014 Conference and Exposition.
- [Julier and Uhlmann 2004] Julier, S. J. and Uhlmann, J. K. (2004). Unscented filtering an nonlinear estimation. *IEEE*, 92(3):401–422.
- [Kuipers 1999] Kuipers, J. C. (1999). *QUATERNIONS AND ROTATION SEQUENCES*. Princeton University Press, Princeton, New Jersey.
- [Leeghim et al. 2010] Leeghim, H., Choi, Y., and Jaroux, B. A. (2010). Uncorrelated unscented filtering for spacecraft attitude determination. *Acta Astronautica*, 67(1-2):135–144.

- [Loiola et al. 2018] Loiola, J. V. L., Ploeg, L. C. V., 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. *2018 IEEE Aerospace Conference*, pages 1–8.
- [Loiola et al. 2017] Loiola, J. V. L., Silva, L. M. B., Battistini, S., Borges, R. A., and Cappelletti, C. (2017). Development of a hardware-in-the-loop test platform for nanosatellites adcs integrated with an ukf. *4th IAA Conference on University Satellite Missions and Cubesats Workshop*, pages 365–373.
- [Menegaz 2016] Menegaz, H. M. T. (2016). *Unscented Kalman Filtering on Euclidian and Riemannian Manifolds*. PhD thesis, Brasília.
- [Microelectronics 2010] Microelectronics, S. (2010). Application note AN3192. using LSM303DLH for a tilt compensated electronic compass.
- [Mirzaei and Roumeliotis 2008] Mirzaei, F. M. and Roumeliotis, S. I. (2008). A kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation. *IEEE transactions on robotics*, 24(5):1143–1156.
- [NOAA 2018] NOAA (2018). NOAA magnetic field calculators. <https://www.ngdc.noaa.gov/geomag-web/#igrfwmm>. Accessed: 2018-05-04.
- [Noureldin et al. 2012] Noureldin, A., Karamat, T. B., and Georgy, J. (2012). *Fundamentals of inertial navigation, satellite-based positioning and their integration*. Springer Science & Business Media.
- [Nussenzveig 2002] Nussenzveig, H. M. (2002). *Curso de física básica*. E. Blucher.
- [Oliveira et al. 2015] Oliveira, A. M., Kuga, H. K., and Carrara, V. (2015). Air bearing platforms for simulation of spacecraft attitude control systems.
- [Ozyagcilar 2012] Ozyagcilar, T. (2012). Implementing a tilt-compensated ecompass using accelerometer and magnetometer sensors. *Freescale semiconductor, ANs*, 4248.
- [Pennec 1998] Penneç, X. (1998). Computing the mean of geometric features application to the mean rotation.
- [Pham et al. 2015] Pham, M. D., Low, K. S., Goh, S. T., and Chen, S. (2015). Gain-scheduled extended kalman filter for nanosatellite attitude determination system. *IEEE Transactions on Aerospace and Electronic Systems*, 51(2):1017–1028.
- [Schwartz et al. 2003] Schwartz, J. L., Peck, M. A., and Hall, C. D. (2003). Historical review of air-bearing spacecraft simulator. *Journal of Guidance, Control and Dynamics*, 26(4):513–522.
- [Selva and Krejci 2012] Selva, D. and Krejci, D. (2012). A survey and assessment of the capabilities of cubesats for earth observation. *Acta Astronautica*, 74:50–68.

- [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 (accepted to publication). *JOURNAL OF AEROSPACE ENGINEERING*.
- [Silva and Rodrigues 2015] Silva, R. C. and Rodrigues, U. A. (2015). *SIMULADOR DE SISTEMA DE DETERMINAÇÃO E CONTROLE DE ATITUDE DE PEQUENOS SATÉLITES*. UnB, Brasília, Brazil.
- [Simon 2006] Simon, D. (2006). *Optimal state estimation*. New Jersey: Wiley-Interscience, Hoboken, New Jersey.
- [Tewari 2007] Tewari, A. (2007). *Atmospheric and space flight dynamics*. Springer.
- [Thomson 1961] Thomson, W. T. (1961). *Introduction to Space Dynamics*. General Publishing Company, New York 11501, New York, 1st edition.
- [Titterton and Weston 1997] Titterton, D. H. and Weston, J. L. (1997). *Strapdown Inertial Navigation Technology*. IET, New York 17, New York, 2nd edition.
- [Wertz 1990] Wertz, J. R. (1990). *Spacecraft Attitude Determination and Control*. kluwer Academic Publishers, Torrance, California, 6nd edition.

Appendices

Appendix A

Sensors

This section will provide fundamental concepts and specifications about the sensors utilized to obtain the platform attitude.

In order to address this issue, the platform was equipped with a inertial measurement unity (IMU) fabricated by Adafruit. This component was placed in the test platform center of rotation as it can be seen in Fig.A.1, where xyz are the IMU axes. This inertial measurement unity provides three different types of output, which are acceleration, magnetic orientation and angular velocity. Each of these measurements is provided through three components in IMU axes. The sum of all output is equal to nine different measurements, that is why this type of sensor is referred as 9DOF.

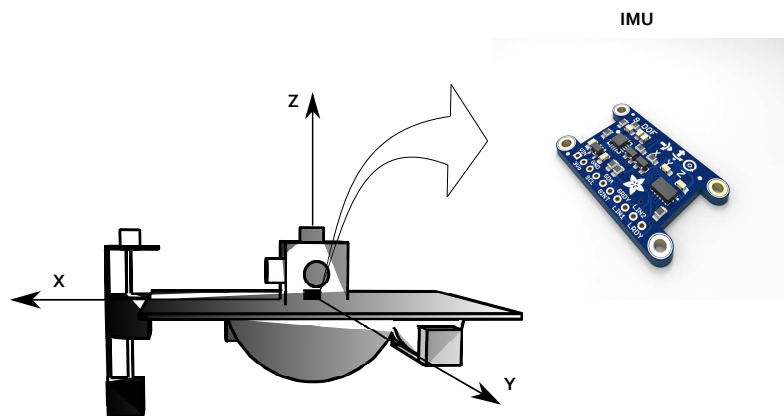


Figure A.1: IMU location

This IMU obtains those measurements from two different electronic devices, which are a LSM303DLHC and a L3GD20. The specifications of those components are presented in table A.1 In addition it is important to mention that those specifications are in the platform expected range operation, but they can be adjusted to different ranges through register settings.

The LSM303DLHC is composed by a 3D digital linear acceleration sensor and a 3D digital magnetic sensor. In order to better understand those components, their work principles

Table A.1: LSM303DLHC and L3GD20 settings

| Components | Sensor | Range | Measurement unit |
|------------|---------------|--------------------|------------------|
| LSM303DLHC | Magnetometer | -1.3 \mapsto 1.3 | Gauss |
| | Accelerometer | -2 \mapsto 2 | g |
| L3GD20 | Gyroscope | -250 \mapsto 250 | degree/s |

will be presented. The accelerometer is composed by three proof masses aligned with each axis of the system case. Each one of those mass is connected to two springs which allow the mass movement along each axis direction as it can be seen in Fig.A.2. In order to measure the displacement from the equilibrium position, the mass is connected to a electric circuit in such way that the mass movement is associated by a variable resistance. And based on the voltage variations it is obtained a relation between the voltage and acceleration on that direction.

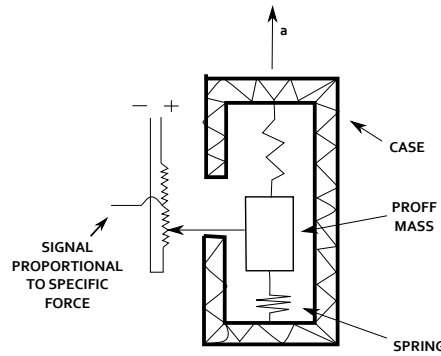


Figure A.2: A single axis accelerometer

The other sensor is the magnetometer, which has two main construction principles. The first one is based in the Hall effect principle, which states that a magnetic field applied perpendicularly to a current flow generates a voltage difference across the conductor shown in Fig.A.3.A. The second principle is the magneto-resistive effect, which is observed in some magnetic field sensitive materials shown in Fig.A.3.B. The main structural configuration of a magnetometer is composed by three of these materials aligned with each axis of the system case.

Finally, the L3GD20 is a low-power three-axis angular rate sensor also known as gyroscope. This sensor is developed based on the Coriolis force. According to [Frautschi et al. 2007], this force is considered as an inertial force that acts on a object that is in motion relative to a rotating frame. In another words, when a object is rotating there is a force orthogonal to the axis of rotation. If the body axes is rotating in clockwise direction, the force acts to the left of the object motion and If the body axes is rotating in counterclockwise direction, the force acts to the right of the object motion. Based on that, a mass, connected by springs, is placed on the plane orthogonal to axis of rotation as shown in Fig.A.4. The displacement from the equilibrium position is used to measure the angular velocity.

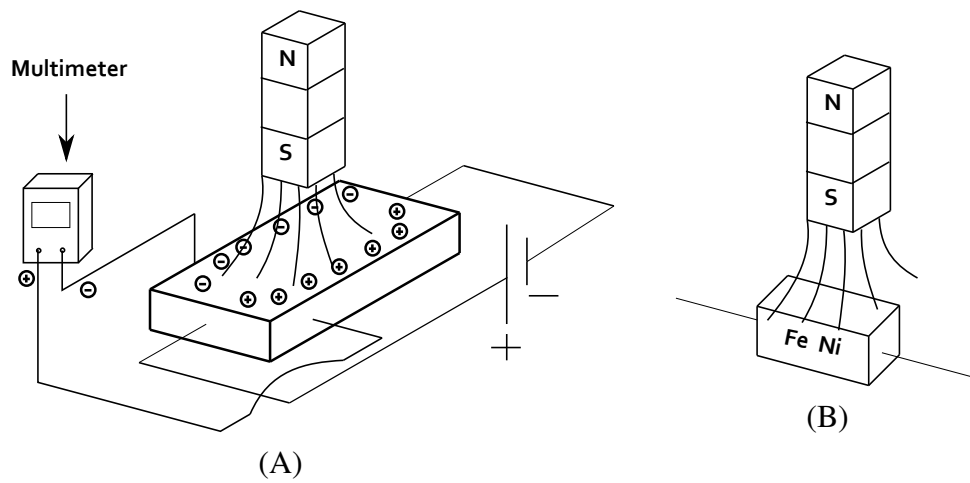


Figure A.3: Theoretical conception of principles used to construct a magnetometer; **A.** Hall effect; **B.** Magneto-resistive effect.

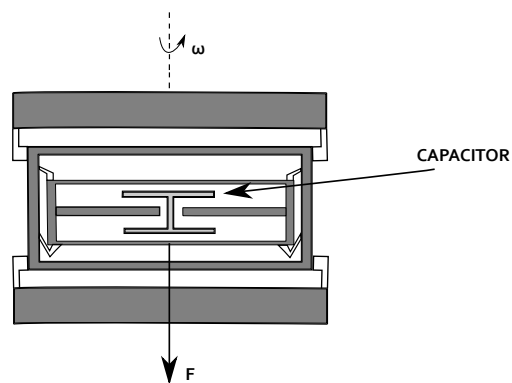


Figure A.4: Coriolis effect

Appendix B

Exponential and logarithm mapping of the S^{n-1} applied in the Riemannian Filter

According to [Menegaz 2016], the exponential mapping ($exp_{\mathbf{q}}^e(g^{TS})$), used to mapping vectors from tangent basis to the sphere(in canonical basis $\{e_1, \dots, e_n\}$) of \mathbb{R}^n) is given by

$$exp_{\mathbf{q}}^e(g^{TS}) := \mathbf{q} \cos(\|g^{TS}\|) + \frac{g^{TS}}{\|g^{TS}\|} \sin(\|g^{TS}\|). \quad (\text{B.1})$$

Now, lets $\mathbf{p} = exp_{\mathbf{q}}^e(g^{TS})$, so, for

$$\theta^{TS} := \arccos(\langle \mathbf{q}, \mathbf{p} \rangle), \quad (\text{B.2})$$

The inverse mapping called logarithm mapping ($ln_{\mathbf{q}}^p$) is given by

$$ln_{\mathbf{q}}(\mathbf{p}) := exp_{\mathbf{q}}^{-1}(v) = \frac{\theta^{TS}}{\sin(\theta^{TS})} \mathbf{p} - \frac{\theta^{TS} \cos(\theta^{TS})}{\sin(\theta^{TS})} \mathbf{q}. \quad (\text{B.3})$$

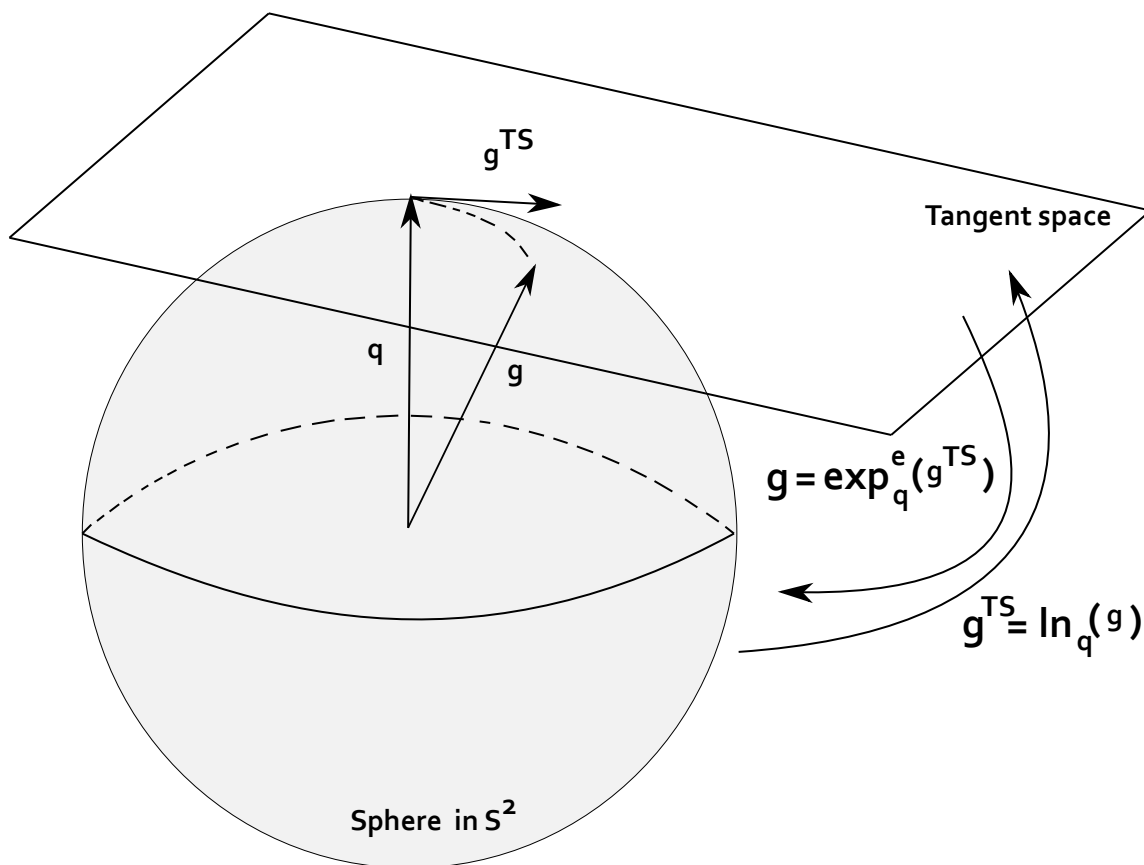


Figure B.1: Exponential and logarithm mapping

Appendix C

Gradient descent algorithm applied in the Riemannian Filter

According to [Menegaz 2016] the predicted quaternion $\hat{\mathbf{x}}_{k|k-1} \in S^3$ can be obtained using the intrinsic gradient descent algorithm presented in [Pennec 1998]. This algorithm consists of the following steps:

Step 1:

Initially choose a threshold $\varepsilon \in \mathbb{R}, \varepsilon > 0$ and choosing the initial candidate

$$\mathbf{q} := f_k(\hat{\mathbf{x}}_{k-1|k-1}) \quad (\text{C.1})$$

Step 2:

For $1 \leq i \leq N$:

$$e_i^v := \sum_{i=1}^N w_i^m QtoRoV(\chi_i^{k|k-1} \otimes \mathbf{q}^{-1}). \quad (\text{C.2})$$

Step 3:

While ($\|\hat{e}^v\| > \varepsilon$)

(a) Choose a new candidate

$$\mathbf{q} := RoVtoQ(\hat{e}^v) \otimes \mathbf{q}, \quad (\text{C.3})$$

After that repeat the step 2.

Step 4:

$$\hat{\mathbf{x}}_{k|k-1} := \mathbf{q}. \quad (\text{C.4})$$

Appendix D

Vector parametrization for unit quaternions applied in the Riemannian Filter

Some times computations of unity quaternions lead to computational problems. According to [Menegaz 2016], one convenient way to deal with this issue is to use vector parameterizations of the S^3 such as rotation vectors. This author shows that quaternions are expressed in rotations vectors as follows. Initially, consider a quaternion $\mathbf{q} := [q_0 \ \varrho]^T$, so:

The equations that express a quaternion in rotation vectors is given by:

$$QtoRoV(\mathbf{q}) := q_{RoV}^v := 2\arccos(q_0) \frac{\varrho}{\|\varrho\|}. \quad (\text{D.1})$$

where q_{RoV}^v is the result of the parameterization and ϱ is the imaginary part of the quaternion.

In addition, the inverse transformation is given by:

$$RoVtoQ(q_{RoV}^v) := \mathbf{q} \quad (\text{D.2})$$

where,

$$\mathbf{q} := \cos\left(\frac{\|q_{RoV}^v\|}{2}\right) + i_m \frac{q_{RoV}^v}{\|q_{RoV}^v\|} \sin\left(\frac{\|q_{RoV}^v\|}{2}\right) \quad (\text{D.3})$$

Appendix E

Sigma sets method applied in the Riemannian Filter

In the Riemannian Filter, the sigma sets were calculated using the methodology presented in [Julier and Uhlmann 2004]. This methodology consist of the following algorithm:

Step 1:

Initially for $(N = 2n + 1)$, where n is the number of states that you have, We have to choose a $w_0 < 1$. For the Riemannian filter we choose $w_0 = \frac{1}{3}$.

Step 2:

For $i=1,\dots,n$:

$$w_i = w_{i+n} = \frac{1 - w_0}{2n}, \quad \chi_0 = \bar{X} \quad (\text{E.1})$$

$$\chi_i = \bar{X} + \left(\sqrt{\frac{n}{1 - w_0} P_{XX}} \right), \quad (\text{E.2})$$

$$\chi_{i+n} = \bar{X} - \left(\sqrt{\frac{n}{1 - w_0} P_{XX}} \right). \quad (\text{E.3})$$

Appendix F

Hard-iron and scale factor compensation

According to [Microelectronics 2010], a magnetometer can be calibrated using the following method. First, considering the soft-iron effect negligible, the ellipsoid generated by 3D rotations are not tilted. The equation of this ellipsoid is given by:

$$\frac{(M_x - X_0)^2}{a^2} + \frac{(M_y - Y_0)^2}{b^2} + \frac{(M_z - Z_0)^2}{c^2} = R^2 \quad (\text{F.1})$$

where

- X_0 , Y_0 and Z_0 are the offset generated by hard-iron distortion.
- M_x , M_y and M_z are the magnetic sensor raw measurements.
- a , b and c are the semi-axis lengths.
- R is the magnetic field strength constant.

From Eq. (F.1)

$$M_x^2 = [M_x \quad M_y \quad M_x \quad -M_y^2 \quad -M_z^2 \quad 1] \begin{bmatrix} 2X_0 \\ \frac{a^2}{b^2} 2Y_0 \\ \frac{a^2}{c^2} 2Z_0 \\ \frac{a^2}{b^2} \\ \frac{a^2}{c^2} \\ a^2 R^2 - X_0^2 - \frac{a^2}{b^2} Y_0^2 - \frac{a^2}{c^2} Z_0^2 \end{bmatrix}, \quad (\text{F.2})$$

After collecting n magnetic field measurements, the Eq. (F.2) can be written as:

$$W_{n \times 1} = [H]_{n \times 6} X_{6 \times 1} \quad (\text{F.3})$$

Applying the last square method

$$X = [H^T H]^{-1} H^T W. \quad (\text{F.4})$$

So, the offset components caused by hard-iron disturbance are given by:

$$X_0 = \frac{X(1)}{2}, Y_0 = \frac{X(2)}{2X(4)}, Z_0 = \frac{X(3)}{2X(5)}. \quad (\text{F.5})$$

Now, let

$$A := a^2 R^2 = X(6) + X_0 + X(4)Y_0^2 + X(5)Z_0^2, \quad (\text{F.6})$$

$$B := \frac{A}{X(4)}, \quad (\text{F.7})$$

$$C := \frac{A}{X(5)}. \quad (\text{F.8})$$

and also let the magnetic measurements less the offsets be defined as:

$$M_{xx} := M_x - X_0, \quad (\text{F.9})$$

$$M_{yy} := M_y - Y_0, \quad (\text{F.10})$$

$$M_{zz} := M_z - Z_0. \quad (\text{F.11})$$

So,

$$\frac{M_{xx}^2}{A} + \frac{M_{yy}^2}{B} + \frac{M_{zz}^2}{C} = 1. \quad (\text{F.12})$$

From the ellipsoid equation theory, the scale factors are given by:

$$SC_x = \sqrt{A}, \quad (\text{F.13})$$

$$SC_y = \sqrt{B}, \quad (\text{F.14})$$

$$SC_z = \sqrt{C}. \quad (\text{F.15})$$

Therefore, the final magnetic field unity vector components are given by:

$$M_{xxx} = \frac{M_{xx}}{SC_x}, \quad (\text{F.16})$$

$$M_{yyy} = \frac{M_{yy}}{SC_y}, \quad (\text{F.17})$$

$$M_{zzz} = \frac{M_{zz}}{SC_z}. \quad (\text{F.18})$$

Then,

$$M_{xxx}^2 + M_{yyy}^2 + M_{zzz}^2 = 1. \quad (\text{F.19})$$

Finally, in order to obtain the corrected magnetic field vector, this unity vector given by the components M_{xxx} , M_{yyy} and M_{zzz} must be multiplied by local magnetic field intensity.

Appendix G

Practical evaluation of consistency

In this section will be presented a method, derived based on linear filters, used to evaluate the consistency of an estimator obtained from [Bar-Shalom et al. 2004]. According to this author a state estimator is consistent if it satisfy the follow assumptions

- First, the expectation of the difference between the state in the instant k minus the the predicted value for that instant must be zero

$$E[x(k) - \hat{x}(k|k)] = 0. \quad (\text{G.1})$$

- Second, The expectation of this difference squared is equals to the state covariance matrix

$$E[[x(k) - \hat{x}(k|k)][x(k) - \hat{x}(k|k)]^T] = P(k|k). \quad (\text{G.2})$$

Under the linear Gaussian assumption:

$$p[x(k)|Z^k] = N[x(k); \hat{x}(k|k), P(k|k)]. \quad (\text{G.3})$$

The normalized state estimation error squared (NEES) is given by

$$d2(k) = [x(k) - \hat{x}(k|k)]^T P(k|k)^{-1} [x(k) - \hat{x}(k|k)]. \quad (\text{G.4})$$

In order to evaluate the filter consistency. The NEES should be calculated for each interaction and about (95%) of these values should be below the probability level. This probability level is obtained from the normalized chi square distribution taking into account the distribution degree of freedom, which is equal to the number of states, and probability desired, which is normally considered to be 0.95.

Appendix H

Parallel transport

In order to represent a vector in different tangent space, the Riemannian filter uses parallel transportations. This method transports vectors and matrices from tangent space of a quaternion to the tangent space of other quaternion. This section will present the parallel transport of vectors and matrices explained in [Menegaz 2016].

H.0.1 Parallel transport of a vector

Given the quaternions \mathbf{q} and $\mathbf{r} \in \mathbb{H}$ and assuming that $\nu = \log_{\mathbf{q}}^{\mathbf{r}}$, the parallel transport of a vector from $\zeta \in T_{\mathbf{q}}S^{n-1}$ to $T_{\mathbf{r}}S^{n-1}$ is given by:

$$\zeta^{T_{\mathbf{r}}S^{n-1}} := -\mathbf{q} \sin(\|\nu\|) \frac{\nu^T}{\|\nu\|} \zeta + \frac{\nu}{\|\nu\|} \cos(\|\nu\|) \mathbf{q}^T \zeta + \left(I - \frac{\nu}{\|\nu\|} \frac{\nu^T}{\|\nu\|} \right) \zeta. \quad (\text{H.1})$$

H.0.2 Parallel transport of a matrix

Assuming that λ and ζ are the eigenvalues and eigenvectors of a matrix $P \in T_{\mathbf{q}}S^3 \times T_{\mathbf{q}}S^3$, the parallel transport from $T_{\mathbf{q}}S^3$ to $T_{\mathbf{r}}S^3$ of the covariance matrix P is obtained following the following steps:

- Step 1: The parallel transport from $T_{\mathbf{q}}S^3$ to $T_{\mathbf{r}}S^3$ of the eigenvectors of P is done using Eq. (H.1)

$$\zeta^{T_{\mathbf{r}}} := -\mathbf{q} \sin(\|\nu\|) \frac{\nu^T}{\|\nu\|} \zeta + \frac{\nu}{\|\nu\|} \cos(\|\nu\|) \mathbf{q}^T \zeta + \left(I - \frac{\nu}{\|\nu\|} \frac{\nu^T}{\|\nu\|} \right) \zeta. \quad (\text{H.2})$$

- Step 2: Using the parallel transportation of eigenvectors from step 1, the parallel trans-

portation of the matrix \mathbf{P} is given by:

$$P^{Tr} = \sum_{i=1}^3 \lambda \zeta^{Tr} (\zeta^{Tr})^T. \quad (\text{H.3})$$

Appendix I

Transformations between tangent basis and canonical basis

According to [Menegaz 2016] the transformation from tangent basis to canonical basis (TBtoCB) and the transformation from canonical basis to tangent basis (CBtoTB) are obtained as follows:

First, it is necessary calculating the transformation matrix $T_{L|C}$, which will send vectors from the tangent basis to canonical basis. This matrix is given by:

$$T_{L|C} = \begin{bmatrix} [X_1^L]_C & [X_2^L]_C & [X_3^L]_C & [X_4^L]_C \end{bmatrix}, \quad (\text{I.1})$$

where $[X_1^L]_C = (\mathbf{q})_C$ and $[X_i^L]_C$ is given by the parallel transport of the vectors e_i of the tangent space of $[1 \ 0 \ 0 \ 0]^T$ para \mathbf{q} , considering $i=2,3$ and 4.

Using $T_{L|C}$, the transformation of basis are given by:

- (TBtoCB)

$$[V]_C = T_{L|C}[V]_L \quad (\text{I.2})$$

- (CBtoTB)

$$[V]_L = (T_{L|C})^{-1}[V]_C \quad (\text{I.3})$$