UNIVERSIDAD DE MURCIA

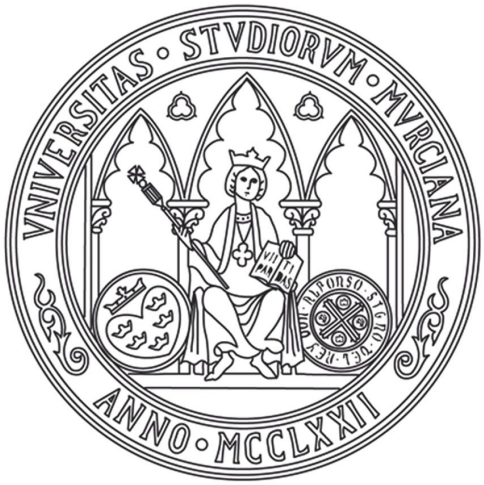
ESCUELA INTERNACIONAL DE DOCTORADO

Kalman Filtering for Orientation Estimation using IMUs

Filtrado Kalman para Estimación de la Orientación usando IMUs

D. Pablo Bernal Polo

2020



Universidad de Murcia
Facultad de Informática

Ph.D. Thesis

Kalman Filtering for Orientation Estimation using IMUs

Filtrado Kalman para Estimación de la Orientación usando IMUs

Author:
Pablo Bernal Polo

Advisor:
Dr. Humberto Martínez Barberá

2020

Abstract

Navigation, automatic vehicle control, robotics, or virtual reality. All are applications in which the knowledge about the orientation plays an important role. Nowadays there are lightweight and cheap sensors that provide information about the orientation of a system. This thesis focuses on obtaining such orientation using one type of those sensors: Inertial Measurement Units (IMUs). This objective raises the two main problems addressed in this work. The first is the study of a method to estimate the orientation using vector measurements. The second is the calibration of IMUs in particular, and triaxial sensors in general, which emanates from the fact that IMU measurements are usually imprecise and inaccurate.

The problem of estimating the orientation of a rigid body using sensor measurements has been broadly addressed. In 1982, a pioneering article was published on orientation estimation through Kalman filtering in which the orientation was described using quaternions, instead of the classic orientation descriptors, such as the Euler angles. Quaternions have properties that make them advantageous over the rest of orientation descriptors. However, the Kalman filter is not directly applicable in this scenario, since only unit quaternions represent orientations. Classicaly, this problem is solved by defining a "multiplicative update", and redefining the covariance matrix using a three-dimensional vector. Recently, it has also been observed that a "covariance correction step" is necessary. This study is also included in the framework of Kalman filtering with quaternions as the orientation descriptors, but introduces a new viewpoint from which the notions of "multiplicative update" and "covariance correction step" are conceived in a natural way.

Although there are other methods used to calibrate triaxial sensors, a frequent starting point is the definition of an optimization problem. This work also uses that strategy, but it differs from many others by incorporating the temperature into the sensor model. In the end, an iterative algorithm is obtained that only needs to know the module of the real vector associated with each sensor measurement to produce a calibration. The algorithm is tested with real sensor measurements obtained using an electro-mechanical device constructed for this purpose. The results obtained from the multiple calibrations allow us to better understand the behavior of IMUs.

Acknowledgements

Gracias a mi director de tesis, Humberto, por su confianza y por darme la libertad e independencia que necesitaba. También por presionarme en los momentos adecuados, porque eso ha resultado en un trabajo de mayor calidad que el que habría resultado si no lo hubiera hecho.

Gracias a la gente del Centro de Transferencia Tecnológica de la Universidad de Murcia, y en especial a Antonio. Gracias por haberme acompañado durante estos años. También por las reflexiones que hemos compartido sobre temas diversos durante los viajes a Fuente Álamo, almuerzos, y demás ratos.

Gracias a los profesores que han contribuido en mi educación; en el desarrollo de mi rigor científico. He disfrutado de cada clase. De cada lección.

Gracias también a los compañeros que me han acompañado, tanto durante la carrera, como en el master. Especialmente a los de aquella última promoción de licenciatura, con los que tantos momentos compartí. Espero no olvidar nunca tan buenos recuerdos.

Gracias a mis amigos del instituto. Nuestros encuentros han amenizado cada año de estudio. Tras tanto tiempo, aún seguimos conservando nuestra amistad, y espero que así siga siendo durante mucho más.

Gracias a toda mi familia, pero en especial a mi familia de El Palmar. Sin su apoyo, me habría sido imposible producir este trabajo.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1. Motivation

In recent years we have witnessed the birth of technologies such as VR (Virtual Reality), or UAVs (Unmanned Aerial Vehicles, colloquially known as drones). One of the main reasons why these technologies emerged were the advances in the manufacture of miniaturized components, which brought the Micro Electro Mechanical Systems (MEMS): mechanical systems on a chip. A particular case of a MEMS is an IMU (Inertial Measurement Unit) that include MEMS accelerometers, which measure proper acceleration, MEMS gyroscopes, which measure angular velocity, and in some cases MEMS magnetometers, which measure the magnetic field. Classical IMUs were large, heavyweight, and expensive devices. Few had access to this equipment, and they could only be installed in specialized systems such as submarines, ships, aircrafts, spacecrafts, satellites, and some boats and cars. MEMS IMUs were born as small, lightweight, and cheap devices. This made it possible to install IMUs in smaller systems such as the previously mentioned VR glasses and drones, and other small systems such as robots, missiles, or motion capture devices.

IMU measurements contain information about its orientation: accelerometer measurements tend to point down, and gyroscope measurements inform us about the rotation speed. Knowing the orientation is important in multiple scenarios:

Control: some autonomous vehicles need to know their orientation to compute the control outputs. For example, a drone needs to know its orientation to compute the speed at which the propellers must rotate to accelerate in a certain direction [1, 2]. Another example is a rocket that needs to know its orientation for the Thrust Vector Control (TVC): the system that controls the direction of thrust from its engine.

Navigation: there are two main reasons why the orientation is important for navigation:

- Heading: the orientation determines the direction of the front of the vehicle, which tends to be the forward direction.
- Inertial navigation: accelerometer measurements can be used to estimate the position of a vehicle [3]. Since the measurements are taken in a non-inertial reference frame (the vehicle), they must first be expressed in an inertial reference frame. For this we need to know the orientation of the vehicle with respect to that reference frame.

Virtual reality: it is necessary to know the orientation of the glasses with respect to a static reference frame to perform the projection of the virtual scene.

The growing interest that these applications arouse leads to research on methods to estimate the orientation of a system. However, there are several factors that make it difficult to obtain the orientation from IMU measurements. First, IMUs produce partial information about the state of the system, which makes it a partially observable system. Then it is necessary to use some type of memory to merge the information carried by the measurements over time. Second, like any other sensor, IMUs produce imperfect measurements, which degrades the estimations produced by any estimation algorithm fed with them. This problem is mitigated with a calibration. However, since IMUs are triaxial sensors (they measure vectors in place of scalar quantities), the calibration process presents additional issues.

This thesis addresses the two problems mentioned above. Namely,

- The estimation of the orientation of a system using IMU measurements.
- The calibration of an IMU.

In the next section we will put these problems in perspective by reviewing the related scientific literature.

1.2. Literature Review

This section introduces the two main problems addressed in this thesis. We review the scientific literature that, in one way or another, have influenced on the development of this work.

1.2.1. Orientation Estimation

To estimate the dynamic state of a system, we usually use the rigid body model. A rigid body is described using 4 mathematical objects: two of them represent its position and velocity, and the other two represent its orientation and angular velocity. However, our sensors do not usually measure these mathematical objects explicitly; rather, they provide measurements that implicitly contain information about the dynamic state. For example, for IMUs, information on angular velocity is provided explicitly, but information on the orientation is provided by "telling us where down is". To extract from the measurements the underlying information about the state we use estimation algorithms.

Although there are other mathematical tools used for estimation [4, 5], the Kalman filter [6] has become the algorithm par excellence in this area. Because of its simplicity, the rigor and elegance in its mathematical derivation, and its recursive nature, it is very attractive for many practical applications. Its non-linear versions have been widely used in orientation estimation: the Extended Kalman Filter (EKF), and the Unscented Kalman Filter (UKF) [7]. However, there are problems arising from the used parametrization to describe the orientation.

The orientation of a system is represented by the rotation transformation that relates two reference frames: a reference frame anchored to our rigid body, and an external reference frame. A thorough survey of orientation representations is provided in [8]. The parametrization used to describe a rotation transformation could be singular, or present discontinuities among others. Table 1.1 summarizes the main characteristics of the most used parametrizations.

Table 1.1: Main characteristics of the most used parametrizations to represent an orientation

Representation	**Parameters**	**Continuous**	**Non-Singular**	**Linear Evolution Equation**
Euler angles	3	✗	✗	✗
Axis-angle	3–4	✗	✗	✗
Rotation matrix	9	✓	✓	✓
Unit quaternion	4	✓	✓	✓

Knowing that the *special orthogonal group* SO(3) has dimension three, ideally we would seek for a continuous and non-singular representation expressed by 3 parameters. However, since 1964 we know that "...it is topologically impossible to have a global 3-dimensional parametrization without singular points for the rotation group" [9]. Knowing this, we would not be wrong to say that unit quaternions are the most convenient representation we have, and that we will have for orientations.

Quaternions are 4-dimensional entities, but only those having unit norm represent a rotation transformation. This fact implies a problem in applying the ordinary Kalman Filter, so different approaches have emerged. Since a quaternion is of dimension 4, one tends to think at first on a 4×4 covariance matrix, and in the direct application of the Kalman Filter [10]. Given that all predictions are contained in the surface defined by the unit constraint, the quaternion distribution shrinks in the orthogonal direction to this surface, which leads to a singular covariance matrix after several updates.

Another perspective was firstly approached in [11], where the literature on attitude estimation is reviewed until 1982, when other parametrizations such as Euler angles were common. It founds the basis of modern quaternion-based attitude estimation, on which this work is supported. The

key idea lies in the definition of an "error-quaternion" that is transformed to a 3-vector. This vector is used to construct the covariance matrix, and one refers to a "3×3 representation of the quaternion covariance matrix". After that work, many others have explored this viewpoint, and have demonstrated its superiority. In [12], the relationship between the error-quaternion and the 3-vector is clarified, and the approach is called "Multiplicative Extended Kalman Filter". A second-order filter is also developed. In [13], they explore a general transformation from error-quaternion to 3-vector under the UKF framework. The developed algorithm is evaluated in simulations using a realistic spacecraft model. In [14] the "Multiplicative Extended Kalman Filter" is compared with the "Additive Extended Kalman Filter", and it is concluded that no valid reason is seen to prefer the second to the first. Similarly, in [15] the "Multiplicative Extended Kalman Filter" is compared to several approaches in which attitude is estimated from unconstrained quaternion estimations, and the same conclusions are reached. In [16] an algorithm based on the UKF is developed to estimate both the attitude state of the system and its parameters. The algorithm is compared with another based on the EKF through simulations. In [17] an algorithm equivalent to the "Multiplicative Extended Kalman Filter" is derived to estimate the orientation of a spacecraft using measurements of an IMU and a sun-sensor. It also reviews the derivations, properties and relationships between the mathematical objects that appear in the algorithm. In [18] an algorithm clearly inspired by the "Multiplicative Extended Kalman Filter" is presented to estimate the orientation of a Miniature Air Vehicle (MAV) using an IMU and a GPS. An algorithm is presented in [19] that uses ideas comparable to those of the "Multiplicative Extended Kalman Filter" to estimate the pose of a vehicle using an IMU and a camera. The algorithm is tested in simulations and in real experiments. The former was the forerunner of many other successful works based on the same principles [20, 21, 22, 23, 24, 25, 26].

Although the "Multiplicative Extended Kalman Filter" has been and is widely used to estimate orientation, there are still details in this adaptation that are currently being developed. Namely, the "covariance correction step" [27].

1.2.2. Calibration of IMUs

A substantial effort has been devoted to the development of calibration algorithms for triaxial sensors and, as a result, several strategies have emerged. When it comes to the calibration of an accelerometer, it is common to exploit the fact that the magnitude of the gravity vector should be independent of the orientation of the sensor [28, 29, 30, 31]. Other works accommodate this idea to calibrate magnetometers [32, 33, 34, 35]. The advantage of this approach is that we do not need to know the actual orientation of the sensor to perform the calibration; only the magnitude of the measured vector is required. Naturally, there are works in which this idea is not used [36].

To calibrate a sensor it is necessary to choose a model. The model will describe the behavior of our sensor; how the measurements relate to the physical quantities we want to measure. It is common to use a linear model for the sensor [28, 29, 30, 32, 33, 34, 31, 35, 36]. But there are works in which non-linear models have been proposed [37]. In [38] even a neural network model is used. However, the use of such a complex model as a neural network would imply a computational overload probably greater than that of an estimation algorithm. A computational overload of that magnitude to obtain the calibrated measurements would be disproportionate.

Calibrations have been performed using both online and offline algorithms. Online calibration algorithms have the advantage of not having to perform a calibration before using the sensor. In opposition, they entail a greater online computational overload, and a greater complexity of the estimation algorithm, what usually implies the simplification of the sensor model. Reference [12] is an example of an online calibration algorithm in which only gyroscope biases are calibrated. On the other hand, offline calibration have to be performed before using the sensor, but they allow the calibration for complex sensor models, and after the calibration the corrected measurements are obtained very efficiently. Thus, this second approach is the preferred choice.

Offline calibration algorithms are usually methods for finding the solution to an optimization problem, which generally consists on the minimization of a cost function. Of course, the strategy of minimizing a cost function is not always used. In [36] they use an offline Kalman filter to find the model parameters.

There are sensors whose measurements are highly dependent on temperature, as is the case shown in Figure 1.1. Still, none of the works mentioned above considers a temperature-dependent calibration. To the best knowledge of the author, there are few works that address this problem.

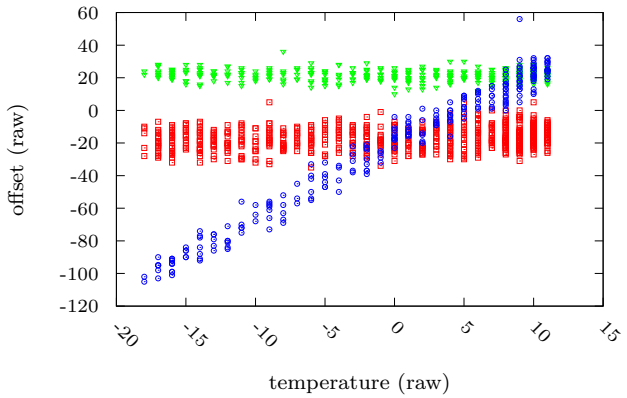


Figure 1.1: Example of a sensor whose measurements strongly depend on temperature. It is the gyroscope contained in the chip L3GD20H (Adafruit 9-DOF). x axis measurements are red squares. y axis measurements are green triangles. z axis measurements are blue circles. The measurements were collected by varying the temperature of a sensor that remained static.

A good example is [39], where a temperature-dependent non-linear calibration of a Fiber-Optic Gyroscope (FOG) is performed. Yet, it is not a calibration of a triaxial sensor.

1.3. Objectives

Once the problems have been stated, and the scientific literature has been reviewed, we can proceed to specify the main objectives of this work:

Orientation estimation: approach the problem of estimating the orientation of a system using the Kalman Filter formalism, and quaternions to describe the rotation transformation that represents an orientation. Formulate the measurement model to be compatible with IMU measurements.

Calibration of IMUs: design a calibration algorithm for triaxial sensors that considers their temperature dependence. Use it to calibrate real IMUs.

The following subsections dissect these main objectives into more specific ones.

1.3.1. Orientation Estimation

This subsection establishes the specific objectives related to the estimation of the orientation:

Learn about the Kalman Filter
As stated above, "the Kalman Filter has become the algorithm par excellence" in the field of estimation. Many state of the art works on orientation estimation choose the Kalman filter as the framework for their estimation algorithms. Then, it seems appropriate to put effort into understanding the underlying mechanisms that make this estimator work.

Looking back now, the author believes that this decision was correct, since he cannot imagine another algorithm that would have been more convenient to address the problem of orientation estimation than the Kalman filter.

Learn about Quaternions
Like the Kalman Filter, quaternions have gained a lot of popularity in applications where orientation is involved. Being so present in the literature, it seems reasonable to study how these hypercomplex numbers are capable of representing three-dimensional rotations.

At first, the author was reluctant to study quaternions, as he believed they were some kind of engineering device. He could not be more wrong. After knowing its definition, glimpsing its properties, and understanding the relationships with other mathematical objects, the author fell in love with quaternions.

Study how the Kalman Filter and Quaternions are combined to estimate orientation
The Kalman filter is designed to estimate states described by random vectors. However, quaternions that describe rotations (unit quaternions) do not form a vector space. It is necessary to understand how it has been possible to estimate orientations described by unit quaternions using the Kalman filter.

Design orientation estimation algorithms
Once we are able to understand how to estimate orientations represented by unit quaternions using the Kalman filter, we will be able to design our own estimation algorithms.

Try to make progress regarding the "covariance correction step"
The "covariance correction step" is a computation located at the end of each iteration of the Kalman Filter when it comes to estimating orientations described by unit quaternions. There are divided opinions on this aspect. The first developers of this approach [12, 13, 4] believed that this computation was not necessary, so at the end of each iteration a "reset step" was performed. On the other hand, works were published [40, 41, 27] claiming that such computation was necessary.

Generate code to estimate the orientation and test it with real IMUs
Having designed orientation estimation algorithms, we can generate their code. That code can be used to test the algorithms with real IMUs. To do this, one must be able to read IMU measurements. It is also necessary to be able to represent human-friendly orientations; for example, to represent a rigid body rotated using the quaternion that results from each algorithm.

1.3.2. Calibration of IMUs

This subsection establishes the specific objectives concerning the calibration of IMUs:

Design a calibration algorithm for triaxial sensors, and generate its code
The orientation estimations provided by the estimation algorithms will worsen due to the imperfections of the measurements provided by real IMUs. A calibration would reduce this effect. For this purpose, we will design a calibration algorithm for triaxial sensors. In particular, an attempt will be made to design a calibration algorithm that takes into account the temperature of the sensor to correct the measurements. We will rely on the concept previously mentioned: the magnitude of the gravity vector should be independent of the orientation of the sensor.

Test the calibration algorithm with real triaxial sensors
Once the calibration algorithm is designed, it will be tested with real IMUs. To perform the calibration, it will be necessary to collect IMU measurements. If we have been able to design a temperature-dependent calibration algorithm, we must engineer a method to take measurements of the IMUs at different temperatures.

Compare the estimations produced with and without calibrated IMUs
Having calibrated some IMU, we will be able to compare the orientations produced with an algorithm fed with calibrated measurements, with the orientations produced by one fed with uncalibrated measurements.

Perform inertial navigation with IMUs
At this point we would have orientation estimation algorithms, and calibrated IMUs. We would be able to transform acceleration measurements from the IMU reference frame to an external inertial frame of reference. Acceleration measurements expressed in an inertial reference frame could be integrated twice to obtain the estimated position of the IMU.

1.4. Contributions

In this section we will review the contributions made during the completion of this thesis.

1.4.1. Achievements Summary

In the previous section, the objectives to be completed during the thesis were proposed. These objectives were preliminary; they were established *a priori*, from an initial point of view. Instead, this section has a later character. This section summarizes the objectives that were completed, and the achievements and developments obtained as a result of this work.

New perspective on the problem of orientation estimation
Time was spent trying to understand how to estimate orientations described by unit quaternions using the Kalman filter. Although the scientific literature on the problem is highly appreciated (since without it I could never have reached this point), the way in which the algorithms were argued and developed was not satisfactory for the author. Finally, it was discovered that, after using some basic concepts imported from *manifold theory* to construct some definitions, orientation estimation algorithms developed naturally; without needing to redefine any aspect of the classical Kalman Filter. That new point of view –that perspective– is considered by the author the greatest contribution of this thesis.

Definition of the mean and covariance matrix of a distribution of unit quaternions
One of the key definitions, used to contemplate the perspective mentioned in the previous point, is that of the mean and the covariance matrix of a distribution of unit quaternions. These are defined using the concept of chart from manifold theory.

Design of orientation estimation algorithms
The new definitions mentioned above are useful for designing algorithms to estimate orientations described with unit quaternions using the Kalman filter. When considering these definitions, the algorithms are naturally conceived from the Kalman filter framework.

To generate the code for the algorithms, practical problems were solved. For example, we had to preserve the positive definiteness of covariance matrices.

Solution to the problem of the "covariance correction step"
The new definitions also lead us to a solution of the problem of the "covariance correction step" in a natural way. In addition, it can be verified that the solutions provided in other works such as [41, 27] turn out to be approximations of those provided in this thesis.

Generation of code to obtain IMU measurements
Although there was code written to read measurements of almost all the IMUs used during the development of this thesis [42], it was convenient to read the measurements of the IMU included in the SenseHAT of the Raspberry Pi[1] using the Processing[2] libraries. With this objective several classes were generated [43].

Design of a temperature-dependent calibration algorithm for triaxial sensors
It was possible to design a temperature-dependent calibration algorithm for triaxial sensors. To produce the calibration, the algorithm takes a set of vector measurements, each associated with a temperature measurement. The development of this algorithm represents an advance with respect to the reviewed scientific literature.

Design and construction of a prototype to automatically collect calibration data
To test the designed algorithm with real IMUs, it was necessary to collect calibration data while the temperature was varied. Since we wanted to calibrate multiple sensors, several times each, we decided to build an electromechanical system to automate the collection of calibration data.

Characterization of real triaxial sensors
Several sets of measurements were collected for different real IMUs. Each set of measurements was used to obtain a temperature-dependent calibration. The resulting set of calibrations was used to draw conclusions about the properties of real IMUs.

[1] `https://www.raspberrypi.org/` [Online; accessed January-2020]
[2] `https://processing.org/` [Online; accessed January-2020]

1.4.2. List of Publications

Many of the results presented in this thesis have been published in several peer-reviewed conferences and journals [44, 45, 46, 47, 48]. These publications are listed below:

- Journal of Physical Agents; SJR: 0.176 (2016), Q4 in "Control and Systems Engineering":

  P. Bernal-Polo and H. Martínez-Barberá, "Orientation estimation by means of extended kalman filter, quaternions, and charts," *Journal of Physical Agents*, vol. 8, no. 1, pp. 11–24, 2017.

- Conference ROBOT'2017:

  P. Bernal-Polo and H. Martínez-Barberá, "Triaxial sensor calibration: A prototype for accelerometer and gyroscope calibration," in *Iberian Robotics conference*, pp. 79–90, Springer, 2017.

- Workshop WAF'2018:

  P. Bernal-Polo and H. Martínez-Barberá, "First steps towards a general algorithm to estimate the mechanical state of a vehicle," in *Workshop of Physical Agents*, pp. 319–332, Springer, 2018.

- MDPI Sensors; JCR: 3.031 (2018), Q1 in "Instruments & Instrumentation":

  P. Bernal-Polo and H. Martínez-Barberá, "Kalman filtering for attitude estimation with quaternions and concepts from manifold theory," *Sensors*, vol. 19, no. 1, p. 149, 2019.

- IEEE Sensors Journal; JCR: 3.076 (2018), Q1 in "Instruments & Instrumentation":

  P. Bernal-Polo and H. Martínez-Barberá, "Temperature-dependent calibration of triaxial sensors: Algorithm, prototype, and some results," *IEEE Sensors Journal*, vol. 20, no. 2, pp. 876–884, 2019.

1.5. Contents Overview

The structure of the rest of this thesis is organized as follows:

Chapter 2: Kalman filtering. We begin by reviewing the concept of a random variable. We recall the definitions of mean and covariance of a random variable. Next, we introduce the basics of Kalman filtering. We also review the two most common Kalman filter extensions for nonlinear systems: the EKF and the UKF.

Chapter 3: Quaternions. We introduce quaternions, and we explore how these hypercomplex numbers are related to 3D rotations. We also present the definitions that we use to describe a distribution of unit quaternions. These definitions are essential for developing orientation estimation algorithms within the Kalman filter formalism.

Chapter 4: System Model. We define the model used to describe the system. We find both the evolution equations for the orientation of a rigid body, and the measurement equations for IMUs.

Chapter 5: Manifold Kalman Filters. We culminate in this chapter using all the concepts introduced in the previous chapters to develop two orientation estimation algorithms. We test the algorithms with real IMU measurements, and we use a simulation to compare both algorithms and to extract some properties about them.

Chapter 6: Triaxial Sensor Calibration. We address the problem of temperature-dependent calibration of triaxial sensors by designing a temperature-dependent calibration algorithm. We write about the prototype designed to collect calibration data from real IMUs. We also present the conclusions drawn from the calibrations performed with the data collected.

Chapter 7: Conclusion. We review the contributions of this thesis, and propose some potential future works.

Chapter 2

Kalman Filtering

2.1. Introduction

The Kalman filter is an algorithm that uses a series of successive measurements to produce an estimation of a variable related to those measurements. In particular, it approximates the probability density function of the variable that we want to estimate with its first two moments: its expected value, and its covariance matrix. It is a simple algorithm with a recursive nature, which allows its execution in real time. The filter was named after Rudolf Emil Kálmán; one of the primary developers of its theory [6].

The Kalman filter is usually broken down into two steps: the prediction step, and the update step. The author of this work prefers to decompose it into three steps: state prediction, measurement prediction, and state update. This allows to separate the unique dynamics of the system from the different equations that describe the measurements of different sensors. Such a structure makes the Kalman filter a convenient framework for sensor fusion algorithms.

In this chapter, we will review the basics of Kalman filtering. We will start reviewing the concept of random variable, and the definitions of mean and covariance in Section 2.2. Then, in Section 2.3, we will provide a derivation of the algorithm. Next, in Section 2.4, we will introduce the natural adaptation of the Kalman filter to non-linear systems: the Extended Kalman Filter (EKF). Finally, in Section 2.5, we will review another popular version of the Kalman filter that can be applied to non-linear systems: the Unscented Kalman Filter (UKF).

2.2. Random Variables

Definition 2.1 (Random Variable)**.** *Consider a random phenomenon with a set of possible outcomes* Ω*. A* random variable X *is a measurable function* $X : \Omega \to E$ *from the set of possible outcomes* Ω *to a measurable space* E*. The probability that* X *takes a value in a set* $S \subseteq E$ *is defined through a function* $P : E \to [0, 1]$*.*

Given a random variable $X : \Omega \to E$, and a function $g : E \to E'$, where E' is a measurable space, we can define a new random variable $Y : \Omega \to E'$, with $Y = g(X)$.

Example 2.1 (Random Variable)**.** The random phenomenon is a coin flip. The possible outcomes are heads (H) or tails (T), so that $\Omega = \{H, T\}$. The random variable X is the number of heads:

$$X(\omega) \quad = \quad \begin{cases} 1 & \text{if} \quad \omega = H \ , \\ 0 & \text{if} \quad \omega = T \ , \end{cases} \qquad \text{with} \quad \omega \in \Omega \quad . \tag{2.1}$$

This is the case of a discrete random variable. If the coin is fair, then, $P(H) = 0.5$ and $P(T) = 0.5$.

Example 2.2 (Random Variable)**.** The random phenomenon is the same as in Example 2.1. This time, if we obtain heads, we get 5 points; if we obtain tails we lose 5 points. The random variable X is our score:

$$X(\omega) \quad = \quad \begin{cases} +5 & \text{if} \quad \omega = H \ , \\ -5 & \text{if} \quad \omega = T \ , \end{cases} \qquad \text{with} \quad \omega \in \Omega \quad . \tag{2.2}$$

Example 2.3 (Random Variable)**.** The random phenomenon is the same as in Example 2.2. The random variable X_N is our score after N flips:

$$X_N(\omega) \quad = \quad X_{N-1} \; + \; X(\omega) \quad . \tag{2.3}$$

This is a case of a random variable X_N defined from other random variables.

Example 2.4 (Random Variable)**.** The random phenomenon is the movement of an object in a one-dimensional space. The possible outcomes are the possible accelerations experienced by the object, so that $\Omega = \mathbb{R}$. The random variable a_t is the acceleration of the object. Then, the random variable is defined through the identity function. This is the case of a continuous random variable. The probability of an acceleration falling within a particular range of values is defined through its probability density function f_{a_t}, which could be, for example, a normal distribution.

Example 2.5 (Random Variable)**.** The random phenomenon is the same as in Example 2.4, but this time there is no acceleration. The possible outcomes are the possible combinations of position and velocity experienced by the object: $\Omega = \mathbb{R}^2$. The random variable $\mathbf{x}_t = \big(x_t \, , \, v_t \big)^T$ is the state of the object at a time t. If we have information about the state of the object at a previous time $t - \Delta t$, then we can define our random variable in a recursive way:

$$\frac{d}{dt} \begin{pmatrix} x_t \\ v_t \end{pmatrix} \quad = \quad \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_{t-\Delta t} \\ v_{t-\Delta t} \end{pmatrix} \quad . \tag{2.4}$$

The random variable has been defined from other random variables. If the relation $\mathbf{x}_t = \mathbf{g}(\, \mathbf{x}_{t-\Delta t} \,)$ is invertible (if $\mathbf{g}^{-1}$ exists), then the new probability density function $f_{\mathbf{x}_t}$ is defined through:

$$f_{\mathbf{x}_t}(\mathbf{x}) \quad = \quad f_{\mathbf{x}_{t-\Delta t}}\big(\, \mathbf{g}^{-1}(\mathbf{x}) \, \big) \left| \frac{d \, \mathbf{g}^{-1}(\mathbf{x})}{d\mathbf{x}} \right| \quad . \tag{2.5}$$

2.2.1. Moments of a Random Variable

The probability distribution of a random variable give us substantial information. It tells us the values that the random variable will most likely take. However, working directly with probability distributions is often cumbersome; take, for example, the case of equation (2.5) in Example 2.5. Due to the analytical nature of these recursive relations, it is difficult to code them (write a computer program). For this reason, we often use other means to encode the probabilistic information of a random variable. One of these means are the moments.

Definition 2.2 (Expected value)**.** *Let X be a random variable with probability density function f_X. Then, the* expected value *of a random variable $Y = g(X)$ is given by:*

$$\mathrm{E}[\, Y \,] \quad := \quad \int_{\mathbb{R}} g(x) \; f_X(x) \; dx \quad . \tag{2.6}$$

Definition 2.3 (Moments)**.** *The n-th* moment *of a random variable X with probability density function f_X about a value c is defined as the expected value of the random variable $Y = (X - c)^n$:*

$$\mu_n \quad := \quad \mathrm{E}[\, (X - c)^n \,] \quad = \tag{2.7a}$$

$$= \quad \int_{-\infty}^{\infty} (x - c)^n \; f_X(x) \; dx \quad . \tag{2.7b}$$

Note that the 0-th moment of any random variable X is the integral of its probability density function f_X, which is always 1.

Definition 2.4 (Mean)**.** *The* mean *of a random variable X with probability density function f_X is defined as its first moment around zero:*

$$\overline{X} \quad := \quad \mathrm{E}[\, X \,] \quad = \tag{2.8a}$$

$$= \quad \int_{-\infty}^{\infty} x \; f_X(x) \; dx \quad . \tag{2.8b}$$

The mean is a measure of the average value of the distribution.

Definition 2.5 (Variance)**.** *The* variance *of a random variable* X *with probability density function* f_X *is defined as its second moment around its mean:*

$$\operatorname{var}(X) := \mathrm{E}[(X-\overline{X})^2] = \tag{2.9a}$$

$$= \int_{-\infty}^{\infty} (x-\overline{X})^2 \, f_X(x) \, dx \quad . \tag{2.9b}$$

The variance is a measure of the dispersion of the distribution with respect to its average value (the mean).

Definition 2.6 (Covariance)**.** *The* covariance *between two random variables* X *and* Y *with joint probability density function* f_{XY} *is defined as:*

$$\operatorname{cov}(X,Y) := \mathrm{E}[(X-\overline{X})(Y-\overline{Y})] = \tag{2.10a}$$

$$= \iint_{-\infty}^{\infty} (x-\overline{X})\,(y-\overline{Y})\, f_{XY}(x,y) \, dx \, dy \quad . \tag{2.10b}$$

The covariance is a measure of the relationship between two random variables. A positive covariance means that when one of the random variables takes greater values, the other also tends to take its greater values. A negative covariance means that when one of the random variables takes greater values, the other tends to take its lower values. Then, the sign of the covariance indicates the tendency in the linear relationship between the random variables. A null covariance would indicate that such a relationship does not exist. In that case, we say that two random variables are independent, or are uncorrelated.

Definition 2.7 (Random vector)**.** *A* random vector *or* multivariate random variable*,* $\mathbf{X}$*, is a vector of random variables. A* random vector $\mathbf{X}$ *containing* N *random variables is itself a random variable with* $\Omega = \Omega_1 \oplus \Omega_2 \oplus \cdots \oplus \Omega_N$*, where* Ω_i *is the set of possible outcomes for the* i*-th random variable, and "*$\oplus$*" represents the direct sum of spaces.*

An example of a random vector is the random variable of Example 2.5.

Definition 2.8 (Mean of a random vector)**.** *The* mean *of a random vector* $\mathbf{X}$ *with probability density function* $f_{\mathbf{X}}$ *is defined as:*

$$\overline{\mathbf{X}} := \mathrm{E}[\mathbf{X}] = \tag{2.11a}$$

$$= \int_{\mathbb{R}^N} \mathbf{x} \, f_{\mathbf{X}}(\mathbf{x}) \, d\mathbf{x} \quad , \tag{2.11b}$$

where N *is the dimension of* $\mathbf{X}$*.*

Definition 2.9 (Covariance matrix of a random vector)**.** *The* covariance matrix *of a random vector* $\mathbf{X}$ *with probability density function* $f_{\mathbf{X}}$ *is defined as:*

$$\operatorname{cov}(\mathbf{X}) := \mathrm{E}[(\mathbf{X}-\overline{\mathbf{X}})(\mathbf{X}-\overline{\mathbf{X}})^T] = \tag{2.12a}$$

$$= \int_{\mathbb{R}^N} (\mathbf{x}-\overline{\mathbf{X}})(\mathbf{x}-\overline{\mathbf{X}})^T \, f_{\mathbf{X}}(\mathbf{x}) \, d\mathbf{x} \quad . \tag{2.12b}$$

The covariance matrix of a random vector is always a positive-semidefinite matrix:

$$\mathbf{v}^T \operatorname{cov}(\mathbf{X})\, \mathbf{v} \;\geq\; 0 \quad , \quad \forall \;\; \mathbf{v} \in \mathbb{R}^N \quad . \tag{2.13}$$

Definition 2.10 (Cross-covariance matrix between two random vectors)**.** *The* cross-covariance matrix *between two random vectors* $\mathbf{X}$ *and* $\mathbf{Y}$ *with joint probability density function* $f_{\mathbf{XY}}$ *is defined as:*

$$\operatorname{cov}(\mathbf{X},\mathbf{Y}) := \mathrm{E}[(\mathbf{X}-\overline{\mathbf{X}})(\mathbf{Y}-\overline{\mathbf{Y}})^T] = \tag{2.14a}$$

$$= \int_{\mathbb{R}^N} \int_{\mathbb{R}^M} (\mathbf{x}-\overline{\mathbf{X}})(\mathbf{y}-\overline{\mathbf{Y}})^T \, f_{\mathbf{XY}}(\mathbf{x},\mathbf{y}) \, d\mathbf{y} \, d\mathbf{x} \quad , \tag{2.14b}$$

where N *is the dimension of* $\mathbf{X}$*, and* M *is the dimension of* $\mathbf{Y}$*.*

These latter definitions are of special interest. The mean and the covariance matrix of a random vector provide a large amount of information. The mean is the average value of the random vector. On the other hand, the covariance matrix contains information about the variance of each scalar random variable contained in the random vector (the diagonal elements of the matrix), and about the relationship between the components: their covariance (the off-diagonal elements of the matrix).

2.2.2. Propagation of the Mean and the Covariance Matrix

The concepts of mean and covariance matrix are not only interesting because of the information they provide about a random vector. They also represent an advantage when characterizing random vectors defined from other random vectors.

2.2.2.1. Random Vector Function of a Single Random Vector

Let $\mathbf{X}$ be a random vector, and let $\mathbf{Y} = \mathbf{g}(\mathbf{X})$ be another random vector defined from the first one. If $\mathbf{g}$ is sufficiently differentiable, then the random vector $\mathbf{Y}$ can be approximated using a Taylor expansion at a point $\mathbf{x}$ as

$$Y_i \approx g_i(\mathbf{x}) + \sum_j \left.\frac{\partial\, g_i(\mathbf{X})}{\partial\, X_j}\right|_{\mathbf{X}=\mathbf{x}} \left(X_j - x_j\right) \equiv \tag{2.15a}$$

$$\equiv \mathbf{g}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\,(\mathbf{X}-\mathbf{x}) \quad , \tag{2.15b}$$

where $\mathbf{G}(\mathbf{x})$ is the *Jacobian matrix* of the function $\mathbf{g}$ evaluated at $\mathbf{X} = \mathbf{x}$. In particular, if we choose $\mathbf{x}$ to be the mean of the random vector $\mathbf{X}$, $\overline{\mathbf{X}}$, then,

$$\mathbf{Y} \approx \mathbf{g}(\overline{\mathbf{X}}) + \mathbf{G}(\overline{\mathbf{X}})\left(\mathbf{X}-\overline{\mathbf{X}}\right) \quad . \tag{2.16}$$

In such a case, the mean of the random vector $\mathbf{Y}$ can be approximated as

$$\overline{\mathbf{Y}} = \mathrm{E}[\mathbf{Y}] \approx \tag{2.17a}$$

$$\approx \mathrm{E}\left[\,\mathbf{g}(\overline{\mathbf{X}}) + \mathbf{G}(\overline{\mathbf{X}})\left(\mathbf{X}-\overline{\mathbf{X}}\right)\right] = \tag{2.17b}$$

$$= \mathbf{g}(\overline{\mathbf{X}}) + \mathbf{G}(\overline{\mathbf{X}})\left(\mathrm{E}[\mathbf{X}]-\overline{\mathbf{X}}\right) = \tag{2.17c}$$

$$= \mathbf{g}(\overline{\mathbf{X}}) + \mathbf{G}(\overline{\mathbf{X}})\left(\overline{\mathbf{X}}-\overline{\mathbf{X}}\right) = \tag{2.17d}$$

$$= \mathbf{g}(\overline{\mathbf{X}}) \quad . \tag{2.17e}$$

On the other hand, the covariance matrix of the random vector $\mathbf{Y}$ can be approximated as

$$\mathrm{cov}(\mathbf{Y}) = \mathrm{E}\left[(\mathbf{Y}-\overline{\mathbf{Y}})(\mathbf{Y}-\overline{\mathbf{Y}})^T\right] \approx \tag{2.18a}$$

$$\approx \mathrm{E}\left[\mathbf{G}(\overline{\mathbf{X}})\left(\mathbf{X}-\overline{\mathbf{X}}\right)\left(\mathbf{X}-\overline{\mathbf{X}}\right)^T\mathbf{G}^T(\overline{\mathbf{X}})\right] = \tag{2.18b}$$

$$= \mathbf{G}(\overline{\mathbf{X}})\,\mathrm{E}\left[\left(\mathbf{X}-\overline{\mathbf{X}}\right)\left(\mathbf{X}-\overline{\mathbf{X}}\right)^T\right]\mathbf{G}^T(\overline{\mathbf{X}}) = \tag{2.18c}$$

$$= \mathbf{G}(\overline{\mathbf{X}})\,\mathrm{cov}(\mathbf{X})\,\mathbf{G}^T(\overline{\mathbf{X}}) \quad . \tag{2.18d}$$

Notice the simplicity of these expressions compared to (2.5). We can obtain the mean and the covariance matrix of the random vector $\mathbf{Y}$ from the mean and the covariance matrix of the random vector $\mathbf{X}$ using only function evaluations and matrix multiplications.

2.2.2.2. Random Vector Function of Multiple Random Vectors

Finally, let us consider the case of a random vector defined from serveral random vectors. Let $\mathbf{X}$ and $\mathbf{Y}$ be random vectors. And let $\mathbf{Z} = \mathbf{g}(\mathbf{X},\mathbf{Y})$ be another random vector defined from the

first two. The expression equivalent to (2.16) would be

$$Z_i \approx g_i(\overline{\mathbf{X}}, \overline{\mathbf{Y}}) + \sum_j \left.\frac{\partial\, g_i(\mathbf{X},\mathbf{Y})}{\partial\, X_j}\right|_{\substack{\mathbf{X}=\overline{\mathbf{X}}\\ \mathbf{Y}=\overline{\mathbf{Y}}}} \left(X_j - \overline{X}_j\right) + + \sum_j \left.\frac{\partial\, g_i(\mathbf{X},\mathbf{Y})}{\partial\, Y_j}\right|_{\substack{\mathbf{X}=\overline{\mathbf{X}}\\ \mathbf{Y}=\overline{\mathbf{Y}}}} \left(Y_j - \overline{Y}_j\right) \equiv \quad (2.19a)$$

$$\equiv \mathbf{g}(\overline{\mathbf{X}}, \overline{\mathbf{Y}}) + \mathbf{G}_\mathbf{X}\left(\mathbf{X} - \overline{\mathbf{X}}\right) + \mathbf{G}_\mathbf{Y}\left(\mathbf{Y} - \overline{\mathbf{Y}}\right) \quad , \quad (2.19b)$$

where it has been obviated that $\mathbf{G}_\mathbf{X}$ and $\mathbf{G}_\mathbf{Y}$ are evaluated at $(\overline{\mathbf{X}}, \overline{\mathbf{Y}})$. Then, the mean of the random vector $\mathbf{Z}$ can be approximated as

$$\overline{\mathbf{Z}} \approx \mathbf{g}(\overline{\mathbf{X}}, \overline{\mathbf{Y}}) \quad , \quad (2.20)$$

and its covariance matrix can be approximated as

$$\begin{aligned}\mathrm{cov}(\mathbf{Z}) \approx\ & \mathbf{G}_\mathbf{X}\, \mathrm{cov}(\mathbf{X})\, \mathbf{G}_\mathbf{X}^T + \mathbf{G}_\mathbf{Y}\, \mathrm{cov}(\mathbf{Y})\, \mathbf{G}_\mathbf{Y}^T + \\ & + \mathbf{G}_\mathbf{X}\, \mathrm{cov}(\mathbf{X}, \mathbf{Y})\, \mathbf{G}_\mathbf{Y}^T + \mathbf{G}_\mathbf{Y}\, \mathrm{cov}(\mathbf{Y}, \mathbf{X})\, \mathbf{G}_\mathbf{X}^T \quad . \end{aligned} \quad (2.21)$$

Note that these approximations are equivalent to (2.17) and (2.18) if we consider the random vector $\mathbf{X}' = (\mathbf{X}, \mathbf{Y})$ with covariance matrix

$$\mathrm{cov}(\mathbf{X}') = \begin{pmatrix} \mathrm{cov}(\mathbf{X}) & \mathrm{cov}(\mathbf{X}, \mathbf{Y}) \\ \mathrm{cov}(\mathbf{Y}, \mathbf{X}) & \mathrm{cov}(\mathbf{Y}) \end{pmatrix} \quad . \quad (2.22)$$

The random vector $\mathbf{Y}' = \mathbf{g}(\mathbf{X}')$ would be approximated using the Jacobian matrix $\mathbf{G} = \begin{pmatrix} \mathbf{G}_\mathbf{X} & \mathbf{G}_\mathbf{Y} \end{pmatrix}$. However, equation (2.21) will be useful when $\mathbf{X}$ and $\mathbf{Y}$ are independent random vectors.

2.3. The Kalman Filter

2.3.1. Kalman Filter Conception

The state of a system at a time t_n is described by a random vector $\mathbf{x}_n$. We assume that the random vector $\mathbf{x}_n$ satisfies the evolution equation

$$\mathbf{x}_n = \mathbf{F}_n\, \mathbf{x}_{n\text{-}1} + \mathbf{q}_n \quad , \quad (2.23)$$

where $\mathbf{F}_n$ is the matrix that describes the state transition, and $\mathbf{q}_n$ is a random vector with mean $\overline{\mathbf{q}}_n$ and covariance matrix $\mathbf{Q}_n$ that represents the process noise. We also assume the independence between $\mathbf{q}_n$ and $\mathbf{x}_{n\text{-}1}$.

At time t_n a measurement $\mathbf{z}_n$ is made. We assume that the measurement $\mathbf{z}_n$ is related to the state of the system $\mathbf{x}_n$ through

$$\mathbf{z}_n = \mathbf{H}_n\, \mathbf{x}_n + \mathbf{r}_n \quad , \quad (2.24)$$

where $\mathbf{H}_n$ is the matrix that describes the measurement model, and $\mathbf{r}_n$ is a random vector with mean $\overline{\mathbf{r}}_n$ and covariance matrix $\mathbf{R}_n$ that represents the measurement noise. We also assume the independence between $\mathbf{r}_n$ and $\mathbf{x}_n$.

We want to estimate the random vector $\mathbf{x}_n$ using the measurement $\mathbf{z}_n$, and the information at a previous time $t_{n\text{-}1}$ about the random vector $\mathbf{x}_{n\text{-}1}$. To that end, we define the random vector $\mathbf{x}_{n|m}$ through the following recursive linear relation:

$$\mathbf{x}_{n|n} = \mathbf{K}'_n\, \mathbf{x}_{n|n\text{-}1} + \mathbf{K}_n\, \mathbf{z}_n \quad . \quad (2.25)$$

We want $\mathbf{x}_{n|m}$ to be an estimation of the random vector $\mathbf{x}_n$ having considered the information provided by all the measurements up to $\mathbf{z}_m$. Then, $\mathbf{x}_{n|m}$ must also satisfy its own version of the prediction equation (2.23), and of the measurement equation (2.24). Matrices $\mathbf{K}'_n$ and $\mathbf{K}_n$ will be determined by the properties we impose on the random vector $\mathbf{x}_{n|m}$.

2.3.2. Prediction

Given the mean $\overline{\mathbf{x}}_{n\text{-}1|n\text{-}1}$ and the covariance matrix $\mathbf{P}_{n\text{-}1|n\text{-}1} := \mathrm{cov}(\,\mathbf{x}_{n\text{-}1|n\text{-}1}\,)$ of the state estimation $\mathbf{x}_{n\text{-}1|n\text{-}1}$ at time $t_{n\text{-}1}$, the Kalman filter uses the evolution equation (2.23) and the measurement equation (2.24) to predict the state of the system and the measurement at time t_n.

2.3.2.1. Prediction of the State

The random vector $\mathbf{x}_{n|n\text{-}1}$ is known as the state prediction at time t_n. It is defined from equation (2.23), and aims to be an estimation of the state at time t_n having considered all the measurements up to $\mathbf{z}_{n\text{-}1}$. Its mean $\overline{\mathbf{x}}_{n|n\text{-}1}$ is computed from equation (2.20), which turns out to be an equality because in our case $\mathbf{g}$ is a linear function:

$$\overline{\mathbf{x}}_{n|n\text{-}1} \quad = \quad \mathbf{F}_n\ \overline{\mathbf{x}}_{n\text{-}1|n\text{-}1} \ +\ \overline{\mathbf{q}}_n \quad . \tag{2.26}$$

Its covariance matrix $\mathbf{P}_{n|n\text{-}1} := \mathrm{cov}(\,\mathbf{x}_{n|n\text{-}1}\,)$ is computed from equation (2.21), which is also an equality for the same reason:

$$\mathbf{P}_{n|n\text{-}1} \quad = \quad \mathbf{F}_n\ \mathbf{P}_{n\text{-}1|n\text{-}1}\ \mathbf{F}_n^T \ +\ \mathbf{Q}_n \quad . \tag{2.27}$$

2.3.2.2. Prediction of the Measurement

The random vector $\mathbf{z}_{n|n\text{-}1}$ is known as the measurement prediction at time t_n. It is defined from equation (2.24), and aims to be an estimation of the measurement at time t_n having considered all the previous measurements up to $\mathbf{z}_{n\text{-}1}$. Its mean $\overline{\mathbf{z}}_{n|n\text{-}1}$ is computed from equation (2.20):

$$\overline{\mathbf{z}}_{n|n\text{-}1} \quad = \quad \mathbf{H}_n\ \overline{\mathbf{x}}_{n|n\text{-}1} \ +\ \overline{\mathbf{r}}_n \quad . \tag{2.28}$$

Its covariance matrix $\mathbf{S}_{n|n\text{-}1} := \mathrm{cov}(\,\mathbf{z}_{n|n\text{-}1}\,)$ is computed from equation (2.21):

$$\mathbf{S}_{n|n\text{-}1} \quad = \quad \mathbf{H}_n\ \mathbf{P}_{n|n\text{-}1}\ \mathbf{H}_n^T \ +\ \mathbf{R}_n \quad . \tag{2.29}$$

2.3.3. Unbiased Estimation

We want our estimation to be accurate. For this purpose, we impose the unbiased estimation property ($\overline{\mathbf{x}}_{n|n} = \mathrm{E}[\,\mathbf{x}_n\,]$):

$$\begin{aligned}
\mathrm{E}[\,\mathbf{x}_n\,] \quad &= \quad \mathrm{E}[\,\mathbf{x}_{n|n}\,] \quad = && \text{(2.30a)}\\
&= \quad \mathrm{E}[\,\mathbf{K}'_n\,\mathbf{x}_{n|n\text{-}1} \ +\ \mathbf{K}_n\,\mathbf{z}_n\,] \quad = && \text{(2.30b)}\\
&= \quad \mathbf{K}'_n\ \mathrm{E}[\,\mathbf{x}_{n|n\text{-}1}\,] \ +\ \mathbf{K}_n\ \mathrm{E}[\,\mathbf{H}_n\,\mathbf{x}_n \ +\ \mathbf{r}_n\,] \quad = && \text{(2.30c)}\\
&= \quad \mathbf{K}'_n\ \mathrm{E}[\,\mathbf{F}_n\,\mathbf{x}_{n\text{-}1|n\text{-}1} \ +\ \mathbf{q}_n\,] \ +\ \mathbf{K}_n\ \mathbf{H}_n\ \mathrm{E}[\,\mathbf{x}_n\,] \ +\ \mathbf{K}_n\ \overline{\mathbf{r}}_n \quad = && \text{(2.30d)}\\
&= \quad \mathbf{K}'_n\ \mathbf{F}_n\ \overline{\mathbf{x}}_{n\text{-}1|n\text{-}1} \ +\ \mathbf{K}'_n\ \overline{\mathbf{q}}_n \ +\ \mathbf{K}_n\ \mathbf{H}_n\ \mathrm{E}[\,\mathbf{x}_n\,] \ +\ \mathbf{K}_n\ \overline{\mathbf{r}}_n \quad . && \text{(2.30e)}
\end{aligned}$$

Assuming that the estimation was unbiased at time $t_{n\text{-}1}$ ($\overline{\mathbf{x}}_{n\text{-}1|n\text{-}1} = \mathrm{E}[\,\mathbf{x}_{n\text{-}1}\,]$),

$$\begin{aligned}
\mathrm{E}[\,\mathbf{x}_n\,] \quad &= \quad \mathbf{K}'_n\ \mathbf{F}_n\ \mathrm{E}[\,\mathbf{x}_{n\text{-}1}\,] \ +\ \mathbf{K}'_n\ \overline{\mathbf{q}}_n \ +\ \mathbf{K}_n\ \mathbf{H}_n\ \mathrm{E}[\,\mathbf{x}_n\,] \ +\ \mathbf{K}_n\ \overline{\mathbf{r}}_n \quad = && \text{(2.31a)}\\
&= \quad \mathbf{K}'_n\ \mathrm{E}[\,\mathbf{F}_n\,\mathbf{x}_{n\text{-}1} \ +\ \mathbf{q}_n\,] \ +\ \mathbf{K}_n\ \mathbf{H}_n\ \mathrm{E}[\,\mathbf{x}_n\,] \ +\ \mathbf{K}_n\ \overline{\mathbf{r}}_n \quad = && \text{(2.31b)}\\
&= \quad \mathbf{K}'_n\ \mathrm{E}[\,\mathbf{x}_n\,] \ +\ \mathbf{K}_n\ \mathbf{H}_n\ \mathrm{E}[\,\mathbf{x}_n\,] \ +\ \mathbf{K}_n\ \overline{\mathbf{r}}_n \quad = && \text{(2.31c)}\\
&= \quad \left(\,\mathbf{K}'_n \ +\ \mathbf{K}_n\,\mathbf{H}_n\,\right)\mathrm{E}[\,\mathbf{x}_n\,] \ +\ \mathbf{K}_n\ \overline{\mathbf{r}}_n \quad . && \text{(2.31d)}
\end{aligned}$$

Now, if we assume that the mean of the measurement noise is zero ($\overline{\mathbf{r}}_n = \mathbf{0}$), then

$$\mathbf{K}'_n \ +\ \mathbf{K}_n\,\mathbf{H}_n \quad = \quad \mathbf{I} \qquad \Rightarrow \qquad \mathbf{K}'_n \quad = \quad \mathbf{I} \ -\ \mathbf{K}_n\,\mathbf{H}_n \quad . \tag{2.32}$$

Therefore, the update equation (2.25) will have to take the form

$$\begin{aligned}
\mathbf{x}_{n|n} \quad &= \quad \left(\,\mathbf{I} \ -\ \mathbf{K}_n\,\mathbf{H}_n\,\right)\mathbf{x}_{n|n\text{-}1} \ +\ \mathbf{K}_n\,\mathbf{z}_n \quad = && \text{(2.33a)}\\
&= \quad \mathbf{x}_{n|n\text{-}1} \ +\ \mathbf{K}_n\,\left(\,\mathbf{z}_n \ -\ \mathbf{H}_n\,\mathbf{x}_{n|n\text{-}1}\,\right) \quad . && \text{(2.33b)}
\end{aligned}$$

2.3.3.1. Mean of the Estimation

The mean of the updated state $\overline{\mathbf{x}}_{n|n}$ will be

$$\begin{aligned} \mathrm{E}[\,\mathbf{x}_{n|n}\,] &= \mathrm{E}\big[\,\mathbf{x}_{n|n\text{-}1} + \mathbf{K}_n\,\big(\,\mathbf{z}_n - \mathbf{H}_n\,\mathbf{x}_{n|n\text{-}1}\,\big)\,\big] = && (2.34a)\\ &= \overline{\mathbf{x}}_{n|n\text{-}1} + \mathbf{K}_n\,\big(\,\overline{\mathbf{z}}_n - \mathbf{H}_n\,\overline{\mathbf{x}}_{n|n\text{-}1}\,\big) \quad . && (2.34b) \end{aligned}$$

Our best estimation of the mean of the measurement $\overline{\mathbf{z}}_n$ is the measurement itself $\mathbf{z}_n$. For that reason, we compute the mean of the updated state as

$$\overline{\mathbf{x}}_{n|n} \approx \overline{\mathbf{x}}_{n|n\text{-}1} + \mathbf{K}_n\,\big(\,\mathbf{z}_n - \mathbf{H}_n\,\overline{\mathbf{x}}_{n|n\text{-}1}\,\big) \quad . \tag{2.35}$$

2.3.3.2. Covariance Matrix of the Estimation

The covariance matrix of the updated state $\mathbf{P}_{n|n} := \mathrm{cov}(\,\mathbf{x}_{n|n}\,)$ will be

$$\begin{aligned} \mathbf{P}_{n|n} &= \mathrm{cov}\big(\,\mathbf{x}_{n|n\text{-}1} + \mathbf{K}_n\,(\,\mathbf{z}_n - \mathbf{H}_n\,\mathbf{x}_{n|n\text{-}1}\,)\,\big) = && (2.36a)\\ &= \mathrm{E}\big[\,\big(\mathbf{x}_{n|n\text{-}1} - \overline{\mathbf{x}}_{n|n\text{-}1} + \mathbf{K}_n\,(\mathbf{z}_n - \overline{\mathbf{z}}_n\,) + \\ &\qquad - \mathbf{K}_n\,\mathbf{H}_n\,(\mathbf{x}_{n|n\text{-}1} - \overline{\mathbf{x}}_{n|n\text{-}1})\big)\,(\star)^T\,\big] = && (2.36b)\\ &= \mathrm{E}\big[\,\big(\,(\,\mathbf{I} - \mathbf{K}_n\,\mathbf{H}_n\,)\,(\mathbf{x}_{n|n\text{-}1} - \overline{\mathbf{x}}_{n|n\text{-}1}) + \mathbf{K}_n\,\mathbf{r}_n\,\big)\,(\star)^T\,\big] = && (2.36c)\\ &= (\,\mathbf{I} - \mathbf{K}_n\,\mathbf{H}_n\,)\,\mathbf{P}_{n|n\text{-}1}\,(\,\mathbf{I} - \mathbf{K}_n\,\mathbf{H}_n\,)^T + \mathbf{K}_n\,\mathbf{R}_n\,\mathbf{K}_n^T \quad , && (2.36d) \end{aligned}$$

where "$\star$" represents the same expression on its left (to avoid rewriting). Recall that $\mathbf{x}_{n|n\text{-}1}$ and $\mathbf{r}_n$ are assumed to be independent.

2.3.4. Minimum Trace of the Covariance Matrix

We not only want our estimation to be accurate; we also want it to be precise. Recall that the variance of a distribution is a measure of its dispersion. Since the diagonal terms of the covariance matrix $\mathbf{P}_{n|n}$ are the variance of each component of the random vector $\mathbf{x}_{n|n}$, we want to find the matrix $\mathbf{K}_n$ that minimizes the trace of the covariance matrix:

$$\mathbf{K}_n = \arg\min\ \mathrm{tr}(\,\mathbf{P}_{n|n}\,) \quad . \tag{2.37}$$

Let us start by rewriting (2.36):

$$\begin{aligned} P_{ij} = \sum_{km} \Big(\delta_{ik} - \sum_{\ell} K_{i\ell}\,H_{\ell k}\Big)\,\widetilde{P}_{km}\,\Big(\delta_{jm} - \sum_{\ell} K_{j\ell}\,H_{\ell m}\Big) + \\ + \sum_{km} K_{ik}\,R_{km}\,K_{jm} \quad , \end{aligned} \tag{2.38}$$

where δ_{ik} is the Kronecker delta. The trace is

$$\begin{aligned} \sum_i P_{ii} = \sum_{ikm} \Big(\delta_{ik} - \sum_{\ell} K_{i\ell}\,H_{\ell k}\Big)\,\widetilde{P}_{km}\,\Big(\delta_{im} - \sum_{\ell} K_{i\ell}\,H_{\ell m}\Big) + \\ + \sum_{ikm} K_{ik}\,R_{km}\,K_{im} \quad . \end{aligned} \tag{2.39}$$

At the extrema of the trace we get

$$\frac{\partial \sum_i P_{ii}}{\partial K_{\alpha\beta}} = 0 \qquad \Longleftrightarrow \tag{2.40a}$$

$$\Longleftrightarrow \quad \sum_{ikm} \Big(-\sum_{\ell} \delta_{i\alpha}\,\delta_{\ell\beta}\,H_{\ell k} \Big)\,\widetilde{P}_{km}\,\Big(\delta_{im} - \sum_{\ell} K_{i\ell}\,H_{\ell m}\Big) + \tag{2.40b}$$
$$+ \sum_{ikm} \Big(\delta_{ik} - \sum_{\ell} K_{i\ell}\,H_{\ell k}\Big)\,\widetilde{P}_{km}\,\Big(-\sum_{\ell} \delta_{i\alpha}\,\delta_{\ell\beta}\,H_{\ell m}\Big) +$$
$$+ \sum_{ikm} \big(\,\delta_{i\alpha}\,\delta_{k\beta}\,R_{km}\,K_{im} + K_{ik}\,R_{km}\,\delta_{i\alpha}\,\delta_{m\beta}\,\big) = 0 \qquad \equiv$$

$$\equiv \quad \sum_{km} \big(-H_{\beta k}\big)\,\widetilde{P}_{km}\,\Big(\delta_{\alpha m} - \sum_{\ell} K_{\alpha\ell}\,H_{\ell m}\Big) + \tag{2.40c}$$
$$+ \sum_{km} \Big(\delta_{\alpha k} - \sum_{\ell} K_{\alpha\ell}\,H_{\ell k}\Big)\,\widetilde{P}_{km}\,\big(-H_{\beta m}\big) +$$
$$+ \sum_{m} R_{\beta m}\,K_{\alpha m} + \sum_{k} K_{\alpha k}\,R_{k\beta} = 0 \qquad \equiv$$

$$\equiv \quad \sum_{km} \big(-H_{\beta k}\big)\,\big(\widetilde{P}_{km} + \widetilde{P}_{mk}\big)\,\Big(\delta_{\alpha m} - \sum_{\ell} K_{\alpha\ell}\,H_{\ell m}\Big) + \tag{2.40d}$$
$$+ \sum_{m} \big(R_{\beta m} + R_{m\beta}\big)\,K_{\alpha m} = 0 \qquad \equiv$$

$$\equiv \quad \sum_{\ell} K_{\alpha\ell}\,\Big(\sum_{km} H_{\ell m}\,\big(\widetilde{P}_{km} + \widetilde{P}_{mk}\big)\,H_{\beta k} + \big(R_{\beta\ell} + R_{\ell\beta}\big)\Big) = \tag{2.40e}$$
$$= \sum_{k} \big(\widetilde{P}_{k\alpha} + \widetilde{P}_{\alpha k}\big)\,H_{\beta k} \qquad \equiv$$

$$\equiv \quad \mathbf{K}\,\Big(\mathbf{H}\,\big(\widetilde{\mathbf{P}}^T + \widetilde{\mathbf{P}}\big)\,\mathbf{H}^T + \big(\mathbf{R}^T + \mathbf{R}\big)\Big) = \big(\widetilde{\mathbf{P}}^T + \widetilde{\mathbf{P}}\big)\,\mathbf{H}^T \quad . \tag{2.40f}$$

In our particular case where $\mathbf{P}_{n|n\text{-}1}$ and $\mathbf{R}_n$ are symmetric matrices ($\mathbf{P}^T_{n|n\text{-}1} = \mathbf{P}_{n|n\text{-}1}$ and $\mathbf{R}^T_n = \mathbf{R}_n$) the extrema is found when

$$\mathbf{K}_n\,\big(\mathbf{H}_n\,\mathbf{P}_{n|n\text{-}1}\,\mathbf{H}^T_n + \mathbf{R}_n\big) = \mathbf{P}_{n|n\text{-}1}\,\mathbf{H}^T_n \qquad \equiv \tag{2.41a}$$

$$\equiv \quad \mathbf{K}_n\,\mathbf{S}_{n|n\text{-}1} = \mathbf{P}_{n|n\text{-}1}\,\mathbf{H}^T_n \quad . \tag{2.41b}$$

In addition, we know that (2.41) leads to a minimum because the tensor $\frac{\partial^2 \sum_i P_{ii}}{\partial K_{\alpha\beta}\,\partial K_{\mu\nu}}$ is positive-definite, what means that the quadratic form used to approximate[1] $\sum_i P_{ii}$ around the extremum is strictly positive, except at the point where the extremum is located. Let us check it:

$$\frac{\partial^2 \sum_i P_{ii}}{\partial K_{\alpha\beta}\,\partial K_{\mu\nu}} = \sum_{km\ell} H_{\beta k}\,\big(\widetilde{P}_{km} + \widetilde{P}_{mk}\big)\,\delta_{\alpha\mu}\,\delta_{\ell\nu}\,H_{\ell m} + \tag{2.42a}$$
$$+ \sum_{m} \big(R_{\beta m} + R_{m\beta}\big)\,\delta_{\alpha\mu}\,\delta_{m\nu} =$$

$$= \delta_{\alpha\mu}\,\Big(\sum_{km} H_{\beta k}\,\big(\widetilde{P}_{km} + \widetilde{P}_{mk}\big)\,H_{\nu m} + \tag{2.42b}$$
$$+ \big(R_{\beta\nu} + R_{\nu\beta}\big)\Big) \qquad \equiv$$

$$\equiv \quad \delta_{\alpha\mu}\,\Big(\mathbf{H}\,\big(\widetilde{\mathbf{P}} + \widetilde{\mathbf{P}}^T\big)\,\mathbf{H}^T + \mathbf{R} + \mathbf{R}^T\Big)_{\beta\nu} = \tag{2.42c}$$

$$= \quad 2\,\delta_{\alpha\mu}\,\Big(\mathbf{H}\,\widetilde{\mathbf{P}}\,\mathbf{H}^T + \mathbf{R}\Big)_{\beta\nu} \quad . \tag{2.42d}$$

[1] $f(\mathbf{K}) \approx f(\mathbf{A}) + \sum_{\alpha\beta} \frac{\partial f(\mathbf{K})}{\partial K_{\alpha\beta}}\Big|_{\mathbf{A}} (K_{\alpha\beta} - A_{\alpha\beta}) + \sum_{\alpha\beta\mu\nu} \frac{\partial f(\mathbf{K})}{\partial K_{\alpha\beta}\,\partial K_{\mu\nu}}\Big|_{\mathbf{A}} (K_{\alpha\beta} - A_{\alpha\beta})\,(K_{\mu\nu} - A_{\mu\nu})$

Then, the tensor $\frac{\partial^2 \sum_i P_{ii}}{\partial K_{\alpha\beta}\, \partial K_{\mu\nu}}$ is positive definite if

$$\sum_{\alpha\beta\mu\nu} \frac{\partial^2 \sum_i P_{ii}}{\partial K_{\alpha\beta}\, \partial K_{\mu\nu}}\, x_{\alpha\beta}\, x_{\mu\nu} \quad > \quad 0 \qquad \Longleftrightarrow \tag{2.43a}$$

$$\Longleftrightarrow \qquad \sum_{\alpha\beta\mu\nu} \delta_{\alpha\mu} \left(\mathbf{H}\, \widetilde{\mathbf{P}}\, \mathbf{H}^T \,+\, \mathbf{R} \right)_{\beta\nu} x_{\alpha\beta}\, x_{\mu\nu} \quad > \quad 0 \qquad \equiv \tag{2.43b}$$

$$\equiv \qquad \sum_{\alpha\beta\nu} x_{\alpha\beta} \left(\mathbf{H}\, \widetilde{\mathbf{P}}\, \mathbf{H}^T \,+\, \mathbf{R} \right)_{\beta\nu} x_{\alpha\nu} \quad > \quad 0 \qquad \equiv \tag{2.43c}$$

$$\equiv \qquad \sum_{\alpha} \mathbf{x}_\alpha \left(\mathbf{H}\, \widetilde{\mathbf{P}}\, \mathbf{H}^T \,+\, \mathbf{R} \right) \mathbf{x}_\alpha^T \quad > \quad 0 \quad . \tag{2.43d}$$

In our particular case, $\mathbf{P}_{n|n\text{-}1}$ and $\mathbf{R}_n$ are not only symmetric matrices but also positive-definite (they are covariance matrices). Then, $\mathbf{S} = \mathbf{H}\, \widetilde{\mathbf{P}}\, \mathbf{H}^T + \mathbf{R}$ is also positive-definite, and (2.43) is satisfied.

2.3.5. Kalman Filter Summary

At time t_n we receive a measurement $\mathbf{z}_n$. Our knowledge about the state of the system at a previous time $t_{n\text{-}1}$, is encoded in the mean $\overline{\mathbf{x}}_{n\text{-}1|n\text{-}1}$ and in the covariance matrix $\mathbf{P}_{n\text{-}1|n\text{-}1}$ of the random variable $\mathbf{x}_{n\text{-}1|n\text{-}1}$. We update our knowledge about the state of the system in three steps:

State prediction

$$\overline{\mathbf{x}}_{n|n\text{-}1} \quad = \quad \mathbf{F}_n\, \overline{\mathbf{x}}_{n\text{-}1|n\text{-}1} \,+\, \overline{\mathbf{q}}_n \quad , \tag{2.44}$$

$$\mathbf{P}_{n|n\text{-}1} \quad = \quad \mathbf{F}_n\, \mathbf{P}_{n\text{-}1|n\text{-}1}\, \mathbf{F}_n^T \,+\, \mathbf{Q}_n \quad . \tag{2.45}$$

Measurement prediction

$$\overline{\mathbf{z}}_{n|n\text{-}1} \quad = \quad \mathbf{H}_n\, \overline{\mathbf{x}}_{n|n\text{-}1} \quad , \tag{2.46}$$

$$\mathbf{S}_{n|n\text{-}1} \quad = \quad \mathbf{H}_n\, \mathbf{P}_{n|n\text{-}1}\, \mathbf{H}_n^T \,+\, \mathbf{R}_n \quad . \tag{2.47}$$

Kalman update

$$\mathbf{K}_n \quad = \quad \mathbf{P}_{n|n\text{-}1}\, \mathbf{H}_n^T\, \mathbf{S}_{n|n\text{-}1}^{-1} \quad , \tag{2.48}$$

$$\overline{\mathbf{x}}_{n|n} \quad = \quad \overline{\mathbf{x}}_{n|n\text{-}1} \,+\, \mathbf{K}_n \left(\mathbf{z}_n \,-\, \overline{\mathbf{z}}_{n|n\text{-}1} \right) \quad , \tag{2.49}$$

$$\mathbf{P}_{n|n} \quad = \quad (\, \mathbf{I} \,-\, \mathbf{K}_n\, \mathbf{H}_n \,)\, \mathbf{P}_{n|n\text{-}1}\, (\, \mathbf{I} \,-\, \mathbf{K}_n\, \mathbf{H}_n \,)^T \,+\, \mathbf{K}_n\, \mathbf{R}_n\, \mathbf{K}_n^T \quad , \tag{2.50}$$

Since $\mathbf{S}_{n|n\text{-}1}$ is a positive definite matrix, it can be Cholesky-decomposed, and the Kalman gain $\mathbf{K}_n$ in (2.48) can be obtained by solving a matrix equation as it is shown in Appendix A.2. To maintain the positive-definiteness of the covariance matrices, it is advisable to compute the matrix products as described in Appendix A.3.

Kalman filter assumptions

1. The evolution of the system can be modeled by the linear equation (2.23).
2. The measurement can be modeled by the linear equation (2.24).
3. Our estimation of the state of the system was unbiased at the previous time $t_{n\text{-}1}$ ($\overline{\mathbf{x}}_{n\text{-}1|n\text{-}1} = \mathrm{E}[\, \mathbf{x}_{n\text{-}1} \,]$).
4. The mean of the measurement noise is zero ($\overline{\mathbf{r}}_n = \mathbf{0}$).
5. The mean of the measurement $\overline{\mathbf{z}}_n$ can be approximated by the measurement $\mathbf{z}_n$.

2.4. Extended Kalman Filter

The Extended Kalman Filter (EKF) is a natural adaptation of the Kalman Filter for non-linear systems. In place of equations (2.23) and (2.24), we assume that the system can be modeled by

$$\mathbf{x}_n = \mathbf{f}(\,\mathbf{x}_{n\text{-}1}\,,\,\mathbf{q}_n\,) \quad , \tag{2.51}$$

$$\mathbf{z}_n = \mathbf{h}(\,\mathbf{x}_n\,,\,\mathbf{r}_n\,) \quad , \tag{2.52}$$

where $\mathbf{f}$ and $\mathbf{h}$ are non-linear, but sufficiently differentiable functions. The prediction equations for the state and for the measurement are derived from applying (2.20) and (2.21) to (2.51) and (2.52):

State prediction

$$\overline{\mathbf{x}}_{n|n\text{-}1} = \mathbf{f}(\,\overline{\mathbf{x}}_{n\text{-}1|n\text{-}1}\,,\,\overline{\mathbf{q}}_n\,) \quad , \tag{2.53}$$

$$\mathbf{F}_n = \left.\frac{\partial\,\mathbf{f}(\,\mathbf{x}\,,\,\mathbf{q}\,)}{\partial\mathbf{x}}\right|_{\substack{\mathbf{x}\,=\,\overline{\mathbf{x}}_{n\text{-}1|n\text{-}1}\\ \mathbf{q}\,=\,\overline{\mathbf{q}}_n}} \quad , \tag{2.54}$$

$$\mathbf{L}_n = \left.\frac{\partial\,\mathbf{f}(\,\mathbf{x}\,,\,\mathbf{q}\,)}{\partial\mathbf{q}}\right|_{\substack{\mathbf{x}\,=\,\overline{\mathbf{x}}_{n\text{-}1|n\text{-}1}\\ \mathbf{q}\,=\,\overline{\mathbf{q}}_n}} \quad , \tag{2.55}$$

$$\mathbf{P}_{n|n\text{-}1} = \mathbf{F}_n\;\mathbf{P}_{n\text{-}1|n\text{-}1}\;\mathbf{F}_n^T \;+\; \mathbf{L}_n\;\mathbf{Q}_n\;\mathbf{L}_n^T \quad . \tag{2.56}$$

Measurement prediction

$$\overline{\mathbf{z}}_{n|n\text{-}1} = \mathbf{h}(\,\overline{\mathbf{x}}_{n|n\text{-}1}\,,\,\mathbf{0}\,) \quad , \tag{2.57}$$

$$\mathbf{H}_n = \left.\frac{\partial\,\mathbf{h}(\,\mathbf{x}\,,\,\mathbf{r}\,)}{\partial\mathbf{x}}\right|_{\substack{\mathbf{x}\,=\,\overline{\mathbf{x}}_{n|n\text{-}1}\\ \mathbf{r}\,=\,\mathbf{0}}} \quad , \tag{2.58}$$

$$\mathbf{M}_n = \left.\frac{\partial\,\mathbf{h}(\,\mathbf{x}\,,\,\mathbf{r}\,)}{\partial\mathbf{r}}\right|_{\substack{\mathbf{x}\,=\,\overline{\mathbf{x}}_{n|n\text{-}1}\\ \mathbf{r}\,=\,\mathbf{0}}} \quad , \tag{2.59}$$

$$\mathbf{S}_{n|n\text{-}1} = \mathbf{H}_n\;\mathbf{P}_{n|n\text{-}1}\;\mathbf{H}_n^T \;+\; \mathbf{M}_n\;\mathbf{R}_n\;\mathbf{M}_n^T \quad . \tag{2.60}$$

We assume that our non-linear system can be described locally as a linear system, so we can apply the Kalman update (equations (2.48), (2.49), and (2.50)).

2.4.1. Continuous-Discrete Extended Kalman Filter

The evolution of most real systems is described using differential equations. On the other hand, the digital systems we use to process information receive measurements at discrete times. This leads us to consider non-linear systems modeled by

$$\frac{d\,\mathbf{x}(t)}{d\,t} = \mathbf{f}\big(\,\mathbf{x}(t)\,,\,\mathbf{w}(t)\,\big) \quad , \tag{2.61}$$

$$\mathbf{z}_n = \mathbf{h}\big(\,\mathbf{x}(t_n)\,,\,\mathbf{r}_n\,\big) \quad , \tag{2.62}$$

being $\mathbf{w}(t)$ a random variable with mean $\overline{\mathbf{w}}(t)$ and whose covariance matrix fulfills

$$\mathrm{E}\Big[\,\big(\,\mathbf{w}(\tau)-\overline{\mathbf{w}}(\tau)\,\big)\big(\,\mathbf{w}(\tau')-\overline{\mathbf{w}}(\tau')\,\big)^T\,\Big] = \mathbf{Q}(t)\;\delta(\tau-\tau') \quad , \tag{2.63}$$

where $\delta(t)$ is the *Dirac delta function*.

Since $\mathbf{x}$ and $\mathbf{w}$ are random variables, equation (2.61) is a *Stochastic Differential Equation* (SDE). These kind of differential equations are generally unsolvable. Currently only some particular cases can be solved using *Itô calculus*. Hence, an approximation of the differential equation is usually solved.

2.4.1.1. Approximate Differential Equation for the Mean

Taking the first terms in the Taylor series of $\mathbf{f}$ around the means $\overline{\mathbf{x}}(t)$ and $\overline{\mathbf{w}}(t)$, equation (2.61) can be approximated as

$$\begin{aligned} \frac{d\,\mathbf{x}(t)}{d\,t} \quad \approx \quad & \mathbf{f}\big(\,\overline{\mathbf{x}}(t)\,,\,\overline{\mathbf{q}}(t)\,\big) \;+\; \mathbf{F}(t)\,\big(\mathbf{x}(t) - \overline{\mathbf{x}}(t)\big) \;+ \\ & +\; \mathbf{L}(t)\,\big(\mathbf{w}(t) - \overline{\mathbf{w}}(t)\big) \quad , \end{aligned} \tag{2.64}$$

with $\mathbf{F}(t) = \frac{\partial\,\mathbf{f}}{\partial\,\mathbf{x}}\big(\,\overline{\mathbf{x}}(t)\,,\,\overline{\mathbf{w}}(t)\,\big)$ and $\mathbf{L}(t) = \frac{\partial\,\mathbf{f}}{\partial\,\mathbf{w}}\big(\,\overline{\mathbf{x}}(t)\,,\,\overline{\mathbf{w}}(t)\,\big)$. Taking expected values in (2.64) we obtain an approximate differential equation for the mean:

$$\frac{d\,\overline{\mathbf{x}}(t)}{d\,t} \quad \approx \quad \mathbf{f}\big(\,\overline{\mathbf{x}}(t)\,,\,\overline{\mathbf{w}}(t)\,\big) \quad . \tag{2.65}$$

2.4.1.2. Approximate Differential Equation for the Covariance Matrix

In order to obtain an approximate differential equation for the covariance matrix $\mathbf{P}(t)$, we first need to define a new random variable as

$$\mathbf{q}(t) \quad = \quad \mathbf{q}(t_0) \;+\; \int_{t_0}^{t} \mathbf{w}(\tau)\;d\tau \quad . \tag{2.66}$$

Then, subtracting (2.65) from (2.64) gives:

$$\frac{d\,\big(\mathbf{x}(t) - \overline{\mathbf{x}}(t)\big)}{d\,t} \quad \approx \quad \mathbf{F}(t)\,\big(\mathbf{x}(t) - \overline{\mathbf{x}}(t)\big) \;+\; \mathbf{L}(t)\,\big(\mathbf{w}(t) - \overline{\mathbf{w}}(t)\big) \quad = \tag{2.67a}$$

$$= \quad \mathbf{F}(t)\,\big(\mathbf{x}(t) - \overline{\mathbf{x}}(t)\big) \;+\; \mathbf{L}(t)\;\frac{d\,\big(\mathbf{q}(t) - \overline{\mathbf{q}}(t)\big)}{d\,t} \quad . \tag{2.67b}$$

From equation (2.67) we obtain an approximate evolution equation for $\mathbf{x}(t) - \overline{\mathbf{x}}(t)$:

$$\begin{aligned} \Delta\mathbf{x}(t+\Delta t) \quad \approx \quad & \Delta\mathbf{x}(t) \;+\; \mathbf{F}(t)\;\Delta\mathbf{x}(t)\;\Delta t \;+ \\ & +\; \mathbf{L}(t)\,\big(\Delta\mathbf{q}(t+\Delta t) - \Delta\mathbf{q}(t)\big) \quad = \end{aligned} \tag{2.68a}$$

$$\begin{aligned} = \quad & \big(\mathbf{I} + \mathbf{F}(t)\,\Delta t\big)\;\Delta\mathbf{x}(t) \;+ \\ & +\; \mathbf{L}(t)\,\big(\Delta\mathbf{q}(t+\Delta t) - \Delta\mathbf{q}(t)\big) \quad , \end{aligned} \tag{2.68b}$$

where we have denoted $\Delta\mathbf{x}(t) = \mathbf{x}(t) - \overline{\mathbf{x}}(t)$ and $\Delta\mathbf{q}(t) = \mathbf{q}(t) - \overline{\mathbf{q}}(t)$. We should notice now that

$$\mathrm{E}\Big[\,\big(\Delta\mathbf{q}(t+\Delta t) - \Delta\mathbf{q}(t)\big)\,(\star)^T\,\Big] \quad = \tag{2.69a}$$

$$= \quad \mathrm{E}\Big[\,\Big(\mathbf{q}(t+\Delta t) - \overline{\mathbf{q}}(t+\Delta t) \;-\; \big(\mathbf{q}(t) - \overline{\mathbf{q}}(t)\big)\Big)(\star)^T\,\Big] \quad = \tag{2.69b}$$

$$= \quad \mathrm{cov}\big(\,\mathbf{q}(t+\Delta t) - \mathbf{q}(t)\,\big) \quad = \tag{2.69c}$$

$$= \quad \int_t^{t+\Delta t} d\tau \;\int_t^{t+\Delta t} d\tau'\;\mathrm{E}[\,\Delta\mathbf{w}(\tau)\;\Delta\mathbf{w}^T(\tau')\,] \quad = \tag{2.69d}$$

$$= \quad \int_t^{t+\Delta t} d\tau \;\int_t^{t+\Delta t} d\tau'\;\mathbf{Q}(\tau)\;\delta(\tau - \tau') \quad = \tag{2.69e}$$

$$= \quad \int_t^{t+\Delta t} d\tau\;\mathbf{Q}(\tau) \quad . \tag{2.69f}$$

where "$\star$" represents the same expression on its left (to avoid rewriting), and $\Delta\mathbf{w}(t) = \mathbf{w}(t) - \overline{\mathbf{w}}(t)$. Then, assuming that $\mathbf{q}(t)$ and $\mathbf{x}(t)$ are independent random variables ($\mathrm{E}[\,\Delta\mathbf{x}(\tau)\;\Delta\mathbf{q}(\tau')\,] = 0$),

we get

$$\mathbf{P}(t+\Delta t) = \mathrm{E}\big[\,\Delta\mathbf{x}(t+\Delta t)\;\Delta\mathbf{x}^T(t+\Delta t)\,\big] \approx \tag{2.70a}$$

$$\approx \big(\mathbf{I}+\mathbf{F}(t)\,\Delta t\,\big)\;\mathbf{P}(t)\;\big(\mathbf{I}+\mathbf{F}(t)\,\Delta t\,\big)^T + + \mathbf{L}(t)\;\mathbf{Q}(t)\;\Delta t\;\mathbf{L}^T(t) = \tag{2.70b}$$

$$= \mathbf{P}(t) + \mathbf{F}(t)\;\mathbf{P}(t)\;\Delta t + \mathbf{P}(t)\;\mathbf{F}^T(t)\;\Delta t + + \mathbf{F}(t)\;\mathbf{P}(t)\;\mathbf{F}^T(t)\;(\Delta t)^2 + \mathbf{L}(t)\;\mathbf{Q}(t)\;\Delta t\;\mathbf{L}^T(t) \quad . \tag{2.70c}$$

Finally, applying the definition of derivative,

$$\frac{d\,\mathbf{P}(t)}{d\,t} = \lim_{\Delta t\to 0}\frac{\mathbf{P}(t+\Delta t)-\mathbf{P}(t)}{\Delta t} \approx \tag{2.71a}$$

$$\approx \lim_{\Delta t\to 0}\frac{1}{\Delta t}\Big[\,\mathbf{F}(t)\;\mathbf{P}(t)\;\Delta t + \mathbf{P}(t)\;\mathbf{F}^T(t)\;\Delta t + + \mathbf{F}(t)\;\mathbf{P}(t)\;\mathbf{F}^T(t)\;(\Delta t)^2 + \mathbf{L}(t)\;\mathbf{Q}(t)\;\Delta t\;\mathbf{L}^T(t)\Big] = \tag{2.71b}$$

$$= \mathbf{F}(t)\;\mathbf{P}(t) + \mathbf{P}(t)\;\mathbf{F}^T(t) + \mathbf{L}(t)\;\mathbf{Q}(t)\;\mathbf{L}^T(t) \quad . \tag{2.71c}$$

2.5. Unscented Kalman Filter

The Unscented Kalman Filter (UKF) was introduced [49] as an alternative to the EKF. In the UKF, the statistical information about the random variables is encoded in a set of points obtained from the mean and the covariance matrix of the estimated state. This set of points is used to predict the mean and covariance matrix of the state and the measurement by evaluating the non-linear functions (2.51) and (2.52). Since in this case it is no longer necessary to compute Jacobians, the non-linear functions does not need to be differentiable.

2.5.1. Unscented Transformation

The weighted mean $\overline{\mathbf{x}}$, and weighted covariance matrix $\mathbf{P}$ of a set of points $\{\boldsymbol{\mathcal{X}}_i\}_i$ are computed through

$$\overline{\mathbf{x}} = \sum_i W_i^m\;\boldsymbol{\mathcal{X}}_i \quad , \tag{2.72}$$

$$\mathbf{P} = \sum_i W_i^c\;(\,\boldsymbol{\mathcal{X}}_i-\overline{\mathbf{x}}\,)\;(\,\boldsymbol{\mathcal{X}}_i-\overline{\mathbf{x}}\,)^T \quad , \tag{2.73}$$

with $\sum_i W_i^m = \sum_i W_i^c = 1$. The set of points $\{\boldsymbol{\mathcal{X}}_i\}_i$ that is commonly used in the *Unscented Transformation* scheme is generally defined as

$$\boldsymbol{\mathcal{X}}_0 = \overline{\mathbf{x}} \quad , \tag{2.74a}$$

$$\boldsymbol{\mathcal{X}}_i = \overline{\mathbf{x}} + \boldsymbol{\sigma}_i \qquad \text{for} \quad i=1\,,\,2\,,\,\ldots\,,\,N \quad , \tag{2.74b}$$

$$\boldsymbol{\mathcal{X}}_{N+i} = \overline{\mathbf{x}} - \boldsymbol{\sigma}_i \qquad \text{for} \quad i=1\,,\,2\,,\,\ldots\,,\,N \quad , \tag{2.74c}$$

being N the number of rows or columns of $\mathbf{P}$. These points are called *sigma points*. Given this definition, and recalling that $\sum_i W_i^m = 1$, we obtain for the mean:

$$\overline{x}_j = \sum_{i=0}^{2N} W_i^m\;\mathcal{X}_{ji} = \tag{2.75a}$$

$$= W_0^m\;\overline{x}_j + \sum_{i=1}^{N} W_i^m\;(\,\overline{x}_j+\sigma_{ji}\,) + \sum_{i=1}^{N} W_{N+i}^m\;(\,\overline{x}_j-\sigma_{ji}\,) = \tag{2.75b}$$

$$= \overline{x}_j + \sum_{i=1}^{N}\sigma_{ji}\;(W_i^m - W_{N+i}^m) \quad . \tag{2.75c}$$

Then, $W_i^m = W_{N+i}^m$ is necessary.

On the other hand, we obtain for the covariance matrix

$$P_{jk} = \sum_{i=0}^{2N} W_i^c \, (\mathcal{X}_{ji} - x_j) \, (\mathcal{X}_{ki} - x_k) \quad = \tag{2.76a}$$

$$= \quad 0 \; + \; \sum_{i=1}^{N} W_i^c \, (x_j + \sigma_{ji} \, - \, x_j)(x_k + \sigma_{ki} \, - \, x_k) \, + \\ + \; \sum_{i=1}^{N} W_{N+i}^c \, (x_j - \sigma_{ji} \, - \, x_j)(x_k - \sigma_{ki} \, - \, x_k) \quad = \tag{2.76b}$$

$$= \quad \sum_{i=1}^{N} W_i^c \; \sigma_{ji} \; \sigma_{ki} \; + \; \sum_{i=1}^{N} W_{N+i}^c \; \sigma_{ji} \; \sigma_{ki} \quad \equiv \tag{2.76c}$$

$$\equiv \quad \boldsymbol{\Sigma} \, (\, \mathbf{W}_+^c \, + \, \mathbf{W}_-^c \,) \, \boldsymbol{\Sigma}^T \quad , \tag{2.76d}$$

where $\mathbf{W}_+^c$ and $\mathbf{W}_-^c$ are diagonal matrices. There are several ways to find the matrix $\boldsymbol{\Sigma}$ and weights $\{W_i^c\}_i$ satisfying $\sum_i W_i^c = 1$:

Eigen-decomposition

$$\mathbf{P} \quad = \quad \mathbf{Q} \, \mathbf{D} \, \mathbf{Q}^T \quad = \tag{2.77a}$$

$$= \quad \left(\sqrt{D_0 + \mathrm{tr}(\mathbf{D})} \; \mathbf{Q} \right) \; \frac{1}{D_0 + \mathrm{tr}(\mathbf{D})} \; \mathbf{D} \; \left(\sqrt{D_0 + \mathrm{tr}(\mathbf{D})} \; \mathbf{Q} \right)^T \quad , \tag{2.77b}$$

where D_0 is a parameter that modulates the importance given to the sigma point $\boldsymbol{\mathcal{X}}_0$: $W_0 = \frac{D_0}{D_0 + \mathrm{tr}(\mathbf{D})}$. Then, the vector $\boldsymbol{\sigma}_i$ would be the i-th column of the matrix $\boldsymbol{\Sigma} = \sqrt{D_0 + \mathrm{tr}(\mathbf{D})} \; \mathbf{Q}$, and the weights would satisfy $W_i^c + W_{N+i}^c = \frac{1}{D_0 + \mathrm{tr}(\mathbf{D})} \; D_i$. If we establish symmetric weights for the sigma points, then $W_i^c = W_{N+i}^c = \frac{1}{2 \, [\, D_0 + \mathrm{tr}(\mathbf{D}) \,]} \; D_i$.

LDLT decomposition

$$\mathbf{P} \quad = \quad \mathbf{L} \, \mathbf{D} \, \mathbf{L}^T \quad = \tag{2.78a}$$

$$= \quad \left(\sqrt{D_0 + \mathrm{tr}(\mathbf{D})} \; \mathbf{L} \right) \; \frac{1}{D_0 + \mathrm{tr}(\mathbf{D})} \; \mathbf{D} \; \left(\sqrt{D_0 + \mathrm{tr}(\mathbf{D})} \; \mathbf{L} \right)^T \quad , \tag{2.78b}$$

where D_0 modulates the importance given to the sigma point $\boldsymbol{\mathcal{X}}_0$: $W_0 = \frac{D_0}{D_0 + \mathrm{tr}(\mathbf{D})}$. Then, the vector $\boldsymbol{\sigma}_i$ would be the i-th column of the matrix $\boldsymbol{\Sigma} = \sqrt{D_0 + \mathrm{tr}(\mathbf{D})} \; \mathbf{L}$, and the weights would satisfy $W_i^c + W_{N+i}^c = \frac{1}{D_0 + \mathrm{tr}(\mathbf{D})} \; D_i$. If we establish symmetric weights for the sigma points, then $W_i^c = W_{N+i}^c = \frac{1}{2 \, [\, D_0 + \mathrm{tr}(\mathbf{D}) \,]} \; D_i$.

Cholesky decomposition

$$\mathbf{P} \quad = \quad \mathbf{L} \, \mathbf{L}^T \quad = \tag{2.79a}$$

$$= \quad \left(\boldsymbol{\Sigma} \, \sqrt{\mathbf{W}_+^c + \mathbf{W}_-^c} \right) \left(\boldsymbol{\Sigma} \, \sqrt{\mathbf{W}_+^c + \mathbf{W}_-^c} \, \right)^T \quad . \tag{2.79b}$$

Then, the vector $\boldsymbol{\sigma}_i$ would be the i-th column of the matrix $\boldsymbol{\Sigma} = \mathbf{L} \, \left(\sqrt{\mathbf{W}_+^c + \mathbf{W}_-^c} \right)^{-1}$. In particular, if we decide to give the same importance to each sigma point for the covariance matrix computation ($W_i^c = W^c$), then $\boldsymbol{\Sigma} = \frac{1}{\sqrt{2 W^c}} \; \mathbf{L}$.

This latter decomposition is generally preferred. There are multiple choices for the weights W_i^m and W_i^c: [7, 49, 50]. In this work the weights will be chosen accordingly to a simple and intuitive rule presented in [50]:

$$W_0^m \quad = \quad W_0^c \quad = \quad W_0 \quad , \tag{2.80a}$$

$$W_i^m \quad = \quad W_{N+i}^m \quad = \quad W_i^c \quad = \quad W_{N+i}^c \quad = \quad \frac{1 - W_0}{2N} \quad , \tag{2.80b}$$

with $i = 1 \, , \, \ldots \, , \, N$.

2.5.2. UKF Prediction

In the UKF prediction step, we first generate the sigma points $\{\boldsymbol{\mathcal{X}}_i\}_i$ using the *augmented system descriptors*:

$$\widetilde{\mathbf{x}}_{n\text{-}1|n\text{-}1} = \begin{pmatrix} \overline{\mathbf{x}}_{n\text{-}1|n\text{-}1} \\ \mathbf{q}_n \\ \mathbf{r}_n \end{pmatrix} , \tag{2.81}$$

$$\widetilde{\mathbf{P}}_{n\text{-}1|n\text{-}1} = \begin{pmatrix} \mathbf{P}_{n\text{-}1|n\text{-}1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_n \end{pmatrix} . \tag{2.82}$$

Then, we obtain the sigma points $\{\boldsymbol{\mathcal{Y}}_i\}_i$ that contain statistical information about the predicted state $\mathbf{x}_{n|n\text{-}1}$:

$$\boldsymbol{\mathcal{Y}}_i = \mathbf{f}(\boldsymbol{\mathcal{X}}_i) , \tag{2.83}$$

and the sigma points $\{\boldsymbol{\mathcal{Z}}_i\}_i$ that contain statistical information about the predicted measurement $\mathbf{z}_{n|n\text{-}1}$:

$$\boldsymbol{\mathcal{Z}}_i = \mathbf{h}(\boldsymbol{\mathcal{Y}}_i) . \tag{2.84}$$

The mean and the covariance matrix of the predicted state $\mathbf{x}_{n|n\text{-}1}$ and the predicted measurement $\mathbf{z}_{n|n\text{-}1}$ are computed using equations (2.72) and (2.73):

$$\overline{\mathbf{x}}_{n|n\text{-}1} = \sum_{i=0}^{2N} W_i^m \, \boldsymbol{\mathcal{Y}}_i , \tag{2.85}$$

$$\overline{\mathbf{z}}_{n|n\text{-}1} = \sum_{i=0}^{2N} W_i^m \, \boldsymbol{\mathcal{Z}}_i , \tag{2.86}$$

$$\mathbf{P}_{n|n\text{-}1} = \sum_{i=0}^{2N} W_i^c \, (\boldsymbol{\mathcal{Y}}_i - \overline{\mathbf{x}}_{n|n\text{-}1})(\boldsymbol{\mathcal{Y}}_i - \overline{\mathbf{x}}_{n|n\text{-}1})^T , \tag{2.87}$$

$$\mathbf{S}_{n|n\text{-}1} = \sum_{i=0}^{2N} W_i^c \, (\boldsymbol{\mathcal{Z}}_i - \overline{\mathbf{z}}_{n|n\text{-}1})(\boldsymbol{\mathcal{Z}}_i - \overline{\mathbf{z}}_{n|n\text{-}1})^T , \tag{2.88}$$

$$\mathbf{P}^{yz}_{n|n\text{-}1} = \sum_{i=0}^{2N} W_i^c \, (\boldsymbol{\mathcal{Y}}_i - \overline{\mathbf{x}}_{n|n\text{-}1})(\boldsymbol{\mathcal{Z}}_i - \overline{\mathbf{z}}_{n|n\text{-}1})^T . \tag{2.89}$$

2.5.3. UKF Update

Recalling that the minimum trace of the covariance matrix for the updated state is found when $\mathbf{K}_n \, \mathbf{S}_{n|n\text{-}1} = \mathbf{P}_{n|n\text{-}1} \, \mathbf{H}_n^T$, equation (2.50) can be rewritten as

$$\mathbf{P}_{n|n} = (\mathbf{I} - \mathbf{K}_n \mathbf{H}_n) \, \mathbf{P}_{n|n\text{-}1} \, (\mathbf{I} - \mathbf{K}_n \mathbf{H}_n)^T + \mathbf{K}_n \, \mathbf{R}_n \, \mathbf{K}_n^T = \tag{2.90a}$$

$$= \mathbf{P}_{n|n\text{-}1} - \mathbf{K}_n \, \mathbf{H}_n \, \mathbf{P}_{n|n\text{-}1} - \mathbf{P}_{n|n\text{-}1} \, \mathbf{H}_n^T \, \mathbf{K}_n^T + \\ + \mathbf{K}_n \, \mathbf{H}_n \, \mathbf{P}_{n|n\text{-}1} \, \mathbf{H}_n^T \, \mathbf{K}_n^T + \mathbf{K}_n \, \mathbf{R}_n \, \mathbf{K}_n^T = \tag{2.90b}$$

$$= \mathbf{P}_{n|n\text{-}1} - \mathbf{K}_n \, \mathbf{S}_{n|n\text{-}1}^T \, \mathbf{K}_n^T - \mathbf{K}_n \, \mathbf{S}_{n|n\text{-}1} \, \mathbf{K}_n^T + \\ + \mathbf{K}_n \, \mathbf{S}_{n|n\text{-}1} \, \mathbf{K}_n^T = \tag{2.90c}$$

$$= \mathbf{P}_{n|n\text{-}1} - \mathbf{K}_n \, \mathbf{S}_{n|n\text{-}1} \, \mathbf{K}_n^T . \tag{2.90d}$$

Noticing that for a linear system $\mathbf{P}_{n|n\text{-}1}\,\mathbf{H}_n^T = \mathbf{P}_{n|n\text{-}1}^{yz}$, and assuming that our non-linear system can be described locally as a linear system, the update equations for the UKF turn out to be

$$\mathbf{K}_n = \mathbf{P}_{n|n\text{-}1}^{yz}\,\mathbf{S}_{n|n\text{-}1}^{-1} \quad , \tag{2.91}$$

$$\mathbf{x}_{n|n} = \mathbf{x}_{n|n\text{-}1} + \mathbf{K}_n\left(\mathbf{z}_n - \overline{\mathbf{z}}_{n|n\text{-}1}\right) \quad , \tag{2.92}$$

$$\mathbf{P}_{n|n} = \mathbf{P}_{n|n\text{-}1} - \mathbf{K}_n\,\mathbf{S}_{n|n\text{-}1}\,\mathbf{K}_n^T \quad . \tag{2.93}$$

2.6. Notes on the EKF and the UKF

- The EKF and the UKF are not optimal estimators due to two reasons:
  - The moments of the random variables $\mathbf{x}_{n|n\text{-}1}$ and $\mathbf{z}_{n|n\text{-}1}$ are computed approximately.
  - Equation (2.33) does not produce an unbiased estimation any more [51] (except in case of $\mathbf{f}$ and $\mathbf{h}$ being linear functions).
- The accuracy of the EKF predictions (given by equations (2.20) and (2.21)) depend on how the degree of uncertainty compares to the amount of local non-linearity of the functions that model the system.
- The UKF update equation (2.93) does not generally produce a positive-definite matrix.

Chapter 3

Quaternions

3.1. Introduction

Quaternions are numbers that extend the complex numbers. Instead of having a single imaginary unit (i), they have 3 ($\{\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}\}$). They were first introduced as an algebra by the Irish mathematician and physicist William Rowan Hamilton in 1843 [52]. Knowing that complex numbers can be interpreted as points in a 2-dimensional space, and that they can be multiplied and divided, Hamilton tried to find numbers with similar properties in a 3-dimensional space. The following is a letter from Hamilton to his son Archibald in which he talks about the circumstances in which he discovered quaternions [53]:

> 'My dear Archibald—(1) I had been wishing for an occasion of corresponding a little with you on Quaternions: and such now presents itself, by your mentioning in your note of yesterday, received this morning, that you "have been reflecting on several points connected with them" (the quaternions), "particularly on the *Multiplication of Vectors*."
>
> '(2) No more important, or indeed fundamental question, in the whole Theory of Quaternions, can be proposed than that which thus inquires *What is such* Multiplication? What are its Rules, its Objects, its Results? What *Analogies* exist between it and other *Operations*, which have received the same general *Name*? And finally, what is (if any) its Utility?
>
> '(3) If I may be allowed to speak of *myself* in connexion with the subject, I might do so in a way which would bring *you* in, by referring to an *ante-quaternionic* time, when you were a mere *child*, but had caught from me the conception of a Vector, as represented by a *Triplet*: and indeed I happen to be able to put the finger of memory upon the year and month—October, 1843—when having recently returned from visits to Cork and Parsonstown, connected with a Meeting of the British Association, the desire to discover the laws of the multiplication referred to regained with me a certain strength and earnestness, which had for years been dormant, but was then on the point of being gratified, and was occasionally talked of with you. Every morning in the early part of the above-cited month, on my coming down to breakfast, your (then) little brother William Edwin, and yourself, used to ask me, "Well, Papa, can you *multiply* triplets"? Whereto I was always obliged to reply, with a sad shake of the head: "No, I can only *add* and subtract them."
>
> '(4) But on the 16th day of the same month—which happened to be a Monday, and a Council day of the Royal Irish Academy—I was walking in to attend and preside, and your mother was walking with me, along the Royal Canal, to which she had perhaps driven; and although she talked with me now and then, yet an *under-current* of thought was going on in my mind, which gave at last a *result*, whereof it is not too much to say that I felt *at once* the importance. An *electric* circuit seemed to *close*; and a spark flashed forth, the herald (as I *foresaw, immediately*) of many long years to come of definitely directed thought and work, by *myself* if spared, and at all events on the part of *others*, if I should even be allowed to live long enough distinctly to communicate the discovery. Nor could I resist the impulse—unphilosophical as it may have been—to cut with a knife on a stone of Brougham Bridge, as we passed it, the fundamental formula

with the symbols, i, j, k; namely,

$$i^2 = j^2 = k^2 = ijk = -1,$$

which contains the *Solution* of the *Problem*, but of course, as an inscription, has long since mouldered away. A more durable notice remains, however, on the Council Books of the Academy for that day (October 16th, 1843), which records the fact, that I then asked for and obtained leave to read a Paper on *Quaternions*, at the *First General Meeting* of the Session: which reading took place accordingly, on Monday the 13th of the November following.

'With this *quaternion of paragraphs* I close this letter I.; but I hope to follow it up very shortly with another.

'Your affectionate Father,
'William Rowan Hamilton.'

Now, we know that his search for the multiplication of triplets was doomed to failure by virtue of Frobenius theorem [54]: the only finite-dimensional associative division algebras over the real numbers are isomorphic to real numbers ($\mathbb{R}$ with real dimension 1), complex numbers ($\mathbb{C}$ with real dimension 2), or quaternions ($\mathbb{H}$ with real dimension 4); and therefore, quaternions were the only solution to the problem.

Prior to Hamilton's work, others glimpsed some quaternion properties: Leonhard Euler obtained the "four-square identity" in 1748 [55]; Olinde Rodrigues parameterized 3-dimensional rotations using the "Euler-Rodrigues formula" in 1840; and Carl Friedrich Gauss even discovered quaternions in 1819 [56], but this work was not published until 1900.

In this chapter, we will start reviewing the basic definitions and properties of quaternions in Section 3.2. Then, in Section 3.3, we will examine how unit quaternions are related to 3D rotations. Finally, we will use basic concepts from manifold theory to encode a distribution of unit quaternions in Section 3.4.

3.2. Definition and Properties

A quaternion is a hypercomplex number usually expressed as

$$\boldsymbol{q} \;=\; q_0 \;+\; q_1\,\boldsymbol{i} \;+\; q_2\,\boldsymbol{j} \;+\; q_3\,\boldsymbol{k} \quad , \tag{3.1}$$

where q_0, q_1, q_2, and q_3 are real numbers, and $\{\,\boldsymbol{i}\,,\,\boldsymbol{j}\,,\,\boldsymbol{k}\,\}$ are three different imaginary units.

3.2.1. Quaternions as a Vector Space

Quaternions can be defined as elements of a vector space if we define for them the operations of addition and scalar multiplication. In such a case, a quaternion would be a 4-dimensional vector ($\boldsymbol{q} \in \mathbb{R}^4$) expressed in the basis $\{\,\boldsymbol{1}\,,\,\boldsymbol{i}\,,\,\boldsymbol{j}\,,\,\boldsymbol{k}\,\}$. Then, a quaternion $\boldsymbol{q}$ may be represented using several notations:

$$\boldsymbol{q} \;=\; q_0 \;+\; q_1\,\boldsymbol{i} \;+\; q_2\,\boldsymbol{j} \;+\; q_3\,\boldsymbol{k} \quad \equiv \tag{3.2a}$$

$$\equiv \quad (\; q_0 \;,\; q_1 \;,\; q_2 \;,\; q_3 \;)^T \quad \equiv \tag{3.2b}$$

$$\equiv \quad (\; q_0 \;,\; \mathbf{q} \;)^T \quad . \tag{3.2c}$$

Notice that in (3.2c) we write the imaginary part of the quaternion as a vector. We will see that this distinction is convenient when defining the quaternion product.

3.2.1.1. Addition

The addition of quaternions is defined as:

$$\boldsymbol{p} \;+\; \boldsymbol{q} \;=\; (\,p_0 \;+\; q_0\,) \;+\; (\,p_1 \;+\; q_1\,)\,\boldsymbol{i} \;+\; (\,p_2 \;+\; q_2\,)\,\boldsymbol{j} \;+\; (\,p_3 \;+\; q_3\,)\,\boldsymbol{k} \quad . \tag{3.3}$$

3.2.1.2. Scalar Multiplication

The scalar multiplication of a quaternion is defined as:

$$\lambda\, \boldsymbol{q} \quad = \quad (\lambda\, q_0) \;+\; (\lambda\, q_1)\, \boldsymbol{i} \;+\; (\lambda\, q_2)\, \boldsymbol{j} \;+\; (\lambda\, q_3)\, \boldsymbol{k} \quad . \tag{3.4}$$

3.2.2. Quaternion Product

The quaternion product is derived from the Hamilton axiom:

$$\boldsymbol{i}^2 \quad = \quad \boldsymbol{j}^2 \quad = \quad \boldsymbol{k}^2 \quad = \quad \boldsymbol{i}\,\boldsymbol{j}\,\boldsymbol{k} \quad = \quad -\mathbf{1} \quad . \tag{3.5}$$

The multiplication table 3.1 is a direct consequence of (3.5)[1]. Note that, like the matrix product, the quaternion product is not commutative (take for example $\boldsymbol{i}\,\boldsymbol{j} \;\neq\; \boldsymbol{j}\,\boldsymbol{i}$).

Table 3.1: Multiplication table for the quaternion product

$*$	$\mathbf{1}$	$\boldsymbol{i}$	$\boldsymbol{j}$	$\boldsymbol{k}$
$\mathbf{1}$	$\mathbf{1}$	$\boldsymbol{i}$	$\boldsymbol{j}$	$\boldsymbol{k}$
$\boldsymbol{i}$	$\boldsymbol{i}$	$-\mathbf{1}$	$\boldsymbol{k}$	$-\boldsymbol{j}$
$\boldsymbol{j}$	$\boldsymbol{j}$	$-\boldsymbol{k}$	$-\mathbf{1}$	$\boldsymbol{i}$
$\boldsymbol{k}$	$\boldsymbol{k}$	$\boldsymbol{j}$	$-\boldsymbol{i}$	$-\mathbf{1}$

Using Table 3.1 we can obtain a general expression for the product of two quaternions:

$$\boldsymbol{p} * \boldsymbol{q} \quad = \quad (p_0 \;+\; p_1\,\boldsymbol{i} \;+\; p_2\,\boldsymbol{j} \;+\; p_3\,\boldsymbol{k})\,(q_0 \;+\; q_1\,\boldsymbol{i} \;+\; q_2\,\boldsymbol{j} \;+\; q_3\,\boldsymbol{k}) \quad = \tag{3.6a}$$

$$\begin{aligned} = \quad & p_0\,q_0 \;+\; p_0\,q_1\,\boldsymbol{i} \;+\; p_0\,q_2\,\boldsymbol{j} \;+\; p_0\,q_3\,\boldsymbol{k} \;+ \\ & +\; p_1\,q_0\,\boldsymbol{i} \;+\; p_1\,q_1\,\boldsymbol{i}\,\boldsymbol{i} \;+\; p_1\,q_2\,\boldsymbol{i}\,\boldsymbol{j} \;+\; p_1\,q_3\,\boldsymbol{i}\,\boldsymbol{k} \;+ \\ & +\; p_2\,q_0\,\boldsymbol{j} \;+\; p_2\,q_1\,\boldsymbol{j}\,\boldsymbol{i} \;+\; p_2\,q_2\,\boldsymbol{j}\,\boldsymbol{j} \;+\; p_2\,q_3\,\boldsymbol{j}\,\boldsymbol{k} \;+ \\ & +\; p_3\,q_0\,\boldsymbol{k} \;+\; p_3\,q_1\,\boldsymbol{k}\,\boldsymbol{i} \;+\; p_3\,q_2\,\boldsymbol{k}\,\boldsymbol{j} \;+\; p_3\,q_3\,\boldsymbol{k}\,\boldsymbol{k} \quad = \end{aligned} \tag{3.6b}$$

$$\begin{aligned} = \quad & p_0\,q_0 - p_1\,q_1 \;-\; p_2\,q_2 \;-\; p_3\,q_3 \;+ \\ & +\; (p_0\,q_1 \;+\; q_0\,p_1 \;+\; p_2\,q_3 \;-\; p_3\,q_2)\,\boldsymbol{i} \;+ \\ & +\; (p_0\,q_2 \;+\; q_0\,p_2 \;+\; p_3\,q_1 \;-\; p_1\,q_3)\,\boldsymbol{j} \;+ \\ & +\; (p_0\,q_3 \;+\; q_0\,p_3 \;+\; p_1\,q_2 \;-\; p_2\,q_1)\,\boldsymbol{k} \quad . \end{aligned} \tag{3.6c}$$

Equation (3.6) can be rewritten using notation (3.2c) as

$$\boldsymbol{p} * \boldsymbol{q} \quad = \quad \begin{pmatrix} p_0\,q_0 \;-\; \mathbf{p}\cdot\mathbf{q} \\ p_0\,\mathbf{q} \;+\; q_0\,\mathbf{p} \;+\; \mathbf{p}\times\mathbf{q} \end{pmatrix} \quad , \tag{3.7}$$

where $(\cdot)$ represents the usual *dot product*, and $(\times)$ represents the 3-vector *cross product*. From (3.7) we see that the quaternion product can also be expressed in matrix form:

$$\boldsymbol{p} * \boldsymbol{q} \quad \equiv \tag{3.8a}$$

$$\equiv \quad [\boldsymbol{p}]_*\;\boldsymbol{q} \quad = \quad \begin{pmatrix} p_0 & -\mathbf{p}^T \\ \mathbf{p} & p_0\,\mathbf{I} \;+\; [\mathbf{p}]_\times \end{pmatrix} \boldsymbol{q} \quad = \quad \begin{pmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & -p_3 & p_2 \\ p_2 & p_3 & p_0 & -p_1 \\ p_3 & -p_2 & p_1 & p_0 \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} \quad \equiv \tag{3.8b}$$

$$\equiv \quad {}_*[\boldsymbol{q}]\;\boldsymbol{p} \quad = \quad \begin{pmatrix} q_0 & -\mathbf{q}^T \\ \mathbf{q} & q_0\,\mathbf{I} \;-\; [\mathbf{q}]_\times \end{pmatrix} \boldsymbol{p} \quad = \quad \begin{pmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix} \quad , \tag{3.8c}$$

[1] Example: $\boldsymbol{i}\,\boldsymbol{j} \;=\; \boldsymbol{i}\,\boldsymbol{j}\,(-\boldsymbol{k}^2) \;=\; \boldsymbol{i}\,\boldsymbol{j}\,\boldsymbol{k}\,(-\boldsymbol{k}) \;=\; (-1)\,(-\boldsymbol{k}) \;=\; \boldsymbol{k}$

where we have defined

$$[\,\mathbf{v}\,]_\times \quad := \quad \begin{pmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{pmatrix} \quad . \tag{3.9}$$

Made this observation, we can verify that the quaternion product is associative, and distributive over the sum:

Associative property

$$\boldsymbol{p} * (\,\boldsymbol{q} * \boldsymbol{r}\,) \;=\; [\,\boldsymbol{p}\,]_* \left(\,[\,\boldsymbol{q}\,]_*\;\boldsymbol{r}\,\right) \;=\; \left(\,[\,\boldsymbol{p}\,]_*\;[\,\boldsymbol{q}\,]_*\,\right)\boldsymbol{r} \;=\; [\,\boldsymbol{p} * \boldsymbol{q}\,]_*\;\boldsymbol{r} \;=\; (\,\boldsymbol{p} * \boldsymbol{q}\,) * \boldsymbol{r} \quad . \tag{3.10}$$

Distributive property

$$\boldsymbol{p} * (\,\boldsymbol{q} + \boldsymbol{r}\,) \;=\; [\,\boldsymbol{p}\,]_*\,(\,\boldsymbol{q} + \boldsymbol{r}\,) \;=\; [\,\boldsymbol{p}\,]_*\;\boldsymbol{q} \;+\; [\,\boldsymbol{p}\,]_*\;\boldsymbol{r} \;=\; \boldsymbol{p} * \boldsymbol{q} \;+\; \boldsymbol{p} * \boldsymbol{r} \quad . \tag{3.11}$$

$$(\,\boldsymbol{p} + \boldsymbol{q}\,) * \boldsymbol{r} \;=\; {}_*[\,\boldsymbol{r}\,]\,(\,\boldsymbol{p} + \boldsymbol{q}\,) \;=\; {}_*[\,\boldsymbol{r}\,]\;\boldsymbol{p} \;+\; {}_*[\,\boldsymbol{r}\,]\;\boldsymbol{q} \;=\; \boldsymbol{p} * \boldsymbol{r} \;+\; \boldsymbol{q} * \boldsymbol{r} \quad . \tag{3.12}$$

3.2.3. Conjugate Quaternion

We define the *conjugate quaternion* of the quaternion $\boldsymbol{q}$ as

$$\boldsymbol{q}^* \quad = \quad q_0 \;-\; q_1\,\boldsymbol{i} \;-\; q_2\,\boldsymbol{j} \;-\; q_3\,\boldsymbol{k} \quad \equiv \tag{3.13a}$$

$$\equiv \quad (\;q_0\;,\;-q_1\;,\;-q_2\;,\;-q_3\;)^T \quad \equiv \tag{3.13b}$$

$$\equiv \quad (\;q_0\;,\;-\mathbf{q}\;)^T \quad . \tag{3.13c}$$

Note that the conjugate of the product of two quaternions is the product of the conjugates in the reverse order:

$$(\boldsymbol{p} * \boldsymbol{q})^* \quad = \quad \begin{pmatrix} p_0\,q_0 \;-\; \mathbf{p}\cdot\mathbf{q} \\ -\,p_0\,\mathbf{q} \;-\; q_0\,\mathbf{p} \;-\; \mathbf{p}\times\mathbf{q} \end{pmatrix} \quad = \tag{3.14a}$$

$$= \quad \begin{pmatrix} p_0\,q_0 \;-\; (-\mathbf{p})\cdot(-\mathbf{q}) \\ p_0\,(-\mathbf{q}) \;+\; q_0\,(-\mathbf{p}) \;-\; (-\mathbf{p})\times(-\mathbf{q}) \end{pmatrix} \quad = \tag{3.14b}$$

$$= \quad \begin{pmatrix} p_0\,q_0 \;-\; (-\mathbf{q})\cdot(-\mathbf{p}) \\ q_0\,(-\mathbf{p}) \;+\; p_0\,(-\mathbf{q}) \;+\; (-\mathbf{q})\times(-\mathbf{p}) \end{pmatrix} \quad = \tag{3.14c}$$

$$= \quad \boldsymbol{q}^* * \boldsymbol{p}^* \quad . \tag{3.14d}$$

3.2.4. Quaternion Norm

We define the *norm of a quaternion* $\boldsymbol{q}$ as

$$\|\boldsymbol{q}\| \quad = \quad \sqrt{q_0^2 \;+\; q_1^2 \;+\; q_2^2 \;+\; q_3^2} \quad . \tag{3.15}$$

Note that the product of a quaternion $\boldsymbol{q}$ and its conjugate quaternion $\boldsymbol{q}^*$ is the squared norm of the quaternion:

$$\boldsymbol{q} * \boldsymbol{q}^* \quad = \quad \boldsymbol{q}^* * \boldsymbol{q} \quad = \tag{3.16a}$$

$$= \quad \begin{pmatrix} q_0\,q_0 \;-\; \mathbf{q}\cdot(-\mathbf{q}) \\ q_0\,(-\mathbf{q}) \;+\; q_0\,\mathbf{q} \;+\; \mathbf{q}\times(-\mathbf{q}) \end{pmatrix} \quad = \tag{3.16b}$$

$$= \quad \begin{pmatrix} q_0^2 \;+\; \|\mathbf{q}\|^2 \\ \mathbf{0} \end{pmatrix} \quad \equiv \tag{3.16c}$$

$$\equiv \quad q_0^2 \;+\; q_1^2 \;+\; q_2^2 \;+\; q_3^2 \quad = \quad \|\boldsymbol{q}\|^2 \quad . \tag{3.16d}$$

Note also that the norm of the product of two quaternions is the product of the norms:

$$\begin{aligned} \|\boldsymbol{p} * \boldsymbol{q}\|^2 &= (\boldsymbol{p} * \boldsymbol{q}) * (\boldsymbol{p} * \boldsymbol{q})^* = && (3.17a)\\ &= \boldsymbol{p} * \boldsymbol{q} * \boldsymbol{q}^* * \boldsymbol{p}^* = && (3.17b)\\ &= \boldsymbol{p} * \|\boldsymbol{q}\|^2 * \boldsymbol{p}^* = && (3.17c)\\ &= \|\boldsymbol{q}\|^2 \, (\boldsymbol{p} * \boldsymbol{p}^*) = && (3.17d)\\ &= \|\boldsymbol{q}\|^2 \, \|\boldsymbol{p}\|^2 \quad . && (3.17e) \end{aligned}$$

Quaternions with unit norm $\{ \boldsymbol{q} \in \mathbb{R}^4 : \|\boldsymbol{q}\| = 1 \}$ are of special interest as we will see in Section 3.3. These quaternions are called *unit quaternions*.

3.2.5. Multiplicative Inverse Quaternion

We define the *multiplicative inverse* or *reciprocal* of a quaternion $\boldsymbol{q}$ as the quaternion $\boldsymbol{q}^{-1}$ that satisfies

$$\boldsymbol{q} * \boldsymbol{q}^{-1} = \boldsymbol{q}^{-1} * \boldsymbol{q} = 1 \quad . \tag{3.18}$$

Using (3.16) we can always obtain the reciprocal of a quaternion $\boldsymbol{q}$ with $\|\boldsymbol{q}\| \neq 0$:

$$\boldsymbol{q}^{-1} = \frac{\boldsymbol{q}^*}{\|\boldsymbol{q}\|^2} \quad . \tag{3.19}$$

Note that the reciprocal of the product of two quaternions is the product of the reciprocals in the reverse order:

$$\begin{aligned} (\boldsymbol{p} * \boldsymbol{q})^{-1} &= \frac{(\boldsymbol{p} * \boldsymbol{q})^*}{\|\boldsymbol{p} * \boldsymbol{q}\|^2} = && (3.20a)\\ &= \frac{\boldsymbol{q}^* * \boldsymbol{p}^*}{\|\boldsymbol{q}\|^2 \, \|\boldsymbol{p}\|^2} = && (3.20b)\\ &= \frac{\boldsymbol{q}^*}{\|\boldsymbol{q}\|^2} * \frac{\boldsymbol{p}^*}{\|\boldsymbol{p}\|^2} = \boldsymbol{q}^{-1} * \boldsymbol{p}^{-1} \quad . && (3.20c) \end{aligned}$$

3.2.6. Quaternion Exponentiation

Analogous to the case of real numbers, and the case of complex numbers, the exponential function of a quaternion is defined as:

$$e^{\boldsymbol{q}} = \sum_{n=0}^{\infty} \frac{\boldsymbol{q}^n}{n!} \quad , \qquad \text{with} \quad \boldsymbol{q}^n = \underbrace{\boldsymbol{q} * \boldsymbol{q} * \cdots * \boldsymbol{q}}_{n \text{ times}} \quad . \tag{3.21}$$

As with complex numbers, this definition leads to a closed-form expression:

$$\begin{aligned} e^{\boldsymbol{q}} &= e^{q_0} \left[\cos \|\mathbf{q}\| + \left(q_1 \, \boldsymbol{i} + q_2 \, \boldsymbol{j} + q_3 \, \boldsymbol{k} \right) \frac{\sin \|\mathbf{q}\|}{\|\mathbf{q}\|} \right] = && (3.22a)\\ &= e^{q_0} \begin{pmatrix} \cos \|\mathbf{q}\| \\ \frac{\mathbf{q}}{\|\mathbf{q}\|} \sin \|\mathbf{q}\| \end{pmatrix} \quad . && (3.22b) \end{aligned}$$

A detailed derivation of expression (3.22) can be found in Appendix A.4.

3.3. Quaternions Representing Rotations

Equation (3.22) bears a strong resemblance to its analog for complex numbers:

$$e^{a + b\,i} = e^a \, (\cos b + i \, \sin b) \quad . \tag{3.23}$$

When one perceives this parallelism, an idea arises; quaternions could describe rotations in a similar way to complex numbers:

$$e^{i\,\theta} \;*\; z \;=\; (\cos\theta \;+\; i\;\sin\theta)\;*\;(x \;+\; i\,y) \;=\; \tag{3.24a}$$

$$=\; x\;\cos\theta \;-\; y\;\sin\theta \;+\; i\,(x\;\sin\theta \;+\; y\;\cos\theta) \;\equiv\; \tag{3.24b}$$

$$\equiv\; \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad . \tag{3.24c}$$

For complex numbers, multiplying by $e^{i\,\theta}$ is equivalent to applying a positive rotation[2] of θ radians to the vector $\mathbf{z} = \left(\begin{smallmatrix} x \\ y \end{smallmatrix}\right)$ in the complex plane.

To explore what happens with quaternions we will consider the exponential of a purely imaginary quaternion $\boldsymbol{u}$:

$$\boldsymbol{q} \;=\; e^{\boldsymbol{u}} \;=\; e^{u_1\,\boldsymbol{i} + u_2\,\boldsymbol{j} + u_3\,\boldsymbol{k}} \;= \tag{3.25a}$$

$$=\; \begin{pmatrix} \cos\theta \\ \widehat{\mathbf{u}}\;\sin\theta \end{pmatrix} \quad , \qquad \text{with} \quad \begin{cases} \theta \;=\; \|\mathbf{u}\| \\ \widehat{\mathbf{u}} \;=\; \frac{\mathbf{u}}{\|\mathbf{u}\|} \end{cases} \quad . \tag{3.25b}$$

The vector $\widehat{\mathbf{u}}$ could be an element of an orthonormal basis $\mathcal{B} = \{\,\widehat{\mathbf{u}}\,,\,\widehat{\mathbf{v}}\,,\,\widehat{\mathbf{w}}\,\}$ for $\mathbb{R}^3$. And the basis $\mathcal{B}$ could be used to build quaternions $\boldsymbol{u} = (\,0\,,\,\widehat{\mathbf{u}}\,)^T$, $\boldsymbol{v} = (\,0\,,\,\widehat{\mathbf{v}}\,)^T$, and $\boldsymbol{w} = (\,0\,,\,\widehat{\mathbf{w}}\,)^T$, that could be part of an orthonormal basis $\mathcal{B}' = \{\,\mathbf{1}\,,\,\boldsymbol{u}\,,\,\boldsymbol{v}\,,\,\boldsymbol{w}\,\}$ of $\mathbb{R}^4$. The basis $\mathcal{B}'$ would satisfy Hamilton's axiom (3.5). Specifically,

$$\boldsymbol{u}^2 \;=\; \boldsymbol{v}^2 \;=\; \boldsymbol{w}^2 \;=\; \boldsymbol{u}\;\boldsymbol{v}\;\boldsymbol{w} \;=\; -\mathbf{1} \quad . \tag{3.26}$$

Then, we could express any quaternion $\boldsymbol{p}$ using this basis:

$$\boldsymbol{p} \;=\; p_0 \;+\; p_1\;\boldsymbol{u} \;+\; p_2\;\boldsymbol{v} \;+\; p_3\;\boldsymbol{w} \quad . \tag{3.27}$$

In particular, the quaternion $\boldsymbol{q}$ in equation (3.25) would be expressed as

$$\boldsymbol{q} \;=\; \cos\theta \;+\; \boldsymbol{u}\;\sin\theta \quad . \tag{3.28}$$

Equations (3.8) will now be useful:

Left product ($\boldsymbol{q} * \boldsymbol{p}$)

From equations (3.8b) and (3.28),

$$\boldsymbol{q} * \boldsymbol{p} \;=\; \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & \cos\theta & -\sin\theta \\ 0 & 0 & \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix} \quad . \tag{3.29}$$

This is, the left product produces two rotations:

- A positive rotation of the plane defined by $\{\,\mathbf{1}\,,\,\boldsymbol{u}\,\}$.
- A positive rotation of the plane defined by $\{\,\boldsymbol{v}\,,\,\boldsymbol{w}\,\}$.

Right product ($\boldsymbol{p} * \boldsymbol{q}$)

From equations (3.8c) and (3.28),

$$\boldsymbol{p} * \boldsymbol{q} \;=\; \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & \cos\theta & \sin\theta \\ 0 & 0 & -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix} \quad . \tag{3.30}$$

This is, the right product produces two rotations:

- A positive rotation of the plane defined by $\{\,\mathbf{1}\,,\,\boldsymbol{u}\,\}$.
- A negative rotation of the plane defined by $\{\,\boldsymbol{v}\,,\,\boldsymbol{w}\,\}$.

[2] Counterclockwise rotation.

3.3.1. Quaternions Representing 3D Rotations

We have found that the product of a unit quaternion (3.28) and another quaternion $\boldsymbol{p}$ produces a rotation of $\boldsymbol{p}$ in the planes defined by $\{\,\mathbf{1}\,,\,\boldsymbol{u}\,\}$ and by $\{\,\boldsymbol{v}\,,\,\boldsymbol{w}\,\}$. If we compose the left product and the right product $(\boldsymbol{q} * \boldsymbol{p} * \boldsymbol{q})$, then their effects in the first plane would double up, while the effects in the second plane would cancel. However, if we use the inverse quaternion $\boldsymbol{q}^{-1}$ in the right product $(\boldsymbol{q} * \boldsymbol{p} * \boldsymbol{q}^{-1})$, then the effects in the first plane would cancel, and the effects in the second plane would double up. In particular, if

$$\boldsymbol{q} \quad = \quad \begin{pmatrix} \cos\frac{\theta}{2} \\ \widehat{\mathbf{u}}\,\sin\frac{\theta}{2} \end{pmatrix} \quad , \tag{3.31}$$

and $\boldsymbol{p} = (\,0\,,\,\mathbf{p}\,)^T$, the result of the quaternion product $\boldsymbol{q} * \boldsymbol{p} * \boldsymbol{q}^{-1}$ would be a quaternion $\boldsymbol{p}' = (\,0\,,\,\mathbf{p}'\,)^T$ in which the component along the vector $\widehat{\mathbf{u}}$ would remain invariant, while the components in the plane defined by $\{\,\widehat{\mathbf{v}}\,,\,\widehat{\mathbf{w}}\,\}$ would be rotated by an angle θ. That looks exactly like a 3D rotation.

3.3.1.1. From a Quaternion to a Rotation Matrix

The quaternion product $\boldsymbol{q} * \boldsymbol{p} * \boldsymbol{q}^{-1}$ can be expressed in matrix form using equations (3.8):

$$\boldsymbol{q} * \boldsymbol{p} * \boldsymbol{q}^{-1} \quad \equiv \quad [\,\boldsymbol{q}\,]_*\;{}_*[\boldsymbol{q}^{-1}]\;\boldsymbol{p} \quad = \quad \frac{1}{\|\boldsymbol{q}\|^2}\;[\,\boldsymbol{q}\,]_*\;{}_*[\boldsymbol{q}^*]\;\boldsymbol{p} \quad = \tag{3.32a}$$

$$= \quad \frac{1}{\|\boldsymbol{q}\|^2}\begin{pmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{pmatrix}\begin{pmatrix} q_0 & q_1 & q_2 & q_3 \\ -q_1 & q_0 & -q_3 & q_2 \\ -q_2 & q_3 & q_0 & -q_1 \\ -q_3 & -q_2 & q_1 & q_0 \end{pmatrix}\boldsymbol{p} \quad \equiv \tag{3.32b}$$

$$\equiv \quad {}_*[\boldsymbol{q}^{-1}]\;[\,\boldsymbol{q}\,]_*\;\boldsymbol{p} \quad = \quad \frac{1}{\|\boldsymbol{q}\|^2}\;{}_*[\boldsymbol{q}^*]\;[\,\boldsymbol{q}\,]_*\;\boldsymbol{p} \quad = \tag{3.32c}$$

$$= \quad \frac{1}{\|\boldsymbol{q}\|^2}\begin{pmatrix} q_0 & q_1 & q_2 & q_3 \\ -q_1 & q_0 & -q_3 & q_2 \\ -q_2 & q_3 & q_0 & -q_1 \\ -q_3 & -q_2 & q_1 & q_0 \end{pmatrix}\begin{pmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{pmatrix}\boldsymbol{p} \quad = \tag{3.32d}$$

$$= \quad \frac{1}{\|\boldsymbol{q}\|^2}\begin{pmatrix} \|\boldsymbol{q}\|^2 & 0 & 0 & 0 \\ 0 & q_1^2+q_0^2-q_3^2-q_2^2 & 2\,(\,q_1\,q_2-q_0\,q_3\,) & 2\,(\,q_1\,q_3+q_0\,q_2\,) \\ 0 & 2\,(\,q_2\,q_1+q_3\,q_0\,) & q_2^2-q_3^2+q_0^2-q_1^2 & 2\,(\,q_2\,q_3-q_0\,q_1\,) \\ 0 & 2\,(\,q_3\,q_1-q_2\,q_0\,) & 2\,(\,q_3\,q_2+q_1\,q_0\,) & q_3^2-q_2^2-q_1^2+q_0^2 \end{pmatrix}\boldsymbol{p} \quad . \tag{3.32e}$$

If $\boldsymbol{q}$ happens to be a unit quaternion, then we obtain the 3D rotation matrix:

$$\mathbf{R}(\,\boldsymbol{q}\,) \quad = \quad \begin{pmatrix} 1-2\,(\,q_2^2+q_3^2\,) & 2\,(\,q_1\,q_2-q_0\,q_3\,) & 2\,(\,q_1\,q_3+q_0\,q_2\,) \\ 2\,(\,q_2\,q_1+q_3\,q_0\,) & 1-2\,(\,q_1^2+q_3^2\,) & 2\,(\,q_2\,q_3-q_0\,q_1\,) \\ 2\,(\,q_3\,q_1-q_2\,q_0\,) & 2\,(\,q_3\,q_2+q_1\,q_0\,) & 1-2\,(\,q_1^2+q_2^2\,) \end{pmatrix} \quad . \tag{3.33}$$

Note that both $\boldsymbol{q}$ and $-\boldsymbol{q}$ represent the same 3D rotation: $\mathbf{R}(\,\boldsymbol{q}\,) = \mathbf{R}(\,-\boldsymbol{q}\,)$.

3.4. Unit Quaternions as Random Variables

When dealing with the Kalman filter, the distribution of a random variable $\mathbf{x}$ is encoded by its mean $\overline{\mathbf{x}} = \mathrm{E}[\,\mathbf{x}\,]$ and its covariance matrix $\mathbf{P}$ defined as

$$\mathbf{P} \quad = \quad \mathrm{E}\Big[\,(\mathbf{x}-\overline{\mathbf{x}})\,(\mathbf{x}-\overline{\mathbf{x}})^T\,\Big] \quad . \tag{3.34}$$

This definition makes sense when our random variables are defined in the Euclidean space. However, unit quaternions does not live in the Euclidean space. Unit quaternions form a group under

multiplication, but not under addition. This means that the addition of two unit quaternions does not result in another unit quaternion. Therefore, the addition of two unit quaternions does not represent a rotation, which causes the definitions of mean and covariance matrix to lose their meaning. For example, what would be the mean if each quaternion was equiprobable in the unit sphere? How can we define the covariance for unit quaternions if $\boldsymbol{q} - \overline{\boldsymbol{q}}$ does not represent a rotation? We cannot redefine the mean and the covariance matrix, because the Kalman filter uses this precise form in its derivations. However, we know that unit quaternions live in the unit sphere of $\mathbb{R}^4$, S^3. This space is a manifold, and some concepts related to these mathematical objects are useful to define the moments of *random unit quaternions*.

Let us retrieve some important definitions:

Definition 3.1 (**Homeomorphism**). *A* homeomorphism *is a function $f : X \to Y$ between two topological spaces X and Y satisfying the following properties:*

- *f is a bijection,*
- *f is continuous,*
- *its inverse function f^{-1} is continuous.*

If such a function exists, we say that X and Y are homeomorphic.

Definition 3.2 (**Manifold**). *A* n-manifold *M^n is a topological space in which each point is locally homeomorphic to the Euclidean space $\mathbb{R}^n$. This is, each point $x \in M^n$ has a neighborhood $N \subset M^n$ for which we can define a homeomorphism $f : N \to B_n$, with B_n the unit ball of $\mathbb{R}^n$.*

Definition 3.3 (**Chart**). *A* chart *for a manifold M^n, is a homeomorphism φ from an open subset $U \subset M^n$, to an open subset of the Euclidean space $V \subset \mathbb{R}^n$. This is, a chart is a function*

$$\varphi : U \subset M^n \to V \subset \mathbb{R}^n \quad , \tag{3.35}$$

with φ a homeomorphism. Traditionally, a chart is expressed as the pair (U, φ).

Given these definitions we can continue our reasoning.

In [12] it talks about four "attitude error representations". Namely, the one we will call *Orthographic* (O), the *Rodrigues Parameters* (RP), the *Modified Rodrigues Parameters* (MRP), and the *Rotation Vector* (RV). The first three are what we know as *stereographic projections* (and are called *Orthographic*, *Gnomonic*, and *Stereographic* respectively). The last one is a projection called *Equidistant*. But all four are charts defining a homeomorphism from the manifold S^3 to the Euclidean space $\mathbb{R}^3$. This is, they map a point $\boldsymbol{q}$ in the manifold with a point $\mathbf{e}$ in $\mathbb{R}^3$. As these charts are widely used in the literature [12, 13, 15, 16, 41] we will also use them in this work. Table 3.2 arranges their definition, together with their domain and image. We must ensure the charts to be bijections so that they properly define a homeomorphism, and that they do not map $\boldsymbol{q}$ and $-\boldsymbol{q}$ with different points of $\mathbb{R}^3$ since they represent the same rotation. We achieve this by the given definition of the domain and image for each chart.

Figure 3.1 shows how points in the sphere S^2 (subspace of the sphere S^3, where quaternions live) are mapped to points in $\mathbb{R}^2$ (subspace of $\mathbb{R}^3$, where the images of the charts are contained) through each one of the named charts. Since our charts are homeomorphisms, it is possible to invert the functions. Figure 3.2 shows how points from $\mathbb{R}^2$ are mapped to points in the manifold through the inverted charts.

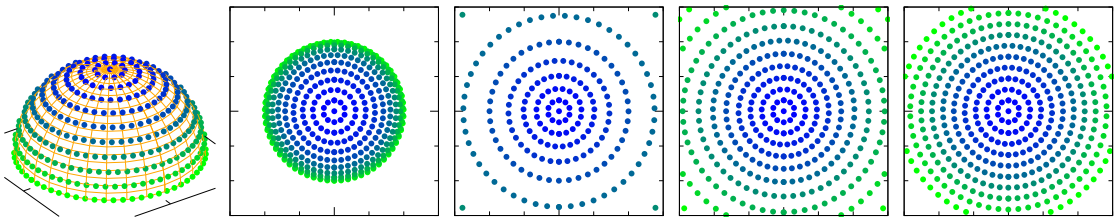
As pointed in [12], all four charts provide the same second-order approximation for a point $\mathbf{e} \in \mathbb{R}^3$ near the origin, to a quaternion $\boldsymbol{q} \in S^3$:

$$\varphi^{-1}(\,\mathbf{e}\,) \quad \approx \quad \left(\, 1 - \frac{\|\mathbf{e}\|^2}{8} \;,\; \frac{\mathbf{e}}{2} \,\right)^T \quad . \tag{3.36}$$

We should notice that having $\mathbb{R}^3$ and S^3 different metrics, a chart φ will inevitably produce a deformation of the space. However, for quaternions in the neighborhood of the identity quaternion (top of the sphere), our charts behave like the identity transformation between the imaginary part of these quaternions, and the points near the origin in $\mathbb{R}^3$, as suggested by (3.36). This is a desirable property, as this means that the space around the identity quaternion closely resembles the Euclidean space, which is the space for which the Kalman filter is designed. But this just

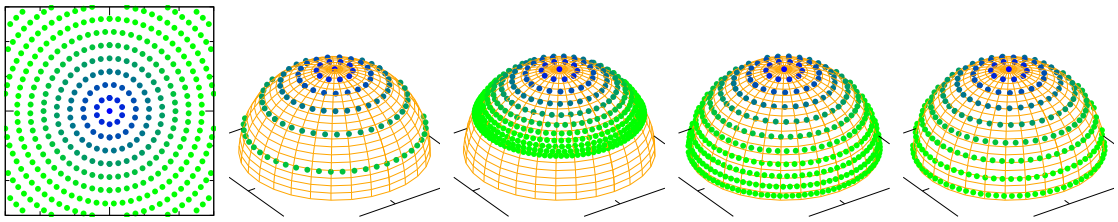
Table 3.2: Main characteristics of the charts studied

Chart	$\mathbf{e} = \boldsymbol{\varphi}(\boldsymbol{q})$	$\boldsymbol{q} = \boldsymbol{\varphi}^{-1}(\mathbf{e})$	**Domain**	**Image**
O	$2\,\mathbf{q}$	$\begin{pmatrix} \sqrt{1-\frac{\|\mathbf{e}\|^2}{4}} \\ \mathbf{e}/2 \end{pmatrix}$	$\{\boldsymbol{q} \in S^3 : q_0 \geq 0\}$	$\{\mathbf{e} \in \mathbb{R}^3 : \|\mathbf{e}\| \leq 2\}$
RP	$2\,\dfrac{\mathbf{q}}{q_0}$	$\dfrac{1}{\sqrt{4+\|\mathbf{e}\|^2}}\begin{pmatrix} 2 \\ \mathbf{e} \end{pmatrix}$	$\{\boldsymbol{q} \in S^3 : q_0 > 0\}$	$\mathbb{R}^3$
MRP	$4\,\dfrac{\mathbf{q}}{1+q_0}$	$\dfrac{1}{16+\|\mathbf{e}\|^2}\begin{pmatrix} 16-\|\mathbf{e}\|^2 \\ 8\,\mathbf{e} \end{pmatrix}$	$\{\boldsymbol{q} \in S^3 : q_0 \geq 0\}$	$\{\mathbf{e} \in \mathbb{R}^3 : \|\mathbf{e}\| \leq 4\}$
RV	$2\,\widehat{\mathbf{q}}\,\arcsin(\|\mathbf{q}\|)$	$\begin{pmatrix} \cos\left(\frac{\|\mathbf{e}\|}{2}\right) \\ \widehat{\mathbf{e}}\,\sin\left(\frac{\|\mathbf{e}\|}{2}\right) \end{pmatrix}$	$\{\boldsymbol{q} \in S^3 : q_0 \geq 0\}$	$\{\mathbf{e} \in \mathbb{R}^3 : \|\mathbf{e}\| \leq \pi\}$



(a) S^2 (b) O (c) RP (d) MRP (e) RV

Figure 3.1: Points in the manifold with $q_3 = 0$ are mapped with points in the Euclidean space through each chart φ.



(a) $\mathbb{R}^2$ (b) O (c) RP (d) MRP (e) RV

Figure 3.2: Points in the Euclidean space with $e_3 = 0$ are mapped with points in the manifold through each chart inverse φ^{-1}.

happens in the neighborhood of the identity quaternion. However, we can extend this property for any quaternion $\overline{\boldsymbol{q}} \in S^3$ noting that any quaternion $\boldsymbol{q} \in S^3$ can be expressed as a "deviation" from the first one through the quaternion product:

$$\boldsymbol{q} \quad = \quad \overline{\boldsymbol{q}} * \boldsymbol{\delta}^{\overline{\boldsymbol{q}}} \quad , \tag{3.37}$$

where $\boldsymbol{\delta}^{\overline{\boldsymbol{q}}}$ represents such a deviation[3]. Then, we define a chart $\varphi_{\overline{\boldsymbol{q}}}$ for each quaternion $\overline{\boldsymbol{q}} \in S^3$ as

$$\mathbf{e}^{\overline{\boldsymbol{q}}} \quad = \quad \varphi_{\overline{\boldsymbol{q}}}(\boldsymbol{q}) \quad = \quad \varphi\left(\boldsymbol{\delta}^{\overline{\boldsymbol{q}}}\right) \quad , \tag{3.38}$$

where $\boldsymbol{\delta}^{\overline{\boldsymbol{q}}} = \overline{\boldsymbol{q}}^* * \boldsymbol{q}$, and where we have denoted the point of the Euclidean space mapped with

[3]This definition is arbitrary: we could have chosen to relate the quaternions through $\boldsymbol{q} = \boldsymbol{\delta}^{\overline{\boldsymbol{q}}} * \overline{\boldsymbol{q}}$, but it is important to establish one of these definitions, and then be consequent with it. However, (3.37) entails some computational advantages that make us prefer it.

the quaternion $\boldsymbol{q} \in S^3$ through the chart $\varphi_{\overline{\boldsymbol{q}}}$ as $\mathbf{e}^{\overline{\boldsymbol{q}}}$. Thus, we will have a set of charts $\{\varphi_{\overline{\boldsymbol{q}}}\}_{\overline{\boldsymbol{q}}}$, each one resembling the Euclidean space around a quaternion $\overline{\boldsymbol{q}} \in S^3$, and mapping this last quaternion to the origin of $\mathbb{R}^3$. Making an abuse of language, we will refer to the Euclidean space associated with the chart $\varphi_{\overline{\boldsymbol{q}}}$ as the $\overline{\boldsymbol{q}}$-centered chart. Thus, the homeomorphism $\varphi_{\overline{\boldsymbol{q}}}^{-1}$ takes a point $\mathbf{e}^{\overline{\boldsymbol{q}}}$ in the $\overline{\boldsymbol{q}}$-centered chart and maps it to a point $\boldsymbol{q}$ in the manifold through

$$\boldsymbol{q} \quad = \quad \varphi_{\overline{\boldsymbol{q}}}^{-1}\big(\,\mathbf{e}^{\overline{\boldsymbol{q}}}\,\big) \quad = \quad \overline{\boldsymbol{q}} * \varphi^{-1}\big(\,\mathbf{e}^{\overline{\boldsymbol{q}}}\,\big) \quad . \tag{3.39}$$

After reviewing these concepts, we can define the mean and the covariance matrix of a distribution of unit quaternions.

3.4.1. Mean and Covariance Matrix of a Unit Quaternion Distribution

Given a unit quaternion $\overline{\boldsymbol{q}}$ and a chart φ, we will define the expected value of a distribution of unit quaternions in the $\overline{\boldsymbol{q}}$-centered chart as

$$\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}} \quad = \quad \mathrm{E}\big[\,\mathbf{e}^{\overline{\boldsymbol{q}}}\,\big] \quad , \tag{3.40}$$

and its covariance matrix as

$$\mathbf{P}^{\overline{\boldsymbol{q}}} \quad = \quad \mathrm{E}\Big[\big(\mathbf{e}^{\overline{\boldsymbol{q}}} - \overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\big)\big(\mathbf{e}^{\overline{\boldsymbol{q}}} - \overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\big)^T\Big] \quad . \tag{3.41}$$

The probability density of each unit quaternion $\boldsymbol{q}$ would be defined through the homeomorphism $\boldsymbol{q} = \varphi_{\overline{\boldsymbol{q}}}^{-1}(\,\mathbf{e}^{\overline{\boldsymbol{q}}}\,)$. Then, a distribution of unit quaternions needs of four mathematical objects to be encoded:

$$\big(\;\varphi\;,\;\overline{\boldsymbol{q}}\;,\;\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\;,\;\mathbf{P}^{\overline{\boldsymbol{q}}}\;\big) \quad . \tag{3.42}$$

Although a distribution of unit quaternions is unique, given this definition, its expected value $\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}$ and its covariance matrix $\mathbf{P}^{\overline{\boldsymbol{q}}}$ may take different values depending on the chosen quaternion $\overline{\boldsymbol{q}}$ and chart φ. However, knowing that the Kalman filter is designed for the Euclidean space, it will be convenient to choose a unit quaternion $\overline{\boldsymbol{q}}$ central in the distribution, so that the manifold space around it closely resembles the most significant region for the covariance matrix in the $\overline{\boldsymbol{q}}$-centered chart. It is particularly convenient to choose a quaternion $\overline{\boldsymbol{q}}$ such that $\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}} = \mathbf{0}$, so that the covariance matrix is centered in the origin of the $\overline{\boldsymbol{q}}$-centered chart.

3.4.2. Transition Maps

At some step of the Kalman filter, we will have a distribution of unit quaternions defined in a $\overline{\boldsymbol{q}}$-centered chart, and we will be interested in expressing our distribution in another $\overline{\boldsymbol{p}}$-centered chart. The concept of transition map is relevant for this purpose.

Definition 3.4 (**Transition map**). *Given two charts $(U_\alpha, \varphi_\alpha)$ and (U_β, φ_β) for a manifold M, with $U_{\alpha\beta} = U_\alpha \cap U_\beta \neq \emptyset$, a function $\varphi_{\alpha\beta} : \varphi_\alpha(\,U_{\alpha\beta}\,) \to \varphi_\beta(\,U_{\alpha\beta}\,)$ can be defined as*

$$\varphi_{\alpha\beta}(x) \quad = \quad \varphi_\beta\left(\varphi_\alpha^{-1}(x)\right), \tag{3.43}$$

with $x \in \varphi_\alpha(\,U_{\alpha\beta}\,)$. The function $\varphi_{\alpha\beta}$ is called a transition map. *Being that φ_α and φ_β are homeomorphisms, so is $\varphi_{\alpha\beta}$.*

For the present case, let us consider two unit quaternions $\overline{\boldsymbol{p}}$ and $\overline{\boldsymbol{q}}$ both related through

$$\overline{\boldsymbol{p}} \quad = \quad \overline{\boldsymbol{q}} * \overline{\boldsymbol{\delta}} \quad . \tag{3.44}$$

These two quaternions define the charts $\varphi_{\overline{\boldsymbol{p}}}$ and $\varphi_{\overline{\boldsymbol{q}}}$. We build the transition map that relates a point $\mathbf{e}^{\overline{\boldsymbol{q}}}$ expressed in the $\overline{\boldsymbol{q}}$-centered chart with a point $\mathbf{e}^{\overline{\boldsymbol{p}}}$ expressed in the $\overline{\boldsymbol{p}}$-centered chart doing

$$\mathbf{e}^{\overline{\boldsymbol{p}}} \quad = \quad \varphi_{\overline{\boldsymbol{p}}}\Big(\,\varphi_{\overline{\boldsymbol{q}}}^{-1}\big(\,\mathbf{e}^{\overline{\boldsymbol{q}}}\,\big)\,\Big) \quad = \tag{3.45a}$$

$$= \quad \varphi\Big(\,\overline{\boldsymbol{p}}^{*} * \overline{\boldsymbol{q}} * \varphi^{-1}\big(\,\mathbf{e}^{\overline{\boldsymbol{q}}}\,\big)\,\Big) \quad = \tag{3.45b}$$

$$= \quad \varphi\Big(\,\overline{\boldsymbol{\delta}}^{*} * \varphi^{-1}\big(\,\mathbf{e}^{\overline{\boldsymbol{q}}}\,\big)\,\Big) \quad . \tag{3.45c}$$

That is to say, first we take the point $\mathbf{e}^{\overline{\boldsymbol{q}}}$ in the $\overline{\boldsymbol{q}}$-centered chart, and we obtain its associated quaternion $\boldsymbol{q}$ in the manifold using $\varphi_{\overline{\boldsymbol{q}}}^{-1}$. Then, we transform this quaternion $\boldsymbol{q}$ to a point $\mathbf{e}^{\overline{\boldsymbol{p}}}$ in the $\overline{\boldsymbol{p}}$-centered chart. Nevertheless, knowing the quaternion $\overline{\boldsymbol{\delta}}$ we do not need to explicitly compute $\boldsymbol{q}$. In fact, being able to express the same quaternion $\boldsymbol{q}$ as two different deviations,

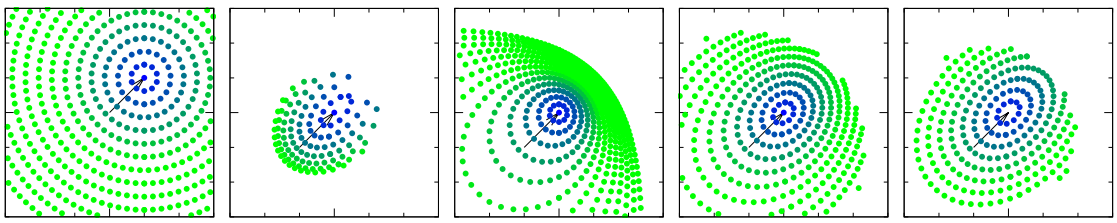
$$\left.\begin{array}{rcl} \boldsymbol{q} & = & \overline{\boldsymbol{q}} * \boldsymbol{\delta}^{\overline{\boldsymbol{q}}} \\ \boldsymbol{q} & = & \overline{\boldsymbol{p}} * \boldsymbol{\delta}^{\overline{\boldsymbol{p}}} \end{array}\right\} \quad \Longrightarrow \quad \boldsymbol{\delta}^{\overline{\boldsymbol{p}}} \;=\; \underbrace{\overline{\boldsymbol{p}}^{*} * \overline{\boldsymbol{q}}}_{\overline{\boldsymbol{\delta}}^{*}} * \boldsymbol{\delta}^{\overline{\boldsymbol{q}}} \quad . \tag{3.46}$$

Note the equivalence of expressions (3.45c) and (3.46).

Table 3.3 displays the transition maps for the charts studied. The detailed derivations of these transition maps can be found in Appendix A.5. Figure 3.3 attempts to provide some insight into how points are transformed through the transition map of each chart.

Table 3.3: Transition maps for the charts studied

Chart	**Transition Map** $\mathbf{e}^{\overline{\boldsymbol{p}}}\big(\,\mathbf{e}^{\overline{\boldsymbol{q}}}\,\big)$
O	$\overline{\delta}_0\,\mathbf{e}^{\overline{\boldsymbol{q}}} \;-\; \sqrt{4 - \|\mathbf{e}^{\overline{\boldsymbol{q}}}\|^2}\;\overline{\boldsymbol{\delta}} \;-\; \overline{\boldsymbol{\delta}} \times \mathbf{e}^{\overline{\boldsymbol{q}}}$
RP	$2\,\dfrac{\overline{\delta}_0\,\mathbf{e}^{\overline{\boldsymbol{q}}} \;-\; 2\,\overline{\boldsymbol{\delta}} \;-\; \overline{\boldsymbol{\delta}} \times \mathbf{e}^{\overline{\boldsymbol{q}}}}{2\,\overline{\delta}_0 \;+\; \overline{\boldsymbol{\delta}} \cdot \mathbf{e}^{\overline{\boldsymbol{q}}}}$
MRP	$4\,\dfrac{8\,\overline{\delta}_0\,\mathbf{e}^{\overline{\boldsymbol{q}}} \;-\; (16 - \|\mathbf{e}^{\overline{\boldsymbol{q}}}\|^2)\,\overline{\boldsymbol{\delta}} \;-\; 8\,\overline{\boldsymbol{\delta}} \times \mathbf{e}^{\overline{\boldsymbol{q}}}}{16 + \|\mathbf{e}^{\overline{\boldsymbol{q}}}\|^2 \;+\; \overline{\delta}_0\,(16 - \|\mathbf{e}^{\overline{\boldsymbol{q}}}\|^2) \;+\; 8\,\overline{\boldsymbol{\delta}} \cdot \mathbf{e}^{\overline{\boldsymbol{q}}}}$
RV	$2\,\dfrac{\boldsymbol{\delta}^{\overline{\boldsymbol{p}}}}{\|\boldsymbol{\delta}^{\overline{\boldsymbol{p}}}\|}\,\arcsin\|\boldsymbol{\delta}^{\overline{\boldsymbol{p}}}\| \quad ,$ with $\quad \boldsymbol{\delta}^{\overline{\boldsymbol{p}}} \;=\; \overline{\delta}_0\,\widehat{\mathbf{e}}^{\overline{\boldsymbol{q}}}\,\sin\left(\frac{\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|}{2}\right) \;-\; \cos\left(\frac{\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|}{2}\right)\overline{\boldsymbol{\delta}} \;-\; \overline{\boldsymbol{\delta}} \times \widehat{\mathbf{e}}^{\overline{\boldsymbol{q}}}\,\sin\left(\frac{\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|}{2}\right)$



(a) $\mathbb{R}^2$ (b) O (c) RP (d) MRP (e) RV

Figure 3.3: Points in a $\overline{\boldsymbol{q}}$-centered chart are transformed using the transition map defined by each chart, and travel to the chart centered in the quaternion mapped with $\mathbf{e}^{\overline{\boldsymbol{q}}} = (1, 1, 0)^T$ in the previous chart.

Chapter 4

System Model

4.1. Introduction

To describe the dynamic state of an object whose structure is fixed or its deformations negligible, we generally use the rigid body model. A rigid body is a system of particles whose positions are constant with respect to some reference frame. This restriction allows expressing the position of each particle through the position and orientation of such reference frame. It also allows to describe the dynamic state of a rigid body using 4 mathematical objects: two of them representing its position and velocity, and the other two representing its orientation and angular velocity. Both position, velocity, and angular velocity are represented by vectors. However, the orientation is represented by a rotation transformation: here is where quaternions come into the scene.

On the other hand, sensors usually do not provide explicit information about the state; instead, their measurements contain implicit information about it. The model of a sensor offers a relation between its measurements and the state of the rigid body. We will use these relations to estimate the dynamic state by extracting the information that underlies the measurements.

In this chapter we will begin by introducing the nomenclature in Section 4.2. Then, in Section 4.3, we will derive the differential equations that describe the evolution of the angular state of a rigid body. Finally, in Section 4.4, we will derive the equations that model the measurements of an IMU: accelerometer, gyroscope, and magnetometer.

4.2. Nomenclature

Table 4.1 summarizes the notation used thorough this chapter and the following one.

Table 4.1: Notation summary

Object	Representation	Description
Vector	$\mathbf{v}$	Bold upright lower case symbols
Matrix	$\mathbf{M}$	Bold upright capital symbols
Quaternion	$\boldsymbol{q}$	Bold italic lower case symbols

A vector $\mathbf{v}$ expressed in a reference frame $\mathcal{B}$ will be represented as ${}^{B}\mathbf{v}$. The same vector $\mathbf{v}$ expressed in a reference frame $\mathcal{S}$ will be represented as ${}^{S}\mathbf{v}$. Both vector expressions are related through a rotation transformation represented by a unit quaternion $\boldsymbol{q}^{BS}$:

$$ {}^{B}\mathbf{v} \quad = \quad \mathbf{R}(\,\boldsymbol{q}^{BS}\,)\;{}^{S}\mathbf{v} \quad . \tag{4.1} $$

The unit quaternion $\boldsymbol{q}^{BS}$ represents the rotation transformation that describes the orientation of reference frame $\mathcal{S}$ with respect to reference frame $\mathcal{B}$.

The position of reference frame $\mathcal{B}$ relative to reference frame $\mathcal{O}$ will be represented by a vector $\mathbf{x}^{OB}$. If the vector $\mathbf{x}^{OB}$ is expressed in reference frame $\mathcal{O}$, then we will write ${}^{O}\mathbf{x}^{OB}$. Figure 4.1

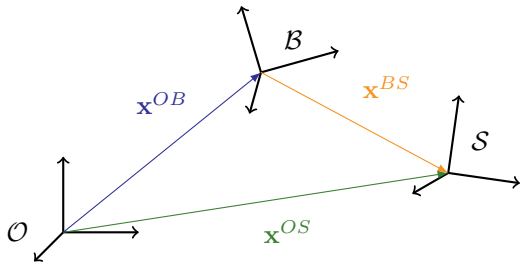


Figure 4.1: Illustration for introducing the nomenclature.

is useful to verify the following equation:

$$
\begin{aligned}
{}^{O}\mathbf{x}^{OS} &= {}^{O}\mathbf{x}^{OB} + {}^{O}\mathbf{x}^{BS} = && \text{(4.2a)}\\
&= {}^{O}\mathbf{x}^{OB} + \mathbf{R}(\boldsymbol{q}^{OB})\ {}^{B}\mathbf{x}^{BS} \quad . && \text{(4.2b)}
\end{aligned}
$$

A vector that represents a quantity measured in reference frame $\mathcal{B}$ will be represented as $\mathbf{v}^{B}$. For example, the magnetic field measured in reference frame $\mathcal{B}$ could be represented as $\mathbf{b}^{B}$. Depending on the reference frame used to express the magnetic field, we could write ${}^{B}\mathbf{b}^{B}$ or ${}^{S}\mathbf{b}^{B}$, but in both cases it would represent the same vector. Similarly, the angular velocity of reference frame $\mathcal{B}$ is represented as $\boldsymbol{\omega}^{B}$.

4.3. Rigid Body

4.3.1. Differential Equation for the Orientation

The concept of temporal derivative of the orientation, originates when a rotating reference frame is considered. Let us analyze the case of a vector $\mathbf{v}$ which is constant when we express it in the inertial reference frame $\mathcal{O}$. The same vector $\mathbf{v}$ will not be constant if we express it in the rotating reference frame $\mathcal{B}$. Figure 4.2 helps to visualize this situation.

(a) Now (t) (b) Later ($t+\Delta t$)

Figure 4.2: A rotating reference frame at two different times.

Figure 4.2 also illustrates that ${}^{B'}\mathbf{u} = {}^{B}\mathbf{v}$. Besides, we observe that if reference frame $\mathcal{B}$ is spinning with a constant angular velocity $\boldsymbol{\omega}^{B}$, then

$$
{}^{B'}\mathbf{u} = \mathbf{R}\Big(\boldsymbol{q}\big({}^{B}\boldsymbol{\omega}^{B}, \Delta t\big)\Big)\ {}^{B'}\mathbf{v} \quad , \tag{4.3}
$$

with

$$
\boldsymbol{q}\big({}^{B}\boldsymbol{\omega}^{B}, \Delta t\big) = \begin{pmatrix} \cos\left(\frac{\|{}^{B}\boldsymbol{\omega}^{B}\|\,\Delta t}{2}\right) \\ \frac{{}^{B}\boldsymbol{\omega}^{B}}{\|{}^{B}\boldsymbol{\omega}^{B}\|}\ \sin\left(\frac{\|{}^{B}\boldsymbol{\omega}^{B}\|\,\Delta t}{2}\right) \end{pmatrix} \quad . \tag{4.4}
$$

Note that in this particular case ${}^{B}\boldsymbol{\omega}^{B} = {}^{B'}\boldsymbol{\omega}^{B}$. All in all, we can write

$$
\begin{aligned}
{}^{O}\mathbf{v} &= \mathbf{R}(\boldsymbol{q}^{OB})\ {}^{B}\mathbf{v} = && \text{(4.5a)}\\
&= \mathbf{R}(\boldsymbol{q}^{OB})\ {}^{B'}\mathbf{u} = && \text{(4.5b)}\\
&= \mathbf{R}(\boldsymbol{q}^{OB})\ \mathbf{R}\Big(\boldsymbol{q}\big({}^{B}\boldsymbol{\omega}^{B}, \Delta t\big)\Big)\ {}^{B'}\mathbf{v} = && \text{(4.5c)}\\
&= \mathbf{R}(\boldsymbol{q}^{OB'})\ {}^{B'}\mathbf{v} \quad , && \text{(4.5d)}
\end{aligned}
$$

and noting that $\boldsymbol{q}^{OB'} = \boldsymbol{q}^{OB}_{t+\Delta t}$, we obtain

$$\boldsymbol{q}^{OB}_{t+\Delta t} \;=\; \boldsymbol{q}^{OB}_{t} \,*\, \boldsymbol{q}\big({}^{B}\boldsymbol{\omega}^{B}, \Delta t\big) \quad . \tag{4.6}$$

Using equation (4.6) we can apply the derivative definition:

$$\frac{d\,\boldsymbol{q}^{OB}}{dt} \;=\; \lim_{\Delta t\to 0} \frac{\boldsymbol{q}^{OB}_{t+\Delta t} - \boldsymbol{q}^{OB}_{t}}{\Delta t} \;= \tag{4.7a}$$

$$= \; \boldsymbol{q}^{OB}_{t} \,*\, \lim_{\Delta t\to 0} \frac{\boldsymbol{q}\big({}^{B}\boldsymbol{\omega}^{B}, \Delta t\big) - 1}{\Delta t} \;= \tag{4.7b}$$

$$= \; \boldsymbol{q}^{OB}_{t} \,*\, \lim_{\Delta t\to 0} \begin{pmatrix} -\frac{\|{}^{B}\boldsymbol{\omega}^{B}\|^2\,\Delta t}{8} + \mathcal{O}\big((\Delta t)^4\big) \\ \frac{{}^{B}\boldsymbol{\omega}^{B}}{\|{}^{B}\boldsymbol{\omega}^{B}\|}\left[\frac{\|{}^{B}\boldsymbol{\omega}^{B}\|}{2} + \mathcal{O}\big((\Delta t)^3\big)\right] \end{pmatrix} \;= \tag{4.7c}$$

$$= \; \frac{1}{2}\,\boldsymbol{q}^{OB}_{t} \,*\, \begin{pmatrix} 0 \\ {}^{B}\boldsymbol{\omega}^{B} \end{pmatrix} \quad . \tag{4.7d}$$

4.3.2. Time Derivative of a Vector in a Non-Inertial Reference Frame

Having found (4.7) it is straightforward to relate two time derivatives of the same vector: one for the vector expressed in an inertial reference frame, and another for the vector expressed in a non-inertial reference frame. Denoting $\boldsymbol{q} = \boldsymbol{q}^{OB}$, $\boldsymbol{\omega} = \begin{pmatrix} 0 \\ {}^{B}\boldsymbol{\omega}^{B} \end{pmatrix}$, $\boldsymbol{v} = \begin{pmatrix} 0 \\ {}^{O}\mathbf{v} \end{pmatrix}$, and $\boldsymbol{v}' = \begin{pmatrix} 0 \\ {}^{B}\mathbf{v} \end{pmatrix}$, and starting from $\boldsymbol{v} = \boldsymbol{q} * \boldsymbol{v}' * \boldsymbol{q}^{*}$, we have

$$\dot{\boldsymbol{v}} \;=\; \dot{\boldsymbol{q}} * \boldsymbol{v}' * \boldsymbol{q}^{*} \;+\; \boldsymbol{q} * \dot{\boldsymbol{v}}' * \boldsymbol{q}^{*} \;+\; \boldsymbol{q} * \boldsymbol{v}' * \dot{\boldsymbol{q}}^{*} \;= \tag{4.8a}$$

$$= \; \frac{1}{2}\,\boldsymbol{q} * \boldsymbol{\omega} * \boldsymbol{v}' * \boldsymbol{q}^{*} \;+\; \boldsymbol{q} * \dot{\boldsymbol{v}}' * \boldsymbol{q}^{*} \;+\; \frac{1}{2}\,\boldsymbol{q} * \boldsymbol{v}' * \boldsymbol{\omega}^{*} * \boldsymbol{q}^{*} \;= \tag{4.8b}$$

$$= \; \boldsymbol{q} * \Big[\, \frac{1}{2}\,\boldsymbol{\omega} * \boldsymbol{v}' \;+\; \dot{\boldsymbol{v}}' \;+\; \frac{1}{2}\,\boldsymbol{v}' * \boldsymbol{\omega}^{*} \,\Big] * \boldsymbol{q}^{*} \;= \tag{4.8c}$$

$$= \; \boldsymbol{q} * \Big[\, \frac{1}{2}\begin{pmatrix} 0 \\ {}^{B}\boldsymbol{\omega}^{B} \end{pmatrix} * \begin{pmatrix} 0 \\ {}^{B}\mathbf{v} \end{pmatrix} \;+\; \begin{pmatrix} 0 \\ {}^{B}\dot{\mathbf{v}} \end{pmatrix} \;+\; \frac{1}{2}\begin{pmatrix} 0 \\ {}^{B}\mathbf{v} \end{pmatrix} * \begin{pmatrix} 0 \\ -{}^{B}\boldsymbol{\omega}^{B} \end{pmatrix} \Big] * \boldsymbol{q}^{*} \;= \tag{4.8d}$$

$$= \; \boldsymbol{q} * \left[\begin{pmatrix} 0 \\ {}^{B}\dot{\mathbf{v}} \end{pmatrix} \;+\; \frac{1}{2}\begin{pmatrix} -{}^{B}\boldsymbol{\omega}^{B} \cdot {}^{B}\mathbf{v} \;+\; {}^{B}\mathbf{v} \cdot {}^{B}\boldsymbol{\omega}^{B} \\ {}^{B}\boldsymbol{\omega}^{B} \times {}^{B}\mathbf{v} \;-\; {}^{B}\mathbf{v} \times {}^{B}\boldsymbol{\omega}^{B} \end{pmatrix} \right] * \boldsymbol{q}^{*} \;= \tag{4.8e}$$

$$= \; \boldsymbol{q} * \Big[\begin{pmatrix} 0 \\ {}^{B}\dot{\mathbf{v}} \end{pmatrix} \;+\; \begin{pmatrix} 0 \\ {}^{B}\boldsymbol{\omega}^{B} \times {}^{B}\mathbf{v} \end{pmatrix} \Big] * \boldsymbol{q}^{*} \quad , \tag{4.8f}$$

where we have applied the "over-dot" notation for the time derivative. Using matrix notation, we conclude:

$$\frac{d\,{}^{O}\mathbf{v}}{dt} \;=\; \frac{d\;\mathbf{R}(\,\boldsymbol{q}^{OB}\,)\,{}^{B}\mathbf{v}}{dt} \;= \tag{4.9a}$$

$$= \; \mathbf{R}(\,\boldsymbol{q}^{OB}\,)\,\Big[\, \frac{d\,{}^{B}\mathbf{v}}{dt} \;+\; {}^{B}\boldsymbol{\omega}^{B} \times {}^{B}\mathbf{v} \,\Big] \quad . \tag{4.9b}$$

4.3.3. Differential Equation for the Angular Velocity

In classical mechanics, the *angular momentum* of a particle system $\mathcal{P}$ relative to an inertial reference frame $\mathcal{O}$ is defined as

$${}^{O}\mathbf{L}^{OP} \;=\; \sum_{i} {}^{O}\mathbf{L}^{Oi} \;=\; \sum_{i} {}^{O}\mathbf{x}^{Oi} \times {}^{O}\mathbf{p}^{Oi} \quad , \tag{4.10}$$

where ${}^{O}\mathbf{x}^{Oi}$ is the position of the i-th particle, and ${}^{O}\mathbf{p}^{Oi} = m_i\,{}^{O}\mathbf{v}^{Oi}$ is its *linear momentum* (with ${}^{O}\mathbf{v}^{Oi}$ its linear velocity) both measured and expressed in $\mathcal{O}$. The *torque* of the particle

system is defined as

$$ {}^{O}\boldsymbol{\tau}^{OP} = \frac{d\,{}^{O}\mathbf{L}^{OP}}{dt} = \tag{4.11a} $$

$$ = \sum_i \underbrace{\frac{d\,{}^{O}\mathbf{x}^{Oi}}{dt} \times {}^{O}\mathbf{p}^{Oi}}_{0} + {}^{O}\mathbf{x}^{Oi} \times \frac{d\,{}^{O}\mathbf{p}^{Oi}}{dt} = \tag{4.11b} $$

$$ = \sum_i {}^{O}\mathbf{x}^{Oi} \times {}^{O}\mathbf{F}^{i} \quad . \tag{4.11c} $$

A rigid body is a particle system in which the position of each particle is constant when measured and expressed in some reference frame $\mathcal{B}$ ($\frac{d\,{}^{B}\mathbf{x}^{Bi}}{dt} = 0$). In such a case we can use (4.9) to obtain

$$ {}^{O}\mathbf{x}^{Oi} = {}^{O}\mathbf{x}^{OB} + {}^{O}\mathbf{x}^{Bi} \quad \Longrightarrow \tag{4.12a} $$

$$ \Longrightarrow \quad \frac{d\,{}^{O}\mathbf{x}^{Oi}}{dt} = \frac{d\,{}^{O}\mathbf{x}^{OB}}{dt} + \frac{d\,{}^{O}\mathbf{x}^{Bi}}{dt} = \tag{4.12b} $$

$$ = {}^{O}\mathbf{v}^{OB} + \mathbf{R}(\boldsymbol{q}^{OB}) \left[\underbrace{\frac{d\,{}^{B}\mathbf{x}^{Bi}}{dt}}_{0} + {}^{B}\boldsymbol{\omega}^{B} \times {}^{B}\mathbf{x}^{Bi} \right] \quad . \tag{4.12c} $$

Then, the angular momentum ${}^{O}\mathbf{L}^{OP}$ can be re-expressed as

$$ {}^{O}\mathbf{L}^{OP} = \sum_i \left({}^{O}\mathbf{x}^{OB} + {}^{O}\mathbf{x}^{Bi} \right) \times m_i \left({}^{O}\mathbf{v}^{OB} + \mathbf{R}(\boldsymbol{q}^{OB}) \left[{}^{B}\boldsymbol{\omega}^{B} \times {}^{B}\mathbf{x}^{Bi} \right] \right) = \tag{4.13a} $$

$$ \begin{aligned} = {}& {}^{O}\mathbf{x}^{OB} \times {}^{O}\mathbf{v}^{OB} \left(\sum_i m_i \right) + \left(\sum_i m_i\, {}^{O}\mathbf{x}^{Bi} \right) \times {}^{O}\mathbf{v}^{OB} + \\ & + {}^{O}\mathbf{x}^{OB} \times \mathbf{R}(\boldsymbol{q}^{OB}) \left[{}^{B}\boldsymbol{\omega}^{B} \times \left(\sum_i m_i\, {}^{B}\mathbf{x}^{Bi} \right) \right] + \\ & + \sum_i m_i\, {}^{O}\mathbf{x}^{Bi} \times \left(\mathbf{R}(\boldsymbol{q}^{OB}) \left[{}^{B}\boldsymbol{\omega}^{B} \times {}^{B}\mathbf{x}^{Bi} \right] \right) \quad . \end{aligned} \tag{4.13b} $$

If we define the position of $\mathcal{B}$ to be the *center of mass*:

$$ {}^{O}\mathbf{x}^{OB} = \frac{\sum_i m_i\, {}^{O}\mathbf{x}^{Oi}}{\sum_i m_i} = \tag{4.14a} $$

$$ = \frac{\sum_i m_i \left({}^{O}\mathbf{x}^{OB} + {}^{O}\mathbf{x}^{Bi} \right)}{\sum_i m_i} \quad , \tag{4.14b} $$

then $\left(\sum_i m_i\, {}^{O}\mathbf{x}^{Bi} \right) = \left(\sum_i m_i\, {}^{B}\mathbf{x}^{Bi} \right) = 0$, and the angular momentum in (4.13) can be decomposed in two terms:

$$ {}^{O}\mathbf{L}^{OP} = {}^{O}\mathbf{L}^{OP}_{orbital} + {}^{O}\mathbf{L}^{OP}_{internal} \quad , \tag{4.15} $$

being

$$ {}^{O}\mathbf{L}^{OP}_{orbital} = {}^{O}\mathbf{x}^{OB} \times {}^{O}\mathbf{p}^{OB} \quad , \tag{4.16} $$

$$ {}^{O}\mathbf{L}^{OP}_{internal} = \sum_i m_i\, {}^{O}\mathbf{x}^{Bi} \times \left(\mathbf{R}(\boldsymbol{q}^{OB}) \left[{}^{B}\boldsymbol{\omega}^{B} \times {}^{B}\mathbf{x}^{Bi} \right] \right) \quad , \tag{4.17} $$

with ${}^{O}\mathbf{p}^{OB} = \left(\sum_i m_i \right) {}^{O}\mathbf{v}^{OB}$. Each term is related to the torque (see equation (4.11)) through

$$ \frac{d\,{}^{O}\mathbf{L}^{OP}_{orbital}}{dt} = {}^{O}\boldsymbol{\tau}^{OP}_{orbital} = {}^{O}\mathbf{x}^{OB} \times \sum_i {}^{O}\mathbf{F}^{i} \quad , \tag{4.18} $$

$$ \frac{d\,{}^{O}\mathbf{L}^{OP}_{internal}}{dt} = {}^{O}\boldsymbol{\tau}^{OP}_{internal} = \sum_i {}^{O}\mathbf{x}^{Bi} \times {}^{O}\mathbf{F}^{i} \quad . \tag{4.19} $$

Using (A.39), we can rewrite (4.17) as

$$
\begin{aligned}
{}^{O}\mathbf{L}^{OP}_{internal} &= \sum_i m_i \left(\mathbf{R}(\,\boldsymbol{q}^{OB}\,)\,{}^{B}\mathbf{x}^{Bi} \right) \times \left(\mathbf{R}(\,\boldsymbol{q}^{OB}\,) \left[\,{}^{B}\boldsymbol{\omega}^{B} \times {}^{B}\mathbf{x}^{Bi}\,\right] \right) \quad = && \text{(4.20a)}\\
&= \mathbf{R}(\,\boldsymbol{q}^{OB}\,) \left(\sum_i m_i\,{}^{B}\mathbf{x}^{Bi} \times \left({}^{B}\boldsymbol{\omega}^{B} \times {}^{B}\mathbf{x}^{Bi} \right) \right) \quad = && \text{(4.20b)}\\
&= \mathbf{R}(\,\boldsymbol{q}^{OB}\,) \left(-\sum_i m_i \left[{}^{B}\mathbf{x}^{Bi}\right]^2_{\times} \right) {}^{B}\boldsymbol{\omega}^{B} \quad = && \text{(4.20c)}\\
&= \mathbf{R}(\,\boldsymbol{q}^{OB}\,)\;\mathbf{I}_B\;{}^{B}\boldsymbol{\omega}^{B} \quad . && \text{(4.20d)}
\end{aligned}
$$

The 3×3 matrix $\mathbf{I}_B$ is known as the *moment of inertia* of the rigid body. Similarly, we can rewrite (4.19) as

$$
\begin{aligned}
{}^{O}\boldsymbol{\tau}^{OP}_{internal} &= \sum_i \left(\mathbf{R}(\,\boldsymbol{q}^{OB}\,)\,{}^{B}\mathbf{x}^{Bi} \right) \times \left(\mathbf{R}(\,\boldsymbol{q}^{OB}\,)\,{}^{B}\mathbf{F}^{i} \right) \quad = && \text{(4.21a)}\\
&= \mathbf{R}(\,\boldsymbol{q}^{OB}\,) \sum_i {}^{B}\mathbf{x}^{Bi} \times {}^{B}\mathbf{F}^{i} \quad . && \text{(4.21b)}
\end{aligned}
$$

Finally, after applying (4.9) to (4.20d), we obtain

$$
\frac{d\,{}^{O}\mathbf{L}^{OP}_{internal}}{dt} = \mathbf{R}(\,\boldsymbol{q}^{OB}\,) \left(\frac{d\left(\mathbf{I}_B\,{}^{B}\boldsymbol{\omega}^{B}\right)}{dt} + {}^{B}\boldsymbol{\omega}^{B} \times \left(\mathbf{I}_B\,{}^{B}\boldsymbol{\omega}^{B}\right) \right) \quad , \tag{4.22}
$$

therefore, equation (4.19) becomes

$$
\mathbf{I}_B\,\frac{d\,{}^{B}\boldsymbol{\omega}^{B}}{dt} + {}^{B}\boldsymbol{\omega}^{B} \times \left(\mathbf{I}_B\,{}^{B}\boldsymbol{\omega}^{B}\right) = \sum_i {}^{B}\mathbf{x}^{Bi} \times {}^{B}\mathbf{F}^{i} \quad . \tag{4.23}
$$

4.3.3.1. Differential Equation for the Angular Velocity of a General Rigid Body

The moment of inertia $\mathbf{I}_B$ of a rigid body $\mathcal{B}$ depends on its specific mass distribution. That means that each rigid body has its own version of (4.23). If we do not know the mass distribution of the rigid body, or we are looking for an equation that is approximately valid for a large number of rigid bodies, perhaps the most appropriate thing would be to choose the moment of inertia of a homogeneous mass distribution; a moment of inertia proportional to the identity: $\mathbf{I}_B = \alpha\,\mathbf{I}$. In such a case (4.23) becomes

$$
\frac{d\,{}^{B}\boldsymbol{\omega}^{B}}{dt} = {}^{B}\widetilde{\boldsymbol{\tau}}^{BP} \quad , \tag{4.24}
$$

with ${}^{B}\widetilde{\boldsymbol{\tau}}^{BP} = \frac{1}{\alpha}\,{}^{B}\boldsymbol{\tau}^{BP} = \frac{1}{\alpha}\sum_i {}^{B}\mathbf{x}^{Bi} \times {}^{B}\mathbf{F}^{i}$.

4.4. Sensors

4.4.1. Gyroscope Measurement

A gyroscope measures its angular velocity with respect to a reference frame $\mathcal{S}$ like the one depicted in Figure 4.1. Let us consider a gyroscope fixed to a rigid body represented by a reference frame $\mathcal{B}$. We know that the angular velocity of the rigid body expressed in its reference frame, ${}^{B}\boldsymbol{\omega}^{B}$, is related to the angular velocity expressed in the reference frame of the sensor, ${}^{S}\boldsymbol{\omega}^{B}$, through

$$
{}^{B}\boldsymbol{\omega}^{B} = \mathbf{R}(\,\boldsymbol{q}^{BS}\,)\,{}^{S}\boldsymbol{\omega}^{B} \quad . \tag{4.25}
$$

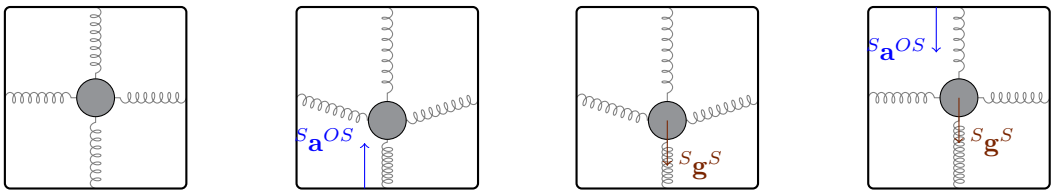
Then, the measurement of a gyroscope can be expressed as

$$
\begin{aligned}
\boldsymbol{\omega}^{m} &= {}^{S}\boldsymbol{\omega}^{B} + \mathbf{r}^{\omega} \quad = && \text{(4.26a)}\\
&= \mathbf{R}^{T}(\,\boldsymbol{q}^{BS}\,)\,{}^{B}\boldsymbol{\omega}^{B} + \mathbf{r}^{\omega} \quad , && \text{(4.26b)}
\end{aligned}
$$

being $\mathbf{r}^{\omega}$ the measurement noise of the gyroscope. Therefore, gyroscope measurements will provide information about the angular velocity of the rigid body, ${}^{B}\boldsymbol{\omega}^{B}$, and about the orientation of the sensor with respect to the rigid body, $\boldsymbol{q}^{BS}$.

4.4.2. Accelerometer Measurement

An accelerometer senses both gravitational accelerations, ${}^{S}\mathbf{g}^{S}$, and coordinate accelerations, ${}^{S}\mathbf{a}^{OS}$.[1] Figure 4.3 presents a useful mental experiment to find the expression of the accelerometer measurement.



(a) (b) (c) (d)

Figure 4.3: Depictions of an accelerometer in different situations. **(a)** No acceleration. **(b)** Being pushed. **(c)** In a gravitational field. **(d)** Free falling.

The measurement of an accelerometer can be expressed as

$$\mathbf{a}^{m} = {}^{S}\mathbf{a}^{OS} - {}^{S}\mathbf{g}^{S} + \mathbf{r}^{a} = \tag{4.27a}$$

$$= \mathbf{R}^{T}(\boldsymbol{q}^{BS})\,\mathbf{R}^{T}(\boldsymbol{q}^{OB})\left({}^{O}\mathbf{a}^{OS} - {}^{O}\mathbf{g}^{S}\right) + \mathbf{r}^{a} \quad , \tag{4.27b}$$

being $\mathbf{r}^{a}$ the measurement noise of the accelerometer. Gravitational acceleration, ${}^{O}\mathbf{g}^{S}$, is usually very similar to ${}^{O}\mathbf{g}^{B}$. On the other hand, coordinate acceleration, ${}^{O}\mathbf{a}^{OS}$, may differ from ${}^{O}\mathbf{a}^{OB}$ substantially due to rotation. Let us consider an accelerometer fixed to a rigid body $\mathcal{B}$ $\left(\frac{d\,{}^{B}\mathbf{x}^{BS}}{dt} = \mathbf{0}\right)$. Its position expressed in an inertial reference frame $\mathcal{O}$ is

$${}^{O}\mathbf{x}^{OS} = {}^{O}\mathbf{x}^{OB} + {}^{O}\mathbf{x}^{BS} = \tag{4.28a}$$

$$= {}^{O}\mathbf{x}^{OB} + \mathbf{R}(\boldsymbol{q}^{OB})\,{}^{B}\mathbf{x}^{BS} \quad . \tag{4.28b}$$

Using (4.9), we obtain the expression for the velocity of the sensor mounted in our rotating reference frame $\mathcal{B}$:

$$\frac{d\,{}^{O}\mathbf{x}^{OS}}{dt} = \frac{d\,{}^{O}\mathbf{x}^{OB}}{dt} + \mathbf{R}(\boldsymbol{q}^{OB})\Big[\underbrace{\frac{d\,{}^{B}\mathbf{x}^{BS}}{dt}}_{\mathbf{0}} + {}^{B}\boldsymbol{\omega}^{B} \times {}^{B}\mathbf{x}^{BS}\Big] \equiv \tag{4.29a}$$

$$\equiv {}^{O}\mathbf{v}^{OS} = {}^{O}\mathbf{v}^{OB} + \mathbf{R}(\boldsymbol{q}^{OB})\left[{}^{B}\boldsymbol{\omega}^{B} \times {}^{B}\mathbf{x}^{BS}\right] \quad . \tag{4.29b}$$

After applying (4.9) again, we obtain the coordinate acceleration:

$$\frac{d^{2}\,{}^{O}\mathbf{x}^{OS}}{dt^{2}} = \frac{d^{2}\,{}^{O}\mathbf{x}^{OB}}{dt^{2}} + \mathbf{R}(\boldsymbol{q}^{OB})\left(\frac{d\left[{}^{B}\boldsymbol{\omega}^{B} \times {}^{B}\mathbf{x}^{BS}\right]}{dt} + \right.$$

$$\left. + {}^{B}\boldsymbol{\omega}^{B} \times \left[{}^{B}\boldsymbol{\omega}^{B} \times {}^{B}\mathbf{x}^{BS}\right]\right) \equiv \tag{4.30a}$$

$$\equiv {}^{O}\mathbf{a}^{OS} = {}^{O}\mathbf{a}^{OB} + \mathbf{R}(\boldsymbol{q}^{OB})\left(\frac{d\,{}^{B}\boldsymbol{\omega}^{B}}{dt} \times {}^{B}\mathbf{x}^{BS} + \right.$$

$$\left. + {}^{B}\boldsymbol{\omega}^{B} \times \left[{}^{B}\boldsymbol{\omega}^{B} \times {}^{B}\mathbf{x}^{BS}\right]\right) \quad . \tag{4.30b}$$

Introducing (4.30) in (4.27) we obtain

$$\mathbf{a}^{m} = \mathbf{R}^{T}(\boldsymbol{q}^{BS})\left({}^{B}\mathbf{a}^{OB} - \mathbf{R}^{T}(\boldsymbol{q}^{OB})\,{}^{O}\mathbf{g}^{S} + \right.$$

$$\left. + \frac{d\,{}^{B}\boldsymbol{\omega}^{B}}{dt} \times {}^{B}\mathbf{x}^{BS} + {}^{B}\boldsymbol{\omega}^{B} \times \left[{}^{B}\boldsymbol{\omega}^{B} \times {}^{B}\mathbf{x}^{BS}\right]\right) + \mathbf{r}^{a} \quad . \tag{4.31}$$

[1] Actually, an accelerometer measures proper acceleration.

Finally, if we introduce (4.23) in (4.31) we can conclude that the accelerometer measures

$$\begin{aligned}\mathbf{a}^m \;=\; \mathbf{R}^T(\,\boldsymbol{q}^{BS}\,)\Big(\,{}^B\mathbf{a}^{OB} \;-\; \mathbf{R}^T(\,\boldsymbol{q}^{OB}\,)\;{}^O\mathbf{g}^S \;+ \\ +\; \mathbf{I}_B^{-1}\Big(\,{}^B\boldsymbol{\tau}^{BP} \;-\; {}^B\boldsymbol{\omega}^B \times \big(\,\mathbf{I}_B\,{}^B\boldsymbol{\omega}^B\,\big)\Big) \times {}^B\mathbf{x}^{BS} \;+ \\ +\; {}^B\boldsymbol{\omega}^B \times \big[\,{}^B\boldsymbol{\omega}^B \times {}^B\mathbf{x}^{BS}\,\big]\Big) \;+\; \mathbf{r}^a \quad . \end{aligned} \tag{4.32}$$

Notice how rich in information is the measurement of an accelerometer. It provides information about the orientation $\boldsymbol{q}^{BS}$ and position ${}^B\mathbf{x}^{BS}$ of the sensor $\mathcal{S}$, and about the orientation $\boldsymbol{q}^{OB}$, angular velocity ${}^B\boldsymbol{\omega}^B$, and moment of inertia $\mathbf{I}_B$ of the rigid body $\mathcal{B}$.

4.4.2.1. Accelerometer Measurement for a General Rigid Body

If we introduce (4.24) in (4.32) we obtain

$$\begin{aligned}\mathbf{a}^m \;=\; \mathbf{R}^T(\,\boldsymbol{q}^{BS}\,)\Big(\,{}^B\mathbf{a}^{OB} \;-\; \mathbf{R}^T(\,\boldsymbol{q}^{OB}\,)\;{}^O\mathbf{g}^S \;+ \\ +\; {}^B\widetilde{\boldsymbol{\tau}}^{BP} \times {}^B\mathbf{x}^{BS} \;+\; {}^B\boldsymbol{\omega}^B \times \big[\,{}^B\boldsymbol{\omega}^B \times {}^B\mathbf{x}^{BS}\,\big]\Big) \;+\; \mathbf{r}^a \quad . \end{aligned} \tag{4.33}$$

4.4.3. Magnetometer Measurement

A magnetometer measures the magnetic field at the particular location of the sensor, with respect to a reference frame $\mathcal{S}$ like the one depicted in Figure 4.1. Let us consider a magnetometer fixed to a rigid body represented by a reference frame $\mathcal{B}$. In such a case, the magnetic field expressed in the reference frame of the sensor ${}^S\mathbf{b}^S$ is related to the magnetic field expressed in the inertial reference frame $\mathcal{O}$ through

$${}^O\mathbf{b}^S \;=\; \mathbf{R}(\,\boldsymbol{q}^{OB}\,)\;\mathbf{R}(\,\boldsymbol{q}^{BS}\,)\;{}^S\mathbf{b}^S \quad . \tag{4.34}$$

If we consider that the magnetic field is uniform in space, then the measurement of a magnetometer can be expressed as

$$\mathbf{b}^m \;=\; \mathbf{R}^T(\,\boldsymbol{q}^{BS}\,)\;\mathbf{R}^T(\,\boldsymbol{q}^{OB}\,)\;\big(\,\mathbf{b}^d \;+\; {}^O\mathbf{b}^S\,\big) \;+\; \mathbf{r}^m \quad , \tag{4.35}$$

where $\mathbf{r}^m$ is the measurement noise of the magnetometer, and $\mathbf{b}^d$ would describe magnetic disturbances, which could be produced, among other causes, by moving irons.

Chapter 5

Manifold Kalman Filters

5.1. Introduction

In previous chapters we introduced the concepts with which we will design Kalman filters to estimate the orientation of a rigid body. In particular, we presented the basics of Kalman filtering in Chapter 2, we introduced quaternions in Chapter 3, we defined the mean and the covariance matrix of a distribution of unit quaternions in Section 3.4, and we derived the motion equations for a rigid body and the measurement equations for IMUs in Chapter 4.

The purpose of this chapter is multiple. First, we intend to design, in a simple context (avoiding complicated models that divert attention), a Kalman filter that includes unit quaternions (those that describe orientation) as part of the state. In this way, extrapolation to more complicated models should be simpler. Secondly, we aim for this simple design to serve as a basis for other Kalman filters whose state is defined in a manifold; other Manifold Kalman filters. Finally, we want to compare the performance of the EKF and the UKF in this particular problem. We also want to compare if there is a chart (see Section 3.4) that offers advantages over the others.

In this chapter, we start by formulating the model in Section 5.2. This model will be a simplification of the models presented in Chapter 4. We also present in this section the description of the state, which contains quaternions. Then, Sections 5.3 and 5.4 present the Manifold Kalman filter designs based on the EKF and the UKF respectively. Finally, in Section 5.5 we describe the experiments used to test the estimation algorithms, and show the results that are extracted from them.

5.2. Formulation of the Model

The angular state of a rigid body $\mathcal{B}$ at a time t is defined by an orientation, encoded with a unit quaternion $\boldsymbol{q}_t^{OB}$, and by an angular velocity ${}^B\boldsymbol{\omega}_t^B$. We will consider them to be random variables, and we will try to estimate their value using a Kalman filter.

Our unit quaternions $\{\, \boldsymbol{q}_t^{OB} \in \mathbb{H} : \|\boldsymbol{q}_t^{OB}\| = 1 \,\}$ will define the rotation transformation that relates a vector ${}^B\mathbf{v}$ expressed in a reference frame $\mathcal{B}$, attached to the rigid body whose state we want to describe, with the same vector ${}^O\mathbf{v}$ expressed in an external reference frame $\mathcal{O}$, as described in Chapter 4:

$$ {}^O\mathbf{v} = \mathbf{R}(\,\boldsymbol{q}_t^{OB}\,)\ {}^B\mathbf{v} \qquad \equiv \tag{5.1a} $$

$$ \equiv \qquad {}^O\boldsymbol{v} = \boldsymbol{q}_t^{OB} * {}^B\boldsymbol{v} * (\boldsymbol{q}_t^{OB})^* \quad . \tag{5.1b} $$

The vector ${}^B\boldsymbol{\omega}_t^B$ will define the angular velocity of the rigid body measured in $\mathcal{B}$. Note that we do not include the bias of the sensors in the state of our system. We will assume that our sensors are calibrated, so the biases are zero.

5.2.1. Prediction Equations

We will assume that the evolution of our random variables $\boldsymbol{q}_t^{OB}$ and ${}^B\boldsymbol{\omega}_t^B$ can be described by equations (4.7) and (4.24). Then, our prediction model is given by the following differential

equations:

$$\frac{d\ {}^{B}\boldsymbol{\omega}^{B}(t)}{d\,t} = \mathbf{w}^{\boldsymbol{\omega}}(t) \quad , \tag{5.2}$$

$$\frac{d\ \boldsymbol{q}^{OB}(t)}{d\,t} = \frac{1}{2}\ \boldsymbol{q}^{OB}(t)\ *\ \begin{pmatrix} 0 \\ {}^{B}\boldsymbol{\omega}^{B}(t) \end{pmatrix} \quad , \tag{5.3}$$

where $\mathbf{w}^{\boldsymbol{\omega}}(t)$ is a random variable that represents the process noise, and is associated with the torque acting on the system, and with its inertia tensor.

5.2.2. Measurement Equations

We will consider that there are two types of sensor attached to our rigid body. The first one gives measurements of a vector $\mathbf{v}_t^m$ whose value ${}^{O}\mathbf{v}_t$, expressed in the external reference frame $\mathcal{O}$, is known. Examples of such sensors would be an accelerometer measuring the gravity vector near the Earth surface (${}^{O}\mathbf{v}_t := -\mathbf{g}$), or a magnetometer measuring the Earth magnetic field (${}^{O}\mathbf{v}_t := \mathbf{B}$). The second sensor gives measurements of angular velocity $\boldsymbol{\omega}_t^m$. An example of such a sensor would be a gyroscope. We will assume that these measurements are related to the variables that describe the state of the system through the following measurement model:

$$\mathbf{v}_t^m = \mathbf{R}^T(\,\boldsymbol{q}_t^{OB}\,)\ \left(\,\mathbf{q}_t^{\mathbf{v}}\ +\ {}^{O}\mathbf{v}_t\,\right)\ +\ \mathbf{r}_t^{\mathbf{v}} \quad , \tag{5.4}$$

$$\boldsymbol{\omega}_t^m = {}^{B}\boldsymbol{\omega}_t^B\ +\ \mathbf{r}_t^{\boldsymbol{\omega}} \quad , \tag{5.5}$$

where $\mathbf{r}_t^{\mathbf{v}}$ and $\mathbf{r}_t^{\boldsymbol{\omega}}$ are random variables with zero mean and covariance matrices $\mathbf{R}_t^{\mathbf{v}}$ and $\mathbf{R}_t^{\boldsymbol{\omega}}$ respectively that represent the measurement noises, and $\mathbf{q}_t^{\mathbf{v}}$ is another random variable with mean $\overline{\mathbf{q}}_t^{\mathbf{v}}$ and covariance matrix $\mathbf{Q}_t^{\mathbf{v}}$ representing external disturbances in the measurement of the vector ${}^{O}\mathbf{v}_t$. For example, it could represent accelerations others than gravity for an accelerometer, or magnetic disturbances produced by moving irons for a magnetometer. If we compare equation (5.4) with equations (4.33) and (4.35) we see that taking our measurement model we are assuming that our sensors give measurements expressed in $\mathcal{B}$, so $\boldsymbol{q}^{BS} = \mathbf{1}$. The same happens if we compare equation (5.5) with (4.26). In addition, for the particular case of an accelerometer, the sensor would be mounted in the center of mass of the rigid body, so that ${}^{B}\mathbf{x}^{BS} = \mathbf{0}$.

5.2.3. Description of the State Distribution

We will assume that the measurements arrive at discrete times $\{\,t_n\,\}_n$. The format $x_{t|t_n}$ will be used to denote the estimation of a random variable x at a time t, having included measurements up to a time t_n with $t > t_n$. For times in which measurements are taken, we will write $x_{t_m|t_n}$ as $x_{m|n}$ for the sake of simplicity. Then, our knowledge about the state at a time t, having included measurements up to a time t_n with $t \geq t_n$, is described by a distribution encoded in the collection of mathematical objects $\left(\,\varphi\,,\,\overline{\boldsymbol{p}}\,,\,\overline{\mathbf{x}}_{t|n}^{\overline{\boldsymbol{p}}}\,,\,\mathbf{P}_{t|n}^{\overline{\boldsymbol{p}}}\,\right)$ as described in Section 3.4. For the present case, $\overline{\mathbf{x}}_{t|n}^{\overline{\boldsymbol{p}}} = \left(\,\overline{\mathbf{e}}_{t|n}^{\overline{\boldsymbol{p}}}\,,\,{}^{B}\overline{\boldsymbol{\omega}}_{t|n}^{B}\,\right)^T$ is the expected value of the distribution, and $\mathbf{P}_{t|n}^{\overline{\boldsymbol{p}}}$ is its 6×6 covariance matrix, both expressing the quaternion distribution in the $\overline{\boldsymbol{p}}$-centered chart. Preferably, $\overline{\boldsymbol{p}}$ will be a unit quaternion central in the distribution, so that the mapping of points from the $\overline{\boldsymbol{p}}$-centered chart to the manifold causes minimal deformation in such distribution. The unit quaternion $\overline{\boldsymbol{q}}_{t|n}^{OB} = \varphi_{\overline{\boldsymbol{p}}}^{-1}\left(\,\overline{\mathbf{e}}_{t|n}^{\overline{\boldsymbol{p}}}\,\right)$ will be our best estimation of the real quaternion $\boldsymbol{q}_t^{OB}$ that defines the orientation of the system with respect to the external reference frame $\mathcal{O}$ at time t.

The following two sections present two different Kalman filters that result from the presented model: a version based on the EKF and another version based on the UKF.

5.3. Manifold Extended Kalman Filter

At time t_n we receive a measurement

$$\mathbf{z}_n = \begin{pmatrix} \mathbf{v}_n^m \\ \boldsymbol{\omega}_n^m \end{pmatrix} \quad . \tag{5.6}$$

Our knowledge about the orientation at a previous time $t_{n\text{-}1}$ is described by a distribution expressed in the $\overline{\boldsymbol{q}}^{OB}_{n\text{-}1|n\text{-}1}$-centered chart. We assume that this distribution has mean

$$\overline{\mathbf{x}}^{\overline{\boldsymbol{q}}^{OB}_{n\text{-}1|n\text{-}1}}_{n\text{-}1|n\text{-}1} \quad = \quad \begin{pmatrix} \overline{\mathbf{e}}^{\overline{\boldsymbol{q}}^{OB}_{n\text{-}1|n\text{-}1}}_{n\text{-}1|n\text{-}1} = \mathbf{0} \\ {}^{B}\overline{\boldsymbol{\omega}}^{B}_{n\text{-}1|n\text{-}1} \end{pmatrix} \quad , \tag{5.7}$$

and covariance matrix $\mathbf{P}^{\overline{\boldsymbol{q}}^{OB}_{n\text{-}1|n\text{-}1}}_{n\text{-}1|n\text{-}1}$. This is, we have an initial four

$$\left(\varphi \ , \ \overline{\boldsymbol{q}}^{OB}_{n\text{-}1|n\text{-}1} \ , \ {}^{B}\overline{\boldsymbol{\omega}}^{B}_{n\text{-}1|n\text{-}1} \ , \ \mathbf{P}^{\overline{\boldsymbol{q}}^{OB}_{n\text{-}1|n\text{-}1}}_{n\text{-}1|n\text{-}1} \right) \quad . \tag{5.8}$$

5.3.1. Prediction of the State

5.3.1.1. Mean of the Prediction

By equation (2.65), in our case we have the following equations that are actually approximate:

$$\frac{d\ {}^{B}\overline{\boldsymbol{\omega}}^{B}(t)}{d\,t} \quad = \quad \overline{\mathbf{w}}^{\boldsymbol{\omega}}(t) \quad , \tag{5.9}$$

$$\frac{d\ \overline{\boldsymbol{q}}^{OB}(t)}{d\,t} \quad = \quad \frac{1}{2}\ \overline{\boldsymbol{q}}^{OB}(t)\ * \begin{pmatrix} 0 \\ {}^{B}\overline{\boldsymbol{\omega}}^{B}(t) \end{pmatrix} \quad . \tag{5.10}$$

If we consider $\overline{\mathbf{w}}^{\boldsymbol{\omega}}$ to be constant in the integration interval, then

$${}^{B}\overline{\boldsymbol{\omega}}^{B}(t) \quad = \quad {}^{B}\overline{\boldsymbol{\omega}}^{B}(0) \ + \ \overline{\mathbf{w}}^{\boldsymbol{\omega}}\ t \quad . \tag{5.11}$$

On the other hand, if ${}^{B}\overline{\boldsymbol{\omega}}^{B}$ were constant in the integration interval ($\overline{\mathbf{w}}^{\boldsymbol{\omega}} = \mathbf{0}$), equation (5.10) could be rewritten in matrix form as

$$\frac{d\ \overline{\boldsymbol{q}}^{OB}(t)}{d\,t} \quad = \quad \frac{1}{2}\ {}_{*}[\ {}^{B}\overline{\boldsymbol{\omega}}^{B}\ ]\ \overline{\boldsymbol{q}}^{OB}(t) \quad , \tag{5.12}$$

where we have expressed the quaternion product in matrix form as in (3.8). The solution to differential equation (5.12) is

$$\overline{\boldsymbol{q}}^{OB}(t) \quad = \quad e^{\frac{1}{2}\ {}_{*}[\ {}^{B}\overline{\boldsymbol{\omega}}^{B}\ ]\ t}\ \overline{\boldsymbol{q}}^{OB}(0) \quad . \tag{5.13}$$

Computing the matrix exponential of $\frac{1}{2}\ {}_{*}[\ {}^{B}\overline{\boldsymbol{\omega}}^{B}\ ]\,t$ (see Appendix A.7) we find out that equation (5.13) can be rewritten using the quaternion product as

$$\overline{\boldsymbol{q}}^{OB}(t) \quad = \quad \overline{\boldsymbol{q}}^{OB}(0)\ *\ \boldsymbol{\delta}^{\boldsymbol{\omega}}(t) \quad = \tag{5.14a}$$

$$= \quad \overline{\boldsymbol{q}}^{OB}(0)\ * \begin{pmatrix} \cos\left(\frac{\| {}^{B}\overline{\boldsymbol{\omega}}^{B} \|\ t}{2} \right) \\ \frac{{}^{B}\overline{\boldsymbol{\omega}}^{B}}{\| {}^{B}\overline{\boldsymbol{\omega}}^{B} \|}\ \sin\left(\frac{\| {}^{B}\overline{\boldsymbol{\omega}}^{B} \|\ t}{2} \right) \end{pmatrix} \quad . \tag{5.14b}$$

In our particular case, ${}^{B}\overline{\boldsymbol{\omega}}^{B} = {}^{B}\overline{\boldsymbol{\omega}}^{B}_{n\text{-}1|n\text{-}1}$, $\overline{\boldsymbol{q}}^{OB}(0) = \overline{\boldsymbol{q}}^{OB}_{n\text{-}1|n\text{-}1}$, $\overline{\boldsymbol{q}}^{OB}(t) = \overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}$, and $t = t_n - t_{n\text{-}1}$. We could have introduced expression (5.11) into (5.10), and find the solution to be

$$\overline{\boldsymbol{q}}^{OB}(t) \quad = \quad e^{\frac{1}{2}\left({}_{*}[\ {}^{B}\overline{\boldsymbol{\omega}}^{B}\ ]\ t\ +\ {}_{*}[\ \overline{\mathbf{w}}\ ]\ \frac{t^2}{2} \right)}\ \overline{\boldsymbol{q}}^{OB}(0) \quad , \tag{5.15}$$

but then we could not find a closed expression for the matrix exponential, unless ${}_{*}[\ {}^{B}\overline{\boldsymbol{\omega}}^{B}\ ]$ and ${}_{*}[\ \overline{\mathbf{w}}\ ]$ commute.

5.3.1.2. Covariance Matrix of the Prediction

We will use (2.71) to obtain the prediction of the covariance matrix. However, given our definition for the covariance matrix of a distribution of unit quaternions (3.41) the matrix $\mathbf{F}(t)$ will take another form than the one adopted in Section 2.4.1. In particular, we need to find the evolution equation for $\mathbf{e}^{\overline{\boldsymbol{q}}}$.

Knowing that any quaternion in the unit sphere can be expressed as a deviation from a central quaternion $\overline{\boldsymbol{q}}$ as $\boldsymbol{q} = \overline{\boldsymbol{q}} * \boldsymbol{\delta}^{\overline{\boldsymbol{q}}}$, and using differential equations (5.3) and (5.10), we can find a differential equation for the quaternion $\boldsymbol{\delta}^{\overline{\boldsymbol{q}}}$:

$$\boldsymbol{q} = \overline{\boldsymbol{q}} * \boldsymbol{\delta}^{\overline{\boldsymbol{q}}} \qquad \Longrightarrow \tag{5.16a}$$

$$\Longrightarrow \qquad \dot{\boldsymbol{q}} = \dot{\overline{\boldsymbol{q}}} * \boldsymbol{\delta}^{\overline{\boldsymbol{q}}} + \overline{\boldsymbol{q}} * \dot{\boldsymbol{\delta}}^{\overline{\boldsymbol{q}}} \qquad \Longrightarrow \tag{5.16b}$$

$$\Longrightarrow \qquad \frac{1}{2}\, \boldsymbol{q} * {}^{B}\boldsymbol{\omega}^{B} \approx \frac{1}{2}\, \overline{\boldsymbol{q}} * {}^{B}\overline{\boldsymbol{\omega}}^{B} * \boldsymbol{\delta}^{\overline{\boldsymbol{q}}} + \overline{\boldsymbol{q}} * \dot{\boldsymbol{\delta}}^{\overline{\boldsymbol{q}}} \quad , \tag{5.16c}$$

where a dot over a symbol represents time derivative, we have denoted $\boldsymbol{q} := \boldsymbol{q}^{OB}$ (this nomenclature will be maintained throughout this particular section), and we have obviated the time dependence. Isolating the time derivative $\dot{\boldsymbol{\delta}}^{\overline{\boldsymbol{q}}}$,

$$\dot{\boldsymbol{\delta}}^{\overline{\boldsymbol{q}}} \approx \frac{1}{2}\, \overbrace{\overline{\boldsymbol{q}}^{*} * \boldsymbol{q}}^{\boldsymbol{\delta}^{\overline{\boldsymbol{q}}}} * {}^{B}\boldsymbol{\omega}^{B} - \frac{1}{2}\, \overbrace{\overline{\boldsymbol{q}}^{*} * \overline{\boldsymbol{q}}}^{\mathbf{1}} * {}^{B}\overline{\boldsymbol{\omega}}^{B} * \boldsymbol{\delta}^{\overline{\boldsymbol{q}}} = \tag{5.17a}$$

$$= \frac{1}{2}\, \left[\, \boldsymbol{\delta}^{\overline{\boldsymbol{q}}} * {}^{B}\boldsymbol{\omega}^{B} - {}^{B}\overline{\boldsymbol{\omega}}^{B} * \boldsymbol{\delta}^{\overline{\boldsymbol{q}}} \,\right] = \tag{5.17b}$$

$$= \frac{1}{2}\, \left[\, \begin{pmatrix} \delta_0^{\overline{\boldsymbol{q}}} \\ \boldsymbol{\delta}^{\overline{\boldsymbol{q}}} \end{pmatrix} * \begin{pmatrix} 0 \\ {}^{B}\boldsymbol{\omega}^{B} \end{pmatrix} - \begin{pmatrix} 0 \\ {}^{B}\overline{\boldsymbol{\omega}}^{B} \end{pmatrix} * \begin{pmatrix} \delta_0^{\overline{\boldsymbol{q}}} \\ \boldsymbol{\delta}^{\overline{\boldsymbol{q}}} \end{pmatrix} \,\right] = \tag{5.17c}$$

$$= \frac{1}{2}\, \begin{pmatrix} -\left({}^{B}\boldsymbol{\omega}^{B} - {}^{B}\overline{\boldsymbol{\omega}}^{B} \right) \cdot \boldsymbol{\delta}^{\overline{\boldsymbol{q}}} \\ \delta_0^{\overline{\boldsymbol{q}}} \left({}^{B}\boldsymbol{\omega}^{B} - {}^{B}\overline{\boldsymbol{\omega}}^{B} \right) - \left({}^{B}\boldsymbol{\omega}^{B} + {}^{B}\overline{\boldsymbol{\omega}}^{B} \right) \times \boldsymbol{\delta}^{\overline{\boldsymbol{q}}} \end{pmatrix} = \tag{5.17d}$$

$$= \frac{1}{2}\, \begin{pmatrix} -\,\Delta^{B}\boldsymbol{\omega}^{B} \cdot \boldsymbol{\delta}^{\overline{\boldsymbol{q}}} \\ \delta_0^{\overline{\boldsymbol{q}}}\; \Delta^{B}\boldsymbol{\omega}^{B} - \left(2\, {}^{B}\overline{\boldsymbol{\omega}}^{B} + \Delta^{B}\boldsymbol{\omega}^{B} \right) \times \boldsymbol{\delta}^{\overline{\boldsymbol{q}}} \end{pmatrix} \quad . \tag{5.17e}$$

Knowing that, for each of our charts, the quaternion $\boldsymbol{\delta}^{\overline{\boldsymbol{q}}}$ can be approximated by (3.36) as $\mathbf{e} \to \mathbf{0}$, then we can obtain an approximate differential equation for a point $\mathbf{e}^{\overline{\boldsymbol{q}}}$ expressed in the $\overline{\boldsymbol{q}}(t)$-centered chart. Using the chain rule for a time derivative and expression (5.17e),

$$\frac{d\, e_i^{\overline{\boldsymbol{q}}}}{d\, t} = \sum_j \overbrace{\frac{\partial\, e_i^{\overline{\boldsymbol{q}}}}{\partial\, \delta_j^{\overline{\boldsymbol{q}}}}}^{\approx 2\,\delta_{ij}} \overbrace{\frac{\partial\, \delta_j^{\overline{\boldsymbol{q}}}}{\partial\, t}}^{\dot{\delta}_j^{\overline{\boldsymbol{q}}}} \equiv \tag{5.18a}$$

$$\equiv \quad \dot{\mathbf{e}}^{\overline{\boldsymbol{q}}} \approx \delta_0^{\overline{\boldsymbol{q}}}\; \Delta^{B}\boldsymbol{\omega}^{B} - \left(2\, {}^{B}\overline{\boldsymbol{\omega}}^{B} + \Delta^{B}\boldsymbol{\omega}^{B} \right) \times \boldsymbol{\delta}^{\overline{\boldsymbol{q}}} \approx \tag{5.18b}$$

$$\approx \left(1 - \frac{\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|^2}{8} \right) \Delta^{B}\boldsymbol{\omega}^{B} - \left(2\, {}^{B}\overline{\boldsymbol{\omega}}^{B} + \Delta^{B}\boldsymbol{\omega}^{B} \right) \times \frac{\mathbf{e}^{\overline{\boldsymbol{q}}}}{2} \quad . \tag{5.18c}$$

Neglecting the contribution of moments higher than the second one, we obtain

$$\frac{d}{d\, t} \begin{pmatrix} \mathbf{e}^{\overline{\boldsymbol{q}}} - \overline{\mathbf{e}}^{\overline{\boldsymbol{q}}} \\ \Delta^{B}\boldsymbol{\omega}^{B} \end{pmatrix} \approx \begin{pmatrix} -\left[{}^{B}\overline{\boldsymbol{\omega}}^{B} \right]_{\times} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{e}^{\overline{\boldsymbol{q}}} - \overline{\mathbf{e}}^{\overline{\boldsymbol{q}}} \\ \Delta^{B}\boldsymbol{\omega}^{B} \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \Delta \mathbf{w}^{\omega} \end{pmatrix} \quad , \tag{5.19}$$

which allows us to identify the matrices that appear in (2.67) as

$$\mathbf{F} := \begin{pmatrix} -\,\boldsymbol{\Omega} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \quad , \tag{5.20}$$

$$\mathbf{L} := \mathbf{I} \quad , \tag{5.21}$$

with $\boldsymbol{\Omega} = \left[{}^{B}\overline{\boldsymbol{\omega}}^{B} \right]_{\times}$. In addition, recalling (2.63) we can infer the form of the covariance matrix of the process noise:

$$\mathbf{Q}(t) = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}^{\omega}(t) \end{pmatrix} \quad . \tag{5.22}$$

We are now in a position to solve differential equation (2.71). Let us consider its homogeneous version first:

$$\frac{d\,\mathbf{P}_H}{d\,t} \;=\; \mathbf{F}\,\mathbf{P}_H \;+\; \mathbf{P}_H\,\mathbf{F}^T \quad . \tag{5.23}$$

The solution to 5.23 is

$$\mathbf{P}_H \;=\; e^{\mathbf{F}\,t}\;\mathbf{C}_0\;e^{\mathbf{F}^T\,t} \quad . \tag{5.24}$$

Recalling the definition of matrix exponential, and observing that

$$\mathbf{F}^n \;=\; \begin{pmatrix} (-\boldsymbol{\Omega})^n & (-\boldsymbol{\Omega})^{n-1} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \quad , \tag{5.25}$$

for $n \geq 1$, we arrive at

$$e^{\mathbf{F}\,t} \;=\; \begin{pmatrix} \sum\limits_{n=0}^{\infty} \frac{(-\boldsymbol{\Omega})^n\,t^n}{n!} & \sum\limits_{n=1}^{\infty} \frac{(-\boldsymbol{\Omega})^{n-1}\,t^n}{n!} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \;\approx \tag{5.26}$$

$$\approx \begin{pmatrix} \mathbf{R}^T\big(\boldsymbol{\delta}^{\boldsymbol{\omega}}(t)\big) & \mathbf{I}\,t \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \quad , \tag{5.27}$$

being $\boldsymbol{\delta}^{\boldsymbol{\omega}}(t)$ the same quaternion that appears in (5.14). We have also assumed that t takes small values so that we can approximate the infinite sum on the right truncating in the first term.

To find the solution of the non-homogeneous differential equation we use the *variation of constants* method:

$$\mathbf{P} \;=\; e^{\mathbf{F}\,t}\;\mathbf{C}(t)\;e^{\mathbf{F}^T\,t} \qquad \Longrightarrow \tag{5.28a}$$

$$\Longrightarrow \qquad \frac{d\,\mathbf{P}}{d\,t} \;=\; \mathbf{F}\;e^{\mathbf{F}\,t}\;\mathbf{C}(t)\;e^{\mathbf{F}^T\,t} \;+\; e^{\mathbf{F}\,t}\;\mathbf{C}(t)\;e^{\mathbf{F}^T\,t}\;\mathbf{F}^T \;+$$

$$+\; e^{\mathbf{F}\,t}\;\frac{d\,\mathbf{C}(t)}{d\,t}\;e^{\mathbf{F}^T\,t} \;= \tag{5.28b}$$

$$= \mathbf{F}\,\mathbf{P} \;+\; \mathbf{P}\,\mathbf{F}^T \;+\; e^{\mathbf{F}\,t}\;\frac{d\,\mathbf{C}(t)}{d\,t}\;e^{\mathbf{F}^T\,t} \quad . \tag{5.28c}$$

Identifying terms with (2.71) we obtain that

$$e^{\mathbf{F}\,t}\;\frac{d\,\mathbf{C}(t)}{d\,t}\;e^{\mathbf{F}^T\,t} \;=\; \mathbf{L}\,\mathbf{Q}\,\mathbf{L}^T \qquad \Longrightarrow \tag{5.29a}$$

$$\Longrightarrow \qquad \frac{d\,\mathbf{C}(t)}{d\,t} \;=\; e^{-\mathbf{F}\,t}\;\mathbf{Q}\;e^{-\mathbf{F}^T\,t} \;= \tag{5.29b}$$

$$= \sum_{n=0}^{\infty}\sum_{m=0}^{\infty} \frac{(-\mathbf{F})^n\,t^n}{n!}\;\mathbf{Q}\;\frac{(-\mathbf{F}^T)^m\,t^m}{m!} \qquad \Longrightarrow \tag{5.29c}$$

$$\Longrightarrow \qquad \mathbf{C}(t) \;=\; \mathbf{C}_0 \;+\; \sum_{n=0}^{\infty}\sum_{m=0}^{\infty} \frac{(-\mathbf{F})^n}{n!}\;\mathbf{Q}\;\frac{(-\mathbf{F}^T)^m}{m!}\;\frac{t^{n+m+1}}{n+m+1} \quad . \tag{5.29d}$$

Note that due to the initial conditions $\mathbf{C}_0 = \mathbf{P}(0)$. Finally, checking that

$$(-\mathbf{F})^0\;\mathbf{Q}\;(-\mathbf{F}^T)^0 \;=\; \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}^{\boldsymbol{\omega}} \end{pmatrix} \quad , \tag{5.30a}$$

$$(-\mathbf{F})^n\;\mathbf{Q}\;(-\mathbf{F}^T)^0 \;=\; (-1)^n \begin{pmatrix} \mathbf{0} & (-\boldsymbol{\Omega})^{n-1}\,\mathbf{Q}^{\boldsymbol{\omega}} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \quad , \tag{5.30b}$$

$$(-\mathbf{F})^0\;\mathbf{Q}\;(-\mathbf{F}^T)^m \;=\; (-1)^m \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{Q}^{\boldsymbol{\omega}}\,(-\boldsymbol{\Omega}^T)^{m-1} & \mathbf{0} \end{pmatrix} \quad , \tag{5.30c}$$

$$(-\mathbf{F})^n\;\mathbf{Q}\;(-\mathbf{F}^T)^m \;=\; (-1)^{n+m} \begin{pmatrix} (-\boldsymbol{\Omega})^{n-1}\,\mathbf{Q}^{\boldsymbol{\omega}}\,(-\boldsymbol{\Omega}^T)^{m-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \quad , \tag{5.30d}$$

we obtain

$$\mathbf{C}(t) = \mathbf{P}(0) + \left(\begin{array}{cc} \sum_{n=1}^{\infty}\sum_{m=1}^{\infty}(-1)^{n+m}\,\frac{(-\boldsymbol{\Omega})^{n-1}}{n!}\,\mathbf{Q}^{\boldsymbol{\omega}}\,\frac{(-\boldsymbol{\Omega}^T)^{m-1}}{m!}\,\frac{t^{n+m+1}}{n+m+1} & \sum_{n=1}^{\infty}(-1)^{n}\,\frac{(-\boldsymbol{\Omega})^{n-1}}{n!}\,\mathbf{Q}^{\boldsymbol{\omega}}\,\frac{t^{n+1}}{n+1} \\ \sum_{m=1}^{\infty}(-1)^{m}\,\mathbf{Q}^{\boldsymbol{\omega}}\,\frac{(-\boldsymbol{\Omega}^T)^{m-1}}{m!}\,\frac{t^{m+1}}{m+1} & \mathbf{Q}^{\boldsymbol{\omega}}\,t \end{array} \right) \approx \tag{5.31a}$$

$$\approx \mathbf{P}(0) + \left(\begin{array}{cc} \mathbf{Q}^{\boldsymbol{\omega}}\,\frac{t^3}{3} & -\mathbf{Q}^{\boldsymbol{\omega}}\,\frac{t^2}{2} \\ -\,\mathbf{Q}^{\boldsymbol{\omega}}\,\frac{t^2}{2} & \mathbf{Q}^{\boldsymbol{\omega}}\,t \end{array} \right) \quad . \tag{5.31b}$$

In our particular case, $\mathbf{P}(0) = \mathbf{P}^{\overline{\boldsymbol{q}}^{OB}_{n-1|n-1}}_{n-1|n-1}$, $\mathbf{P}(t) = \mathbf{P}^{\overline{\boldsymbol{q}}^{OB}_{n-1|n-1}}_{n|n-1}$, and $t = t_n - t_{n-1}$.

5.3.2. Prediction of the Measurement

5.3.2.1. Mean of the Predicted Measurement

Applying (2.57) to (5.4) and (5.5) results in

$$\overline{\mathbf{v}}^m_t = \mathbf{R}^T(\,\overline{\boldsymbol{q}}^{OB}_t\,)\,\left(\,\overline{\mathbf{q}}^{\mathbf{v}}_t + {}^{O}\mathbf{v}_t\,\right) \quad , \tag{5.32}$$

$$\overline{\boldsymbol{\omega}}^m_t = {}^{B}\overline{\boldsymbol{\omega}}^{B}_t \quad . \tag{5.33}$$

5.3.2.2. Covariance Matrix of the Predicted Measurement

Recalling (2.58) and (2.59), and using the result obtained in Appendix A.8.4 we get

$$\mathbf{H}_t = \left(\begin{array}{cc} [\,\overline{\mathbf{v}}^m_t\,]_\times & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{array} \right) \quad , \tag{5.34}$$

$$\mathbf{M}_t = \left(\begin{array}{ccc} \mathbf{R}^T(\,\overline{\boldsymbol{q}}^{OB}_t\,) & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{array} \right) \quad , \tag{5.35}$$

for our particular state $\mathbf{x}_t = (\,\boldsymbol{q}_t\,,\,\boldsymbol{\omega}_t\,)^T$, measurement $\mathbf{z}_t = (\,\mathbf{v}^m_t\,,\,\boldsymbol{\omega}^m_t\,)^T$, and measurement noise $\mathbf{r}_t = (\,\mathbf{q}^{\mathbf{v}}_t\,,\,\mathbf{r}^{\mathbf{v}}_t\,,\,\mathbf{r}^{\boldsymbol{\omega}}_t\,)^T$. As a result, the covariance matrix of the predicted measurement can be computed through

$$\mathbf{S}_t = \mathbf{H}_t\,\mathbf{P}_t\,\mathbf{H}_t^T + \left(\begin{array}{cc} \mathbf{R}^T(\,\overline{\boldsymbol{q}}^{OB}_t\,)\,\mathbf{Q}^{\mathbf{v}}_t\,\mathbf{R}(\,\overline{\boldsymbol{q}}^{OB}_t\,) + \mathbf{R}^{\mathbf{v}}_t & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^{\boldsymbol{\omega}} \end{array} \right) \quad , \tag{5.36}$$

where we have assumed the independence of $\mathbf{q}^{\mathbf{v}}_t$, $\mathbf{r}^{\mathbf{v}}_t$, and $\mathbf{r}^{\boldsymbol{\omega}}_t$.

5.3.3. Kalman Update

At this point, we can compute the Kalman update in the usual way (equations (2.48) to (2.50)). However, we should notice that the orientation is updated in the $\overline{\boldsymbol{q}}^{OB}_{n|n-1}$-centered chart. We still need to compute the mean and the covariance matrix in the $\overline{\boldsymbol{q}}^{OB}_{n|n}$-centered chart, so that the distribution is expressed in the same conditions as at the beginning of the iteration (5.8).

5.3.4. Chart Update

Through (2.49) we obtain the mean $\overline{\mathbf{x}}^{\overline{\boldsymbol{q}}^{OB}_{n|n-1}}_{n|n} = \left(\,\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}^{OB}_{n|n-1}}_{n|n}\,,\,{}^{B}\overline{\boldsymbol{\omega}}^{B}_{n|n}\,\right)^T$. The point $\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}^{OB}_{n|n-1}}_{n|n}$ that is defined in the $\overline{\boldsymbol{q}}^{OB}_{n|n-1}$-centered chart, correspond to a unit quaternion in the manifold. This is the updated unit quaternion $\overline{\boldsymbol{q}}^{OB}_{n|n}$ which we are looking for:

$$\overline{\boldsymbol{q}}^{OB}_{n|n} = \varphi^{-1}_{\overline{\boldsymbol{q}}^{OB}_{n|n-1}}\left(\,\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}^{OB}_{n|n-1}}_{n|n}\,\right) = \tag{5.37a}$$

$$= \overline{\boldsymbol{q}}^{OB}_{n|n-1} * \varphi^{-1}\left(\,\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}^{OB}_{n|n-1}}_{n|n}\,\right) = \tag{5.37b}$$

$$= \overline{\boldsymbol{q}}^{OB}_{n|n-1} * \overline{\boldsymbol{\delta}}_n \quad . \tag{5.37c}$$

Knowing that the Kalman update (2.49) could produce any point in the $\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}$-centered chart we will need to "saturate" to the closest point contained in the image of each chart. The point $\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}}_{n|n}$ in the $\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}$-centered chart is the origin in the $\overline{\boldsymbol{q}}^{OB}_{n|n}$-centered chart. Then, the expected value of the state in this new chart will be given by

$$\overline{\mathbf{x}}^{\overline{\boldsymbol{q}}^{OB}_{n|n}}_{n|n} \quad = \quad \left(\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}^{OB}_{n|n}}_{n|n} = \mathbf{0} \ , \ {}^{B}\overline{\boldsymbol{\omega}}^{B}_{n|n} \right)^{T} \quad , \tag{5.38}$$

as at the beginning of the iteration.

To update the covariance matrix we need to consider its definition (3.41). We want to compute $\mathbf{P}^{\overline{\boldsymbol{q}}^{OB}_{n|n}}$ having $\mathbf{P}^{\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}}$ and knowing the relation $\mathbf{e}^{\overline{\boldsymbol{p}}}\big(\mathbf{e}^{\overline{\boldsymbol{q}}}\big)$ provided by the transition maps in Table 3.3. Continuing with the EKF philosophy, the update for the covariance matrix will be found by linearizing $\mathbf{e}^{\overline{\boldsymbol{p}}}\big(\mathbf{e}^{\overline{\boldsymbol{q}}}\big)$ around the point where the majority of information is comprised (in our case, the point $\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}} = \overline{\mathbf{e}}^{\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}}_{n|n}$):

$$e^{\overline{\boldsymbol{p}}}_{i}\big(\mathbf{e}^{\overline{\boldsymbol{q}}}\big) \quad \approx \quad e^{\overline{\boldsymbol{p}}}_{i}\big(\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\big) \ + \ \sum_{j} \left. \frac{\partial\, e^{\overline{\boldsymbol{p}}}_{i}\big(\mathbf{e}^{\overline{\boldsymbol{q}}}\big)}{\partial\, e^{\overline{\boldsymbol{q}}}_{j}} \right|_{\mathbf{e}^{\overline{\boldsymbol{q}}} = \overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}} \big(e^{\overline{\boldsymbol{q}}}_{j} - \overline{e}^{\overline{\boldsymbol{q}}}_{j} \big) \quad , \tag{5.39}$$

In particular, if we define the matrix

$$(\mathbf{T})_{ij} \quad = \quad \left. \frac{\partial\, e^{\overline{\boldsymbol{p}}}_{i}\big(\mathbf{e}^{\overline{\boldsymbol{q}}}\big)}{\partial\, e^{\overline{\boldsymbol{q}}}_{j}} \right|_{\mathbf{e}^{\overline{\boldsymbol{q}}} = \overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}} \quad , \tag{5.40}$$

then,

$$\mathbf{e}^{\overline{\boldsymbol{p}}} - \overline{\mathbf{e}}^{\overline{\boldsymbol{p}}} \quad \approx \quad \mathbf{e}^{\overline{\boldsymbol{p}}}\big(\mathbf{e}^{\overline{\boldsymbol{q}}}\big) \ - \ \mathbf{e}^{\overline{\boldsymbol{p}}}\big(\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\big) \quad \approx \tag{5.41}$$

$$\approx \quad \mathbf{T} \, \big(\mathbf{e}^{\overline{\boldsymbol{q}}} - \overline{\mathbf{e}}^{\overline{\boldsymbol{q}}} \big) \quad , \tag{5.42}$$

and the chart update for the covariance matrix will be computed through

$$\mathbf{P}^{\overline{\boldsymbol{q}}^{OB}_{n|n}}_{n|n} \quad = \quad \mathrm{E}\Big[\big(\mathbf{x}^{\overline{\boldsymbol{q}}^{OB}_{n|n}}_{n|n} - \overline{\mathbf{x}}^{\overline{\boldsymbol{q}}^{OB}_{n|n}}_{n|n}\big) \big(\mathbf{x}^{\overline{\boldsymbol{q}}^{OB}_{n|n}}_{n|n} - \overline{\mathbf{x}}^{\overline{\boldsymbol{q}}^{OB}_{n|n}}_{n|n}\big)^{T} \Big] \quad \approx \tag{5.43a}$$

$$\approx \quad \begin{pmatrix} \mathbf{T}(\,\overline{\boldsymbol{\delta}}_{n}\,) & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \mathbf{P}^{\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}}_{n|n} \begin{pmatrix} \mathbf{T}(\,\overline{\boldsymbol{\delta}}_{n}\,) & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}^{T} \quad . \tag{5.43b}$$

Table 5.1 summarizes the resulting $\mathbf{T}$-matrix for each chart introduced in Section 3.4, along with its application domain. A detailed derivation of each $\mathbf{T}$-matrix can be found in Appendix A.9.

Table 5.1: $\mathbf{T}$-matrices for the transition maps of the charts studied.

Chart	**$\mathbf{T}(\,\overline{\boldsymbol{\delta}}\,)$ Matrix**	**Domain**
O	$\overline{\delta}_0\,\mathbf{I} - \big[\,\overline{\boldsymbol{\delta}}\,\big]_{\times} + \dfrac{\overline{\boldsymbol{\delta}}\,\overline{\boldsymbol{\delta}}^{T}}{\overline{\delta}_0}$	$\{\ \overline{\boldsymbol{\delta}} \in S^3 \ : \ \overline{\delta}_0 > 0\ \}$
RP	$\overline{\delta}_0 \, \Big(\overline{\delta}_0\,\mathbf{I} - \big[\,\overline{\boldsymbol{\delta}}\,\big]_{\times} \Big)$	$\{\ \overline{\boldsymbol{\delta}} \in S^3 \ : \ \overline{\delta}_0 \neq 0\ \}$
MRP	$\dfrac{1}{2} \, \Big[\big(1 + \overline{\delta}_0\big) \Big(\overline{\delta}_0\,\mathbf{I} - \big[\,\overline{\boldsymbol{\delta}}\,\big]_{\times} \Big) + \overline{\boldsymbol{\delta}}\,\overline{\boldsymbol{\delta}}^{T} \Big]$	$\{\ \overline{\boldsymbol{\delta}} \in S^3 \ : \ \overline{\delta}_0 \geq 0\ \}$
RV	$\left[\overline{\delta}_0 \left(\mathbf{I} - \widehat{\overline{\boldsymbol{\delta}}}\,\widehat{\overline{\boldsymbol{\delta}}}^{T} \right) - \big[\,\overline{\boldsymbol{\delta}}\,\big]_{\times} \right] \dfrac{\|\overline{\boldsymbol{\delta}}\|}{\arcsin \|\overline{\boldsymbol{\delta}}\|} + \widehat{\overline{\boldsymbol{\delta}}}\,\widehat{\overline{\boldsymbol{\delta}}}^{T}$	$\{\ \overline{\boldsymbol{\delta}} \in S^3 \ : \ \overline{\delta}_0 \geq 0 \ , \ \|\overline{\boldsymbol{\delta}}\| \neq 0\ \}$

After the final computations we obtain the four

$$\left(\varphi \, , \, \overline{\boldsymbol{q}}^{OB}_{n|n} \, , \, {}^{B}\overline{\boldsymbol{\omega}}^{B}_{n|n} \, , \, \mathbf{P}^{\overline{\boldsymbol{q}}^{OB}_{n|n}}_{n|n} \right) \quad , \tag{5.44}$$

that is a condition equivalent to (5.8) in which we started the iteration.

5.3.5. MEKF Summary

At time t_n we receive a measurement $\mathbf{z}_n$. Our knowledge about the state of the system at a previous time t_{n-1}, is encoded in the mean $\overline{\mathbf{x}}_{n\text{-}1|n\text{-}1}$ and in the covariance matrix $\mathbf{P}_{n\text{-}1|n\text{-}1}$ of the random variable $\mathbf{x}_{n\text{-}1|n\text{-}1}$. We update our knowledge about the state of the system in four steps:

State prediction

$$\begin{aligned}
{}^{B}\overline{\boldsymbol{\omega}}^{B}_{n|n\text{-}1} &= {}^{B}\overline{\boldsymbol{\omega}}^{B}_{n\text{-}1|n\text{-}1} \quad , && (5.45)\\
\boldsymbol{\delta}^{\boldsymbol{\omega}}_{n} &= \begin{pmatrix} \cos\left(\frac{\| {}^{B}\overline{\boldsymbol{\omega}}^{B}_{n|n\text{-}1} \| \, \Delta t_n}{2} \right) \\ \frac{{}^{B}\overline{\boldsymbol{\omega}}^{B}_{n|n\text{-}1}}{\| {}^{B}\overline{\boldsymbol{\omega}}^{B}_{n|n\text{-}1} \|} \, \sin\left(\frac{\| {}^{B}\overline{\boldsymbol{\omega}}^{B}_{n|n\text{-}1} \| \, \Delta t_n}{2} \right) \end{pmatrix} \quad , && (5.46)\\
\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1} &= \overline{\boldsymbol{q}}^{OB}_{n\text{-}1|n\text{-}1} \, * \, \boldsymbol{\delta}^{\boldsymbol{\omega}}_{n} \quad , && (5.47)\\
\mathbf{F}_n &= \begin{pmatrix} \mathbf{R}^T(\boldsymbol{\delta}^{\boldsymbol{\omega}}_{n}) & \mathbf{I} \, \Delta t_n \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \quad , && (5.48)\\
\mathbf{P}^{\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}}_{n|n\text{-}1} &= \mathbf{F}_n \left[\mathbf{P}^{\overline{\boldsymbol{q}}^{OB}_{n\text{-}1|n\text{-}1}}_{n\text{-}1|n\text{-}1} + \mathbf{Q}_n \right] \mathbf{F}_n^T \quad , && (5.49)
\end{aligned}$$

with $\Delta t_n = t_n - t_{n\text{-}1}$, and

$$\mathbf{Q}_n = \begin{pmatrix} \mathbf{Q}^{\boldsymbol{\omega}}_{n} \, \frac{(\Delta t_n)^3}{3} & -\mathbf{Q}^{\boldsymbol{\omega}}_{n} \, \frac{(\Delta t_n)^2}{2} \\ -\mathbf{Q}^{\boldsymbol{\omega}}_{n} \, \frac{(\Delta t_n)^2}{2} & \mathbf{Q}^{\boldsymbol{\omega}}_{n} \, \Delta t_n \end{pmatrix} \quad . \tag{5.50}$$

Measurement prediction

$$\begin{aligned}
\overline{\mathbf{v}}^{m}_{n|n\text{-}1} &= \mathbf{R}^T\left(\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1} \right) \left(\overline{\mathbf{q}}^{\mathbf{v}}_{n} + {}^{O}\mathbf{v}_n \right) \quad , && (5.51)\\
\overline{\boldsymbol{\omega}}^{m}_{n|n\text{-}1} &= {}^{B}\overline{\boldsymbol{\omega}}^{B}_{n|n\text{-}1} \quad , && (5.52)\\
\overline{\mathbf{z}}_{n|n\text{-}1} &= \begin{pmatrix} \overline{\mathbf{v}}^{m}_{n|n\text{-}1} \\ \overline{\boldsymbol{\omega}}^{m}_{n|n\text{-}1} \end{pmatrix} \quad , && (5.53)\\
\mathbf{H}_n &= \begin{pmatrix} \left[\overline{\mathbf{v}}^{m}_{n|n\text{-}1} \right]_{\times} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \quad , && (5.54)\\
\mathbf{R}_n &= \begin{pmatrix} \mathbf{R}^T\left(\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1} \right) \mathbf{Q}^{\mathbf{v}}_{n} \, \mathbf{R}\left(\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1} \right) + \mathbf{R}^{\mathbf{v}}_{n} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^{\boldsymbol{\omega}}_{n} \end{pmatrix} \quad , && (5.55)\\
\mathbf{S}_{n|n\text{-}1} &= \mathbf{H}_n \, \mathbf{P}^{\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}}_{n|n\text{-}1} \, \mathbf{H}_n^T + \mathbf{R}_n \quad , && (5.56)
\end{aligned}$$

where $[\mathbf{v}]_{\times}$ have been defined previously (3.9).

Kalman update

$$\mathbf{K}_n = \mathbf{P}_{n|n\text{-}1}^{\overline{\boldsymbol{q}}_{n|n\text{-}1}^{OB}} \mathbf{H}_n^T \mathbf{S}_{n|n\text{-}1}^{-1} \quad , \tag{5.57}$$

$$\overline{\mathbf{x}}_{n|n}^{\overline{\boldsymbol{q}}_{n|n\text{-}1}^{OB}} = \overline{\mathbf{x}}_{n|n\text{-}1}^{\overline{\boldsymbol{q}}_{n|n\text{-}1}^{OB}} + \mathbf{K}_n \left(\mathbf{z}_n - \overline{\mathbf{z}}_{n|n\text{-}1} \right) \quad , \tag{5.58}$$

$$\mathbf{P}_{n|n}^{\overline{\boldsymbol{q}}_{n|n\text{-}1}^{OB}} = \left(\mathbf{I} - \mathbf{K}_n \mathbf{H}_n \right) \mathbf{P}_{n|n\text{-}1}^{\overline{\boldsymbol{q}}_{n|n\text{-}1}^{OB}} \left(\mathbf{I} - \mathbf{K}_n \mathbf{H}_n \right)^T + \mathbf{K}_n \mathbf{R}_n \mathbf{K}_n^T \quad , \tag{5.59}$$

where $\overline{\mathbf{x}}_{n|n\text{-}1}^{\overline{\boldsymbol{q}}_{n|n\text{-}1}^{OB}} = \left(\overline{\mathbf{e}}_{n|n\text{-}1}^{\overline{\boldsymbol{q}}_{n|n\text{-}1}^{OB}} = \mathbf{0} \, , \, {}^B\overline{\boldsymbol{\omega}}_{n|n\text{-}1}^B \right)^T$.

Chart update

$$\overline{\boldsymbol{\delta}}_n = \varphi^{-1}\left(\overline{\mathbf{e}}_{n|n}^{\overline{\boldsymbol{q}}_{n|n\text{-}1}^{OB}} \right) \quad , \tag{5.60}$$

$$\overline{\boldsymbol{q}}_{n|n}^{OB} = \overline{\boldsymbol{q}}_{n|n\text{-}1}^{OB} * \overline{\boldsymbol{\delta}}_n \quad , \tag{5.61}$$

$$\mathbf{P}_{n|n}^{\overline{\boldsymbol{q}}_{n|n}^{OB}} = \begin{pmatrix} \mathbf{T}(\overline{\boldsymbol{\delta}}_n) & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \mathbf{P}_{n|n}^{\overline{\boldsymbol{q}}_{n|n\text{-}1}^{OB}} \begin{pmatrix} \mathbf{T}(\overline{\boldsymbol{\delta}}_n) & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}^T \quad , \tag{5.62}$$

being φ and $\mathbf{T}(\overline{\boldsymbol{\delta}}_n)$ defined from the chosen chart (Tables 3.2 and 5.1 respectively).

5.4. Manifold Unscented Kalman Filter

At time t_n we receive a measurement $\mathbf{z}_n$. Our knowledge about the orientation at a previous time $t_{n\text{-}1}$ is described by a distribution expressed in the $\overline{\boldsymbol{q}}_{n\text{-}1|n\text{-}2}^{OB}$-centered chart. This distribution is encoded in the four

$$\left(\varphi \, , \, \overline{\boldsymbol{q}}_{n\text{-}1|n\text{-}2}^{OB} \, , \, \overline{\mathbf{x}}_{n\text{-}1|n\text{-}1}^{\overline{\boldsymbol{q}}_{n\text{-}1|n\text{-}2}^{OB}} \, , \, \mathbf{P}_{n\text{-}1|n\text{-}1}^{\overline{\boldsymbol{q}}_{n\text{-}1|n\text{-}2}^{OB}} \right) . \tag{5.63}$$

5.4.1. Unscented Transformation

The first step in the UKF is to create the augmented $N \times 1$ mean $\widetilde{\mathbf{x}}_n$ and $N \times N$ covariance matrix $\widetilde{\mathbf{P}}_n$. Since the measurement equations are linear for the random variables $\mathbf{r}_t^{\boldsymbol{\omega}}$ and $\mathbf{r}_t^{\mathbf{v}}$, we can leave their covariance matrices out of the augmented one and add them later:

$$\widetilde{\mathbf{x}}_n = \begin{pmatrix} \overline{\mathbf{x}}_{n\text{-}1|n\text{-}1}^{\overline{\boldsymbol{q}}_{n\text{-}1|n\text{-}2}^{OB}} \\ \overline{\mathbf{q}}_n^{\boldsymbol{\omega}} \\ \overline{\mathbf{q}}_n^{\mathbf{v}} \end{pmatrix} \quad , \tag{5.64a}$$

$$\widetilde{\mathbf{P}}_n = \begin{pmatrix} \mathbf{P}_{n\text{-}1|n\text{-}1}^{\overline{\boldsymbol{q}}_{n\text{-}1|n\text{-}2}^{OB}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_n^{\boldsymbol{\omega}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_n^{\mathbf{v}} \end{pmatrix} \quad . \tag{5.64b}$$

Then, we obtain the matrix $\mathbf{L}_n$ which satisfies $\mathbf{L}_n \mathbf{L}_n^T = \widetilde{\mathbf{P}}_n$ and we use it to generate the $2N+1$ sigma points $\{\mathcal{X}_j\}_{j=0}^{2N}$ as described in Section 2.5.1:

$$\mathcal{X}_{i,0} = (\widetilde{\mathbf{x}}_n)_i \quad , \tag{5.65a}$$

$$\mathcal{X}_{i,j} = (\widetilde{\mathbf{x}}_n)_i + \frac{(\mathbf{L}_n)_{ij}}{\sqrt{2W_j}} \qquad \text{for} \quad j = 1, \ldots, N \quad , \tag{5.65b}$$

$$\mathcal{X}_{i,j+N} = (\widetilde{\mathbf{x}}_n)_i - \frac{(\mathbf{L}_n)_{ij}}{\sqrt{2W_j}} \qquad \text{for} \quad j = 1, \ldots, N \quad , \tag{5.65c}$$

being $W_j = (1 - W_0)/(2N)$ for $j \neq 0$ and W_0 regulates the importance given to the sigma point $\mathcal{X}_0$ in the computation of the mean. These sigma points $\{\mathcal{X}_j\}_j$ are expressed in the

$\overline{\boldsymbol{q}}^{OB}_{n\text{-}1|n\text{-}2}$-centered chart. We need to express them in the manifold before applying the evolution equations and the measurement equations:

$$\boldsymbol{\mathcal{X}}^{\boldsymbol{q}}_j = \varphi^{-1}_{\overline{\boldsymbol{q}}^{OB}_{n\text{-}1|n\text{-}2}}\big(\boldsymbol{\mathcal{X}}^{\mathbf{e}}_j\big) = \tag{5.66a}$$

$$= \overline{\boldsymbol{q}}^{OB}_{n\text{-}1|n\text{-}2} * \varphi^{-1}\big(\boldsymbol{\mathcal{X}}^{\mathbf{e}}_j\big) \quad , \tag{5.66b}$$

where for the j-th sigma point, $\boldsymbol{\mathcal{X}}^{\mathbf{e}}_j$ is its chart point part and $\boldsymbol{\mathcal{X}}^{\boldsymbol{q}}_j$ is the quaternion with which it is mapped. Note that when applying the inverse chart φ^{-1} we will need to "saturate" $\boldsymbol{\mathcal{X}}^{\mathbf{e}}_j$ to the closest point in the image of φ.

5.4.2. Prediction

We compute the predicted state and measurement associated with each sigma point $\boldsymbol{\mathcal{X}}_j$ through

$$\boldsymbol{\mathcal{Y}}^{\boldsymbol{\omega}}_j = \boldsymbol{\mathcal{X}}^{\boldsymbol{\omega}}_j + \boldsymbol{\mathcal{X}}^{\mathbf{q}^{\boldsymbol{\omega}}}_j \, \Delta t_n \quad , \tag{5.67}$$

$$\boldsymbol{\mathcal{Y}}^{\boldsymbol{q}}_j = \boldsymbol{\mathcal{X}}^{\boldsymbol{q}}_j * \begin{pmatrix} \cos\left(\frac{\|\boldsymbol{\mathcal{Y}}^{\boldsymbol{\omega}}_j\| \, \Delta t_n}{2}\right) \\ \widehat{\boldsymbol{\mathcal{Y}}}^{\boldsymbol{\omega}}_j \, \sin\left(\frac{\|\boldsymbol{\mathcal{Y}}^{\boldsymbol{\omega}}_j\| \, \Delta t_n}{2}\right) \end{pmatrix} \quad , \tag{5.68}$$

$$\boldsymbol{\mathcal{Z}}^{\mathbf{v}}_j = \mathbf{R}^T\big(\boldsymbol{\mathcal{X}}^{\boldsymbol{q}}_j\big)\,\big(\boldsymbol{\mathcal{X}}^{\mathbf{v}}_j + {}^{O}\mathbf{v}_t\big) \quad , \tag{5.69}$$

$$\boldsymbol{\mathcal{Z}}^{\boldsymbol{\omega}}_j = \boldsymbol{\mathcal{Y}}^{\boldsymbol{\omega}}_j \quad , \tag{5.70}$$

where for the j-th sigma point, $\boldsymbol{\mathcal{X}}^{\boldsymbol{\omega}}_j$ is its angular velocity part, $\boldsymbol{\mathcal{X}}^{\mathbf{q}^{\boldsymbol{\omega}}}_j$ is its angular velocity noise part, $\boldsymbol{\mathcal{Y}}^{\boldsymbol{\omega}}_j$ is its angular velocity prediction, $\boldsymbol{\mathcal{Y}}^{\boldsymbol{q}}_j$ is the quaternion part of its prediction[1], $\boldsymbol{\mathcal{X}}^{\mathbf{v}}_j$ is its process noise part, $\boldsymbol{\mathcal{Z}}^{\mathbf{v}}_j$ is its vector measurement prediction, $\boldsymbol{\mathcal{Z}}^{\boldsymbol{\omega}}_j$ is its angular velocity measurement prediction, and $\Delta t_n = t_n - t_{n\text{-}1}$.

Having these new sigma points, we can obtain the means and covariance matrices of the distributions present in the UKF. First, defining $\boldsymbol{\mathcal{Z}}_j := \big(\boldsymbol{\mathcal{Z}}^{\mathbf{v}}_j \, , \, \boldsymbol{\mathcal{Z}}^{\boldsymbol{\omega}}_j\big)^T$, the means are computed through

$$\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1} = \frac{\sum_j W_j \, \boldsymbol{\mathcal{Y}}^{\boldsymbol{q}}_j}{\|\sum_j W_j \, \boldsymbol{\mathcal{Y}}^{\boldsymbol{q}}_j\|} \quad , \tag{5.71}$$

$${}^{B}\overline{\boldsymbol{\omega}}^{B}_{n|n\text{-}1} = \sum_j W_j \, \boldsymbol{\mathcal{Y}}^{\boldsymbol{\omega}}_j \quad , \tag{5.72}$$

$$\overline{\mathbf{x}}^{\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}}_{n|n\text{-}1} = \begin{pmatrix} \varphi_{\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}}\big(\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}\big) = \mathbf{0} \\ {}^{B}\overline{\boldsymbol{\omega}}^{B}_{n|n\text{-}1} \end{pmatrix} \quad , \tag{5.73}$$

$$\overline{\mathbf{z}}_{n|n\text{-}1} = \sum_j W_j \, \boldsymbol{\mathcal{Z}}_j \quad . \tag{5.74}$$

where we have used a variation of the result provided in [57]. Namely,

$$\overline{\boldsymbol{q}} \approx \frac{\sum_j \boldsymbol{q}_j}{\|\sum_j \boldsymbol{q}_j\|} \quad , \tag{5.75}$$

with $\boldsymbol{q}_j \cdot \boldsymbol{q}_k > 0$ for $j, k = 0, \ldots, 2N$. This result is shown to minimize the fourth order approximation of the distance defined as the sum of squared angles between the rotation transformation represented by each quaternion $\boldsymbol{q}_j$, and the one represented by $\overline{\boldsymbol{q}}$. This approach to compute the mean quaternion is extremely efficient, and the derivation is elegant and simple. In order to ensure that $\boldsymbol{q}_j \cdot \boldsymbol{q}_k > 0$, it is useful to remember the property that both $\boldsymbol{q}$ and $-\boldsymbol{q}$ represent the same rotation. This property is also useful for introducing the quaternions in the domain of φ to execute the next step of the filter.

[1] We have assumed that the angular velocity $\boldsymbol{\mathcal{Y}}^{\boldsymbol{\omega}}_j$ is constant in the time interval $[t_{n\text{-}1}, t_n)$ so that we can use (5.14b).

After this, we use the obtained mean quaternion $\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}$ to express each sigma point in the $\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}$-centered chart, and compute the covariance matrices:

$$\boldsymbol{\mathcal{Y}}^{\mathbf{e}}_j = \varphi_{\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}}\big(\boldsymbol{\mathcal{Y}}^{\boldsymbol{q}}_j\big) = \varphi\big((\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1})^* * \boldsymbol{\mathcal{Y}}^{\boldsymbol{q}}_j\big) \quad , \tag{5.76}$$

$$\mathbf{P}^{\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}}_{n|n\text{-}1} = \sum_j W_j\, \boldsymbol{\mathcal{Y}}_j\, \boldsymbol{\mathcal{Y}}_j^T \quad , \tag{5.77}$$

$$\mathbf{P}^{yz}_{n|n\text{-}1} = \sum_j W_j\, \boldsymbol{\mathcal{Y}}_j \left(\boldsymbol{\mathcal{Z}}_j - \overline{\mathbf{z}}_{n|n\text{-}1}\right)^T \quad , \tag{5.78}$$

$$\mathbf{S}_{n|n\text{-}1} = \sum_j W_j\, \left(\boldsymbol{\mathcal{Z}}_j - \overline{\mathbf{z}}_{n|n\text{-}1}\right)\left(\boldsymbol{\mathcal{Z}}_j - \overline{\mathbf{z}}_{n|n\text{-}1}\right)^T + \begin{pmatrix} \mathbf{R}^{\mathbf{v}}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^{\boldsymbol{\omega}}_n \end{pmatrix} \quad , \tag{5.79}$$

where we have denoted $\boldsymbol{\mathcal{Y}}_j := \big(\, \boldsymbol{\mathcal{Y}}^{\mathbf{e}}_j \;,\; \boldsymbol{\mathcal{Y}}^{\boldsymbol{\omega}}_j - {}^B\overline{\boldsymbol{\omega}}^B_{n|n\text{-}1} \,\big)^T$.

5.4.3. Kalman Update

Finally, we compute the UKF version of the Kalman gain $\mathbf{K}_n$, and we use it to obtain the optimal estimation of the state:

$$\mathbf{K}_n = \mathbf{P}^{yz}_{n|n\text{-}1}\, \mathbf{S}^{-1}_{n|n\text{-}1} \quad , \tag{5.80}$$

$$\overline{\mathbf{x}}^{\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}}_{n|n} = \overline{\mathbf{x}}^{\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}}_{n|n\text{-}1} + \mathbf{K}_n \left(\mathbf{z}_n - \overline{\mathbf{z}}_{n|n\text{-}1}\right) \quad , \tag{5.81}$$

$$\mathbf{P}^{\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}}_{n|n} = \mathbf{P}^{\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}}_{n|n\text{-}1} - \mathbf{K}_n\, \mathbf{S}_{n|n\text{-}1}\, \mathbf{K}_n^T \quad , \tag{5.82}$$

arriving at the same conditions in which we began the iteration, with a distribution expressed in the $\overline{\boldsymbol{q}}_{n|n\text{-}1}$-centered chart, and encoded by the four

$$\big(\; \varphi \;,\; \overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1} \;,\; \overline{\mathbf{x}}^{\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}}_{n|n} \;,\; \mathbf{P}^{\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}}_{n|n} \;\big) \quad . \tag{5.83}$$

Our best estimation for the orientation at this time is

$$\overline{\boldsymbol{q}}^{OB}_{n|n} = \varphi^{-1}_{\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}}\big(\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}}_{n|n}\big) = \tag{5.84a}$$

$$= \overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1} * \varphi^{-1}\Big(\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}}_{n|n}\Big) \quad , \tag{5.84b}$$

being $\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}}_{n|n}$ the part of the mean $\overline{\mathbf{x}}^{\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}}_{n|n}$ that represents the quaternion in the $\overline{\boldsymbol{q}}^{OB}_{n|n\text{-}1}$-centered chart.

Note that setting $\overline{\boldsymbol{q}}^{OB}_{n\text{-}1|n\text{-}2} := \overline{\boldsymbol{q}}^{OB}_{n\text{-}1|n\text{-}1}$ and $\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}^{OB}_{n\text{-}1|n\text{-}2}}_{n\text{-}1|n\text{-}1} := \mathbf{0}$ at the beginning of each iteration yields the traditional version of the algorithm, where a "reset operation" is performed instead of the covariance matrix update.

5.5. Tests and Results

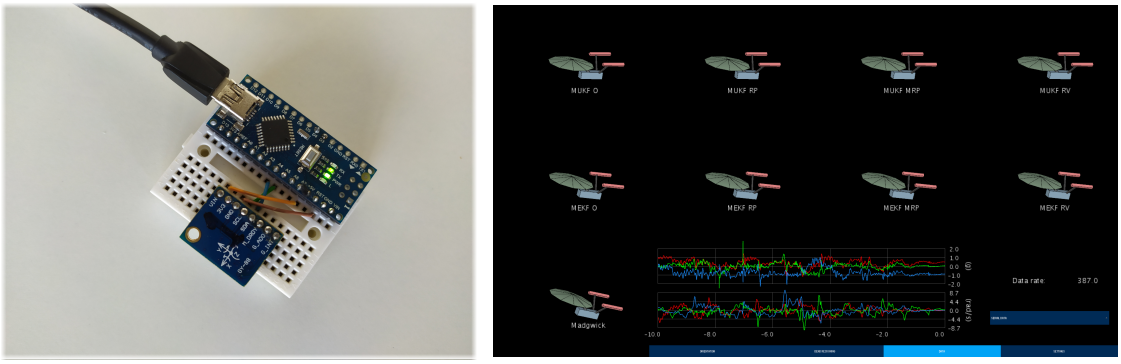
We developed two estimators: one based on the EKF, and another based on the UKF. In order to validate them, they underwent two different tests. The first was an experimental test with which we qualitatively verified that the estimators worked correctly. The second was a simulation test from which we extracted quantitative measures on the performance of each estimator.

5.5.1. Experimental Test

The MEKF and MUKF code was written in Processing[2] (Java). We used a library[3] to read data from an IMU (MPU6050) using an Arduino[4]. The data was sent from the Arduino to a PC

[2] `https://processing.org/` [Online; accessed January-2020]
[3] `https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050` [Online; accessed January-2020]
[4] `https://www.arduino.cc/` [Online; accessed January-2020]

using the serial port. The PC received the measurements, and entered them into the MKFs and the popular Madgwick algorithm [5] that was used to compare. This test helped to debug the code and qualitatively validate the estimation algorithms. The source code can be found in GitHub[5]. Figure 5.1 shows the hardware and a screenshot of the running Processing application.



(a) (b)

Figure 5.1: Images related to the experimental test used to qualitatively characterize the MKFs. **(a)** Hardware used to retrieve the IMU (MPU6050) measurements. **(b)** Screenshot of the running Processing sketch.

5.5.2. Simulation Test and Results

Once the MKFs have been validated, we would like to select the one that produces the best estimations. This section presents the results of the simulations used to measure the accuracy of each estimator. Simulations are chosen instead of real experiments because a real system entails an uncertainty in the measurement of the true attitude: the attitude that is used to compare with that estimated by the algorithms. There are sources of error ranging from a miscalibration of the measurement system to a possible bias in the "true attitude" produced by another attitude estimator, which makes it problematic to define an adequate metric to measure the accuracy of the algorithms. For this reason, the author consider that using a simulation is more reliable to avoid possible biases in the results.

Others have performed similar tests [13, 58, 16]. However, the results do not seem to be statistically conclusive: only the estimations of some orientation trajectories are shown. We perform our comparison through a simulation in which we do have an absolute knowledge of the attitude of the system: a true oracle exists in a simulation. Therefore, we can compare the real orientation with the attitude estimated by the algorithms having fed them only with simulated measurements that we obtain from such known orientations. We will extract our performance metrics from a wide set of orientation trajectories in order to obtain statistically conclusive results.

We try to answer three questions with the simulation test. The first question is, is there a chart for which we get a greater accuracy in attitude estimation? The second one is, what algorithm produces the most accurate attitude estimation; the MEKF or the MUKF? The last question stems from the fact that previous algorithms on attitude estimation, such as the Multiplicative Extended Kalman Filter, did not contemplate updating the distribution from one chart to another as done at (5.62) in the MEKF. However, their estimators performed well [11, 13]. Then the third question is, does this "chart update" imply an improvement in the accuracy of the attitude estimation?

5.5.2.1. Performance Metric

We have already described a quaternion $\boldsymbol{q}$ as a deviation from another quaternion $\overline{\boldsymbol{q}}$ as $\boldsymbol{q} = \overline{\boldsymbol{q}} * \boldsymbol{\delta}^{\overline{\boldsymbol{q}}}$ (3.37). Now we define the instantaneous error between an estimated attitude, represented by a unit quaternion $\overline{\boldsymbol{q}}$, and the real attitude, represented by the unit quaternion $\overset{\star}{\boldsymbol{q}}$, as the angle we have to rotate one of them to transform it into the other. This is, the angle of the rotation transformation defined by the quaternion $\boldsymbol{\delta}_e$ such that $\overset{\star}{\boldsymbol{q}} = \overline{\boldsymbol{q}} * \boldsymbol{\delta}_e$. Recalling (3.31), this

[5] `https://github.com/PBernalPolo/test_MKF` [Online; accessed January-2020]

angle can be computed as:

$$\theta_e \quad = \quad 2 \arccos\left[\left(\overline{\boldsymbol{q}}^* * \overset{\star}{\boldsymbol{q}} \right)_0 \right] \quad = \tag{5.85a}$$

$$= \quad 2 \arccos\left(\overline{\boldsymbol{q}} \cdot \overset{\star}{\boldsymbol{q}} \right) \quad , \tag{5.85b}$$

having previously ensured that $\overline{\boldsymbol{q}} \cdot \overset{\star}{\boldsymbol{q}} \geq 0$ using the fact that both $\boldsymbol{q}$ and $-\boldsymbol{q}$ represent the same rotation transformation. The angle θ_e will vary along an orientation trajectory. Then, we will define the mean error in orientation estimation for a given trajectory starting at time $t = 0$ and ending at time $t = T$ as

$$e_\theta \quad = \quad \frac{1}{T} \int_0^T \theta_e(t) \, dt \quad . \tag{5.86}$$

Finally, e_θ will depend on the trajectory followed, and the set of measurements generated with that trajectory. We will need to generate several orientation trajectories to obtain the mean value $\overline{e}_\theta$ and the variance $\sigma^2_{\overline{e}_\theta}$ that characterize the distribution of the error in orientation estimation, e_θ, for each algorithm. We will define the confidence interval for the value $\overline{e}_\theta$ as

$$\left[\, \overline{e}_\theta - 3\,\sigma_{\overline{e}_\theta}/\sqrt{N_s} \;\; , \;\; \overline{e}_\theta + 3\,\sigma_{\overline{e}_\theta}/\sqrt{N_s} \, \right] \quad , \tag{5.87}$$

where N_s is the number of samples taken for the $\overline{e}_\theta$ computation, so that $\sigma^2_{\overline{e}_\theta}/N_s$ is the variance of the sample mean distribution. Being that the lower the better, the value of $\overline{e}_\theta$ gives us a measure of how well an algorithm estimates the orientation. We will consider that the performance of an algorithm A is better than the performance of other algorithm B if $\overline{e}_\theta(\mathrm{A}) < \overline{e}_\theta(\mathrm{B})$ and their confidence intervals do not overlap.

5.5.2.2. Simulation Scheme

To compute the performance metrics we will need to generate a large number of simulations. Each independent simulation will consist of three steps: initialization, convergence, and estimation.

In the initialization step we set up the initial conditions accordingly to the chosen simulation parameters. This includes generating the initial unit quaternion $\overset{\star}{\boldsymbol{q}}_0$ from a uniform distribution in S^3, setting the initial angular velocity ${}^B\overset{\star}{\boldsymbol{\omega}}^B_0$ to zero, setting the update frequency f_{update}, generating the variances of the process noises $\sigma^2_{\boldsymbol{\omega}}$ and $\sigma^2_{\mathbf{v}}$ from a uniform distribution in the intervals $(0, Q^{\boldsymbol{\omega}}_{\max}]$ and $(0, Q^{\mathbf{v}}_{\max}]$ respectively, and initializing the estimation algorithm. The initialization of the MEKF includes setting $\overline{\boldsymbol{q}}^{OB}_{0|0} = \mathbf{1}$, ${}^B\overline{\boldsymbol{\omega}}^B_{0|0} = \mathbf{0}\,\mathrm{rad/s}$, and $\mathbf{P}^{\overline{\boldsymbol{q}}^{OB}_{0|0}}_{0|0} = 10^2\,\mathbf{I}$. On the other hand, the initialization of the MUKF includes setting $\overline{\boldsymbol{q}}^{OB}_{0|-1} = \mathbf{1}$, $\mathbf{e}^{\overline{\boldsymbol{q}}^{OB}_{0|-1}}_{0|0} = \mathbf{0}$, ${}^B\overline{\boldsymbol{\omega}}^B_{0|0} = (1,1,1)^T\,\mathrm{rad/s}$, and $\mathbf{P}^{\overline{\boldsymbol{q}}^{OB}_{0|-1}}_{0|0} = 10^2\,\mathbf{I}$. The angular velocity is not initialized to $\mathbf{0}$ in the MUKF because it has been observed that it is sometimes necessary to "break the symmetry" for the algorithm to converge; especially when we do not apply the chart update (when we perform the "reset operation") for the RV chart. The covariance matrices that appear in both algorithms are initialized as $\mathbf{Q}^{\boldsymbol{\omega}}_n = \mathbf{I}\,\mathrm{rads}^2/\mathrm{s}^3$, $\mathbf{Q}^{\mathbf{v}}_n = 10^{-2}\,\mathbf{I}\,\mathrm{p.d.u.}^6$, $\mathbf{R}^{\boldsymbol{\omega}}_n = R^{\boldsymbol{\omega}}\,\mathbf{I}\,\mathrm{rads}^2/\mathrm{s}^2$, $\mathbf{R}^{\mathbf{v}}_n = R^{\mathbf{v}}\,\mathbf{I}\,\mathrm{p.d.u.}$, where $R^{\boldsymbol{\omega}}$ and $R^{\mathbf{v}}$ are the variances of the measurement noise that will be used in the simulation. We give this information about the measurement noise to the algorithms because it can be obtained offline, while the information about the process noise cannot. Given that a priori we cannot know how the system will behave, the values of $\mathbf{Q}^{\boldsymbol{\omega}}_n$ and $\mathbf{Q}^{\mathbf{v}}_n$ have been chosen according to what we understand could be normal. Choosing these values we are assuming that after a second it is normal for the angular velocity to have changed by $1\,\mathrm{rad/s}$, and also that it is normal to find external noises added to the vector ${}^O\mathbf{v}_t$ of magnitude $10^{-1}\,\mathrm{p.d.u.}$. For the mean values we set $\overline{\mathbf{q}}^{\boldsymbol{\omega}}_n = \mathbf{0}\,\mathrm{rad/s}$, and $\overline{\mathbf{q}}^{\mathbf{v}} = \mathbf{0}\,\mathrm{p.d.u.}$.

In the convergence step we keep the system in the initial orientation $\overset{\star}{\boldsymbol{q}}_0$. Simulated measurements are generated using (5.4) and (5.5). For each measurement, a different ${}^O\mathbf{v}_t$ is sampled from a uniform distribution in the unit sphere of $\mathbb{R}^3$. The values for each component of $\mathbf{q}^{\mathbf{v}}_t$, $\mathbf{r}^{\mathbf{v}}_t$, and $\mathbf{r}^{\boldsymbol{\omega}}_t$ are obtained from normal distributions with zero mean and variances $\sigma^2_{\mathbf{v}}$, $R^{\mathbf{v}}$, and $R^{\boldsymbol{\omega}}$ respectively. The term $\mathbf{R}^T(\boldsymbol{q}_t)$ in (5.4) is obtained from the true attitude $\overset{\star}{\boldsymbol{q}}_t$, which in the convergence

[6] "p.d.u." stands for "Procedure Defined Unit". In our case it depends on the definition of the vector ${}^O\mathbf{v}$.

step takes the value $\overset{\star}{\boldsymbol{q}}_t = \overset{\star}{\boldsymbol{q}}_0$. The term ${}^B\boldsymbol{\omega}^B_t$ in (5.5) is the true angular velocity, which in the convergence step takes the value ${}^B\overset{\star}{\boldsymbol{\omega}}^B_t = \mathbf{0}$. The tested algorithm updates its state estimation until the inequality $\theta_e(t) < \theta^0_e$ is satisfied, where $\theta_e(t)$ is the value of the error (5.85), and θ^0_e is a parameter in the simulation. The convergence step could have been replaced by an initialization of the attitude estimated by the algorithm $\overline{\boldsymbol{q}}_t$ to the real value $\overset{\star}{\boldsymbol{q}}_t$, but then it would have also been necessary to fix a certain covariance matrix. Since the metric of the space generated by each chart is different, it is difficult to set a covariance matrix that provides the same information for every chart. The author considered it more natural to allow the algorithm to find the true attitude by its own means, and for the covariance matrix to converge to a value in each case.

Finally, in the estimation step we generate a random but continuous orientation sequence using a Wiener process for the angular velocity:

$$ {}^B\overset{\star}{\boldsymbol{\omega}}^B_t = {}^B\overset{\star}{\boldsymbol{\omega}}^B_{t\text{-}\delta t} + \mathbf{n}_t \sqrt{\delta t} \quad , \tag{5.88} $$

$$ \overset{\star}{\boldsymbol{q}}_t = \overset{\star}{\boldsymbol{q}}_{t\text{-}\delta t} * \begin{pmatrix} \cos\left(\frac{\| {}^B\overset{\star}{\boldsymbol{\omega}}^B_t \| \, \delta t}{2}\right) \\ \frac{{}^B\overset{\star}{\boldsymbol{\omega}}^B_t}{\| {}^B\overset{\star}{\boldsymbol{\omega}}^B_t \|} \sin\left(\frac{\| {}^B\overset{\star}{\boldsymbol{\omega}}^B_t \| \, \delta t}{2}\right) \end{pmatrix} \quad , \tag{5.89} $$

where $\mathbf{n}_t$ is a random vector whose components are sampled from a normal distribution with zero mean and variance $\sigma^2_{\boldsymbol{\omega}}$, and δt is the simulation time step that is related to the algorithm time step Δt trough $\texttt{dtdtsim}\,\delta t = \Delta t$, being `dtdtsim` an integer parameter that determines the simulation updates per algorithm update. Note that we multiply $\mathbf{n}_t$ by $\sqrt{\delta t}$ and not by δt. We do it this way so that the covariance matrix after k steps does not depend on the simulation time step δt. In fact, after a time $T = k\,\delta t$ the covariance matrix of the angular velocity will have grown by $\Delta\mathbf{P}^{\boldsymbol{\omega}} = k\,\mathbf{I}\,\sigma^2_{\boldsymbol{\omega}}\,\delta t = \mathbf{I}\,\sigma^2_{\boldsymbol{\omega}}\,T$, and not by $(\Delta\mathbf{P}^{\boldsymbol{\omega}})' = k\,\mathbf{I}\,\sigma^2_{\boldsymbol{\omega}}\,(\delta t)^2 = \mathbf{I}\,\sigma^2_{\boldsymbol{\omega}}\,T\,\delta t$. After each `dtdtsim` simulation updates, a simulated measurement is generated in the same way it was done in the convergence step, and the algorithm is updated with it. The simulation runs for a time $\texttt{Tsim} = k'\,\Delta t$, where k' is an integer number. This way we perform the last algorithm update at the end of the simulation. The error (5.85) is evaluated after each algorithm update, and it is added up through the simulation to obtain the averaged error (5.86). After each simulation, we obtain a sample for the computation of $\overline{e}_\theta$ and $\sigma^2_{\overline{e}_\theta}$. We perform N_s of these simulations to obtain the confidence interval (5.87).

5.5.2.3. Simulation Results and Discussion

In this section we present the results of the simulations. The algorithms are tested for update frequencies $f_{\text{update}} = 1/\Delta t$ in the interval $[2, 1000]\,\text{Hz}$. This range has been chosen thinking about the possible limitations of a real system. For example, the maximum data rate of a low cost IMU is around $1000\,\text{Hz}$. On the other hand, the update frequency may be limited by processing. The computational cost of each estimator has been evaluated in two platforms: an Arduino MEGA 2560, and a Raspberry Pi 3 Model B. The code has been written in c++. The resulting maximum update frequencies are presented in Figure 5.2, which indicates that the MEKF can be executed approximately 3 times faster than the MUKF.

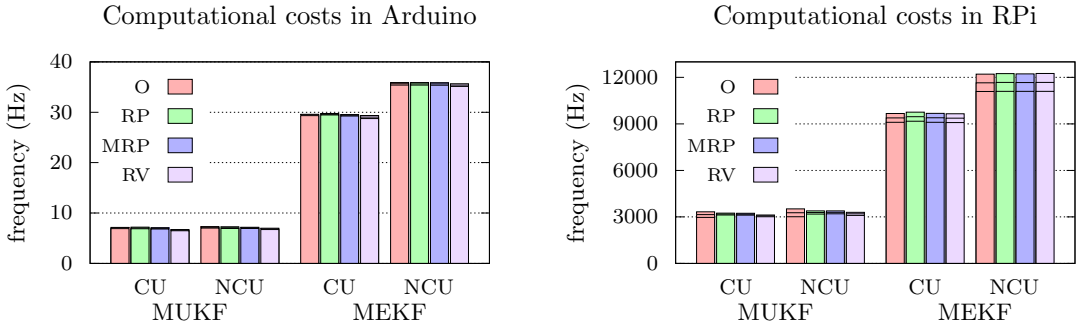


Figure 5.2: Maximum update frequency for each approach. The lines at the top represent the mean and the deviation (3σ) of the distribution of maximum update frequencies. "CU" stands for Chart Update, while "NCU" stands for No Chart Update.

Although the algorithms have been developed allowing a different Δt_n for each update, the simulations are performed using a constant Δt, and the simulation parameters depicted in Table 5.2.

Table 5.2: Parameters used in the simulations.

Parameter	**Value**
θ_e^0	1°
`Tsim`	10 s
`dtdtsim`	100
N_s	1000
$Q^{\boldsymbol{\omega}}_{\max}$	$10^2\,\mathrm{rad}^2/\mathrm{s}^3$
$Q^{\mathbf{v}}_{\max}$	1 p.d.u.
R	$\{\,10^{-2}\,,\,10^{-4}\,,\,10^{-6}\,\}$
$R^{\boldsymbol{\omega}}$	$R\,\mathrm{rad}^2/\mathrm{s}^2$
$R^{\mathbf{v}}$	R p.d.u.
W_0	$1/25$

The parameters θ_e^0, `Tsim`, `dtdtsim`, and N_s have been chosen trying to reach a compromise between the precision of the results, and the execution time of the simulation. The values for $Q^{\boldsymbol{\omega}}_{\max}$ and $Q^{\mathbf{v}}_{\max}$ have been chosen in such a way that the estimation algorithms face both normal situations ($\mathbf{Q}^{\boldsymbol{\omega}}_n \approx \sigma^2_{\boldsymbol{\omega}}\mathbf{I}$ and $\mathbf{Q}^{\mathbf{v}}_n \approx \sigma^2_{\mathbf{v}}\mathbf{I}$) and situations that were not foreseen ($\mathbf{Q}^{\boldsymbol{\omega}}_n \not\approx \sigma^2_{\boldsymbol{\omega}}\mathbf{I}$ or $\mathbf{Q}^{\mathbf{v}}_n \not\approx \sigma^2_{\mathbf{v}}\mathbf{I}$). A typical low cost IMU has $R^{\boldsymbol{\omega}} \approx 10^{-4}\,\mathrm{rad}^2/\mathrm{s}^2$ and $R^{\mathbf{v}} \approx 10^{-4}\,g^2$. The values chosen for R represent an imprecise sensor (10^{-2}), a normal sensor (10^{-4}), and a precise sensor (10^{-6}). The value of W_0 has been chosen so that all sigma points have the same importance, but very similar results, if not identical, have been obtained for other selections of W_0.

Chart choice

Results: The results of the simulation are presented in Figure 5.3. The average of the performance metric is shown along with its confidence interval for each of the selected update frequencies. The results of the MEKF and the MUKF are shown in different graphs, but drawn in the same one are the results for each chart and for a given MKF. In this way we are able to distinguish if a chart has an advantage over the others.

Discussion: We observe that there is no chart that is especially advantageous. All things being equal, we would opt for the RP chart. For this chart it is not necessary to worry about the domain since it maps $\boldsymbol{q}$ and $-\boldsymbol{q}$ with the same point of $\mathbb{R}^3$ and with the same $\mathbf{T}$-matrix; or of the image since it is all $\mathbb{R}^3$. In addition, the expressions of φ^{-1} and the $\mathbf{T}$-matrix for the MEKF are simpler for the RP chart. These computational advantages make us prefer the RP chart over the others.

MEKF vs. MUKF

Results: Figure 5.4 also presents the results of the simulations. This time, we display on the same graph the resulting performance metrics for the MUKF and the MEKF when the RP chart is used. In this way, we can distinguish if one MKF has an advantage over the other.

Discussion: We note that the MEKF performs the same or better than the MUKF. Similar results were previously reported in [58, 21]. This differs from the usual experience, in which the UKF outperforms the EKF in traditional non-linear estimation applications. The fact that the charts resemble the Euclidean space near the origin (see Section 3.4) might be favoring the MEKF,

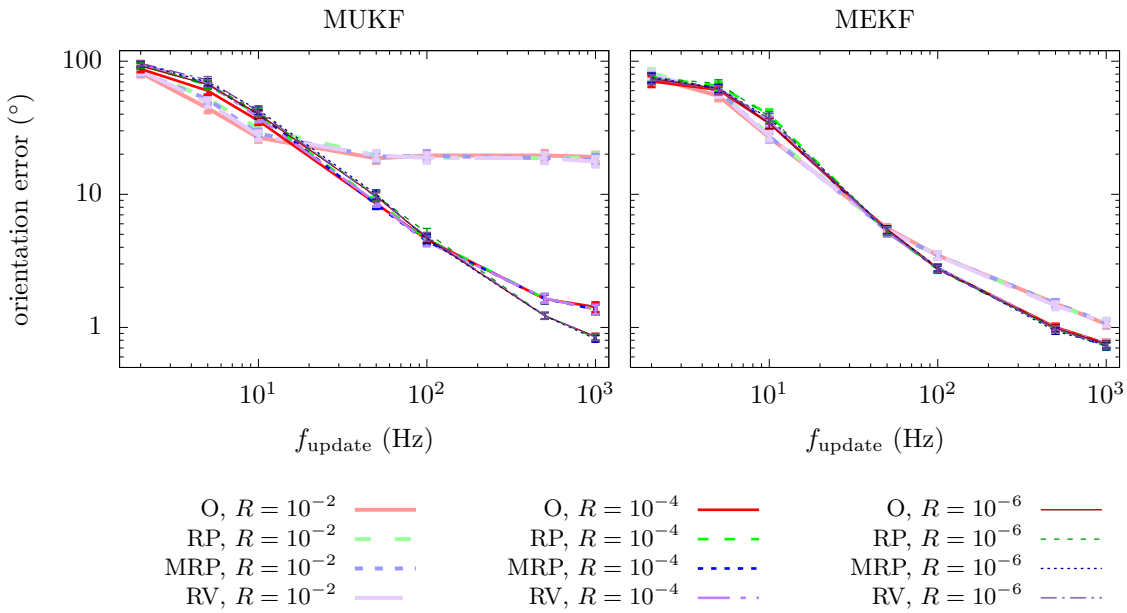


Figure 5.3: Mean of the performance metric for each approach. Results from different charts are plotted in the same graph. Results from different MKF are plotted in different graphs. Bars represent the confidence interval (3σ) for the mean computation.

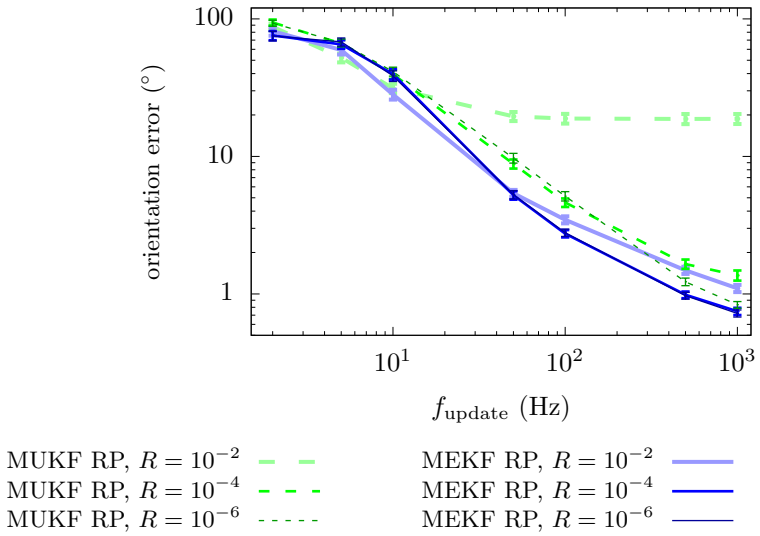


Figure 5.4: Mean of the performance metric for each MKF. Only results for the RP chart are plotted. Bars represent the confidence interval (3σ) in the mean computation.

since the Jacobian matrices, used to approximate the non-linear functions, are defined at that point. On the other hand, the sigma points generated for the MUKF are sampled far from the origin of the chart, where the non-linearities become notorious. We are facing a very particular scenario in which the model is approximately linear for the MEKF, while for the MUKF it is not. In addition, due to the difference in computational cost (see Figure 5.2), the MUKF update frequencies will generally be lower than those of the MEKF, which will imply worse accuracy in its estimations. Then, the MEKF with the RP chart seems to be our best option.

Chart update vs. no chart update

Results: Figure 5.5 presents the results of each MKF with each chart in a different graph, but displayed in the same one are the results using the "chart update" and the results without using it.

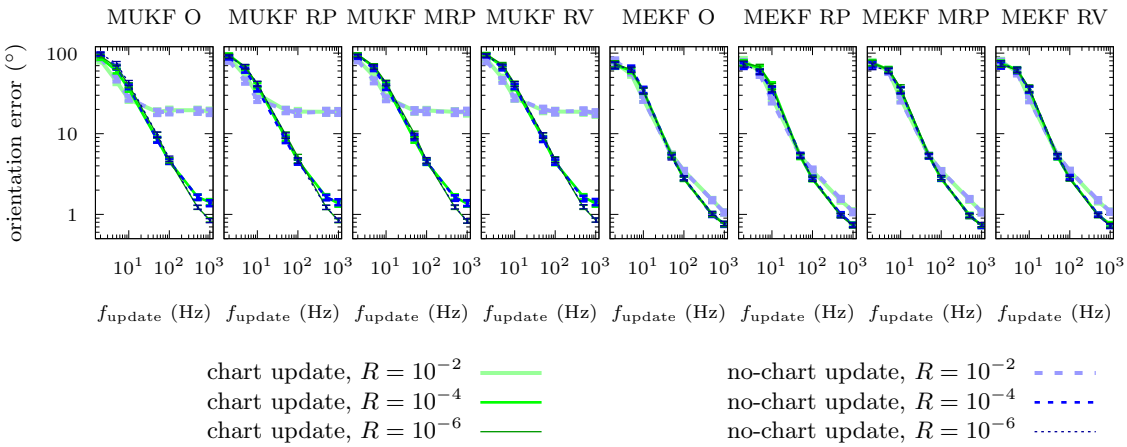


Figure 5.5: Mean of the performance metric for each approach. Results from the approach in which we apply the chart update, and those of which we do not apply it are plotted together. Bars represent the confidence interval (3σ) in the mean computation.

Discussion: We can observe that there is almost no difference between using the "chart update" and not using it. Recall that, although there are publications that presented operations similar to the "chart update", the first developers of these algorithms considered that such update was not necessary. The concepts used in this work have helped us to deeply understand the mechanisms of the MKF and, ultimately, to arrive at the concepts of "multiplicative update", and of "covariance correction step" with the $\mathbf{T}$-matrix definition: the chart update is conceptually necessary. However, in practice one can choose whether to apply the latest update (5.62) or not: we will obtain essentially the same accuracy in our estimations. This could explain why the first developers of this approach might think that the "chart update" was not necessary.

Chapter 6

Triaxial Sensor Calibration

6.1. Introduction

Until now, we have considered sensors whose only imperfection was noise in their measurements. However, that remains an idealized version of real sensors. Measurements of real sensors are deviated from the values that would be expected due to manufacturing technological limitations. For IMUs, those inaccuracies lead to a deterioration in orientation estimations, and to catastrophic results when it comes to position estimation. Normally, these problems are lessen with a calibration; but finding a calibration for sensors that produce vector measurements (triaxial sensors) is a particularly complex task.

In this chapter, we present a method to calibrate a triaxial sensor considering its dependence on temperature. We start by reviewing the sensor model in Section 6.2. This leads us to propose a hybrid model, with a linear dependence of the sensor outputs, but a non-linear dependence on the temperature. We continue, in Section 6.3, defining the calibration algorithm, which is an offline iterative method. We also comment on the prototypes that were designed to collect calibration data from real triaxial sensors in Section 6.4. The calibration data collected with the final prototype version is used to test the calibration algorithm. Finally, some results are presented in Section 6.5. Namely, we study the temperature dependence of the sensors, how calibration changes over time, and how calibration delays the deterioration of speed and position estimations.

6.2. Triaxial Sensor Model

In this section we present the model we adopt to describe the sensor measurements. A biaxial version of the sensor model is depicted in Figure 6.1.

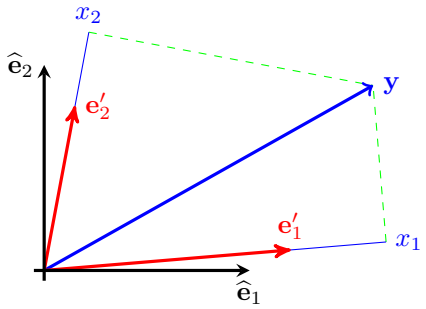


Figure 6.1: Bi-dimensional representation of the sensor model.

We are interested in measuring a vector $\mathbf{y}$ expressed with respect to an orthonormal basis $\mathcal{B} = \{\,\widehat{\mathbf{e}}_1\,,\,\widehat{\mathbf{e}}_2\,,\,\widehat{\mathbf{e}}_3\,\}$ anchored to the sensor. The triaxial sensor measures a vector $\mathbf{x}$, which is different from $\mathbf{y}$ due to miscalibration and noise in the measurements. Triaxial sensors require calibration due to the different sensitivity and bias of each independent axis, and also because of the misalignment between axes. In addition, the properties of a sensor could change with its temperature, which would generate the need for a temperature-dependent calibration.

Let $\mathcal{B}' = \{ \widehat{\mathbf{e}}'_1 , \widehat{\mathbf{e}}'_2 , \widehat{\mathbf{e}}'_3 \}$ be the vectors that define each of the sensor measurement axes expressed with respect to the basis $\mathcal{B}$. We will assume that the i-th sensor axis measures the projection of the vector $\mathbf{y}$ into the unit vector $\widehat{\mathbf{e}}'_i$ that defines such axis. These projections can be expressed in the basis $\mathcal{B}$ through $\{ \widehat{\mathbf{e}}'_i \, \widehat{\mathbf{e}}'^T_i \, \mathbf{y} \}_i$. Since the sensor will measure $\mathbf{y}$ with respect to the basis $\mathcal{B}'$, the vectors $\mathbf{x}$ and $\mathbf{y}$ will be related through

$$\mathbf{x} \quad = \quad \begin{pmatrix} — & \widehat{\mathbf{e}}'^T_1 & — \\ — & \widehat{\mathbf{e}}'^T_2 & — \\ — & \widehat{\mathbf{e}}'^T_3 & — \end{pmatrix} \mathbf{y} \quad . \tag{6.1}$$

However, triaxial sensors also have a particular sensitivity and a bias for each axis. With these considerations, the model can be restated as

$$\mathbf{x} \quad = \quad \begin{pmatrix} — & s_1 \, \widehat{\mathbf{e}}'^T_1 & — \\ — & s_2 \, \widehat{\mathbf{e}}'^T_2 & — \\ — & s_3 \, \widehat{\mathbf{e}}'^T_3 & — \end{pmatrix} \mathbf{y} \, + \, \mathbf{b} \, + \, \mathbf{r} \quad = \tag{6.2a}$$

$$= \quad \mathbf{S}\,\mathbf{y} \, + \, \mathbf{b} \, + \, \mathbf{r} \quad , \tag{6.2b}$$

where the scale factor s_i describes the sensitivity of the i-th axis[1], the vector $\mathbf{b}$ represents the measurement bias, and $\mathbf{r}$ is a random vector with null mean[2] that represents the measurement noise. Finally, assuming that the parameters of our model could depend on the temperature, T,

$$\mathbf{x} \quad = \quad \mathbf{S}(T)\,\mathbf{y} \, + \, \mathbf{b}(T) \, + \, \mathbf{r} \quad . \tag{6.3}$$

Note that our model does not expect non-linear relations between $\mathbf{x}$ and $\mathbf{y}$, variations of the model over time, or hysteresis phenomena such as the one shown in Figure 6.2. Therefore, we must ensure that, if these behaviors exist in our sensor, they are negligible in our operation domain.

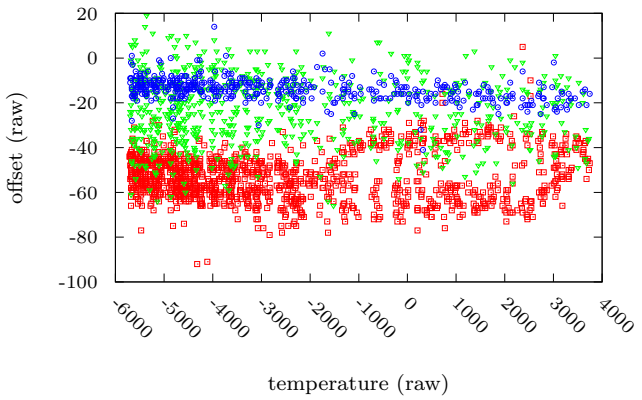


Figure 6.2: Example of a triaxial sensor showing a hysteresis phenomena. It is the case of the gyroscope contained in a MPU-6050 chip (ID 13). The hysteresis is remarkable in the measurements of the x axis, drawn with red squares. It could be due to a too fast temperature change, which could cause a difference between the temperature measured by the sensor and the temperature of the sensor core.

6.3. Triaxial Sensor Calibration

Once our sensor model is defined, we are interested in finding the parameters that characterize a particular sensor. This section presents the method used to find those parameters: the calibration algorithm.

When we evaluate the module of the measurements of an accelerometer, it is observed that it varies depending on the orientation of the sensor. This fact evidences the need for calibration

[1] Assuming that the sensitivities do not depend on the magnitude of $\mathbf{y}$.
[2] If $\mathbf{r}$ had an expected value other than $\mathbf{0}$ its contribution would be delegated to $\mathbf{b}$.

of triaxial sensors such as an accelerometer, since the module of the measured vector should be invariant under rotations. However, it also leads us to the starting point to develop the calibration algorithm. Let us consider the module of the vector $\mathbf{y}$:

$$\|\mathbf{y}\| = \| \mathbf{S}^{-1}(T) \, [\, \mathbf{x} - \mathbf{b}(T) - \mathbf{r} \,] \| = \tag{6.4a}$$

$$= \| \mathbf{K}(T)\, \mathbf{x} + \mathbf{c}(T) + \mathbf{r}' \| = \tag{6.4b}$$

$$= \| \mathbf{A}(T)\, \widetilde{\mathbf{x}} + \mathbf{r}' \| \quad , \tag{6.4c}$$

where $\mathbf{K} = \mathbf{S}^{-1}$ is the calibration matrix, $\mathbf{c} = -\mathbf{S}^{-1}\mathbf{b}$ is the calibration offset, $\mathbf{r}' = -\mathbf{S}^{-1}\mathbf{r}$ is the measurement noise expressed in the basis $\mathcal{B}$, $\mathbf{A} = (\, \mathbf{K} \,|\, \mathbf{c} \,)$ is the extended calibration matrix, and $\widetilde{\mathbf{x}} = (\, \mathbf{x} \,,\, 1 \,)^T$ is the measured vector expressed in homogeneous coordinates. The temperature dependence of $\mathbf{A}$ will be modeled as a polynomial matrix:

$$\mathbf{A}(T) = \sum_{n=0}^{N} \mathbf{A}^{(n)}\, T^n \quad . \tag{6.5}$$

The calibration parameters of a sensor should satisfy equation (6.4) for any orientation, provided that the sensor is in a static condition[3].By collecting a set of measurements taken under static conditions and for a wide range of orientations, we will obtain enough constraints to narrow the solution. Then, our calibration problem turns into a fitting problem. We will define a cost function whose minimum we want to find. The location of that minimum will be the solution to our problem: the calibration parameters.

6.3.1. Cost Function Definition

Let us consider a set of triplets

$$\left\{\, \tau_m = (\, \mathbf{x}_m \,,\, T_m \,,\, y_m \,) \,\right\}_{m=1}^{M} \tag{6.6}$$

taken in M static conditions. For each static condition, $\mathbf{x}_m$ is the measurement taken with the triaxial sensor, T_m is its temperature, and y_m is the module of the vector $\mathbf{y}_m$, that is assumed to be known. We define the cost function as

$$F(\, \mathbf{A} \,) = \sum_{m=1}^{M} \left[\, y_m^2 - \| \, \mathbf{A}(T_m)\, \widetilde{\mathbf{x}}_m \, \|^2 \,\right]^2 \quad . \tag{6.7}$$

We choose to minimize the distance between squared modules rather than the distance between modules because it greatly simplifies mathematical computations. Note that given a solution $\mathbf{A}^* = \left(\, \mathbf{K}^* \,|\, \mathbf{c}^* \,\right)$ that minimizes the cost function (6.7), then $\mathbf{B} = \mathbf{R}\,\mathbf{A}^*$, with $\mathbf{R}$ such that $\mathbf{R}^T\,\mathbf{R} = \mathbf{I}$, would also be a solution to the minimization problem[4]. The solution is not unique.

6.3.2. Model Redefinition

In order to limit the number of solutions, we can add constraints to the calibration matrix $\mathbf{K}$. These constraints will take the form of relations between the vectors of $\mathcal{B}$ and the ones of $\mathcal{B}'$. The vector $\widehat{\mathbf{e}}_1$ of the basis $\mathcal{B}$ will be chosen so that it points in the direction of $\widehat{\mathbf{e}}'_1$ of the basis $\mathcal{B}'$. The vector $\widehat{\mathbf{e}}_2$ will be chosen so that it is orthogonal to $\widehat{\mathbf{e}}_1$, and points in the direction closest to $\widehat{\mathbf{e}}'_2$. Finally, the vector $\widehat{\mathbf{e}}_3$ will be chosen so that it is orthogonal to $\widehat{\mathbf{e}}_1$ and $\widehat{\mathbf{e}}_2$, and points in the direction closest to $\widehat{\mathbf{e}}'_3$. Then, the set $\{\widehat{\mathbf{e}}'_i\}_i$ will fulfill:

$$\widehat{\mathbf{e}}_1 = \widehat{\mathbf{e}}'_1 \quad \Longrightarrow \quad \widehat{\mathbf{e}}'_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} , \tag{6.8a}$$

$$\widehat{\mathbf{e}}_2 \propto \left(\, \mathbf{I} - \widehat{\mathbf{e}}_1\, \widehat{\mathbf{e}}_1^T \,\right) \widehat{\mathbf{e}}'_2 \quad \equiv \quad \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \propto \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \widehat{\mathbf{e}}'_2 \quad \Longrightarrow \quad \widehat{\mathbf{e}}'_2 = \begin{pmatrix} * \\ \star \\ 0 \end{pmatrix} , \tag{6.8b}$$

$$\widehat{\mathbf{e}}_3 \propto \left(\, \mathbf{I} - \widehat{\mathbf{e}}_1\, \widehat{\mathbf{e}}_1^T - \widehat{\mathbf{e}}_2\, \widehat{\mathbf{e}}_2^T \,\right) \widehat{\mathbf{e}}'_3 \quad \equiv \quad \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \propto \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \widehat{\mathbf{e}}'_3 \quad \Longrightarrow \quad \widehat{\mathbf{e}}'_3 = \begin{pmatrix} * \\ * \\ \star \end{pmatrix} , \tag{6.8c}$$

[3] An accelerometer is in a static condition if it is not measuring another acceleration than gravity; a magnetometer is in a static condition if it is only measuring the reference magnetic field.

[4] Note that $\mathbf{R}$ could be a rotation matrix, but also a reflection.

where $*$ stands for *any real value*, and $\star$ means *a real value different than zero*. This redefinition will make our calibration matrix take the following form:

$$\mathbf{S} = \begin{pmatrix} — & s_1\,\widehat{\mathbf{e}}_1'^T & — \\ — & s_2\,\widehat{\mathbf{e}}_2'^T & — \\ — & s_3\,\widehat{\mathbf{e}}_3'^T & — \end{pmatrix} = \begin{pmatrix} s_1 & 0 & 0 \\ * & \star & 0 \\ * & * & \star \end{pmatrix} \quad \Longrightarrow \tag{6.9a}$$

$$\Longrightarrow \quad \mathbf{K} = \begin{pmatrix} \star & 0 & 0 \\ * & \star & 0 \\ * & * & \star \end{pmatrix} \quad , \tag{6.9b}$$

where we have recalled that the inverse of a non-singular[5] lower triangular matrix is lower triangular.

Others have modeled their sensor using an upper triangular matrix [28, 32, 33]. Choosing $\widehat{\mathbf{e}}_3 = \widehat{\mathbf{e}}_3'$ instead of (6.8a), and defining $\widehat{\mathbf{e}}_2$ and $\widehat{\mathbf{e}}_1$ through the same process followed in (6.8b) and (6.8c), one obtains an upper triangular matrix. That would be another option to make the multiple solutions converge into a single one.

Finally, we want the measurements to be expressed in a right-handed reference frame. In such a reference frame, the result of $(\widehat{\mathbf{e}}_1' \times \widehat{\mathbf{e}}_2') \cdot \widehat{\mathbf{e}}_3'$ must be positive, and this expression turns out to be proportional to the determinant of the calibration matrix:

$$\det(\mathbf{S}) \quad = \quad (\, s_1\,\widehat{\mathbf{e}}_1' \times s_2\,\widehat{\mathbf{e}}_2' \,) \cdot s_3\,\widehat{\mathbf{e}}_3' \quad . \tag{6.10}$$

If the sensitivities s_i take positive values then, knowing that $\det(\mathbf{S}) = [\det(\mathbf{S}^{-1})]^{-1} = [\det(\mathbf{K})]^{-1}$, and that for a triangular matrix $\det(\mathbf{K}) = \prod_i K_{ii}$, we can ensure a right-handed orientation by forcing positive values on the diagonal of $\mathbf{K}$.

6.3.3. Calibration Algorithm

The final triaxial sensor model needs of $9\,(N+1)$ parameters to be characterized; there are 9 parameters for each of the coefficients in (6.5):

$$\begin{aligned} \boldsymbol{\theta} \quad &= \quad \Big(K_{11}^{(0)}\,,\, K_{21}^{(0)}\,,\, K_{22}^{(0)}\,,\, K_{31}^{(0)}\,,\, K_{32}^{(0)}\,,\, K_{33}^{(0)} \;\;,\;\; c_1^{(0)}\,,\, c_2^{(0)}\,,\, c_3^{(0)}\,,\,\dots\,, \\ &\qquad K_{11}^{(N)}\,,\, K_{21}^{(N)}\,,\, K_{22}^{(N)}\,,\, K_{31}^{(N)}\,,\, K_{32}^{(N)}\,,\, K_{33}^{(N)} \;\;,\;\; c_1^{(N)}\,,\, c_2^{(N)}\,,\, c_3^{(N)} \Big)^T \;\; = \end{aligned} \tag{6.11a}$$

$$\begin{aligned} &= \quad \Big(A_{11}^{(0)}\,,\, A_{21}^{(0)}\,,\, A_{22}^{(0)}\,,\, A_{31}^{(0)}\,,\, A_{32}^{(0)}\,,\, A_{33}^{(0)} \;\;,\;\; A_{14}^{(0)}\,,\, A_{24}^{(0)}\,,\, A_{34}^{(0)}\,,\,\dots\,, \\ &\qquad A_{11}^{(N)}\,,\, A_{21}^{(N)}\,,\, A_{22}^{(N)}\,,\, A_{31}^{(N)}\,,\, A_{32}^{(N)}\,,\, A_{33}^{(N)} \;\;,\;\; A_{14}^{(N)}\,,\, A_{24}^{(N)}\,,\, A_{34}^{(N)} \Big)^T \quad . \end{aligned} \tag{6.11b}$$

We will use the Levenberg-Marquardt algorithm (see Appendix A.10) to find the minimum of (6.7), which is located at the sensor model parameters (6.11). In our particular case, $\mathbf{y}$ and $\mathbf{f}(\,\boldsymbol{\theta}^{(k)}\,)$ are $M \times 1$ matrices, $\boldsymbol{\theta}^{(k)}$ is $9\,(N+1) \times 1$, and $\mathbf{J}^{(k)}$ is $M \times 9\,(N+1)$. If we define the tensors

$$X_{ijkl}^n \quad := \quad \sum_m \widetilde{x}_{mi}\;\widetilde{x}_{mj}\;\widetilde{x}_{mk}\;\widetilde{x}_{ml}\;T_m^n \quad , \tag{6.12a}$$

$$Y_{ij}^n \quad := \quad \sum_m \widetilde{x}_{mi}\;\widetilde{x}_{mj}\;y_m^2\;T_m^n \quad . \tag{6.12b}$$

then, the matrix products $\mathbf{J}^T\,\mathbf{J}$ and $\mathbf{J}^T\,[\,\mathbf{z} - \mathbf{f}\,]$ can be computed without needing to go over all the measurements again, which speeds up the algorithm. Details on how to compute matrices $\mathbf{J}^T\,\mathbf{J}$ and $\mathbf{J}^T\,[\,\mathbf{z} - \mathbf{f}\,]$ can be found in Appendix A.11.

The squared norm of $\boldsymbol{\delta}^{(k)}$ can be defined as a measure of the convergence of the algorithm. Figure 6.3 shows an example of the convergence measure $\|\boldsymbol{\delta}^{(k)}\|^2$ as a function of the iteration number. Normally, the algorithm converges with less than 20 iterations.

[5]The matrix $\mathbf{S}$ must be non-singular; otherwise, the set $\{\widehat{\mathbf{e}}_i'\}_i$ would not be a basis, and the sensor would measure a projection onto a subspace of $\mathbb{R}^3$, instead of a 3D vector.

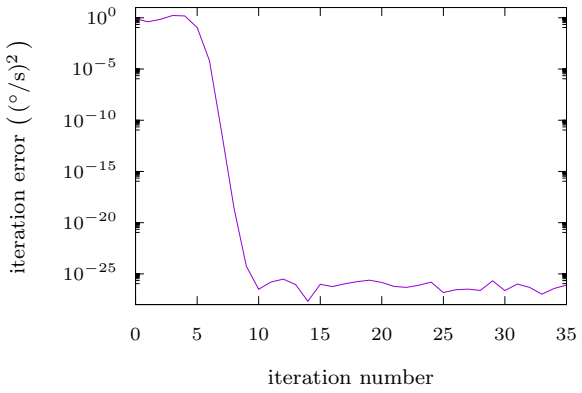


Figure 6.3: Example of the algorithm converging towards the solution that calibrates a gyroscope.

This calibration algorithm allows to perform a calibration whose dependence on temperature is modeled with a polynomial of order N, as noted in equation (6.5). However, if we choose a polynomial order that is too high, the result could be an overfitted calibration. To solve the problem of overfitting we use a strategy inspired by a widely used method for training neural networks: cross-validation. We divide the calibration data set (6.6) into two subsets:

- The calibration subset:

$$\left\{ \tau_m \ : \ T_m \in \left(T_{\text{center}} - \Delta T \, , \, T_{\text{center}} + \Delta T \right) \right\} \quad . \tag{6.13}$$

- The validation subset:

$$\left\{ \tau_m \ : \ T_m \notin \left(T_{\text{center}} - \Delta T \, , \, T_{\text{center}} + \Delta T \right) \right\} \quad . \tag{6.14}$$

with $T_{\text{center}} = \frac{T_{\max} + T_{\min}}{2}$ and $\Delta T = p \, \frac{T_{\max} - T_{\min}}{2}$, where p represents the proportion of calibration data. We use the calibration subset to perform a calibration for each polynomial order up to a maximum $N_{\max}$. Then, we use the validation subset to measure how well the model generalizes; how well the model extrapolates for temperatures not covered in the calibration subset. We will consider the *Mean Absolute Error* (MAE):

$$\text{MAE}(\, \mathbf{A} \,) \ = \ \frac{1}{M} \sum_{m=1}^{M} \left| \, y_m \ - \ \| \, \mathbf{A}(T_m) \, \widetilde{\mathbf{x}}_m \, \| \, \right| \quad , \tag{6.15}$$

as the measure of the performance of a calibration $\mathbf{A}$.

Figure 6.4 shows an example of the calibration and validation errors obtained from several calibrations for a specific sensor using a proportion of calibration data of $p = 0.75$, and $N_{\max} = 5$.

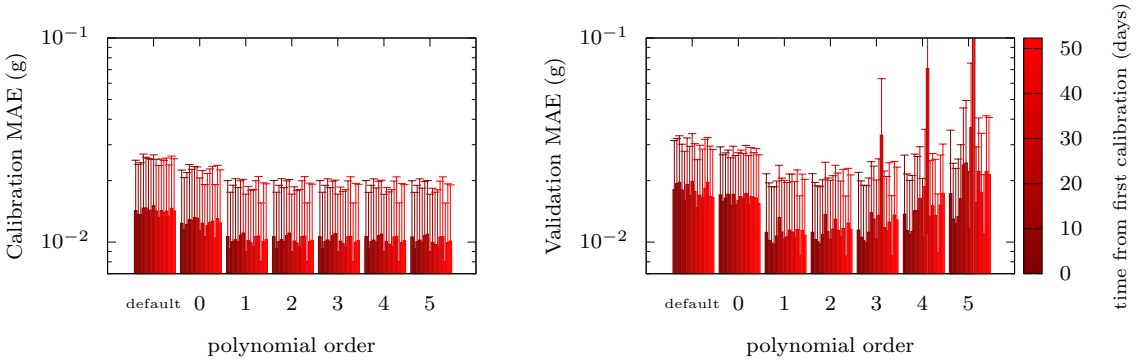


Figure 6.4: Calibration and validation errors obtained from several calibrations of an accelerometer. The errors produced with the default calibration (the one presented in the data sheet) are also shown. Bars represent the 1-σ bounds in the computation of the MAE. It is the case of the MPU-6050 contained in the board GY-88 (ID 11).

Similar figures have been included in Appendix B.2 for each triaxial sensor. Note that although the temperature-independent calibration (the one of order 0) produces less error than the default calibration (the one presented in the data sheet), the temperature-dependent calibration is the one that drastically reduces the error (in this particular case, the ones of order 1 or 2). In fact, if we plot the temperature along with the error in each measurement (each one of the terms in the sum (6.15)) as in Figure 6.5, we can appreciate that the temperature plays an important role in the measurement error.

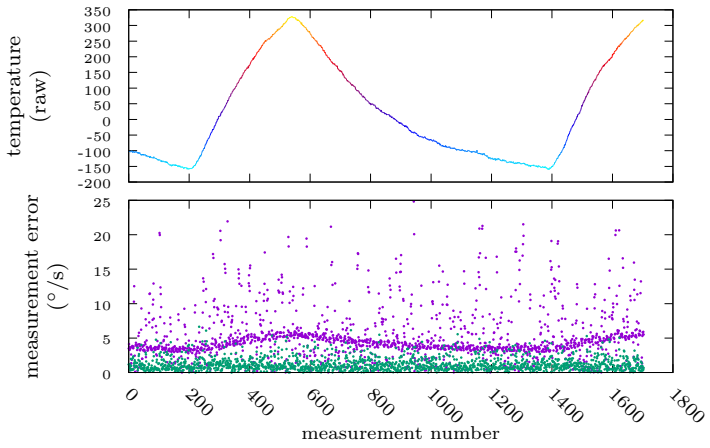


Figure 6.5: Temperature (top) and measurement error (bottom) for each measurement of a calibration set. Errors produced by the default calibration are drawn in purple. Errors produced by the optimal calibration are drawn in green. These particular measurements have been produced by the gyroscope contained in the Sense HAT.

The source code of the algorithms used to calibrate triaxial sensors can be found in GitHub[6].

6.4. Data Collection System (DCS)

We wanted to calibrate multiple real sensors, several times each. Then, collecting real calibration data could be a repetitive task. We decided to design a system capable of automatically obtaining such calibration data to ease the process. This section describes the different versions of the system that were developed.

We needed the system to perform several tasks:

- place the sensors in various orientations with respect to the external reference vectors (gravity, angular velocity, magnetic field, ...),
- rotate the sensors with a chosen angular velocity,
- modify the temperature of the sensors.

Note that it was not necessary to know the orientation of the sensors since our algorithm exploits the information provided by the module of the measured vector.

We developed a first version of the system that was able to perform these tasks, and collect calibration data for a single sensor. Later, we developed a second version of the system that was able to collect calibration data for three sensors; two mounted on a platform and the third being the Sense HAT, which was mounted on a Raspberry Pi.

6.4.1. Data Collection System v1

The first DCS, shown in Figure 6.6, consisted of 2 nodes. The first node was placed inside a temperature insulating chamber, and was composed of an Arduino Mini connected to a triaxial sensor and a radio transmitter (RF 433MHz), all powered by batteries. The orientation of the

[6]Triaxial calibration algorithm: `https://github.com/PBernalPolo/triaxialSensorCalibration2019/tree/master/temperatureCalibrationAlgorithm` [Online; accessed January-2020]

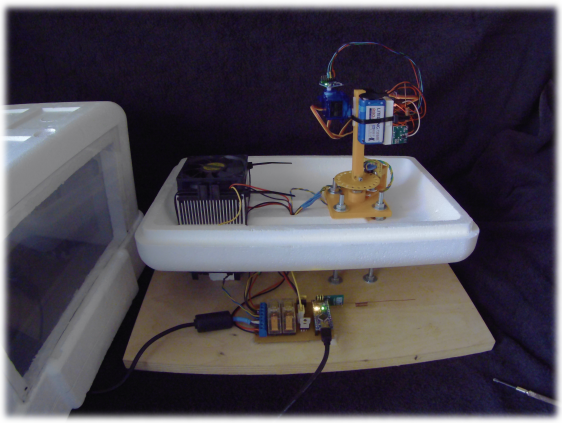
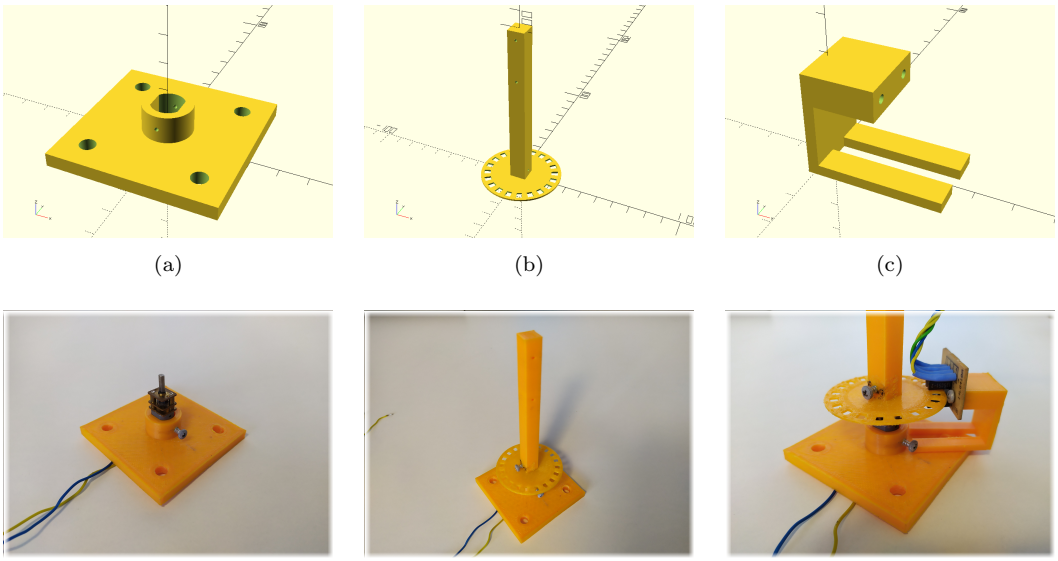


Figure 6.6: First prototype of the Data Collection System.

triaxial sensor was established by two servos (SG90). The Arduino Mini controlled the servos, and requested measurements by I2C to send them through the RF transmitter. Several parts were designed[7] and printed[8] to mount this node, and allow it to rotate with the chosen angular velocity (see Figure 6.7). The angular velocity of the node was measured using a tachometer that was mounted on the piece shown in Figure 6.7c, and that counted the holes of the piece shown in Figure 6.7b. The node was rotated using an electric motor that was mounted on the piece shown in Figure 6.7a, and which was controlled by the second node.



(a) (b) (c)

(d) (e) (f)

Figure 6.7: Parts designed for the first version of the DCS. **(a)** Design of the base used to hold the electric motor. **(b)** Design of the pole used to place the first node. **(c)** Design of the tachometer support. **(d)** 3D-printed base with the electric motor attached to it. **(e)** 3D-printed pole mounted on top of the electric motor. **(f)** 3D-printed tachometer support positioned to detect the holes of the pole disc.

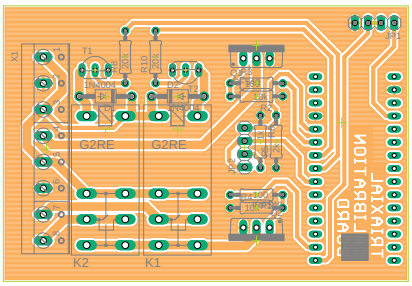
The second node was composed of an Arduino Nano and a radio receiver (RF 433MHz) both connected to a custom design board[9] (TCB from Triaxial Calibration Board. See Figure 6.8. Appendix A.12.1 contains the schematics and the real-size board design). It was in charge of

[7]The parts were designed using OpenSCAD: `https://www.openscad.org/` [Online; accessed January-2020]

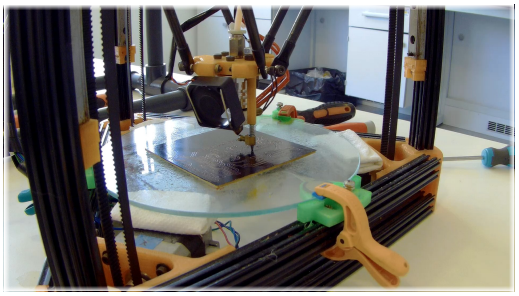
[8]The parts were printed using a delta 3D printer named Kossel: `https://reprap.org/wiki/Kossel` [Online; accessed January-2020]

[9]The board was designed using EAGLE PCB design software: `https://www.autodesk.com/products/eagle/overview` [Online; accessed January-2020]. It was made by isolating routing using a combination of 3D printer, permanent marker, and etching: `http://www.lamja.com/?p=635` [Online; accessed January-2020]

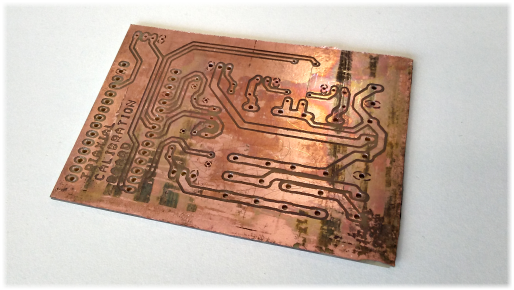
controlling the angular velocity of the electric motor, and the temperature of the chamber using a Peltier device. It also received the measurements sent through the radio transmitter, and forwarded them through the serial port to the computer where they were stored. The computer was running a program that managed the data collection procedure by setting the angular velocity and the target temperature.



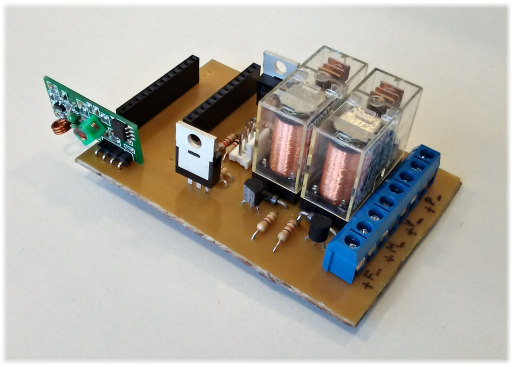
(a)



(b)



(c)



(d)

Figure 6.8: TCB built for the first version of the DCS. It was make by isolating routing with a 3D printer, permanent marker, and etching. **(a)** TCB1 Eagle design. **(b)** TCB1 during marker scraping. **(c)** TCB1 after etching and drilling. **(d)** Finished TCB1 with welded components.

It was necessary to develop some software, among which were the communication protocols for the radio frequency transmission, the measurement of the angular velocity of the motor, and the control algorithms for such an angular velocity and for the temperature of the chamber[10].

6.4.2. Data Collection System v2

The second DCS, shown in Figure 6.9, was developed to ease the data collection process; a platform was built to obtain calibration data from several sensors.

This second DCS was also composed of 2 nodes. As in the previous DCS, the first node was placed inside the temperature insulating chamber. However, the structure of this node was more complex. An Arduino Nano was connected to each sensor. Each Arduino Nano requested measurements by I2C and, once received, these were sent to a Raspberry Pi 3 (RPi) through the serial port. The RPi stored the measurements in a file for later calibration. The sensors, the Arduinos Nano and the RPi were fixed to the platform. The platform was attached to a device similar to a multi-axis trainer that will be called "gimbal" (see Figure 6.10b). The orientation of the platform was established by two servos (SG90) attached to the gimbal, and controlled directly by the RPi through the GPIO pins. The angular velocity of the platform was set by a stepper motor (17HS4401; driver: Keyes L298N) on which the gimbal was mounted. With the stepper motor, the need for a tachometer and the noise in its angular velocity measurements were eliminated with respect to the first version.

[10] At this stage of the project, we had not noticed that the temperature-dependent calibration could be obtained directly from the calibration algorithm, so the strategy was to perform several calibrations each for a given temperature, and then find a polynomial fit for each parameter.

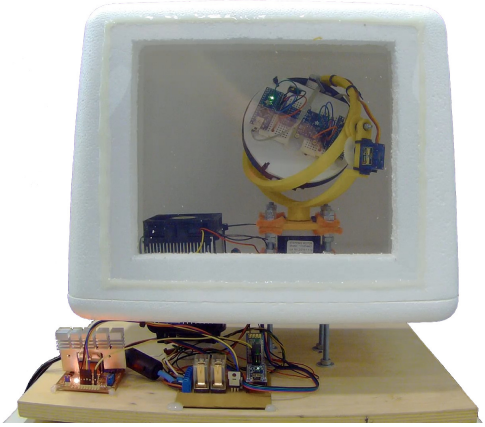
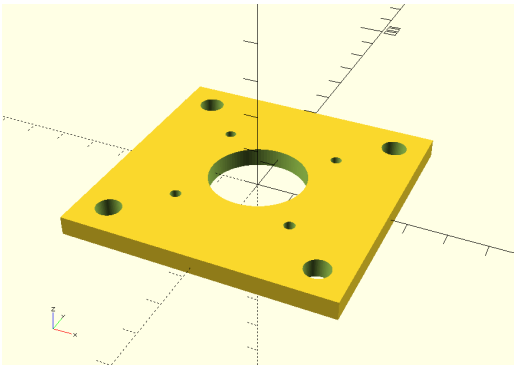
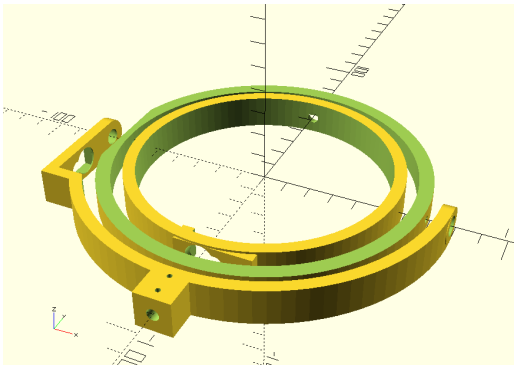


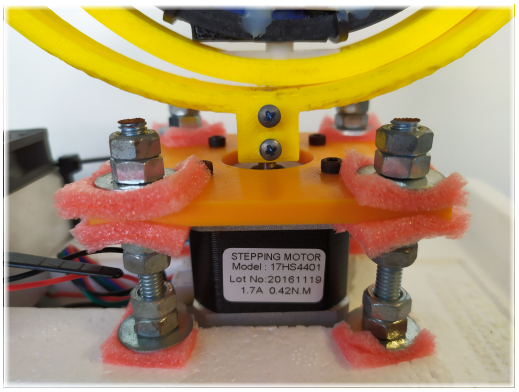
Figure 6.9: Second prototype of the Data Collection System.



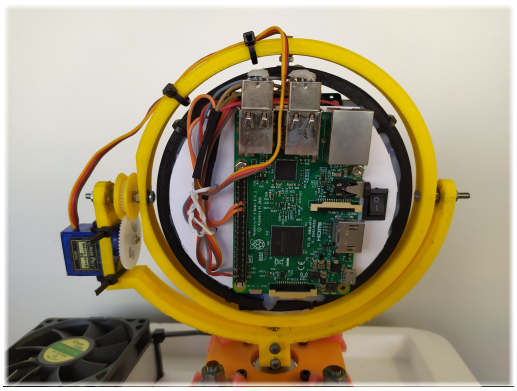
(a)



(b)



(c)

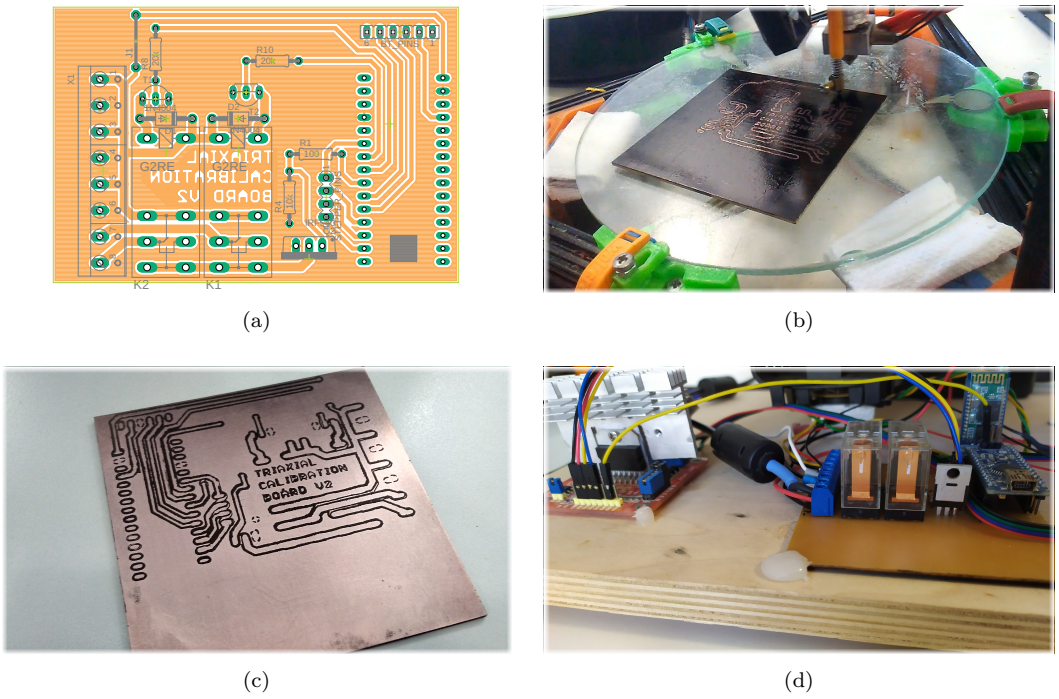


(d)

Figure 6.10: Parts designed for the second version of the DCS. **(a)** Design of the base used to place the stepper motor. **(b)** Design of the gimbal used to place the platform. **(c)** 3D-printed base with stepper motor attached to it. **(d)** 3D-printed gimbal mounted on top of the stepper motor.

The second node was composed of an Arduino Nano, a bluetooth module (HC-05), and the stepper motor driver (Keyes L298N), all connected to a custom design board (see Figure 6.11. Appendix A.12.2 contains the schematics and the real-size board design). The same Peltier device as in the previous DCS was used to modify the temperature of the chamber. Both the stepper motor and the Peltier device were controlled by the Arduino Nano. This Arduino Nano established the angular velocity and the temperature according to requests made by the RPi through the bluetooth device.

It was necessary to develop new software, among which was the stepper motor control (mi-



(a) (b)

(c) (d)

Figure 6.11: TCB built for the second version of the DCS. It was make by isolating routing with a 3D printer, permanent marker, and etching. **(a)** TCB2 Eagle design. **(b)** TCB2 during marker scraping. **(c)** TCB2 after etching. **(d)** Finished TCB2 with welded components and connected to the stepper motor driver.

crostepping). The source code generated to control the TCS can be found in GitHub[11]. The OpenSCAD designs for both versions of the DCS can be found in Thingiverse[12].

6.5. Experimental Results

In this section we present the results extracted from multiple triaxial sensor calibrations. Figure 6.12 shows and Table 6.1 summarizes the 7 boards, which contain triaxial sensors, calibrated during this research. Also, Appendix A.13 gathers some of the characteristics presented in the datasheets of the sensors used during the development of this thesis.

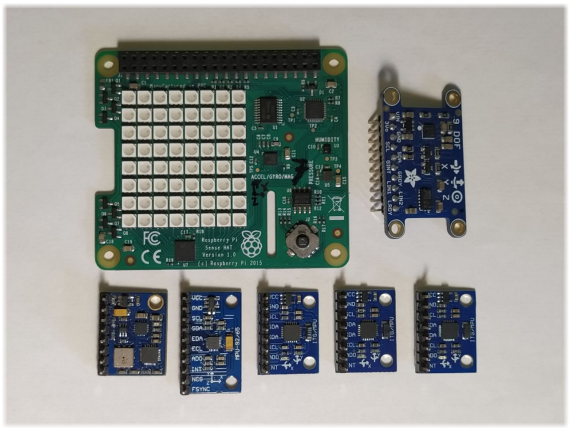


Figure 6.12: Triaxial sensors calibrated in this research.

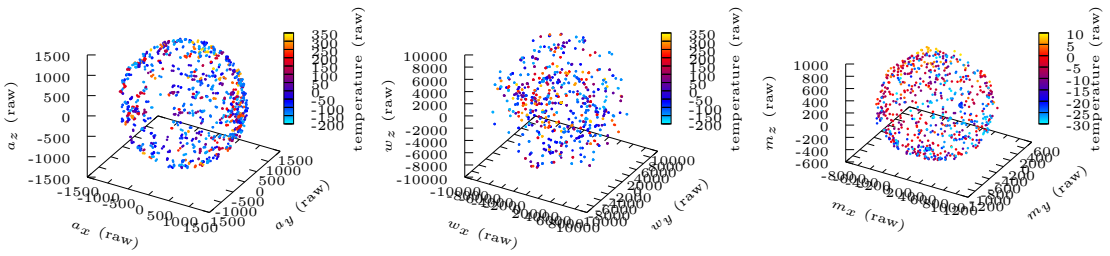
[11]Triaxial calibration system code: `https://github.com/PBernalPolo/triaxialSensorCalibration2019/tree/master/calibrationSystem` [Online; accessed January-2020]

[12]OpenSCAD designs: `https://www.thingiverse.com/thing:4063556` [Online; accessed January-2020]

Table 6.1: Triaxial sensors calibrated during this research. The number at the right of each triaxial sensor is the optimal polynomial order of the overall calibration.

ID	Board	Accelerometer	Gyroscope	Magnetometer
11	GY-88	MPU-6050 (3)	MPU-6050 (2)	HMC5883L (1)
12	GY-9250	MPU-9250 (3)	MPU-9250 (1)	-
13	GY-521	MPU-6050 (1)	MPU-6050 (1)	-
14	GY-521	MPU-6050 (2)	MPU-6050 (2)	-
15	GY-521	MPU-6050 (1)	MPU-6050 (2)	-
16	Adafruit 9-DOF	LSM303DLHC (2)	L3GD20H (4)	LSM303DLHC (0)
17	Sense HAT	LSM9DS1 (1)	LSM9DS1 (3)	LSM9DS1 (0)

We used the system presented in Section 6.4 to collect several sets of measurements of accelerometers and gyroscopes for more than a month varying the temperature from about 15°C to about 45°C. To reduce the disturbances, the data was taken in the afternoon, since it was a period of less hustle in the building and its surroundings. Each data collection lasted approximately 1 hour. Magnetometer data were collected by hand because some pieces of iron in the TCS disturbed the homogeneity of the magnetic field. Since the HMC5883L chip does not contain a temperature sensor, the temperature readings of the MPU-6050 chip, mounted on the same board, were used. Figure 6.13 shows examples of the data sets that were collected. Several more examples of calibration data are contained in Appendix B. Each set of calibration data was used to perform a calibration by applying the algorithm described in Section 6.3.



(a) (b) (c)

Figure 6.13: Examples of sets of measurements produced by a triaxial sensor. They are used to calibrate the triaxial sensors contained in the Sense HAT board. **(a)** Accelerometer measurements. **(b)** Gyroscope measurements. **(c)** Magnetometer measurements.

6.5.1. Calibration Polynomial Orders

Results: Comparing the MAE produced by multiple calibrations, we can obtain information about the trend of the optimal calibration order. Figure 6.14 shows the histogram of optimal calibration orders for each of the triaxial sensors arranged in Table 6.1.

Discussion: We observe that the optimal order of the polynomial is 1 for most calibrations, and that it is rare to find an optimal polynomial order greater than 2.

6.5.2. Calibration over Time

Results: As the measurements were taken for more than a month, we can also obtain information on how the calibration changes over time. Figure 6.15 shows examples of the calibration parameters obtained at different moments in time. The calibrations are evaluated at the same temperature: the center of the calibration interval. The calibration parameters are grouped in the graphs considering the similarity of the values that were expected. In particular, K_{11}, K_{22}, and K_{33} are called *sensitivities*, and are represented with (□), (▽), and (○) respectively. K_{21}, K_{31},

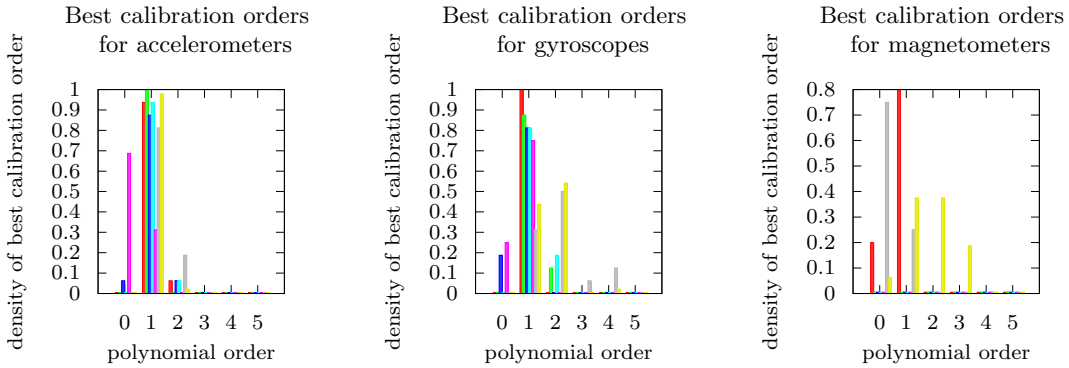


Figure 6.14: Density of best calibration orders computed from multiple calibrations of different triaxial sensors. Red is sensor 11. Blue is sensor 12. Green is sensor 13. Cyan is sensor 14. Magenta is sensor 15. Gray is sensor 16. Yellow is sensor 17.

and K_{32} are called *misalignments*, and are represented with $(+)$, $(\times)$, and $(*)$ respectively. c_1, c_2, and c_3 are called *offsets*, and are represented with $(\square)$, $(\triangledown)$, and $(\bigcirc)$ respectively. Similar figures for each triaxial sensor have been included in Appendix B.3.

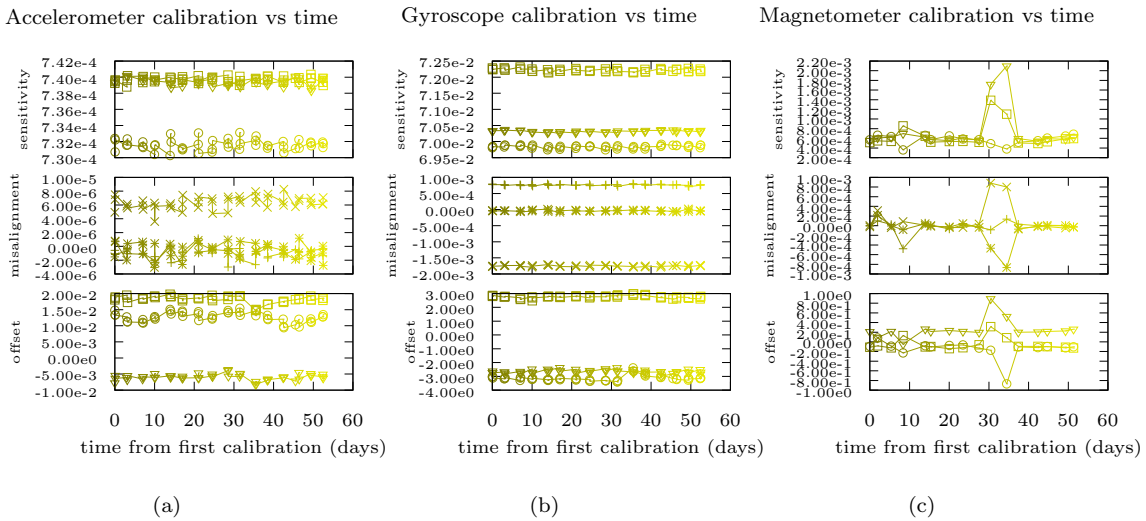


Figure 6.15: Examples of calibrations for a triaxial sensor ordered over time. They are the case of the triaxial sensors contained in the Sense HAT board. **(a)** Accelerometer calibrations (g). **(b)** Gyroscope calibrations (deg/s). **(c)** Magnetometer calibrations (G).

Discussion: We note that the calibration parameters are scattered around certain values (except in some calibrations probably computed with faulty data), and that those values do not seem to change over time. That suggests that if we are able to generate a good calibration, it will be valid time after the moment it was generated. It is worth mentioning that after the first month, and before each calibration, the triaxial sensors were subjected to extreme accelerations (hitting them against the table), and that did not change the trend of the results. This contrasts with the common belief that calibrations (in particular, offsets) change over time[13], or between turn on/off of the devices.

[13] At least not in short periods of time of the order of one month.

6.5.3. Dead Reckoning with Calibrated IMUs

After observing that the calibrations do not seem to change over time, all the data sets were used to generate a single calibration for each sensor. At the right of the name of each triaxial sensor in Table 6.1 we write the optimal polynomial order obtained with these calibrations.

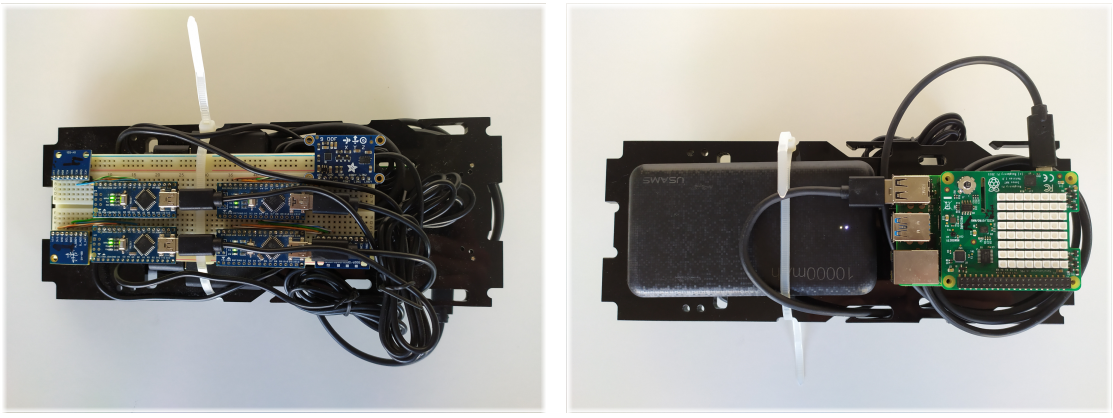
Results: In order to experimentally test the calibrations obtained, the triaxial sensors were used in a specific application: the estimation of their position. The sensors were mounted on a common platform (see Figure 6.16), and we performed several similar trajectories. These trajectories consisted of 4 approximate stages: 2 static seconds, 5 seconds of rotation without translation, 1 second of translation and return to the initial position, and 2 static seconds. We used the orientation estimation algorithm derived in Chapter 5 (MEKF with RP chart) to estimate the orientation $\boldsymbol{q}^{GS}$ that relates the expressions of a vector in two different reference frames:

$$ {}^{G}\mathbf{u} \quad = \quad \mathbf{R}(\,\boldsymbol{q}^{GS}\,)\;{}^{S}\mathbf{u} \quad , \tag{6.16} $$

being ${}^{S}\mathbf{u}$ the vector expressed in the sensor reference frame (the one expressed with respect to the basis $\mathcal{B}$ in this chapter), and ${}^{G}\mathbf{u}$ the same vector expressed in a reference frame $\mathcal{G}$ whose z-axis points in the same direction than gravity. We used the orientation $\boldsymbol{q}^{GS}$ to obtain the coordinate acceleration ${}^{G}\mathbf{a}$ expressed in the quasi-inertial reference frame $\mathcal{G}$ (see equation (5.4)):

$$ {}^{G}\mathbf{a} \quad = \quad \mathbf{R}(\,\boldsymbol{q}^{GS}\,)\;{}^{S}\mathbf{a}^{m} \;-\; {}^{G}\mathbf{g} \quad , \tag{6.17} $$

where ${}^{S}\mathbf{a}^{m}$ is the accelerometer measurement, and ${}^{G}\mathbf{g}$ is the gravitational acceleration expressed in $\mathcal{G}$. Then, we integrated the coordinate acceleration ${}^{G}\mathbf{a}$ to obtain the sensor velocity ${}^{G}\mathbf{v}$, and we integrated the sensor velocity to obtain the sensor position ${}^{G}\mathbf{x}$. Since the platform was static most of the time, we would expect the speed ${}^{G}v = \|{}^{G}\mathbf{v}\|$ and the distance to the origin ${}^{G}d = \|{}^{G}\mathbf{x}\|$ to jiggle around zero. However, due to inaccuracies in the calibration or the orientation estimation, or simply due to the integration of the measurement noise[14], the speed and the distance to the origin will drift away from the origin. The better the calibration and the orientation estimation, the less they will contribute to the speed and distance to the origin moving away from zero. The source code used for this test can be found in GitHub[15].



(a) (b)

Figure 6.16: Platform where the triaxial sensors were mounted. **(a)** Triaxial sensors side. **(b)** Raspberry Pi side.

Figure 6.17 shows the trajectories obtained for the speed ${}^{G}v$ of each sensor and their distance to the origin ${}^{G}d$ with the default calibrations, and with the calibrations obtained in this work. Every 10 seconds ${}^{G}v$ and ${}^{G}d$ are reset to zero.

Discussion: We observe that by using the calibrations obtained in this work the drift is significantly reduced, although not eliminated.

[14] Wiener process, or Brownian motion.

[15] Dead reckoning test code: `https://github.com/PBernalPolo/triaxialSensorCalibration2019/tree/master/calibrationTestD` [Online; accessed January-2020]

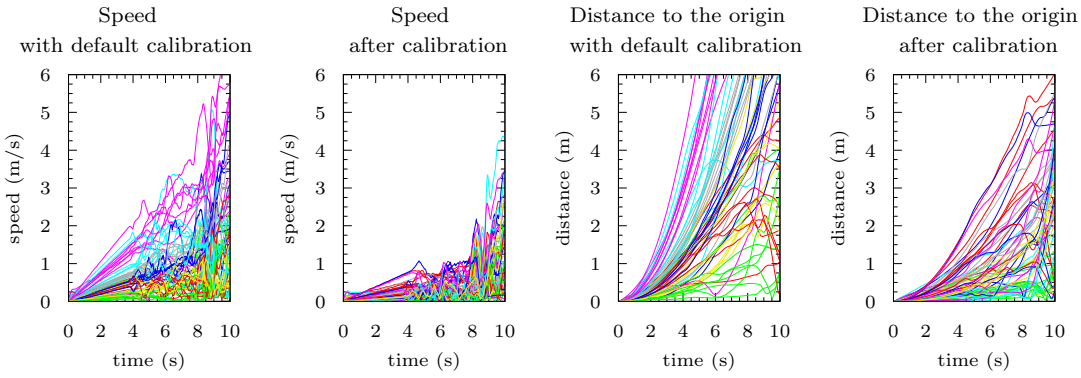


Figure 6.17: Speed and distance to the origin estimated using the triaxial sensors. We display the estimations obtained using their default calibration, and the estimations obtained with the optimal calibration. The orientation of the sensors changes, but the position remains approximately static.

Chapter 7

Conclusion

In this chapter we will summarize the content of this thesis highlighting the conclusions obtained. We will also propose possible lines of future research.

7.1. Conclusions

This thesis approaches the problem of estimating the orientation of a system using IMU measurements. In particular, two problems arising from the main one are addressed:

- the estimation of the orientation of a system using the Kalman filter and quaternions as orientation descriptors,
- and the calibration of IMUs.

The conclusions drawn for each of the aforementioned problems are summarized below.

Orientation estimation
We chose the Kalman filter formalism as the basis for the estimation algorithms. The orientation was represented using unit quaternions.

After reviewing the principles of Kalman filtering, and the basics of quaternions, we used concepts from manifold theory to define the expected value, and the covariance matrix of a distribution of unit quaternions in the unit sphere of $\mathbb{R}^4$, S^3, using the concept of chart. Those definitions can generally be applied to define the moments of distributions in other manifolds. In the problem at hand, these definitions provided natural means to develop orientation estimation algorithms based on the Kalman filter. Specifically, two estimators were developed: one based on the EKF (the MEKF), and another based on the UKF (the MUKF). The definitions also led us to a solution for the "covariance correction step". The case of the MEKF is particularly interesting, since we obtained the chart update matrices ($\mathbf{T}$-matrices), which were either previously unknown or only approximate expressions had been obtained [41, 27].

From all the possible charts that could be used to define the distribution of unit quaternions describing the orientation of the system, four were chosen to test the algorithms. Each combination of MEKF or MUKF with each of the four charts was tested in simulations. The following conclusions were drawn from the results:

- There is no chart that presents a clear advantage over the others, but the one called "RP chart" has some properties that motivate us to prefer it.
- The MEKF is preferable to the MUKF due to its lower computational cost and its greater accuracy in orientation estimation.
- The "covariance correction step" is not necessary for the MKF in practice. Although conceptually necessary, algorithms produce the same performance whether applied or not.

Taking these conclusions into account, we could select the MEKF with the RP chart and without applying the "covariance correction step" as our best orientation estimator according to the metric adopted to test the algorithms in the simulations. This algorithm resembles the conventional "Multiplicative Extended Kalman Filter", but we developed the MEKF naturally; without having to redefine any aspect of the classic Kalman filter.

Calibration of IMUs
To calibrate IMUs, we chose to design an iterative algorithm that would run offline. The fact that the magnitude of the vector measured by a triaxial sensor must be independent of its orientation was exploited. The calibration resulting from the algorithm takes into account the temperature of the sensor to correct its measurements.

We began by defining the sensor model as a linear combination of the measurements. However, the coefficients in the linear combination were defined as a polynomial function of temperature. The calibration problem was transformed into an optimization problem. A cost function was defined whose minimum would locate the optimal calibration. The Levenberg-Marquardt algorithm was used to find the position of the minimum of the cost function. We defined some tensors to increase the computational efficiency of the algorithm. Cross-validation was also used to obtain the most appropriate polynomial degree of temperature dependence for calibration. The resulting algorithm is able to compute temperature-dependent calibrations efficiently.

An electro-mechanical system was designed and built to collect calibration data from real IMUs. The system was used to collect sets of calibration data for more than a month. The datasets were introduced in the designed algorithm to obtain the associated calibrations. We used the calibrations to study the temperature dependence of the calibrated IMUs. The results showed that the temperature dependence of the sensors can almost always be described with a low order polynomial (from 0 to 3) for the temperature range in which the sensors were calibrated. We also studied the dependence of the calibrations over time. We concluded that the calibrations do not change appreciably over time, which contrasts with the common belief that they do. Finally, a dead-reckoning experiment was performed: we estimated the position of each IMU integrating the measured acceleration using its default calibration, and its optimal calibration. The results showed that the estimations improved with the optimal calibrations.

In a nutshell, this work presents a solid foundation for the development of algorithms used to estimate the dynamic state of a system in which orientation is considered part of the state. It also satisfactorily addresses the problem of temperature-dependent triaxial sensor calibration.

7.2. Future Work

This section proposes possible lines of future research related to the content of this thesis.

Fusion of information provided by several IMUs
In Chapter 4 we obtained expressions for the measurements of accelerometers and gyroscopes, which depended on their position and orientation within the system. Then, in Chapter 5, the models were simplified to focus on the development of the algorithms. However, now that we understand how to apply the Kalman filter to estimate orientations, we could include the information about the position and orientation of each sensor to merge the information from multiple IMUs. This seems to be a natural direction to expand this work.

Fusion of information provided by several sensors
IMUs are not the only sensors capable of providing information about the dynamic state of a system. Other types of sensors can also be included. For example, a GPS measures its position, a pressure sensor provides information about the height at which a rigid body is when it is surrounded by a fluid immerse in a gravitational field, and a camera provides information on both the orientation and the position of a rigid body. Each sensor capable of providing measurements related to the state of the rigid body is an opportunity to extend this work.

Fusion of information provided by external sensors
The information on the dynamic state does not need to be obtained exclusively by sensors that are mounted on the rigid body: it could be provided by external sensors. External cameras, or wireless beacons that provide distance measurements are examples of sensors that need external equipment, and are also suitable for supplying information on the dynamic state of the rigid body.

Fusion of information provided by other rigid bodies
Two rigid bodies could have sensors that impose constraints between their states. For example,

they could have wireless beacons that would impose distance constraints between them. They could also have cameras that obtained measurements of relative position and orientation between rigid bodies. Another example would be that of two rigid bodies that get attached together. In such a case, a constraint would be created between the position and orientation of the two bodies.

Orientation control
Many control algorithms use the feedback loop structure. An error is defined, and the control outputs are modulated in order to minimize that error. To control the orientation of a rigid body, one could define the error between the reference orientation and the estimated orientation in a chart: in the same space used in this work to define the distribution of unit quaternions.

Appendix A

Appendices

A.1. Cholesky Decomposition

Every positive-definite matrix $\mathbf{A}$ can be decomposed as the product of a lower triangular matrix $\mathbf{L}$, and its transposed $\mathbf{L}^T$:

$$\mathbf{A} \quad = \quad \mathbf{L}\,\mathbf{L}^T \quad . \tag{A.1}$$

This decomposition is called *Cholesky decomposition* in honor of its discoverer. For each positive-definite matrix there is a unique Cholesky decomposition.

A.1.1. Computation

Let us take for example a 4×4 matrix:

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{pmatrix} = \tag{A.2a}$$

$$= \begin{pmatrix} L_{11} & 0 & 0 & 0 \\ L_{21} & L_{22} & 0 & 0 \\ L_{31} & L_{32} & L_{33} & 0 \\ L_{41} & L_{42} & L_{43} & L_{44} \end{pmatrix} \begin{pmatrix} L_{11} & L_{21} & L_{31} & L_{41} \\ 0 & L_{22} & L_{32} & L_{42} \\ 0 & 0 & L_{33} & L_{43} \\ 0 & 0 & 0 & L_{44} \end{pmatrix} \quad = \tag{A.2b}$$

$$= \begin{pmatrix} L_{11}^2 & L_{11}\ L_{21} & L_{11}\ L_{31} & L_{11}\ L_{41} \\ L_{21}\ L_{11} & L_{21}^2+L_{22}^2 & L_{21}\ L_{31}+L_{22}\ L_{32} & L_{21}\ L_{41}+L_{22}\ L_{42} \\ L_{31}\ L_{11} & L_{31}\ L_{21}+L_{32}\ L_{22} & L_{31}^2+L_{32}^2+L_{33}^2 & L_{31}\ L_{41}+L_{32}\ L_{42}+L_{33}\ L_{43} \\ L_{41}\ L_{11} & L_{41}\ L_{21}+L_{42}\ L_{22} & L_{41}\ L_{31}+L_{42}\ L_{32}+L_{43}\ L_{33} & L_{41}^2+L_{42}^2+L_{43}^2+L_{44}^2 \end{pmatrix} . \tag{A.2c}$$

From (A.2) we deduce:

$$L_{jj} \quad = \quad \sqrt{A_{jj} \ - \ \sum_{k=1}^{j-1} L_{jk}^2} \quad , \tag{A.3}$$

$$L_{ij} \quad = \quad \left(A_{ij} \ - \ \sum_{k=1}^{j-1} L_{ik}\ L_{jk} \right) \frac{1}{L_{jj}} \qquad \text{for} \quad i > j \quad . \tag{A.4}$$

A.1.2. Algorithm

The algorithm can be applied in-place. The input is the $N \times N$ matrix $\mathbf{A}$. The output is the matrix $\mathbf{L}$ stored in the $\mathbf{A}$ space.

```
1  void Cholesky( Matrix A ) {
2    for each column j=1,...,N
3      sumD = 0
4      // first, we fill with 0 until diagonal
5      for each row i = 1, ... , j-1
6        A(i,j) = 0
7        // we compute the sum in (A.4) at the same time
8        sumD = sumD - A(j,i) * A(j,i)
9      end for each row i
10     // we compute (A.3)
11     sumD = sumD + A(j,j)
12     A(j,j) = sqrt( sumD )
13     // we compute the terms below the diagonal
14     for each row i = j+1, ... , N
15       sumL = 0
16       for each column k = 1, ... , j-1
17         sumL = sumL - A(j,k) * A(i,k)
18       end for each column k
19       sumL = sumL + A(i,j)
20       A(i,j) = sum / A(j,j)
21     end for each row i
22   end for each column j
23 end Cholesky
```

A.2. Solving a Matrix Equation Involving a Positive-Definite Matrix

We consider a matrix equation of the form

$$\mathbf{X}\,\mathbf{A} \quad = \quad \mathbf{B} \quad , \tag{A.5}$$

where $\mathbf{A}$ is a positive-definite matrix, $\mathbf{B}$ is a known $M \times N$ matrix, and $\mathbf{X}$ is an unknown $M \times N$ matrix.

Equation (A.5) can be decomposed into M matrix equations:

$$\mathbf{x}_i\,\mathbf{A} \quad = \quad \mathbf{b}_i \quad , \tag{A.6}$$

being $\mathbf{x}_i$ the i-th row of $\mathbf{X}$, and $\mathbf{b}_i$ the i-th row of $\mathbf{B}$.

Since $\mathbf{A}$ is positive-definite, it can be Cholesky-decomposed as it is shown in Appendix A.1, and equations (A.6) can be rewritten as

$$\mathbf{x}_i\,\mathbf{L}\,\mathbf{L}^T \quad = \quad \mathbf{b}_i \quad . \tag{A.7}$$

If we define $\mathbf{y}_i = \mathbf{x}_i\,\mathbf{L}$, equations (A.7) can be solved in two steps:

$$\mathbf{y}_i\,\mathbf{L}^T \quad = \quad \mathbf{b}_i \quad , \tag{A.8a}$$

$$\mathbf{x}_i\,\mathbf{L} \quad = \quad \mathbf{y}_i \quad . \tag{A.8b}$$

A.2.1. Computation

Let us take for example a 4×4 matrix:

First step:

$$\begin{pmatrix} y_{i1} & y_{i2} & y_{i3} & y_{i4} \end{pmatrix} \begin{pmatrix} L_{11} & L_{21} & L_{31} & L_{41} \\ 0 & L_{22} & L_{32} & L_{42} \\ 0 & 0 & L_{33} & L_{43} \\ 0 & 0 & 0 & L_{44} \end{pmatrix} \quad = \quad \begin{pmatrix} b_{i1} & b_{i2} & b_{i3} & b_{i4} \end{pmatrix} \quad , \tag{A.9}$$

that makes us realize

$$y_{ij} \quad = \quad \left(b_{ij} \; - \sum_{k=1}^{j-1} y_{ik}\; L_{jk} \right) \frac{1}{L_{jj}} \qquad \text{with} \quad j = 1\,,\,2\,,\,\ldots\,,\,N \quad . \tag{A.10}$$

Second step

$$\begin{pmatrix} x_{i1} & x_{i2} & x_{i3} & x_{i4} \end{pmatrix} \begin{pmatrix} L_{11} & 0 & 0 & 0 \\ L_{21} & L_{22} & 0 & 0 \\ L_{31} & L_{32} & L_{33} & 0 \\ L_{41} & L_{42} & L_{43} & L_{44} \end{pmatrix} \quad = \quad \begin{pmatrix} y_{i1} & y_{i2} & y_{i3} & y_{i4} \end{pmatrix} \quad , \tag{A.11}$$

that makes us realize

$$x_{ij} \quad = \quad \left(y_{ij} \; - \sum_{k=j+1}^{N} x_{ik}\; L_{kj} \right) \frac{1}{L_{jj}} \qquad \text{with} \quad j = N\,,\,N-1\,,\,\ldots\,,\,1 \quad . \tag{A.12}$$

A.2.2. Algorithm

The algorithm can be applied in-place. The inputs are the $N \times N$ matrix $\mathbf{A}$, and the $M \times N$ matrix $\mathbf{B}$. The output is the matrix $\mathbf{X}$ stored in the $\mathbf{B}$ space.

```
void solve_Cholesky( Matrix A , Matrix B ) {
  // first, we compute the Cholesky decomposition
  Cholesky( A )  // now L is stored in the space of A
  // then we take each pair of rows of X and B independently
  for each row i = 1, ... , M
    // we solve (A.8a)
    for each column j = 1, ... , N
      sum1 = B(i,j)
      for each column k = 1, ... , j-1
        sum1 = sum1 - B(i,k) * A(j,k)
      end for each column k
      B(i,j) = sum1 / A(j,j)
    end for each column j
    // we solve (A.8b)
    for each column j = N, ... , 1
      sum2 = B(i,j)
      for each column k = j+1, ... , N
        sum2 = sum2 - B(i,k) * A(k,j)
      end for each column k
      B(i,j) = sum2 / A(j,j)
    end for each column j
  end for each row i
end solve_Cholesky
```

A.3. Preserving the Positive-Definiteness of a Transformed Covariance Matrix

In the Kalman Filter we find several expressions of the form

$$\mathbf{C} = \mathbf{F}\,\mathbf{A}\,\mathbf{F}^T + \mathbf{Q} = \tag{A.13a}$$

$$= \mathbf{B} + \mathbf{Q} \quad , \tag{A.13b}$$

relating covariance matrices (real positive-definite) $\mathbf{A}$, $\mathbf{Q}$, and $\mathbf{C}$. As the matrix $\mathbf{A}$ is positive-definite it can be Cholesky-decomposed into $\mathbf{A} = \mathbf{L}\,\mathbf{L}^T$. Then, the matrix $\mathbf{B}$ is also positive definite:

$$\mathbf{x}^T\,\mathbf{B}\,\mathbf{x} = \mathbf{x}^T\,\mathbf{F}\,\mathbf{A}\,\mathbf{F}^T\,\mathbf{x} = \tag{A.14a}$$

$$= \mathbf{x}^T\,\mathbf{F}\,\mathbf{L}\,\mathbf{L}^T\,\mathbf{F}^T\,\mathbf{x} = \tag{A.14b}$$

$$= \|\,\mathbf{L}^T\,\mathbf{F}^T\,\mathbf{x}\,\|^2 > 0 \quad . \tag{A.14c}$$

However, if we compute $\mathbf{B}$ through the product $(\mathbf{F}\,\mathbf{A})\,\mathbf{F}^T$ or $\mathbf{F}\,(\mathbf{A}\,\mathbf{F}^T)$, we will not obtain a symmetric matrix due to round-off errors. On the other hand, if we compute $\mathbf{B}$ through

$$\mathbf{A} = \mathbf{L}\,\mathbf{L}^T \quad , \tag{A.15}$$

$$\mathbf{M} = \mathbf{F}\,\mathbf{L} \quad , \tag{A.16}$$

$$\mathbf{B} = \mathbf{M}\,\mathbf{M}^T \quad , \tag{A.17}$$

the result will be a positive-definite matrix; even with round-off errors.

Finally, as the sum of positive-definite matrices is positive-definite:

$$\mathbf{x}^T\,\mathbf{C}\,\mathbf{x} = \mathbf{x}^T\,(\mathbf{B} + \mathbf{Q})\,\mathbf{x} = \tag{A.18a}$$

$$= \mathbf{x}^T\,\mathbf{B}\,\mathbf{x} + \mathbf{x}^T\,\mathbf{Q}\,\mathbf{x} > 0 \quad , \tag{A.18b}$$

the matrix $\mathbf{C}$ will be positive-definite.

A.4. Derivation of the Quaternion Exponentiation Formula

First, we decompose a quaternion $\boldsymbol{q}$ into its real part $\boldsymbol{q}_r$, and its imaginary part $\boldsymbol{q}_i$:

$$\boldsymbol{q} \;=\; \boldsymbol{q}_r \;+\; \boldsymbol{q}_i \;=\; \begin{pmatrix} q_0 \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} 0 \\ \mathbf{q} \end{pmatrix} \quad . \tag{A.19}$$

From the exponential definition (3.21), if $\boldsymbol{q}_r$ and $\boldsymbol{q}_i$ commute ($\boldsymbol{q}_r * \boldsymbol{q}_i = \boldsymbol{q}_i * \boldsymbol{q}_r$), then $e^{\boldsymbol{q}} = e^{\boldsymbol{q}_r}\, e^{\boldsymbol{q}_i} = e^{q_0}\, e^{\boldsymbol{q}_i}$. The first powers of $\boldsymbol{q}_i$ are displayed in Table A.1.

Table A.1: Powers of a pure imaginary quaternion.

n	0	1	2	3	4	5	6	$\cdots$
$\boldsymbol{q}_i^n$	$\begin{pmatrix} 1 \\ \mathbf{0} \end{pmatrix}$	$\begin{pmatrix} 0 \\ \mathbf{q} \end{pmatrix}$	$\begin{pmatrix} -\Vert\mathbf{q}\Vert^2 \\ \mathbf{0} \end{pmatrix}$	$\begin{pmatrix} 0 \\ -\Vert\mathbf{q}\Vert^2\ \mathbf{q} \end{pmatrix}$	$\begin{pmatrix} \Vert\mathbf{q}\Vert^4 \\ \mathbf{0} \end{pmatrix}$	$\begin{pmatrix} 0 \\ \Vert\mathbf{q}\Vert^4\ \mathbf{q} \end{pmatrix}$	$\begin{pmatrix} -\Vert\mathbf{q}\Vert^6 \\ \mathbf{0} \end{pmatrix}$	$\cdots$

We get by induction:

$$\boldsymbol{q}_i^{2n} \;=\; (-1)^n\, \Vert\mathbf{q}\Vert^{2n} \quad , \tag{A.20a}$$

$$\boldsymbol{q}_i^{2n+1} \;=\; (-1)^n\, \Vert\mathbf{q}\Vert^{2n}\, \mathbf{q} \quad . \tag{A.20b}$$

Then, equation (3.21) takes the form:

$$e^{\boldsymbol{q}} \;=\; e^{q_0} \left[\sum_{n=0}^{\infty} \frac{(-1)^n}{2n}\, \Vert\mathbf{q}\Vert^{2n} \;+\; \mathbf{q} \sum_{n=0}^{\infty} (-1)^n\, \Vert\mathbf{q}\Vert^{2n} \right] \;= \tag{A.21a}$$

$$\;=\; e^{q_0} \left[\sum_{n=0}^{\infty} \frac{(-1)^n}{2n}\, \Vert\mathbf{q}\Vert^{2n} \;+\; \frac{\mathbf{q}}{\Vert\mathbf{q}\Vert} \sum_{n=0}^{\infty} (-1)^n\, \Vert\mathbf{q}\Vert^{2n+1} \right] \quad . \tag{A.21b}$$

Recognizing the Taylor series expansions of the sine and cosine, we conclude:

$$e^{\boldsymbol{q}} \;=\; e^{q_0} \left[\cos\Vert\mathbf{q}\Vert \;+\; \frac{\mathbf{q}}{\Vert\mathbf{q}\Vert}\, \sin\Vert\mathbf{q}\Vert \right] \quad . \tag{A.22}$$

A.5. Derivation of the Transition Maps

This appendix contains the derivation of the transition map for each chart.

A.5.1. Orthographic

Using the inverse of the transformation that defines the chart, φ^{-1},

$$\boldsymbol{\delta}^{\overline{\boldsymbol{q}}} = \begin{pmatrix} \sqrt{1 - \frac{\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|^2}{4}} \\ \mathbf{e}^{\overline{\boldsymbol{q}}}/2 \end{pmatrix} . \tag{A.23}$$

Introducing (A.23) into (3.46),

$$\boldsymbol{\delta}^{\overline{\boldsymbol{p}}} = \overline{\boldsymbol{\delta}}^{*} * \boldsymbol{\delta}^{\overline{\boldsymbol{q}}} = \begin{pmatrix} \overline{\delta}_0 \sqrt{1 - \frac{\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|^2}{4}} + \overline{\boldsymbol{\delta}} \cdot \frac{\mathbf{e}^{\overline{\boldsymbol{q}}}}{2} \\ \overline{\delta}_0 \frac{\mathbf{e}^{\overline{\boldsymbol{q}}}}{2} - \sqrt{1 - \frac{\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|^2}{4}}\, \overline{\boldsymbol{\delta}} - \overline{\boldsymbol{\delta}} \times \frac{\mathbf{e}^{\overline{\boldsymbol{q}}}}{2} \end{pmatrix} . \tag{A.24}$$

Finally, applying the chart definition,

$$\mathbf{e}^{\overline{\boldsymbol{p}}} = 2\,\delta^{\overline{\boldsymbol{p}}} = \overline{\delta}_0\, \mathbf{e}^{\overline{\boldsymbol{q}}} - \sqrt{4 - \|\mathbf{e}^{\overline{\boldsymbol{q}}}\|^2}\, \overline{\boldsymbol{\delta}} - \overline{\boldsymbol{\delta}} \times \mathbf{e}^{\overline{\boldsymbol{q}}} . \tag{A.25}$$

A.5.2. Rodrigues Parameters

Using the inverse of the transformation that defines the chart, φ^{-1},

$$\boldsymbol{\delta}^{\overline{\boldsymbol{q}}} = \frac{1}{\sqrt{4 + \|\mathbf{e}^{\overline{\boldsymbol{q}}}\|^2}} \begin{pmatrix} 2 \\ \mathbf{e}^{\overline{\boldsymbol{q}}} \end{pmatrix} . \tag{A.26}$$

Introducing (A.26) into (3.46),

$$\boldsymbol{\delta}^{\overline{\boldsymbol{p}}} = \overline{\boldsymbol{\delta}}^{*} * \boldsymbol{\delta}^{\overline{\boldsymbol{q}}} = \frac{1}{\sqrt{4 + \|\mathbf{e}^{\overline{\boldsymbol{q}}}\|^2}} \begin{pmatrix} 2\,\overline{\delta}_0 + \overline{\boldsymbol{\delta}} \cdot \mathbf{e}^{\overline{\boldsymbol{q}}} \\ \overline{\delta}_0\, \mathbf{e}^{\overline{\boldsymbol{q}}} - 2\,\overline{\boldsymbol{\delta}} - \overline{\boldsymbol{\delta}} \times \mathbf{e}^{\overline{\boldsymbol{q}}} \end{pmatrix} . \tag{A.27}$$

Finally, applying the chart definition,

$$\mathbf{e}^{\overline{\boldsymbol{p}}} = 2\,\frac{\delta^{\overline{\boldsymbol{p}}}}{\delta_0^{\overline{\boldsymbol{p}}}} = 2\,\frac{\overline{\delta}_0\, \mathbf{e}^{\overline{\boldsymbol{q}}} - 2\,\overline{\boldsymbol{\delta}} - \overline{\boldsymbol{\delta}} \times \mathbf{e}^{\overline{\boldsymbol{q}}}}{2\,\overline{\delta}_0 + \overline{\boldsymbol{\delta}} \cdot \mathbf{e}^{\overline{\boldsymbol{q}}}} . \tag{A.28}$$

A.5.3. Modified Rodrigues Parameters

Using the inverse of the transformation that defines the chart, φ^{-1},

$$\boldsymbol{\delta}^{\overline{\boldsymbol{q}}} = \frac{1}{16 + \|\mathbf{e}^{\overline{\boldsymbol{q}}}\|^2} \begin{pmatrix} 16 - \|\mathbf{e}^{\overline{\boldsymbol{q}}}\|^2 \\ 8\,\mathbf{e}^{\overline{\boldsymbol{q}}} \end{pmatrix} . \tag{A.29}$$

Introducing (A.29) into (3.46),

$$\boldsymbol{\delta}^{\overline{\boldsymbol{p}}} = \overline{\boldsymbol{\delta}}^{*} * \boldsymbol{\delta}^{\overline{\boldsymbol{q}}} = \frac{1}{16 + \|\mathbf{e}^{\overline{\boldsymbol{q}}}\|^2} \begin{pmatrix} \overline{\delta}_0 \left(16 - \|\mathbf{e}^{\overline{\boldsymbol{q}}}\|^2\right) + 8\,\overline{\boldsymbol{\delta}} \cdot \mathbf{e}^{\overline{\boldsymbol{q}}} \\ 8\,\overline{\delta}_0\, \mathbf{e}^{\overline{\boldsymbol{q}}} - \left(16 - \|\mathbf{e}^{\overline{\boldsymbol{q}}}\|^2\right) \overline{\boldsymbol{\delta}} - 8\,\overline{\boldsymbol{\delta}} \times \mathbf{e}^{\overline{\boldsymbol{q}}} \end{pmatrix} . \tag{A.30}$$

Finally, applying the chart definition,

$$\mathbf{e}^{\overline{\boldsymbol{p}}} = 4\,\frac{\delta^{\overline{\boldsymbol{p}}}}{1 + \delta_0^{\overline{\boldsymbol{p}}}} = 4\,\frac{8\,\overline{\delta}_0\, \mathbf{e}^{\overline{\boldsymbol{q}}} - \left(16 - \|\mathbf{e}^{\overline{\boldsymbol{q}}}\|^2\right) \overline{\boldsymbol{\delta}} - 8\,\overline{\boldsymbol{\delta}} \times \mathbf{e}^{\overline{\boldsymbol{q}}}}{16 + \|\mathbf{e}^{\overline{\boldsymbol{q}}}\|^2 + \overline{\delta}_0 \left(16 - \|\mathbf{e}^{\overline{\boldsymbol{q}}}\|^2\right) + 8\,\overline{\boldsymbol{\delta}} \cdot \mathbf{e}^{\overline{\boldsymbol{q}}}} . \tag{A.31}$$

A.5.4. Rotation Vector

Using the inverse of the transformation that defines the chart, φ^{-1},

$$\boldsymbol{\delta}^{\overline{\boldsymbol{q}}} \;=\; \begin{pmatrix} \cos\left(\frac{\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|}{2}\right) \\ \widehat{\mathbf{e}}^{\overline{\boldsymbol{q}}}\,\sin\left(\frac{\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|}{2}\right) \end{pmatrix} \;. \tag{A.32}$$

Introducing (A.32) into (3.46),

$$\boldsymbol{\delta}^{\overline{\boldsymbol{p}}} \;=\; \overline{\boldsymbol{\delta}}^{*} * \boldsymbol{\delta}^{\overline{\boldsymbol{q}}} \;=\; \begin{pmatrix} \overline{\delta}_0\,\cos\left(\frac{\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|}{2}\right) \,+\, \overline{\boldsymbol{\delta}}\cdot\widehat{\mathbf{e}}^{\overline{\boldsymbol{q}}}\,\sin\left(\frac{\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|}{2}\right) \\ \overline{\delta}_0\,\widehat{\mathbf{e}}^{\overline{\boldsymbol{q}}}\,\sin\left(\frac{\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|}{2}\right) \,-\, \cos\left(\frac{\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|}{2}\right)\,\overline{\boldsymbol{\delta}} \,-\, \overline{\boldsymbol{\delta}}\times\widehat{\mathbf{e}}^{\overline{\boldsymbol{q}}}\,\sin\left(\frac{\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|}{2}\right) \end{pmatrix} \;. \tag{A.33}$$

Finally, applying the chart definition,

$$\mathbf{e}^{\overline{\boldsymbol{p}}} \;=\; 2\,\frac{\delta^{\overline{\boldsymbol{p}}}}{\|\delta^{\overline{\boldsymbol{p}}}\|}\,\arcsin\|\delta^{\overline{\boldsymbol{p}}}\| \;, \tag{A.34}$$

with

$$\delta^{\overline{\boldsymbol{p}}} \;=\; \overline{\delta}_0\,\widehat{\mathbf{e}}^{\overline{\boldsymbol{q}}}\,\sin\left(\frac{\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|}{2}\right) \,-\, \cos\left(\frac{\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|}{2}\right)\,\overline{\boldsymbol{\delta}} \,-\, \overline{\boldsymbol{\delta}}\times\widehat{\mathbf{e}}^{\overline{\boldsymbol{q}}}\,\sin\left(\frac{\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|}{2}\right) \;. \tag{A.35}$$

Note that all transition maps are expressed using the quaternion $\overline{\boldsymbol{\delta}}$. Given that $\overline{\mathbf{e}} = \varphi\big(\overline{\boldsymbol{\delta}}\big)$ we could also have expressed them using $\overline{\mathbf{e}}$. However, our choice makes transition maps to take a simpler form.

A.6. Proof: $\left(\mathbf{A}\,\mathbf{x} \times \mathbf{A}\,\mathbf{y}\right) = \det(\mathbf{A})\,\mathbf{A}^{-T}\left(\mathbf{x} \times \mathbf{y}\right)$

We begin by noticing that

$$\mathbf{u} \cdot (\mathbf{x} \times \mathbf{y}) = \det\begin{pmatrix} - & \mathbf{u} & - \\ - & \mathbf{x} & - \\ - & \mathbf{y} & - \end{pmatrix} = \det\begin{pmatrix} | & | & | \\ \mathbf{u} & \mathbf{x} & \mathbf{y} \\ | & | & | \end{pmatrix} \quad . \tag{A.36}$$

Then, for any 3×3 matrix $\mathbf{A}$,

$$\begin{aligned} \mathbf{u} \cdot \mathbf{A}^T\left(\mathbf{A}\,\mathbf{x} \times \mathbf{A}\,\mathbf{y}\right) &= (\mathbf{A}\,\mathbf{u}) \cdot \left(\mathbf{A}\,\mathbf{x} \times \mathbf{A}\,\mathbf{y}\right) = && \text{(A.37a)} \\ &= \det\begin{pmatrix} | & | & | \\ \mathbf{A}\,\mathbf{u} & \mathbf{A}\,\mathbf{x} & \mathbf{A}\,\mathbf{y} \\ | & | & | \end{pmatrix} = && \text{(A.37b)} \\ &= \det\left(\mathbf{A}\begin{pmatrix} | & | & | \\ \mathbf{u} & \mathbf{x} & \mathbf{y} \\ | & | & | \end{pmatrix}\right) = && \text{(A.37c)} \\ &= \det(\mathbf{A})\,\det\begin{pmatrix} | & | & | \\ \mathbf{u} & \mathbf{x} & \mathbf{y} \\ | & | & | \end{pmatrix} = && \text{(A.37d)} \\ &= \det(\mathbf{A})\,\left[\mathbf{u} \cdot (\mathbf{x} \times \mathbf{y})\right] \quad . && \text{(A.37e)} \end{aligned}$$

Since this is true for all $\mathbf{u} \in \mathbb{R}^3$, then

$$\mathbf{A}^T\left(\mathbf{A}\,\mathbf{x} \times \mathbf{A}\,\mathbf{y}\right) = \det(\mathbf{A})\,(\mathbf{x} \times \mathbf{y}) \quad . \tag{A.38}$$

For the special case of a rotation matrix $\mathbf{R}$, for which $\mathbf{R}^{-T} = \mathbf{R}$ and $\det(\mathbf{R}) = +1$,

$$\left(\mathbf{R}\,\mathbf{x} \times \mathbf{R}\,\mathbf{y}\right) = \mathbf{R}\,(\mathbf{x} \times \mathbf{y}) \quad . \tag{A.39}$$

A.7. Exponential of $\frac{1}{2}\ {}_*[\boldsymbol{\omega}]\ t$

Let us start by computing the first powers. Using (3.8c) we obtain:

$$\frac{1}{2}\ {}_*[\boldsymbol{\omega}]\ t \quad := \quad \frac{t}{2}\begin{pmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{pmatrix} \quad , \tag{A.40a}$$

$$\Big(\frac{1}{2}\ {}_*[\boldsymbol{\omega}]\ t\Big)^2 \quad = \quad -\Big(\frac{t}{2}\Big)^2\ (\omega_1^2+\omega_2^2+\omega_3^2)\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad , \tag{A.40b}$$

$$\Big(\frac{1}{2}\ {}_*[\boldsymbol{\omega}]\ t\Big)^3 \quad = \quad -\Big(\frac{t}{2}\Big)^3\ (\omega_1^2+\omega_2^2+\omega_3^2)\ {}_*[\boldsymbol{\omega}] \quad , \tag{A.40c}$$

$$\Big(\frac{1}{2}\ {}_*[\boldsymbol{\omega}]\ t\Big)^4 \quad = \quad \Big(\frac{t}{2}\Big)^4\ (\omega_1^2+\omega_2^2+\omega_3^2)^2\ \mathbf{I} \quad , \tag{A.40d}$$

$$\Big(\frac{1}{2}\ {}_*[\boldsymbol{\omega}]\ t\Big)^5 \quad = \quad \Big(\frac{t}{2}\Big)^5\ (\omega_1^2+\omega_2^2+\omega_3^2)^2\ {}_*[\boldsymbol{\omega}] \quad , \tag{A.40e}$$

$$\Big(\frac{1}{2}\ {}_*[\boldsymbol{\omega}]\ t\Big)^6 \quad = \quad -\Big(\frac{t}{2}\Big)^6\ (\omega_1^2+\omega_2^2+\omega_3^2)^3\ \mathbf{I} \quad , \ \ldots \tag{A.40f}$$

By induction we notice that

$$\Big(\frac{1}{2}\ {}_*[\boldsymbol{\omega}]\ t\Big)^{2n} \quad = \quad \Big(\frac{t}{2}\Big)^{2n}\ (-1)^n\ \big(\|\boldsymbol{\omega}\|^2\big)^n\ \mathbf{I} \quad , \tag{A.41}$$

$$\Big(\frac{1}{2}\ {}_*[\boldsymbol{\omega}]\ t\Big)^{2n+1} \quad = \quad \Big(\frac{t}{2}\Big)^{2n+1}\ (-1)^n\ \big(\|\boldsymbol{\omega}\|^2\big)^n\ {}_*[\boldsymbol{\omega}] \quad , \tag{A.42}$$

for $n = 0, 1, 2, \ldots$. Then, the matrix exponential will take the following form:

$$e^{\frac{1}{2}\,{}_*[\boldsymbol{\omega}]\,t} \quad = \quad \sum_{n=0}^{\infty}\ \frac{\big(\frac{1}{2}\ {}_*[\boldsymbol{\omega}]\ t\big)^n}{n!} \quad = \tag{A.43a}$$

$$= \quad \sum_{n=0}^{\infty}\ \Big(\frac{t}{2}\Big)^{2n}\ \frac{(-1)^n}{(2n)!}\ \big(\|\boldsymbol{\omega}\|^2\big)^n\ \mathbf{I}\ + \\ + \quad \sum_{n=0}^{\infty}\ \Big(\frac{t}{2}\Big)^{2n+1}\ \frac{(-1)^n}{(2n+1)!}\ \big(\|\boldsymbol{\omega}\|^2\big)^n\ {}_*[\boldsymbol{\omega}] \quad = \tag{A.43b}$$

$$= \quad \sum_{n=0}^{\infty}\ \Big(\frac{t}{2}\Big)^{2n}\ \frac{(-1)^n}{(2n)!}\ \|\boldsymbol{\omega}\|^{2n}\ \mathbf{I}\ + \\ + \quad \sum_{n=0}^{\infty}\ \Big(\frac{t}{2}\Big)^{2n+1}\ \frac{(-1)^n}{(2n+1)!}\ \|\boldsymbol{\omega}\|^{2n+1}\ \frac{{}_*[\boldsymbol{\omega}]}{\|\boldsymbol{\omega}\|} \quad = \tag{A.43c}$$

$$= \quad \cos\Big(\frac{\|\boldsymbol{\omega}\|\ t}{2}\Big)\ \mathbf{I}\ +\ \sin\Big(\frac{\|\boldsymbol{\omega}\|\ t}{2}\Big)\ \frac{{}_*[\boldsymbol{\omega}]}{\|\boldsymbol{\omega}\|} \quad = \tag{A.43d}$$

$$= \quad {}_*\left[\begin{pmatrix} \cos\Big(\frac{\|\boldsymbol{\omega}\|\ t}{2}\Big) \\ \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}\ \sin\Big(\frac{\|\boldsymbol{\omega}\|\ t}{2}\Big) \end{pmatrix}\right] \quad . \tag{A.43e}$$

A.8. Useful Relations for the MEKF

A.8.1. Expression for $\left.\frac{d\ \mathbf{R}(\boldsymbol{q})}{d\,\boldsymbol{q}}\right|_{\boldsymbol{q}=\mathbf{1}}$

From (3.33) we have

$$\frac{d\ \mathbf{R}(\boldsymbol{q})}{d\,q_k} = 2\begin{pmatrix} 0\,\delta_{0k}+ & -q_3\,\delta_{0k}+ & q_2\,\delta_{0k}+ \\ +0\,\delta_{1k}+ & +q_2\,\delta_{1k}+ & +q_3\,\delta_{1k}+ \\ -2\,q_2\,\delta_{2k}+ & +q_1\,\delta_{2k}+ & +q_0\,\delta_{2k}+ \\ -2\,q_3\,\delta_{3k} & -q_0\,\delta_{3k} & +q_1\,\delta_{3k} \\ \\ q_3\,\delta_{0k}+ & 0\,\delta_{0k}+ & -q_1\,\delta_{0k}+ \\ +q_2\,\delta_{1k}+ & -2\,q_1\,\delta_{1k}+ & -q_0\,\delta_{1k}+ \\ +q_1\,\delta_{2k}+ & +0\,\delta_{2k}+ & +q_3\,\delta_{2k}+ \\ +q_0\,\delta_{3k} & -2\,q_3\,\delta_{3k} & +q_2\,\delta_{3k} \\ \\ -q_2\,\delta_{0k}+ & q_1\,\delta_{0k}+ & 0\,\delta_{0k}+ \\ +q_3\,\delta_{1k}+ & +q_0\,\delta_{1k}+ & -2\,q_1\,\delta_{1k}+ \\ -q_0\,\delta_{2k}+ & +q_3\,\delta_{2k}+ & -2\,q_2\,\delta_{2k}+ \\ +q_1\,\delta_{3k} & +q_2\,\delta_{3k} & +0\,\delta_{3k} \end{pmatrix} . \tag{A.44}$$

Evaluating (A.44) at $\boldsymbol{q}=\mathbf{1}$ we obtain

$$\left.\frac{d\ \mathbf{R}(\boldsymbol{q})}{d\,\boldsymbol{q}}\right|_{\boldsymbol{q}=\mathbf{1}} = 2\begin{pmatrix} 0 & -\delta_{3k} & \delta_{2k} \\ \delta_{3k} & 0 & -\delta_{1k} \\ -\delta_{2k} & \delta_{1k} & 0 \end{pmatrix} . \tag{A.45}$$

Expression (A.45) is similar to the matrix expression of a cross product. With this in mind we can rewrite

$$\left.\frac{d\ R_{il}(\boldsymbol{q})}{d\,q_k}\right|_{\boldsymbol{q}=\mathbf{1}} = 2\sum_n \varepsilon_{inl}\ \delta_{nk} = \tag{A.46a}$$

$$= 2\,\varepsilon_{ikl} . \tag{A.46b}$$

where ε_{ikl} is the Levi-Civita symbol.

A.8.2. Expression for $\sum_k \left.\frac{d\ \mathbf{R}(\boldsymbol{\delta}^{\overline{\boldsymbol{q}}})}{d\,\delta_k^{\overline{\boldsymbol{q}}}}\right|_{\boldsymbol{\delta}^{\overline{\boldsymbol{q}}}=\mathbf{1}} \left.\frac{d\ \delta_k^{\overline{\boldsymbol{q}}}}{d\,\mathbf{e}^{\overline{\boldsymbol{q}}}}\right|_{\mathbf{e}^{\overline{\boldsymbol{q}}}=\mathbf{0}}$

From (3.36) we have

$$\left.\frac{d\ \boldsymbol{\delta}^{\overline{\boldsymbol{q}}}}{d\,e_j^{\overline{\boldsymbol{q}}}}\right|_{\mathbf{e}^{\overline{\boldsymbol{q}}}=\mathbf{0}} \approx \left.\begin{pmatrix} -\,e_j^{\overline{\boldsymbol{q}}}/4 \\ \delta_{1j}/2 \\ \delta_{2j}/2 \\ \delta_{3j}/2 \end{pmatrix}\right|_{\mathbf{e}^{\overline{\boldsymbol{q}}}=\mathbf{0}} = \frac{1}{2}\begin{pmatrix} 0 \\ \delta_{1j} \\ \delta_{2j} \\ \delta_{3j} \end{pmatrix} . \tag{A.47}$$

Using (A.46) and (A.47) we get

$$\sum_k \left.\frac{d\ R_{il}(\boldsymbol{\delta}^{\overline{\boldsymbol{q}}})}{d\,\delta_k^{\overline{\boldsymbol{q}}}}\right|_{\boldsymbol{\delta}^{\overline{\boldsymbol{q}}}=\mathbf{1}} \left.\frac{d\ \delta_k^{\overline{\boldsymbol{q}}}}{d\,e_j^{\overline{\boldsymbol{q}}}}\right|_{\mathbf{e}^{\overline{\boldsymbol{q}}}=\mathbf{0}} = 2\sum_k \varepsilon_{ikl}\,\frac{1}{2}\,\delta_{kj} = \tag{A.48a}$$

$$= \varepsilon_{ijl} . \tag{A.48b}$$

A.8.3. Proof of $\left.\frac{d\ \mathbf{R}(\boldsymbol{q})\,\mathbf{v}}{d\,\mathbf{e}^{\overline{\boldsymbol{q}}}}\right|_{\boldsymbol{q}=\overline{\boldsymbol{q}}} = \mathbf{R}(\overline{\boldsymbol{q}})\,[-\mathbf{v}]_\times$

We start from

$$\mathbf{R}(\boldsymbol{q})\,\mathbf{v} = \mathbf{R}(\overline{\boldsymbol{q}})\,\mathbf{R}(\boldsymbol{\delta}^{\overline{\boldsymbol{q}}})\,\mathbf{v} . \tag{A.49}$$

Applying the chain rule to (A.49),

$$\frac{d\,\mathbf{R}(\boldsymbol{q})\,\mathbf{v}}{d\,e_j^{\overline{\boldsymbol{q}}}} = \sum_k \frac{d\,\mathbf{R}(\boldsymbol{q})}{d\,\delta_k^{\overline{\boldsymbol{q}}}}\,\frac{d\,\delta_k^{\overline{\boldsymbol{q}}}}{d\,e_j^{\overline{\boldsymbol{q}}}}\,\mathbf{v} = \tag{A.50a}$$

$$= \sum_k \mathbf{R}(\overline{\boldsymbol{q}})\,\frac{d\,\mathbf{R}(\boldsymbol{\delta}^{\overline{\boldsymbol{q}}})}{d\,\delta_k^{\overline{\boldsymbol{q}}}}\,\frac{d\,\delta_k^{\overline{\boldsymbol{q}}}}{d\,e_j^{\overline{\boldsymbol{q}}}}\,\mathbf{v} \quad . \tag{A.50b}$$

Noting that evaluating at $\boldsymbol{q}=\overline{\boldsymbol{q}}$ is the same as evaluating at $\boldsymbol{\delta}^{\overline{\boldsymbol{q}}}=\mathbf{1}$ and $\mathbf{e}^{\overline{\boldsymbol{q}}}=\mathbf{0}$,

$$\left.\frac{d\,\mathbf{R}(\boldsymbol{q})\,\mathbf{v}}{d\,e_j^{\overline{\boldsymbol{q}}}}\right|_{\boldsymbol{q}=\overline{\boldsymbol{q}}} = \sum_k \mathbf{R}(\overline{\boldsymbol{q}})\,\left.\frac{d\,\mathbf{R}(\boldsymbol{\delta}^{\overline{\boldsymbol{q}}})}{d\,\delta_k^{\overline{\boldsymbol{q}}}}\right|_{\boldsymbol{\delta}^{\overline{\boldsymbol{q}}}=\mathbf{1}}\left.\frac{d\,\delta_k^{\overline{\boldsymbol{q}}}}{d\,e_j^{\overline{\boldsymbol{q}}}}\right|_{\mathbf{e}^{\overline{\boldsymbol{q}}}=\mathbf{0}}\mathbf{v} \quad . \tag{A.51}$$

Introducing (A.48) in (A.51) we obtain

$$\left.\frac{d\,\left(\mathbf{R}(\boldsymbol{q})\,\mathbf{v}\right)_\alpha}{d\,e_j^{\overline{\boldsymbol{q}}}}\right|_{\boldsymbol{q}=\overline{\boldsymbol{q}}} = \sum_{ikl} R_{\alpha i}(\overline{\boldsymbol{q}})\,\left.\frac{d\,R_{il}(\boldsymbol{\delta}^{\overline{\boldsymbol{q}}})}{d\,\delta_k^{\overline{\boldsymbol{q}}}}\right|_{\boldsymbol{\delta}^{\overline{\boldsymbol{q}}}=\mathbf{1}}\left.\frac{d\,\delta_k^{\overline{\boldsymbol{q}}}}{d\,e_j^{\overline{\boldsymbol{q}}}}\right|_{\mathbf{e}^{\overline{\boldsymbol{q}}}=\mathbf{0}} v_l = \tag{A.52a}$$

$$= \sum_{il} R_{\alpha i}(\overline{\boldsymbol{q}})\,\varepsilon_{ijl}\,v_l = \tag{A.52b}$$

$$= -\sum_{il} R_{\alpha i}(\overline{\boldsymbol{q}})\,\varepsilon_{ilj}\,v_l \quad . \tag{A.52c}$$

After having identified the cross product in equation (A.52), it can be re-expressed in matrix form as

$$\left.\frac{d\,\mathbf{R}(\boldsymbol{q})\,\mathbf{v}}{d\,\mathbf{e}^{\overline{\boldsymbol{q}}}}\right|_{\boldsymbol{q}=\overline{\boldsymbol{q}}} = \mathbf{R}(\overline{\boldsymbol{q}})\,\left[-\mathbf{v}\right]_\times \quad . \tag{A.53}$$

A.8.4. Proof of $\left.\frac{d\,\mathbf{R}^T(\boldsymbol{q})\,\mathbf{v}}{d\,\mathbf{e}^{\overline{\boldsymbol{q}}}}\right|_{\boldsymbol{q}=\overline{\boldsymbol{q}}} = \left[\mathbf{R}^T(\overline{\boldsymbol{q}})\,\mathbf{v}\right]_\times$

We start from

$$\mathbf{R}^T(\boldsymbol{q})\,\mathbf{v} = \mathbf{R}^T(\boldsymbol{\delta}^{\overline{\boldsymbol{q}}})\,\mathbf{R}^T(\overline{\boldsymbol{q}})\,\mathbf{v} \quad . \tag{A.54}$$

Applying the chain rule to (A.54),

$$\frac{d\,\mathbf{R}^T(\boldsymbol{q})\,\mathbf{v}}{d\,e_j^{\overline{\boldsymbol{q}}}} = \sum_k \frac{d\,\mathbf{R}^T(\boldsymbol{q})}{d\,\delta_k^{\overline{\boldsymbol{q}}}}\,\frac{d\,\delta_k^{\overline{\boldsymbol{q}}}}{d\,e_j^{\overline{\boldsymbol{q}}}}\,\mathbf{v} = \tag{A.55a}$$

$$= \sum_k \frac{d\,\mathbf{R}^T(\boldsymbol{\delta}^{\overline{\boldsymbol{q}}})}{d\,\delta_k^{\overline{\boldsymbol{q}}}}\,\frac{d\,\delta_k^{\overline{\boldsymbol{q}}}}{d\,e_j^{\overline{\boldsymbol{q}}}}\,\mathbf{R}^T(\overline{\boldsymbol{q}})\,\mathbf{v} \quad . \tag{A.55b}$$

Noting that evaluating at $\boldsymbol{q}=\overline{\boldsymbol{q}}$ is the same as evaluating at $\boldsymbol{\delta}^{\overline{\boldsymbol{q}}}=\mathbf{1}$ and $\mathbf{e}^{\overline{\boldsymbol{q}}}=\mathbf{0}$,

$$\left.\frac{d\,\mathbf{R}^T(\boldsymbol{q})\,\mathbf{v}}{d\,e_j^{\overline{\boldsymbol{q}}}}\right|_{\boldsymbol{q}=\overline{\boldsymbol{q}}} = \sum_k \left.\frac{d\,\mathbf{R}^T(\boldsymbol{\delta}^{\overline{\boldsymbol{q}}})}{d\,\delta_k^{\overline{\boldsymbol{q}}}}\right|_{\boldsymbol{\delta}^{\overline{\boldsymbol{q}}}=\mathbf{1}}\left.\frac{d\,\delta_k^{\overline{\boldsymbol{q}}}}{d\,e_j^{\overline{\boldsymbol{q}}}}\right|_{\mathbf{e}^{\overline{\boldsymbol{q}}}=\mathbf{0}}\mathbf{R}^T(\overline{\boldsymbol{q}})\,\mathbf{v} \quad . \tag{A.56}$$

Introducing (A.48) in (A.56) we obtain

$$\left.\frac{d\,\left(\mathbf{R}^T(\,\boldsymbol{q}\,)\,\mathbf{v}\,\right)_i}{d\,e_j^{\overline{\boldsymbol{q}}}}\right|_{\boldsymbol{q}=\overline{\boldsymbol{q}}} = \sum_{ikl} \left.\frac{d\,R_{il}^T(\,\boldsymbol{\delta}^{\overline{\boldsymbol{q}}}\,)}{d\,\delta_k^{\overline{\boldsymbol{q}}}}\right|_{\boldsymbol{\delta}^{\overline{\boldsymbol{q}}}=\mathbf{1}} \left.\frac{d\,\delta_k^{\overline{\boldsymbol{q}}}}{d\,e_j^{\overline{\boldsymbol{q}}}}\right|_{\mathbf{e}^{\overline{\boldsymbol{q}}}=\mathbf{0}} \left(\mathbf{R}^T(\,\overline{\boldsymbol{q}}\,)\,\mathbf{v}\,\right)_l = \tag{A.57a}$$

$$= \sum_{ikl} \left.\frac{d\,R_{li}(\,\boldsymbol{\delta}^{\overline{\boldsymbol{q}}}\,)}{d\,\delta_k^{\overline{\boldsymbol{q}}}}\right|_{\boldsymbol{\delta}^{\overline{\boldsymbol{q}}}=\mathbf{1}} \left.\frac{d\,\delta_k^{\overline{\boldsymbol{q}}}}{d\,e_j^{\overline{\boldsymbol{q}}}}\right|_{\mathbf{e}^{\overline{\boldsymbol{q}}}=\mathbf{0}} \left(\mathbf{R}^T(\,\overline{\boldsymbol{q}}\,)\,\mathbf{v}\,\right)_l = \tag{A.57b}$$

$$= \sum_{il} \varepsilon_{lji}\,\left(\mathbf{R}^T(\,\overline{\boldsymbol{q}}\,)\,\mathbf{v}\,\right)_l = \tag{A.57c}$$

$$= -\sum_{il} \varepsilon_{ijl}\,\left(\mathbf{R}^T(\,\overline{\boldsymbol{q}}\,)\,\mathbf{v}\,\right)_l = \tag{A.57d}$$

$$= \sum_{il} \varepsilon_{ilj}\,\left(\mathbf{R}^T(\,\overline{\boldsymbol{q}}\,)\,\mathbf{v}\,\right)_l \quad . \tag{A.57e}$$

After having identified the cross product in equation (A.57), it can be re-expressed in matrix form as

$$\left.\frac{d\,\mathbf{R}^T(\,\boldsymbol{q}\,)\,\mathbf{v}}{d\,\mathbf{e}^{\overline{\boldsymbol{q}}}}\right|_{\boldsymbol{q}=\overline{\boldsymbol{q}}} = \left[\,\mathbf{R}^T(\,\overline{\boldsymbol{q}}\,)\,\mathbf{v}\,\right]_\times \quad . \tag{A.58}$$

A.9. Derivation of the T-matrices

This appendix contains the derivation of the **T**-matrix (5.40) for each chart.

A.9.1. Orthographic

Our transition map (A.25) can be rewritten as

$$e_i^{\overline{\mathbf{p}}}(\mathbf{e}^{\overline{\mathbf{q}}}) \;=\; \overline{\delta}_0\; e_i^{\overline{\mathbf{q}}} \;-\; \sqrt{4 \,-\, \sum\nolimits_k (e_k^{\overline{\mathbf{q}}})^2}\;\; \overline{\boldsymbol{\delta}}_i \;-\; \sum_{lm} \varepsilon_{ilm}\; \overline{\boldsymbol{\delta}}_l\; e_m^{\overline{\mathbf{q}}} \quad , \tag{A.59}$$

being ε_{ilm} the Levi-Civita symbol. Computing (5.40) for (A.59) we obtain

$$(\mathbf{T})_{ij} \;=\; \left[\; \overline{\delta}_0\; \delta_{ij} \;-\; \frac{-\sum_k e_k^{\overline{\mathbf{q}}}\, \delta_{kj}}{\sqrt{4 \,-\, \sum_k (e_k^{\overline{\mathbf{q}}})^2}}\; \overline{\boldsymbol{\delta}}_i \;-\; \sum_{lm} \varepsilon_{ilm}\; \overline{\boldsymbol{\delta}}_l\; \delta_{mj} \;\right]_{\mathbf{e}^{\overline{\mathbf{q}}} = \overline{\mathbf{e}}^{\overline{\mathbf{q}}}} \;=\; \tag{A.60a}$$

$$\;=\; \overline{\delta}_0\; \delta_{ij} \;+\; \frac{\overline{e}_j^{\overline{\mathbf{q}}}}{\sqrt{4 \,-\, \sum_k (\overline{e}_k^{\overline{\mathbf{q}}})^2}}\; \overline{\boldsymbol{\delta}}_i \;-\; \sum_l \varepsilon_{ilj}\; \overline{\boldsymbol{\delta}}_l \quad , \tag{A.60b}$$

This expression can be rewritten in matrix form as

$$\mathbf{T} \;=\; \overline{\delta}_0\; \mathbf{I} \;+\; \frac{\overline{\boldsymbol{\delta}}\; (\overline{\mathbf{e}}^{\overline{\mathbf{q}}})^T}{\sqrt{4 \,-\, \|\overline{\mathbf{e}}^{\overline{\mathbf{q}}}\|^2}} \;-\; \left[\, \overline{\boldsymbol{\delta}} \,\right]_{\times} \quad . \tag{A.61}$$

Finally, recalling that for this chart $\overline{\delta}_0 = \sqrt{1 \,-\, \|\overline{\mathbf{e}}^{\overline{\mathbf{q}}}\|^2/4}$ and $\overline{\boldsymbol{\delta}} = \overline{\mathbf{e}}^{\overline{\mathbf{q}}}/2$, we arrive at the final expression

$$\mathbf{T} \;=\; \overline{\delta}_0\; \mathbf{I} \;+\; \frac{\overline{\boldsymbol{\delta}}\; \overline{\boldsymbol{\delta}}^T}{\overline{\delta}_0} \;-\; \left[\, \overline{\boldsymbol{\delta}} \,\right]_{\times} \quad . \tag{A.62}$$

A.9.2. Rodrigues Parameters

First, let us denote the numerator of (A.28) as $\mathbf{N}(\mathbf{e}^{\overline{\mathbf{q}}})$, and its denominator as $D(\mathbf{e}^{\overline{\mathbf{q}}})$:

$$\mathbf{N}\big(\,\mathbf{e}^{\overline{\mathbf{q}}}\,\big) \;:=\; \overline{\delta}_0\; \mathbf{e}^{\overline{\mathbf{q}}} \;-\; 2\,\overline{\boldsymbol{\delta}} \;-\; \overline{\boldsymbol{\delta}} \times \mathbf{e}^{\overline{\mathbf{q}}} \quad , \tag{A.63}$$

$$D\big(\,\mathbf{e}^{\overline{\mathbf{q}}}\,\big) \;:=\; 2\,\overline{\delta}_0 \;+\; \overline{\boldsymbol{\delta}} \cdot \mathbf{e}^{\overline{\mathbf{q}}} \quad . \tag{A.64}$$

Now let us evaluate (A.63) at $\overline{\mathbf{e}}^{\overline{\mathbf{q}}}$:

$$\mathbf{N}\big(\,\overline{\mathbf{e}}^{\overline{\mathbf{q}}}\,\big) \;=\; \overbrace{\overline{\delta}_0\; \underbrace{\overline{\mathbf{e}}^{\overline{\mathbf{q}}}}_{2\,\overline{\boldsymbol{\delta}}/\overline{\delta}_0} \;-\; 2\,\overline{\boldsymbol{\delta}}}^{\mathbf{0}} \;-\; \overbrace{\overline{\boldsymbol{\delta}} \times \overline{\mathbf{e}}^{\overline{\mathbf{q}}}}^{=\,\mathbf{0}\;\;(\,\overline{\boldsymbol{\delta}} \,\|\, \overline{\mathbf{e}}^{\overline{\mathbf{q}}}\,)} \;=\; \mathbf{0} \quad . \tag{A.65}$$

Then, the approximation of $\mathbf{N}\big(\,\mathbf{e}^{\overline{\mathbf{q}}}\,\big)$ does not have terms of order $\mathcal{O}(1)$. This means that we will only need to approximate $D\left(\mathbf{e}^{\overline{\mathbf{q}}}\right)$ to the zeroth order. Any further approximation would produce, after multiplying by the linear approximation of $\mathbf{N}\big(\,\mathbf{e}^{\overline{\mathbf{q}}}\,\big)$, a higher order term. Let us then calculate each approximation. We can rewrite (A.63) as

$$N_i\big(\mathbf{e}^{\overline{\mathbf{q}}}\big) \;=\; \overline{\delta}_0\; e_i^{\overline{\mathbf{q}}} \;-\; 2\,\overline{\boldsymbol{\delta}}_i \;-\; \sum_{kl} \varepsilon_{ikl}\; \overline{\boldsymbol{\delta}}_k\; e_l^{\overline{\mathbf{q}}} \quad , \tag{A.66}$$

with ε_{ikl} the Levi-Civita symbol. Applying (5.39) to (A.66),

$$N_i\big(\,\mathbf{e}^{\overline{\mathbf{q}}}\,\big) \;\approx\; \sum_j \left[\; \overline{\delta}_0\; \delta_{ij} \;-\; \sum_{kl} \varepsilon_{ikl}\; \overline{\boldsymbol{\delta}}_k\; \delta_{lj} \;\right]_{\mathbf{e}^{\overline{\mathbf{q}}} = \overline{\mathbf{e}}^{\overline{\mathbf{q}}}} \big(\, e_j^{\overline{\mathbf{q}}} - \overline{e}_j^{\overline{\mathbf{q}}} \,\big) \;=\; \tag{A.67a}$$

$$\;=\; \sum_j \left[\; \overline{\delta}_0\; \delta_{ij} \;-\; \sum_k \varepsilon_{ikj}\; \overline{\boldsymbol{\delta}}_k \;\right] \big(\, e_j^{\overline{\mathbf{q}}} - \overline{e}_j^{\overline{\mathbf{q}}} \,\big) \quad , \tag{A.67b}$$

being δ_{ij} the Kronecker delta. Returning to matrix notation, the linear approximation of $\mathbf{N}\left(\mathbf{e}^{\overline{\boldsymbol{q}}}\right)$ is

$$\mathbf{N}\big(\mathbf{e}^{\overline{\boldsymbol{q}}}\big) \;=\; \Big[\,\overline{\delta}_0\,\mathbf{I}\;-\;\big[\,\overline{\boldsymbol{\delta}}\,\big]_\times\,\Big]\,\big(\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\big)\;+\;\mathcal{O}\big(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|^2\big)\quad . \tag{A.68}$$

On the other hand, evaluating (A.64) at $\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}$ we obtain the zeroth order approximation:

$$D\big(\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\big) \;=\; 2\,\overline{\delta}_0\;+\;\overline{\boldsymbol{\delta}}\cdot 2\,\frac{\overline{\boldsymbol{\delta}}}{\overline{\delta}_0}\;+\;\mathcal{O}\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|\right) \;=\; \tag{A.69a}$$

$$=\; \frac{2}{\overline{\delta}_0}\,\underbrace{\left(\overline{\delta}_0^2\;+\;\|\overline{\boldsymbol{\delta}}\|^2\right)}_{1}+\mathcal{O}\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|\right) \;=\; \tag{A.69b}$$

$$=\; \frac{2}{\overline{\delta}_0}\;+\;\mathcal{O}\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|\right). \tag{A.69c}$$

Finally, combining (A.68) and (A.69c) we can compute the linear approximation of (A.28):

$$\mathbf{e}^{\overline{\boldsymbol{p}}}(\mathbf{e}^{\overline{\boldsymbol{q}}}) \;=\; 2\left\{\Big[\,\overline{\delta}_0\,\mathbf{I}\;-\;\big[\,\overline{\boldsymbol{\delta}}\,\big]_\times\,\Big]\left(\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\right)\;+\;\mathcal{O}\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|^2\right)\right\}\left[\,\frac{\overline{\delta}_0}{2}\;+\;\mathcal{O}\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|\right)\right] \;=\; \tag{A.70a}$$

$$=\; \overline{\delta}_0\,\Big[\,\overline{\delta}_0\,\mathbf{I}\;-\;\big[\,\overline{\boldsymbol{\delta}}\,\big]_\times\,\Big]\left(\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\right)\;+\;\mathcal{O}\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|^2\right)\quad . \tag{A.70b}$$

A.9.3. Modified Rodrigues Parameters

First, let us denote the numerator of (A.31) as $\mathbf{N}(\mathbf{e}^{\overline{\boldsymbol{q}}})$, and its denominator as $D(\mathbf{e}^{\overline{\boldsymbol{q}}})$:

$$\mathbf{N}\big(\mathbf{e}^{\overline{\boldsymbol{q}}}\big) \;=\; 8\,\overline{\delta}_0\;\mathbf{e}^{\overline{\boldsymbol{q}}}\;-\;\big(16\;-\;\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|^2\big)\;\overline{\boldsymbol{\delta}}\;-\;8\,\overline{\boldsymbol{\delta}}\times\mathbf{e}^{\overline{\boldsymbol{q}}}\quad , \tag{A.71}$$

$$D\big(\mathbf{e}^{\overline{\boldsymbol{q}}}\big) \;=\; 16\;+\;\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|^2\;+\;\overline{\delta}_0\;\big(16\;-\;\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|^2\big)\;+\;8\,\overline{\boldsymbol{\delta}}\cdot\mathbf{e}^{\overline{\boldsymbol{q}}}\quad . \tag{A.72}$$

Now let us evaluate (A.71) at $\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}$:

$$\mathbf{N}\big(\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\big) \;=\; 8\,\overline{\delta}_0\;\overbrace{\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}}^{4\overline{\boldsymbol{\delta}}/(1+\overline{\delta}_0)}\;-\big(16\;-\;\overbrace{\|\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|^2}^{16\|\overline{\boldsymbol{\delta}}\|^2/(1+\overline{\delta}_0)^2}\big)\;\overline{\boldsymbol{\delta}}\;-\;8\;\overbrace{\overline{\boldsymbol{\delta}}\times\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}}^{=\,\mathbf{0}\,(\,\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\parallel\overline{\boldsymbol{\delta}}\,)} \;=\; \tag{A.73a}$$

$$=\; \frac{16\,\overline{\boldsymbol{\delta}}}{1+\overline{\delta}_0}\left[\,2\,\overline{\delta}_0\;-\;\left(1\;+\;\overline{\delta}_0\;-\;\frac{\overbrace{\|\overline{\boldsymbol{\delta}}\|^2}^{1-\overline{\delta}_0^2}}{1+\overline{\delta}_0}\right)\right] \;=\; \mathbf{0}\quad . \tag{A.73b}$$

Then, as with the RP chart, the approximation of $\mathbf{N}\left(\mathbf{e}^{\overline{\boldsymbol{q}}}\right)$ does not have terms of order $\mathcal{O}(1)$, and we will only need to approximate $D\left(\mathbf{e}^{\overline{\boldsymbol{q}}}\right)$ to the zeroth order. We can write (A.71) as

$$N_i\big(\mathbf{e}^{\overline{\boldsymbol{q}}}\big) \;=\; 8\,\overline{\delta}_0\;e_i^{\overline{\boldsymbol{q}}}\;-\;\left(16-\sum_k(e_k^{\overline{\boldsymbol{q}}})^2\right)\,\overline{\boldsymbol{\delta}}_i\;-\;8\sum_{lm}\varepsilon_{ilm}\;\overline{\boldsymbol{\delta}}_l\;e_m^{\overline{\boldsymbol{q}}}\quad , \tag{A.74}$$

with ε_{ilm} the Levi-Civita symbol. Applying (5.39) to (A.74),

$$N_i\big(\mathbf{e}^{\overline{\boldsymbol{q}}}\big) \;\approx\; \sum_j\left[\,8\,\overline{\delta}_0\;\delta_{ij}\;+\;\sum_k 2\,e_k^{\overline{\boldsymbol{q}}}\;\delta_{kj}\;\overline{\boldsymbol{\delta}}_i\;-\;8\sum_{lm}\varepsilon_{ilm}\;\overline{\boldsymbol{\delta}}_l\;\delta_{mj}\right]_{\mathbf{e}^{\overline{\boldsymbol{q}}}=\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}}\left(e_j^{\overline{\boldsymbol{q}}}-\overline{e}_j^{\overline{\boldsymbol{q}}}\right) \;=\; \tag{A.75a}$$

$$=\; \sum_j\left[\,8\,\overline{\delta}_0\;\delta_{ij}\;+\;2\,\overline{e}_j^{\overline{\boldsymbol{q}}}\;\overline{\boldsymbol{\delta}}_i\;-\;8\sum_l\varepsilon_{ilj}\;\overline{\boldsymbol{\delta}}_l\right]\left(e_j^{\overline{\boldsymbol{q}}}-\overline{e}_j^{\overline{\boldsymbol{q}}}\right)\quad , \tag{A.75b}$$

being δ_{ij} the Kronecker delta. Returning to matrix notation, the linear approximation of $\mathbf{N}\big(\mathbf{e}^{\overline{\boldsymbol{q}}}\big)$ is

$$\mathbf{N}\big(\mathbf{e}^{\overline{\boldsymbol{q}}}\big) \;=\; \Big[\,8\,\overline{\delta}_0\;\mathbf{I}\;+\;2\,\overline{\boldsymbol{\delta}}\;(\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}})^T\;-\;8\,\big[\,\overline{\boldsymbol{\delta}}\,\big]_\times\,\Big]\,\big(\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\big)\;+\;\mathcal{O}\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|^2\right) \;=\; \tag{A.76a}$$

$$=\; 8\left[\,\overline{\delta}_0\;\mathbf{I}\;+\;\frac{\overline{\boldsymbol{\delta}}\;\overline{\boldsymbol{\delta}}^T}{1+\overline{\delta}_0}\;-\;\big[\,\overline{\boldsymbol{\delta}}\,\big]_\times\,\right]\big(\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\big)\;+\;\mathcal{O}\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|^2\right)\quad . \tag{A.76b}$$

On the other hand, evaluating (A.72) at $\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}$ we obtain the zeroth order approximation:

$$D\big(\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\big) \;\approx\; 16 \;+\; 16\,\frac{\|\overline{\boldsymbol{\delta}}\|^2}{(1+\overline{\delta}_0)^2} \;+\; \overline{\delta}_0\left(16 \;-\; 16\,\frac{\|\overline{\boldsymbol{\delta}}\|^2}{(1+\overline{\delta}_0)^2}\right) \;+\; 8\,\overline{\boldsymbol{\delta}}\cdot 4\,\frac{\overline{\boldsymbol{\delta}}}{1+\overline{\delta}_0} \;=\; \tag{A.77a}$$

$$=\; \frac{16}{1+\overline{\delta}_0}\left[(1+\overline{\delta}_0) \;+\; \frac{\|\overline{\boldsymbol{\delta}}\|^2}{1+\overline{\delta}_0} \;+\; \overline{\delta}_0\left((1+\overline{\delta}_0) \;-\; \frac{\|\overline{\boldsymbol{\delta}}\|^2}{1+\overline{\delta}_0}\right) \;+\; 2\,\|\overline{\boldsymbol{\delta}}\|^2\right] \;=\; \tag{A.77b}$$

$$=\; \frac{16}{1+\overline{\delta}_0}\left[2 \;+\; \overline{\delta}_0\left(2\,\overline{\delta}_0\right) \;+\; 2\left(1-\overline{\delta}_0^2\right)\right] \;=\; \frac{64}{1+\overline{\delta}_0}\quad, \tag{A.77c}$$

where we have used the equality $\|\overline{\boldsymbol{\delta}}\|^2 = 1-\overline{\delta}_0^2$ for unit quaternions. Finally, combining (A.76b) and (A.77c) we can compute the linear approximation of (A.31):

$$\mathbf{e}^{\overline{\boldsymbol{p}}}\big(\mathbf{e}^{\overline{\boldsymbol{q}}}\big) \;=\; 4\left\{8\left[\overline{\delta}_0\,\mathbf{I} \;+\; \frac{\overline{\boldsymbol{\delta}}\,\overline{\boldsymbol{\delta}}^T}{1+\overline{\delta}_0} \;-\; \big[\,\overline{\boldsymbol{\delta}}\,\big]_\times\right]\big(\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\big) \;+\; \mathcal{O}\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|^2\right)\right\}\left[\frac{1+\overline{\delta}_0}{64} \;+\; \mathcal{O}\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|\right)\right] = \tag{A.78a}$$

$$=\; \frac{1+\overline{\delta}_0}{2}\left[\overline{\delta}_0\,\mathbf{I} \;+\; \frac{\overline{\boldsymbol{\delta}}\,\overline{\boldsymbol{\delta}}^T}{1+\overline{\delta}_0} \;-\; \big[\,\overline{\boldsymbol{\delta}}\,\big]_\times\right]\big(\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\big) \;+\; \mathcal{O}\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|^2\right) \;=\; \tag{A.78b}$$

$$=\; \frac{1}{2}\Big[\left(1+\overline{\delta}_0\right)\left(\overline{\delta}_0\,\mathbf{I} \;-\; \big[\,\overline{\boldsymbol{\delta}}\,\big]_\times\right) \;+\; \overline{\boldsymbol{\delta}}\;\overline{\boldsymbol{\delta}}^T\Big]\big(\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\big) \;+\; \mathcal{O}\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|^2\right)\quad. \tag{A.78c}$$

A.9.4. Rotation Vector

Let us start evaluating the vector $\boldsymbol{\delta}^{\overline{\boldsymbol{p}}}$ in (A.34) and (A.35) at the point $\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}$:

$$\boldsymbol{\delta}^{\overline{\boldsymbol{p}}}\big(\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\big) \;=\; \overbrace{\overline{\delta}_0\,\underbrace{\widehat{\overline{\mathbf{e}}}^{\overline{\boldsymbol{q}}}\,\sin\left(\frac{\|\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|}{2}\right)}_{\overline{\boldsymbol{\delta}}} \;-\; \underbrace{\cos\left(\frac{\|\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|}{2}\right)}_{\overline{\delta}_0}\,\overline{\boldsymbol{\delta}}}^{=\,\mathbf{0}} \;-\; \overbrace{\overline{\boldsymbol{\delta}}\times\underbrace{\widehat{\overline{\mathbf{e}}}^{\overline{\boldsymbol{q}}}\,\sin\left(\frac{\|\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|}{2}\right)}_{\overline{\boldsymbol{\delta}}}}^{=\,\mathbf{0}} \;=\; \tag{A.79}$$

$$=\; \mathbf{0}\quad. \tag{A.80}$$

Then, the first order approximation of $\boldsymbol{\delta}^{\overline{\boldsymbol{p}}}$ around $\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}$ will have the form

$$\boldsymbol{\delta}^{\overline{\boldsymbol{p}}} \;=\; \widetilde{\mathbf{T}}\,\big(\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\big) \;+\; \mathcal{O}\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|^2\right)\quad, \tag{A.81}$$

and $\|\boldsymbol{\delta}^{\overline{\boldsymbol{p}}}\|\to 0$ as $\mathbf{e}^{\overline{\boldsymbol{q}}}\to\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}$. Taking the Taylor series of the $\arcsin x$,

$$\frac{\arcsin\|\boldsymbol{\delta}^{\overline{\boldsymbol{p}}}\|}{\|\boldsymbol{\delta}^{\overline{\boldsymbol{p}}}\|} \;=\; \frac{\|\boldsymbol{\delta}^{\overline{\boldsymbol{p}}}\| \;+\; \mathcal{O}\big(\|\boldsymbol{\delta}^{\overline{\boldsymbol{p}}}\|^3\big)}{\|\boldsymbol{\delta}^{\overline{\boldsymbol{p}}}\|} \;=\; \tag{A.82a}$$

$$=\; 1 \;+\; \mathcal{O}\left(\|\boldsymbol{\delta}^{\overline{\boldsymbol{p}}}\|^2\right) \;=\; \tag{A.82b}$$

$$=\; 1 \;+\; \mathcal{O}\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|^2\right)\quad, \tag{A.82c}$$

so that (A.34) is linearized as

$$2\,\frac{\boldsymbol{\delta}^{\overline{\boldsymbol{p}}}}{\|\boldsymbol{\delta}^{\overline{\boldsymbol{p}}}\|}\arcsin\|\boldsymbol{\delta}^{\overline{\boldsymbol{p}}}\| \;=\; 2\,\widetilde{\mathbf{T}}\,\big(\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\big) \;+\; \mathcal{O}\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|^2\right)\quad. \tag{A.83}$$

We only lack the $\widetilde{\mathbf{T}}$ matrix. We will need the linear approximations of $\cos\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|/2\right)$ and $\widehat{\mathbf{e}}^{\overline{\boldsymbol{q}}}\sin\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|/2\right)$ around $\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}$. To this end we will first obtain the linear approximation of $\|\mathbf{x}\|$:

$$\|\mathbf{x}\| = \sqrt{\sum\nolimits_k x_k^2} = \tag{A.84a}$$

$$= \|\overline{\mathbf{x}}\| + \sum_j \left[\frac{\sum_k x_k\,\delta_{kj}}{\sqrt{\sum_k x_k^2}}\right]_{\mathbf{x}=\overline{\mathbf{x}}} (x_j-\overline{x}_j) + \mathcal{O}\left(\|\mathbf{x}-\overline{\mathbf{x}}\|^2\right) = \tag{A.84b}$$

$$= \|\overline{\mathbf{x}}\| + \sum_j \left[\frac{\overline{x}_j}{\sqrt{\sum_k \overline{x}_k^2}}\right] (x_j-\overline{x}_j) + \mathcal{O}\left(\|\mathbf{x}-\overline{\mathbf{x}}\|^2\right) = \tag{A.84c}$$

$$= \|\overline{\mathbf{x}}\| + \widehat{\overline{\mathbf{x}}}^T\left(\mathbf{x}-\overline{\mathbf{x}}\right) + \mathcal{O}\left(\|\mathbf{x}-\overline{\mathbf{x}}\|^2\right) \quad . \tag{A.84d}$$

Noticing that

$$\frac{\partial\,\|\mathbf{x}\|}{\partial\,\mathbf{x}} = \widehat{\overline{\mathbf{x}}}^T + \mathcal{O}\big(\|\mathbf{x}-\overline{\mathbf{x}}\|\big) \quad , \tag{A.85}$$

our computations are straightforward:

$$\cos\left(\frac{\|\mathbf{x}\|}{2}\right) = \cos\left(\frac{\|\overline{\mathbf{x}}\|}{2}\right) - \left[\sin\left(\frac{\|\mathbf{x}\|}{2}\right)\frac{1}{2}\left[\widehat{\overline{\mathbf{x}}}^T + \mathcal{O}\left(\|\mathbf{x}-\overline{\mathbf{x}}\|\right)\right]\right]_{\mathbf{x}=\overline{\mathbf{x}}} (\mathbf{x}-\overline{\mathbf{x}}) + \mathcal{O}\left(\|\mathbf{x}-\overline{\mathbf{x}}\|^2\right) = \tag{A.86a}$$

$$= \cos\left(\frac{\|\overline{\mathbf{x}}\|}{2}\right) - \frac{1}{2}\sin\left(\frac{\|\overline{\mathbf{x}}\|}{2}\right)\widehat{\overline{\mathbf{x}}}^T(\mathbf{x}-\overline{\mathbf{x}}) + \mathcal{O}\left(\|\mathbf{x}-\overline{\mathbf{x}}\|^2\right) \quad . \tag{A.86b}$$

For our particular case,

$$\cos\left(\frac{\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|}{2}\right) = \overline{\delta}_0 - \frac{1}{2}\,\overline{\boldsymbol{\delta}}^T\left(\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\right) + \mathcal{O}\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|^2\right) \quad . \tag{A.87}$$

On the other hand,

$$\frac{\sin\left(\frac{\|\mathbf{x}\|}{2}\right)}{\|\mathbf{x}\|} = \frac{\sin\left(\frac{\|\overline{\mathbf{x}}\|}{2}\right)}{\|\overline{\mathbf{x}}\|} + \\ + \left[\left(\frac{\cos\left(\frac{\|\mathbf{x}\|}{2}\right)}{\|\mathbf{x}\|}\frac{1}{2} - \frac{\sin\left(\frac{\|\mathbf{x}\|}{2}\right)}{\|\mathbf{x}\|^2}\right)\left[\widehat{\overline{\mathbf{x}}}^T + \mathcal{O}\left(\|\mathbf{x}-\overline{\mathbf{x}}\|\right)\right]\right]_{\mathbf{x}=\overline{\mathbf{x}}} (\mathbf{x}-\overline{\mathbf{x}}) + \mathcal{O}\left(\|\mathbf{x}-\overline{\mathbf{x}}\|^2\right) = \tag{A.88a}$$

$$= \frac{\sin\left(\frac{\|\overline{\mathbf{x}}\|}{2}\right)}{\|\overline{\mathbf{x}}\|} + \left[\left(\frac{\cos\left(\frac{\|\overline{\mathbf{x}}\|}{2}\right)}{\|\overline{\mathbf{x}}\|}\frac{1}{2} - \frac{\sin\left(\frac{\|\overline{\mathbf{x}}\|}{2}\right)}{\|\overline{\mathbf{x}}\|^2}\right)\widehat{\overline{\mathbf{x}}}^T\right](\mathbf{x}-\overline{\mathbf{x}}) + \mathcal{O}\left(\|\mathbf{x}-\overline{\mathbf{x}}\|^2\right) \quad . \tag{A.88b}$$

Now, taking $\mathbf{x} = \overline{\mathbf{x}} + (\mathbf{x}-\overline{\mathbf{x}})$ we arrive at

$$\frac{\mathbf{x}}{\|\mathbf{x}\|}\sin\left(\frac{\|\mathbf{x}\|}{2}\right) = [\overline{\mathbf{x}} + (\mathbf{x}-\overline{\mathbf{x}})]\frac{\sin\left(\frac{\|\mathbf{x}\|}{2}\right)}{\|\mathbf{x}\|} = \tag{A.89a}$$

$$= \widehat{\overline{\mathbf{x}}}\sin\left(\frac{\|\overline{\mathbf{x}}\|}{2}\right) + \\ + \left[\frac{\sin\left(\frac{\|\overline{\mathbf{x}}\|}{2}\right)}{\|\overline{\mathbf{x}}\|} + \frac{1}{2}\cos\left(\frac{\|\overline{\mathbf{x}}\|}{2}\right)\widehat{\overline{\mathbf{x}}}\widehat{\overline{\mathbf{x}}}^T - \frac{\sin\left(\frac{\|\overline{\mathbf{x}}\|}{2}\right)}{\|\overline{\mathbf{x}}\|}\widehat{\overline{\mathbf{x}}}\widehat{\overline{\mathbf{x}}}^T\right](\mathbf{x}-\overline{\mathbf{x}}) + \mathcal{O}\left(\|\mathbf{x}-\overline{\mathbf{x}}\|^2\right) \quad . \tag{A.89b}$$

For our particular case,

$$\frac{\mathbf{e}^{\overline{\boldsymbol{q}}}}{\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|}\sin\left(\frac{\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|}{2}\right) = \overline{\boldsymbol{\delta}} + \\ + \left[\frac{\|\overline{\boldsymbol{\delta}}\|}{2\arcsin\|\overline{\boldsymbol{\delta}}\|} + \frac{\overline{\delta}_0}{2}\widehat{\overline{\boldsymbol{\delta}}}\widehat{\overline{\boldsymbol{\delta}}}^T - \frac{\|\overline{\boldsymbol{\delta}}\|}{2\arcsin\|\overline{\boldsymbol{\delta}}\|}\widehat{\overline{\boldsymbol{\delta}}}\widehat{\overline{\boldsymbol{\delta}}}^T\right]\left(\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\right) + \mathcal{O}\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|^2\right) = \tag{A.90a}$$

$$= \overline{\boldsymbol{\delta}} + \frac{1}{2}\left[\left(\mathbf{I} - \widehat{\overline{\boldsymbol{\delta}}}\widehat{\overline{\boldsymbol{\delta}}}^T\right)\frac{\|\overline{\boldsymbol{\delta}}\|}{\arcsin\|\overline{\boldsymbol{\delta}}\|} + \overline{\delta}_0\,\widehat{\overline{\boldsymbol{\delta}}}\widehat{\overline{\boldsymbol{\delta}}}^T\right]\left(\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\right) + \mathcal{O}\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|^2\right) \quad . \tag{A.90b}$$

Finally, we just have to replace (A.87) and (A.90b) in (A.35) to obtain the required linear approximation. Returning to the original notation we have

$$2\,\frac{\boldsymbol{\delta}^{\overline{\boldsymbol{p}}}}{\|\boldsymbol{\delta}^{\overline{\boldsymbol{p}}}\|}\arcsin\|\boldsymbol{\delta}^{\overline{\boldsymbol{p}}}\| \;=\; 2\,\boldsymbol{\delta}^{\overline{\boldsymbol{p}}} \;+\; \mathcal{O}\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|^2\right) \;= \tag{A.91a}$$

$$= \;2\,\overline{\delta}_0\,\widehat{\mathbf{e}}^{\overline{\boldsymbol{q}}}\sin\left(\frac{\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|}{2}\right) \;-\; 2\cos\left(\frac{\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|}{2}\right)\overline{\boldsymbol{\delta}} \;-\; 2\,\overline{\boldsymbol{\delta}}\times\widehat{\mathbf{e}}^{\overline{\boldsymbol{q}}}\sin\left(\frac{\|\mathbf{e}^{\overline{\boldsymbol{q}}}\|}{2}\right) \;+\; \mathcal{O}\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|^2\right) \;= \tag{A.91b}$$

$$\begin{aligned}= \;& 2\,\overline{\delta}_0\,\overline{\boldsymbol{\delta}} \;+\; \overline{\delta}_0\left[\left(\mathbf{I}-\widehat{\overline{\boldsymbol{\delta}}}\,\widehat{\overline{\boldsymbol{\delta}}}^{T}\right)\frac{\|\overline{\boldsymbol{\delta}}\|}{\arcsin\|\overline{\boldsymbol{\delta}}\|} \;+\; \overline{\delta}_0\,\widehat{\overline{\boldsymbol{\delta}}}\,\widehat{\overline{\boldsymbol{\delta}}}^{T}\right]\left(\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\right) \;-\; \left(2\,\overline{\delta}_0 \;-\; \overline{\boldsymbol{\delta}}^{T}\left(\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\right)\right)\overline{\boldsymbol{\delta}} \;+ \\ & -\; \overline{\boldsymbol{\delta}}\times\left\{2\,\overline{\boldsymbol{\delta}} \;+\; \left[\left(\mathbf{I}-\widehat{\overline{\boldsymbol{\delta}}}\,\widehat{\overline{\boldsymbol{\delta}}}^{T}\right)\frac{\|\overline{\boldsymbol{\delta}}\|}{\arcsin\|\overline{\boldsymbol{\delta}}\|} \;+\; \overline{\delta}_0\,\widehat{\overline{\boldsymbol{\delta}}}\,\widehat{\overline{\boldsymbol{\delta}}}^{T}\right]\left(\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\right)\right\} \;+\; \mathcal{O}\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|^2\right) \;=\end{aligned} \tag{A.91c}$$

$$= \;\left[\left(\overline{\delta}_0\left(\mathbf{I}-\widehat{\overline{\boldsymbol{\delta}}}\,\widehat{\overline{\boldsymbol{\delta}}}^{T}\right) \;-\; \left[\,\overline{\boldsymbol{\delta}}\,\right]_{\times}\right)\frac{\|\overline{\boldsymbol{\delta}}\|}{\arcsin\|\overline{\boldsymbol{\delta}}\|} \;+\; \overline{\delta}_0^{2}\,\widehat{\overline{\boldsymbol{\delta}}}\,\widehat{\overline{\boldsymbol{\delta}}}^{T} \;+\; \overline{\boldsymbol{\delta}}\,\overline{\boldsymbol{\delta}}^{T}\right]\left(\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\right) \;+\; \mathcal{O}\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|^2\right) \;= \tag{A.91d}$$

$$= \;\left[\left(\overline{\delta}_0\left(\mathbf{I}-\widehat{\overline{\boldsymbol{\delta}}}\,\widehat{\overline{\boldsymbol{\delta}}}^{T}\right) \;-\; \left[\,\overline{\boldsymbol{\delta}}\,\right]_{\times}\right)\frac{\|\overline{\boldsymbol{\delta}}\|}{\arcsin\|\overline{\boldsymbol{\delta}}\|} \;+\; \widehat{\overline{\boldsymbol{\delta}}}\,\widehat{\overline{\boldsymbol{\delta}}}^{T}\right]\left(\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\right) \;+\; \mathcal{O}\left(\|\mathbf{e}^{\overline{\boldsymbol{q}}}-\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}\|^2\right) \quad . \tag{A.91e}$$

Note that the linear approximations of our transition maps are valid for $\mathbf{e}^{\overline{\boldsymbol{q}}}$ near of $\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}$. However, we have not made any assumption about the $\overline{\boldsymbol{\delta}}$ quaternion. This means that our linear approximations are exact for any $\overline{\boldsymbol{\delta}}=\varphi^{-1}(\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}})$ in the domain of each **T**-matrix, provided that $\mathbf{e}^{\overline{\boldsymbol{q}}}$ is close enough to $\overline{\mathbf{e}}^{\overline{\boldsymbol{q}}}$ (the distribution is not too scattered).

A.10. Levenberg-Marquardt Algorithm

The *Levenberg-Marquardt algorithm* is an iterative method used to solve non-linear least squares problems. In particular, given a set of M data pairs $\mathcal{D} = \{ (\mathbf{x}_m , y_m) \}_{m=1}^{M}$, it finds the function $f(\mathbf{x}, \boldsymbol{\theta})$ parametrized with the vector $\boldsymbol{\theta}$, that best models the set $\mathcal{D}$. To that end, the following cost function is defined:

$$F(\boldsymbol{\theta}) \quad = \quad \sum_{m=1}^{M} \big(y_m - f(\mathbf{x}_m , \boldsymbol{\theta}) \big)^2 \quad . \tag{A.92}$$

Given an approximate solution $\boldsymbol{\theta}^{(k)}$, we look for a better approximation through

$$\boldsymbol{\theta}^{(k+1)} \quad = \quad \boldsymbol{\theta}^{(k)} + \boldsymbol{\delta}^{(k)} \quad . \tag{A.93}$$

If $\boldsymbol{\delta} \approx \mathbf{0}$, then the function $f(\mathbf{x}_m , \boldsymbol{\theta} + \boldsymbol{\delta})$ can be approximated by

$$f(\mathbf{x}_m , \boldsymbol{\theta} + \boldsymbol{\delta}) \quad \approx \quad f(\mathbf{x}_m , \boldsymbol{\theta}) + \mathbf{J}\, \boldsymbol{\delta} \quad , \tag{A.94}$$

with

$$J_{mi} \quad = \quad \frac{\partial\, f(\mathbf{x}_m , \boldsymbol{\theta})}{\partial\, \theta_i} \quad . \tag{A.95}$$

Introducing (A.94) into (A.92) results in

$$F(\boldsymbol{\theta} + \boldsymbol{\delta}) \quad \approx \quad \sum_{m=1}^{M} \big(y_m - f(\mathbf{x}_m , \boldsymbol{\theta}) - \mathbf{J}\, \boldsymbol{\delta} \big)^2 \quad \equiv \tag{A.96a}$$

$$\equiv \quad \sum_{m=1}^{M} \Big(\big(y_m - f(\mathbf{x}_m , \boldsymbol{\theta}) \big) - \sum_i J_{mi}\, \delta_i \Big)^2 \quad = \tag{A.96b}$$

$$= \quad \sum_{m=1}^{M} \Big(\big(y_m - f(\mathbf{x}_m , \boldsymbol{\theta}) \big)^2 + \\ - 2 \big(y_m - f(\mathbf{x}_m , \boldsymbol{\theta}) \big) \sum_i J_{mi}\, \delta_i + \\ + \sum_{ij} J_{mi}\, J_{mj}\, \delta_i\, \delta_j \Big) \quad . \tag{A.96c}$$

And looking for the minimum condition:

$$\frac{d\, F(\boldsymbol{\theta} + \boldsymbol{\delta})}{d\, \delta_\alpha} \quad = \quad 0 \quad \approx \tag{A.97a}$$

$$\approx \quad \sum_{m=1}^{M} \Big(- 2 \big(y_m - f(\mathbf{x}_m , \boldsymbol{\theta}) \big) \sum_i J_{mi}\, \delta_{i\alpha} + \\ + \sum_{ij} J_{mi}\, J_{mj} \big(\delta_{i\alpha}\, \delta_j + \delta_i\, \delta_{j\alpha} \big) \Big) \quad = \tag{A.97b}$$

$$= \quad \sum_{m=1}^{M} \Big(- 2 \big(y_m - f(\mathbf{x}_m , \boldsymbol{\theta}) \big)\, J_{m\alpha} + \\ + 2 \sum_i J_{m\alpha}\, J_{mi}\, \delta_i \Big) \quad \equiv \tag{A.97c}$$

$$\equiv \quad -2\, \mathbf{J}^T \big(\mathbf{y} - \mathbf{f}(\boldsymbol{\theta}) \big) + 2\, \mathbf{J}^T\, \mathbf{J}\, \boldsymbol{\delta} \quad , \tag{A.97d}$$

where δ_{ij} is the Kronecker delta, $\mathbf{y} = (y_1 , \ldots , y_M)^T$, and $\mathbf{f}(\boldsymbol{\theta}) = \big(f(\mathbf{x}_1 , \boldsymbol{\theta}) , \ldots , f(\mathbf{x}_M , \boldsymbol{\theta}) \big)^T$. From (A.97) it results that

$$\mathbf{J}^T\, \mathbf{J}\, \boldsymbol{\delta} \quad = \quad \mathbf{J}^T \big(\mathbf{y} - \mathbf{f}(\boldsymbol{\theta}) \big) \quad . \tag{A.98}$$

Then, in each iteration, the solution will progressively approach the optimal value through:

$$\mathbf{y} = (y_1, \ldots, y_M)^T \quad , \tag{A.99}$$

$$\mathbf{f}(\boldsymbol{\theta}^{(k)}) = \left(f(\mathbf{x}_1, \boldsymbol{\theta}^{(k)}), \ldots, f(\mathbf{x}_M, \boldsymbol{\theta}^{(k)})\right)^T \quad , \tag{A.100}$$

$$\mathbf{J}^{(k)} = \left.\frac{\partial \mathbf{f}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(k)}} \quad , \tag{A.101}$$

$$\boldsymbol{\delta}^{(k)} = \left[\mathbf{J}^T \mathbf{J}\right]^{-1} \mathbf{J}^T \left(\mathbf{y} - \mathbf{f}(\boldsymbol{\theta}^{(k)})\right) \quad , \tag{A.102}$$

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \boldsymbol{\delta}^{(k)} \quad . \tag{A.103}$$

There are other versions of the algorithm. Take for example the damped version, where (A.98) is replaced by

$$\left(\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I}\right) \boldsymbol{\delta} = \mathbf{J}^T \left(\mathbf{y} - \mathbf{f}(\boldsymbol{\theta})\right) \quad . \tag{A.104}$$

If λ is small, (A.104) is similar to (A.98). On the other hand, if λ is large, the step $\boldsymbol{\delta}$ is taken on the gradient direction $\mathbf{J}^T \left(\mathbf{y} - \mathbf{f}(\boldsymbol{\theta})\right)$.

Another second version uses

$$\left(\mathbf{J}^T \mathbf{J} + \lambda \, \mathrm{diag}(\mathbf{J}^T \mathbf{J})\right) \boldsymbol{\delta} = \mathbf{J}^T \left(\mathbf{y} - \mathbf{f}(\boldsymbol{\theta})\right) \quad , \tag{A.105}$$

which, for large λ values, scales each component of the gradient according to the curvature. Then, the step takes larger values in the directions where the gradient is smaller, which prevents slow convergence in those directions, and makes the algorithm scale invariant.

A.11. Computation of $\mathbf{J}^T\,\mathbf{J}$ and $\mathbf{J}^T\,[\,\mathbf{z}-\mathbf{f}\,]$

We start from:

$$f_m(\,\mathbf{A}\,) = \|\,\mathbf{A}(T_m)\;\widetilde{\mathbf{x}}_m\,\|^2 = \tag{A.106a}$$

$$= \sum_i \Big[\sum_j A_{ij}(T_m)\;\widetilde{x}_{mj}\Big]^2 = \tag{A.106b}$$

$$= \sum_i \Big[\sum_j A_{ij}(T_m)\;\widetilde{x}_{mj}\Big]\Big[\sum_k A_{ik}(T_m)\;\widetilde{x}_{mk}\Big] = \tag{A.106c}$$

$$= \sum_{ijk} A_{ij}(T_m)\;A_{ik}(T_m)\;\widetilde{x}_{mj}\;\widetilde{x}_{mk} = \tag{A.106d}$$

$$= \sum_{ijkln} A^{(n)}_{ij}\;A^{(l)}_{ik}\;\widetilde{x}_{mj}\;\widetilde{x}_{mk}\;T_m^{n+l} \quad . \tag{A.106e}$$

The derivatives with respect to the parameters are:

$$\frac{\partial\,f_m}{\partial A^{(\gamma)}_{\alpha\beta}} = \sum_{ijkln}\Big(\delta_{i\alpha}\;\delta_{j\beta}\;\delta_{n\gamma}\;A^{(l)}_{ik} + A^{(n)}_{ij}\;\delta_{i\alpha}\;\delta_{k\beta}\;\delta_{l\gamma}\Big)\;\widetilde{x}_{mj}\;\widetilde{x}_{mk}\;T_m^{n+l} = \tag{A.107a}$$

$$= \sum_{kl} A^{(l)}_{\alpha k}\;\widetilde{x}_{m\beta}\;\widetilde{x}_{mk}\;T_m^{\gamma+l} + \sum_{jn} A^{(n)}_{\alpha j}\;\widetilde{x}_{mj}\;\widetilde{x}_{m\beta}\;T_m^{n+\gamma} = \tag{A.107b}$$

$$= 2\sum_{jn} A^{(n)}_{\alpha j}\;\widetilde{x}_{mj}\;\widetilde{x}_{m\beta}\;T_m^{n+\gamma} \quad . \tag{A.107c}$$

Now we can compute the terms

$$(\mathbf{J}^T\;\mathbf{J})_{ij} = \sum_m J^T_{im}\;J_{mj} = \tag{A.108a}$$

$$= \sum_m J_{mi}\;J_{mj} \quad , \tag{A.108b}$$

$$\big(\,\mathbf{J}^T\,[\,\mathbf{z}-\mathbf{f}\,]\,\big)_i = \sum_m J^T_{im}\,[\,z_m - f_m\,] = \tag{A.109a}$$

$$= \sum_m J_{mi}\,[\,z_m - f_m\,] \quad , \tag{A.109b}$$

using the definitions (A.99), (A.100), (A.101) and (6.12):

$$\sum_m \frac{\partial\,f_m}{\partial A^{(\gamma)}_{\alpha\beta}}\,\frac{\partial\,f_m}{\partial A^{(\gamma')}_{\alpha'\beta'}} = 4\sum_{jnj'n'} A^{(n)}_{\alpha j}\;A^{(n')}_{\alpha' j'}\;X^{n+\gamma+n'+\gamma'}_{j\beta j'\beta'} \quad . \tag{A.110}$$

$$\sum_m \frac{\partial\,f_m}{\partial A^{(\gamma)}_{\alpha\beta}}\,[y_m^2 - f_m] = 2\sum_{n'}\Big[\sum_{j'} A^{(n')}_{\alpha j'}\,Y^{n'+\gamma}_{j'\beta} + - \sum_{ijklnj'} A^{(n')}_{\alpha j'}\;A^{(n)}_{ij}\;A^{(l)}_{ik}\;X^{n'+\gamma+n+l}_{j'\beta jk}\Big] \quad . \tag{A.111}$$

A.12. Triaxial Calibration Boards

This appendix contains the schematics and the real-size board designs for the two versions of the Triaxial Calibration Boards.

A.12.1. Triaxial Calibration Board v1

This appendix contains the schematics A.1, and the real-size board design A.2 for the first version of the Triaxial Calibration Board.

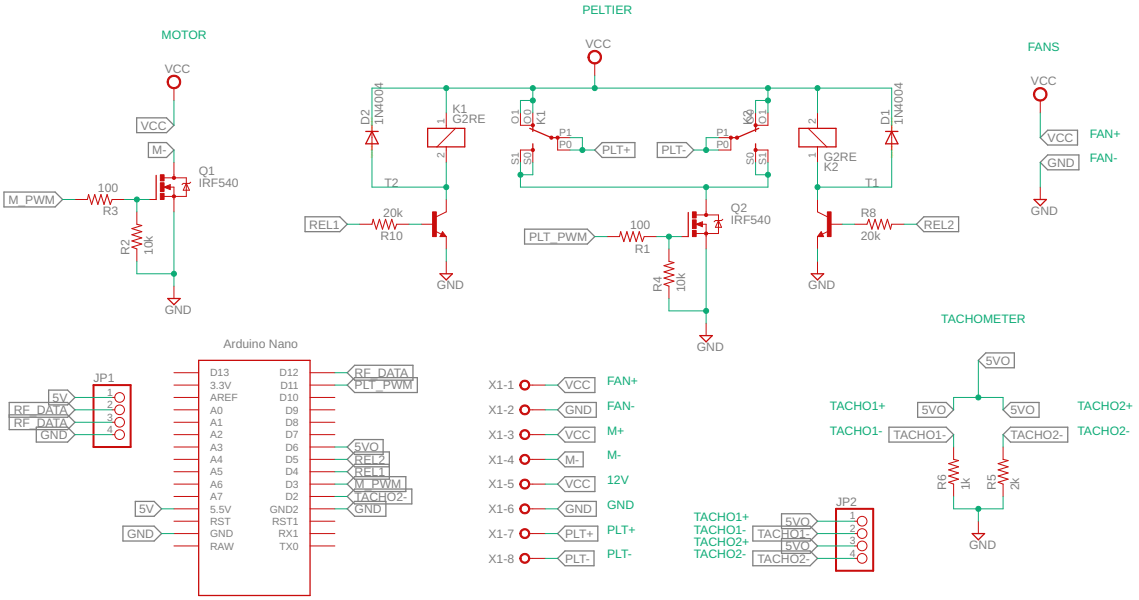


Figure A.1: Schematics for the first version of the TCB.

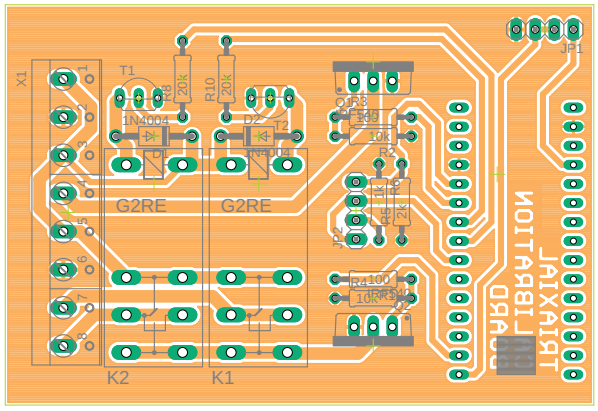


Figure A.2: Board design for the first version of the TCB (real size).

A.12.2. Triaxial Calibration Board v2

This appendix contains the schematics A.3, and the real-size board design A.4 for the second version of the Triaxial Calibration Board.

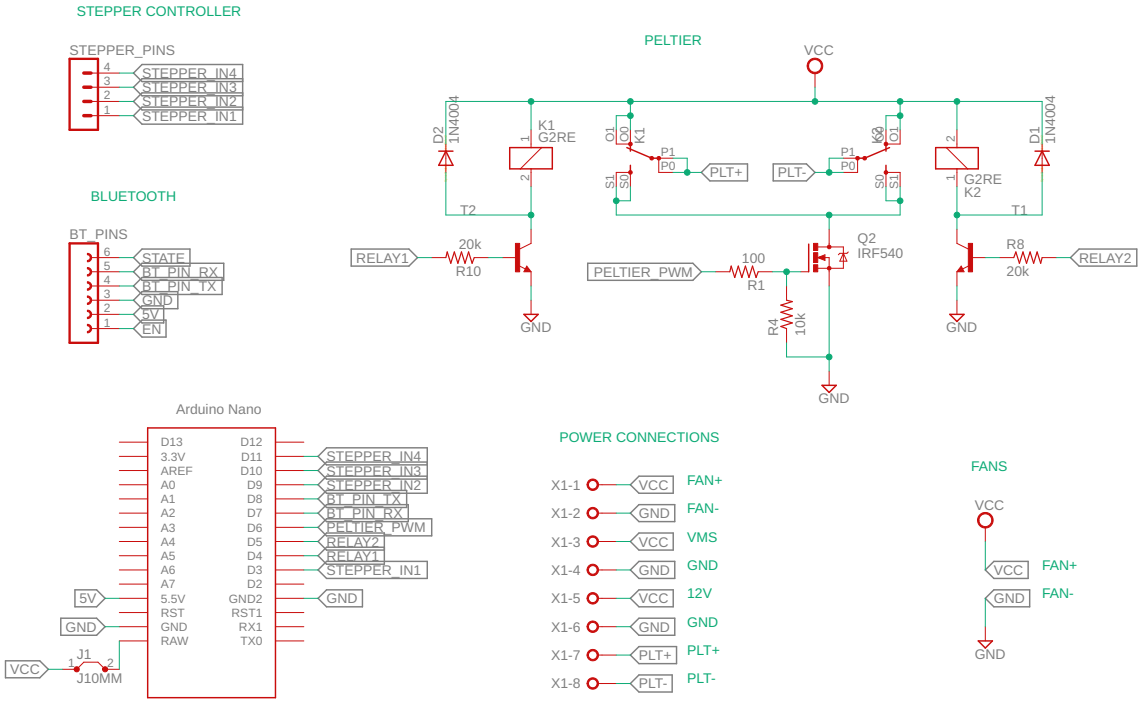


Figure A.3: Schematics for the second version of the TCB.

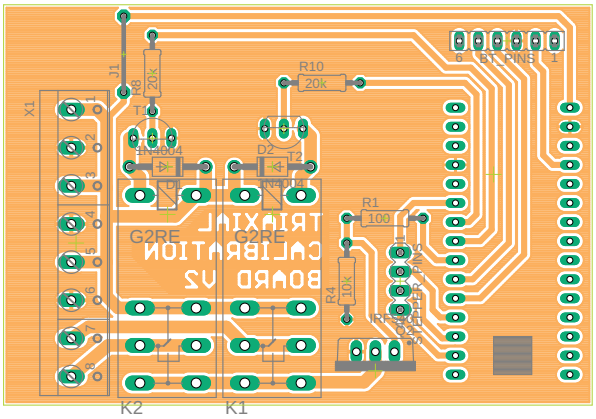


Figure A.4: Board design for the second version of the TCB (real size).

A.13. Comparison of Triaxial Sensors

This appendix contains a comparison of characteristics of the triaxial sensors used (calibrated or not) during the development of this thesis. We begin by reviewing the terminology commonly used in the datasheets. Then, we present Table A.2 that contains a brief description of the IMUs used during the development of this work, and then we provide 4 more tables that compare the main characteristics provided in the datasheets, and some others measured experimentally (accelerometers in Table A.3, gyroscopes in Table A.4, magnetometers in Table A.5, and built-in temperature sensors in Table A.6).

A.13.1. Terminology

This section reviews the terminology commonly used in datasheets of triaxial sensors.

Range: the *range* is the set of values that the sensor can measure before saturating. The range is usually configurable. The more configurations available, the more freedom there is to choose the saturation value and the resolution of the sensor.

Resolution: the *resolution* of a sensor is the smallest measurable increment in the measured quantity. We say that a sensor has high resolution when it can distinguish small changes in the measurements. It is expressed in the same units as the sensor measurements, but it is often preferable to express it in bits (those of the ADC). For example, an accelerometer measuring in a range of $\pm 16\,\mathrm{g}$ with an ADC of 16 bits would be able to measure differences of $\big(+16\,\mathrm{g} - (-16\,\mathrm{g})\big)/2^{16} = 16\,\mathrm{g}/2^{15} \approx 0.5\,\mathrm{mg}$. The more bits the better the resolution.

Raw noise: even if a sensor has a high resolution, there is often a threshold caused by electrical noise in the measurements. The *raw noise* of a sensor is the standard deviation of its measurements collected in a static situation:

$$\sigma \;=\; \sqrt{\frac{1}{N}\sum_{n=1}^{N}(x_n - \overline{x})^2} \quad , \qquad \text{with} \quad \overline{x} \;=\; \frac{1}{N}\sum_{n=1}^{N} x_n \quad . \tag{A.112}$$

ODR: the *Output Data Rate* is the rate at which the sensor samples the values being measured. It is measured in Hertz (Hz). The ODR is usually configurable, but as shown in Section 5.5.2.3, a higher ODR (higher f_{update}) will be preferable to a lower one.

Noise SD: in case the ODR is higher than the frequency at which we want to take measurements, the sensor can take advantage of such excess to reduce its noise by averaging the samples. For example, if the ODR is $1000\,\mathrm{Hz}$ and we want to take measurements at $100\,\mathrm{Hz}$, the sensor could average $N = 1000/100 = 10$ samples. The variance of that average would be related to the variance of the samples (raw noise) through $\sigma_N^2 = \frac{\sigma^2}{N}$. Then, the new noise would have a standard deviation $\sigma_N = \frac{\sigma}{\sqrt{\text{ODR}}}\sqrt{f}$ with f the new sampling frequency. The *Noise Spectral Density* is defined as the term $\frac{\sigma}{\sqrt{\text{ODR}}}$ which is a measure to compare the noise of sensors with different ODR at the same sampling frequency.

Sensitivity: the *sensitivity* measures how the measured quantity changes per each LSB (Least Significant Bit). It is used to transform the raw measurements (usually expressed in the range $\pm 2^{15}$ when the sensor has a resolution of 9 to 16 bits) into the measurements expressed in their actual units.

Sensitivity tolerance: due to manufacturing limitations, the sensitivity provided by the manufacturer may be different from that of the sensor. The *sensitivity tolerance* is a measure of the relative difference between the sensitivity of the sensor and the sensitivity provided by the manufacturer, and is expressed as a percentage:

$$ST \;=\; \frac{|s_S - s_M|}{s_M}\,100 \quad , \tag{A.113}$$

where s_S is the sensor sensitivity, and s_M is the sensitivity provided by the manufacturer.

Sensitivity change with temperature: the sensitivity of a sensor could change as a function of its temperature. The *sensitivity change with temperature* is a measure of the magnitude of such a change, and is expressed as a percentage:

$$SCT \quad = \quad \max_T \frac{|s_S(T) - s_M|}{s_M} \; 100 \quad , \tag{A.114}$$

where $s_S(T)$ is the sensitivity of the sensor as a function of the operating temperature, and s_M is the sensitivity provided by the manufacturer.

Nonlinearity: the sensitivity of a sensor could change as a function of the magnitude of the measured quantity: when one plots the raw measurements versus the measurements that should be measured, the relation is not usually perfectly linear. The *nonlinearity* is a measure of the magnitude of such a change, and is expressed as a percentage:

$$NL \quad = \quad \max_V \frac{|s(V) - s_L|}{s_L} \; 100 \quad , \tag{A.115}$$

where $s(V)$ is the sensitivity of the sensor as a function of the magnitude of the measured quantity, and s_L is the sensitivity obtained from the best linear fit of the data (real measurements vs. raw measurements).

Cross-axis sensitivity: for triaxial sensors, and due to manufacturing limitations, an axis could produce measurements different than zero when measuring a vector that is actually "orthogonal" to that axis (it should measure zero). The *cross-axis sensitivity* measures how sensitive an axis is to vectors "orthogonal" to itself, and is expressed as a percentage:

$$CAS \quad = \quad \frac{s_O}{s_A} \; 100 \quad , \tag{A.116}$$

where s_A is the sensitivity of the axis, and s_O is the sensitivity of the same axis to vectors "orthogonal" to itself.

Offset: due to manufacturing limitations, and a null vector being measured, the sensor may produce nonzero values. The *offset* is the value produced by the sensor when the vector to be measured is actually null. The offset is often referred to as *zero output*.

Offset tolerance: the *offset tolerance* measures the magnitude of the offsets produced by the sensor. It is usually expressed in the measurement units.

Offset change with temperature: the offset of a sensor could change as a function of its temperature. The *offset change with temperature* is a measure of the magnitude of such a change, and is expressed in the measurement units:

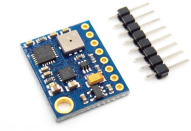
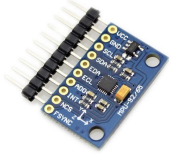
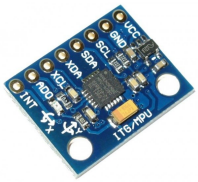
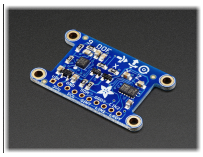
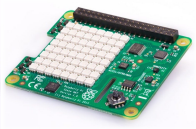
$$OCT \quad = \quad \max_T |\, o_S(T) \,| \quad , \tag{A.117}$$

where $o_S(T)$ is the offset of the sensor as a function of the operating temperature.

A.13.2. IMUs Description

The following table provides a brief description of the IMUs used during the development of this thesis. Information for fields that contain a "-" has not been found.

Table A.2: Descriptive table of IMUs used during this research.

ID	IMU	Manufacturer	Chips included	Price ($)	Year purchased
11	GY-88 	Ronghai Electronics	MPU6050 HMC5883L BMP085	~ 10	2014
12	GY-9250 	Umean Technology	MPU9250	~ 5	2015
13 14 15	GY-521 	Umean Technology	MPU6050	~ 1	2017
16	Adafruit 9-DOF 	Adafruit	LSM303DLHC L3GD20H	~ 20	2016
17	Sense HAT 	Raspberry Pi	LSM9DS1 LPS25H HTS221	~ 30	2016
19	VG400CC-100 	Crossbow	-	~ 10000	1999
20	MTi-G 	Xsens	-	~ 4400	2010

A.13.3. Comparison of Accelerometer Characteristics

The following table compares the main characteristics of the accelerometers used during the development of this thesis. The values have been extracted from the datasheets, except for the values marked with an "*" that have been measured experimentally. Fields that contain a "-" are not provided in the datasheets nor have been measured experimentally. The characteristics that are superior compared to the rest have been colored green. Those that are inferior compared to the rest have been colored red. The rest have been colored orange.

Table A.3: Comparison table of accelerometers used during this research.

Chip	Range (g)	Resolution (bits)	Raw noise (mg)	Maximum ODR (Hz)	Noise SD ($\mu g/\sqrt{Hz}$)	Sensitivity tolerance (%)	Sensitivity change with temp. (%)	Sensitivity non-linearity (%)	Cross-axis sensitivity (%)	Offset tolerance (mg)	Offset change with temp. (mg)
MPU6050	±2 ±4 ±8 ±16	16	7*	1000	221*	3	2.5	0.5	2	60	45
MPU9250	±2 ±4 ±8 ±16	16	10*	4000	158*	3	3.25	0.5	2	67	94
LSM303DLHC	±2 ±4 ±8 ±16	10	72*	220	4854*	-	1.25	-	-	60	62.5
LSM9DS1	±2 ±4 ±8 ±16	16	8*	952	259*	-	-	-	-	90	-
VG400CC-100	2	14	1.2*	75	139*	1	-	1	-	8.5	-
Xsens	±1.7 ±5 ±18	16	1.6*	100	160*	0.03	0	0.2	0.1	2	0

A.13.4. Comparison of Gyroscope Characteristics

The following table compares the main characteristics of the gyroscopes used during the development of this thesis. The values have been extracted from the datasheets, except for the values marked with an "*" that have been measured experimentally. Fields that contain a "-" are not provided in the datasheets nor have been measured experimentally. The characteristics that are superior compared to the rest have been colored green. Those that are inferior compared to the rest have been colored red. The rest have been colored orange.

Table A.4: Comparison table of gyroscopes used during this research.

Chip	Range (dps)	Resolution (bits)	Raw noise (dps)	Maximum ODR (Hz)	Noise SD (dps/$\sqrt{\text{Hz}}$)	Sensitivity tolerance (%)	Sensitivity change with temp. (%)	Sensitivity non-linearity (%)	Cross-axis sensitivity (%)	Offset tolerance (dps)	Offset change with temp. (dps)
MPU6050	±250 ±500 ±1000 ±2000	16	0.3*	8000	0.004*	3	2	0.2	2	20	20
MPU9250	±250 ±500 ±1000 ±2000	16	0.3*	8000	0.004*	3	4	0.1	2	5	30
L3GD20H	±245 ±500 ±2000	16	1*	757.6	0.036*	-	2	0.2	-	25	5
LSM9DS1	±245 ±500 ±2000	16	0.3*	952	0.010*	-	-	-	-	30	-
VG400CC-100	±100	14	0.4*	75	0.046*	1	-	0.3	-	1	-
Xsens	±150 ±300 ±1200	16	0.6*	100	0.060*	-	0	0.1	0.1	1	0

A.13.5. Comparison of Magnetometer Characteristics

The following table compares the main characteristics of the magnetometers used during the development of this thesis. The values have been extracted from the datasheets, except for the values marked with an "*" that have been measured experimentally. Fields that contain a "-" are not provided in the datasheets nor have been measured experimentally. The characteristics that are superior compared to the rest have been colored green. Those that are inferior compared to the rest have been colored red. The rest have been colored orange.

Table A.5: Comparison table of magnetometers used during this research.

Chip	Range (G)	Resolution (bits)	Raw noise (mG)	Maximum ODR (Hz)	Noise SD ($\mu G/\sqrt{Hz}$)	Sensitivity tolerance (%)	Sensitivity change with temp. (%)	Sensitivity non-linearity (%)	Cross-axis sensitivity (%)	Offset tolerance (mG)	Offset change with temp. (mG)
HMC5883L	0.88 1.3 1.9 2.5 4.0 4.7 5.6 8.1	12	5*	75	578*	5	17.25	0.1	0.2	-	-
MPU9250	48	14	14*	100	1400*	-	-	-	-	800	-
LSM303DLHC	1.3 1.9 2.5 4.0 4.7 5.6 8.1	16	4*	220	270*	-	-	-	1	-	-
LSM9DS1	±4 ±8 ±12 ±16	16	16*	80	1789*	-	-	-	-	14	-
Xsens	±0.75	16	4*	100	400*	0.5	0	0.2	0.1	0.1	0

A.13.6. Comparison of Temperature Sensor Characteristics

The following table compares the main characteristics of the built-in temperature sensors used during the development of this thesis. The values have been extracted from the datasheets. Fields that contain a "-" are not provided in the datasheets. The characteristics that are superior compared to the rest have been colored green. Those that are inferior compared to the rest have been colored red. The rest have been colored orange.

Table A.6: Comparison table of temperature sensors used during this research.

Chip	Range (°C)	Resolution (bits)	Maximum ODR (Hz)
MPU6050	-40 – 85	12	-
MPU9250	-40 – 105	16	4000
LSM303DLHC	-40 – 85	12	220
L3GD20H	-40 – 85	8	1
LSM9DS1	-40 – 85	16	59.5
VG400CC-100	-40 – 71	12	-
Xsens	-40 – 85	12	-

Appendix B

Triaxial Sensor Calibration Data

B.1. Sample Data graphics

B.1.1. Accelerometers

Figures are presented with the format of Table B.1

Table B.1: Arrangement of the figures of accelerometers.

A	B	C
D	E	

The details of each part are explained below:

A Temperature VS measurement number. Very cold temperatures are cyan. Cold temperatures are blue. Warm temperatures are red. Hot temperatures are yellow.

B Measurements taken with the triaxial sensor. The color of the point represents its temperature (see **A**).

C Measurement surfaces. The green sphere represents the surface where the measurements should appear (if the sensors where calibrated). The red ellipse represents the surface where the measurements actually appear (due to the miscalibration).

D Measurement error VS measurement number. Errors produced by the default calibration are drawn in purple. Errors produced by the optimal calibration are drawn in green.

E Algorithm convergence. $\|\boldsymbol{\delta}^{(k)}\|^2$ as a function of the iteration number.

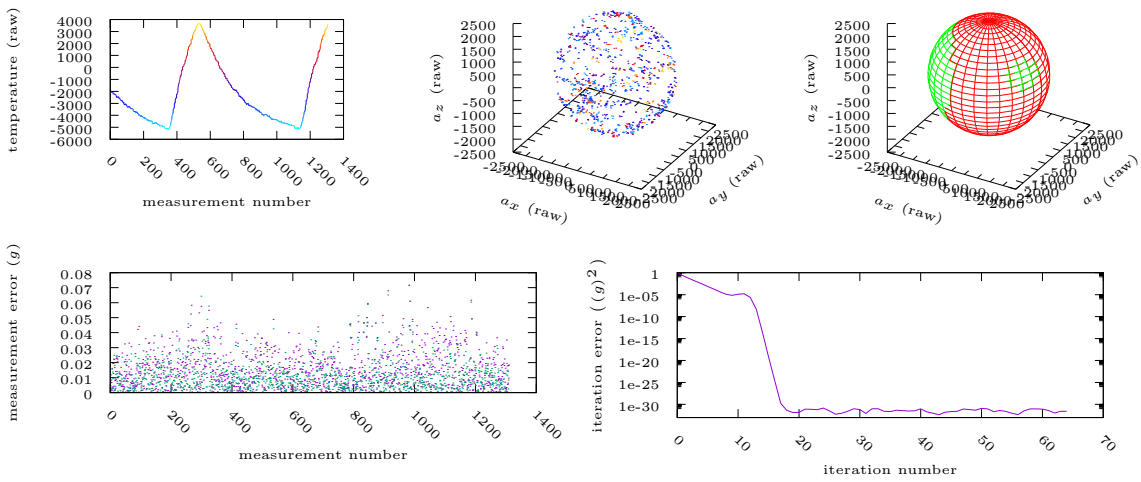


Figure B.1: Data representations: ID 11a_20190408191309 .

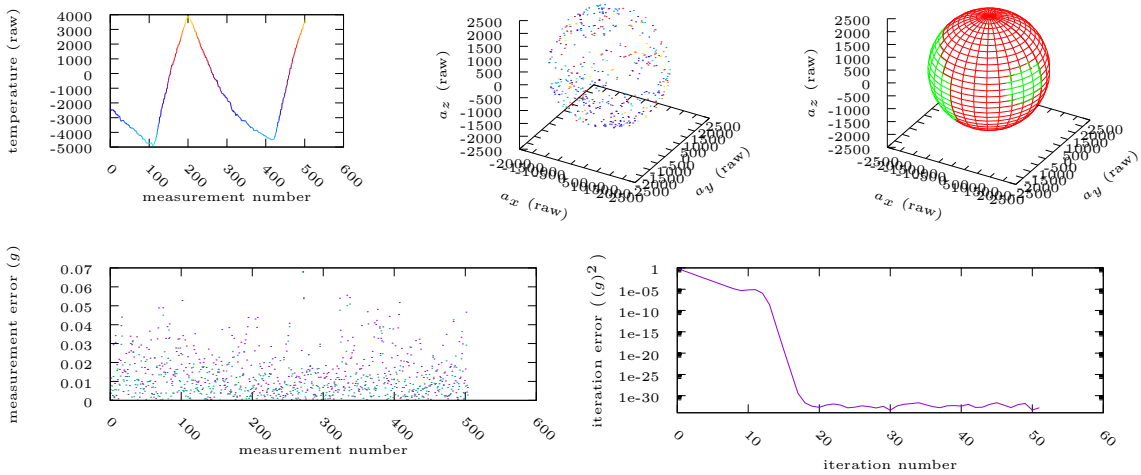


Figure B.2: Data representations: ID 11a_20190411220405 .

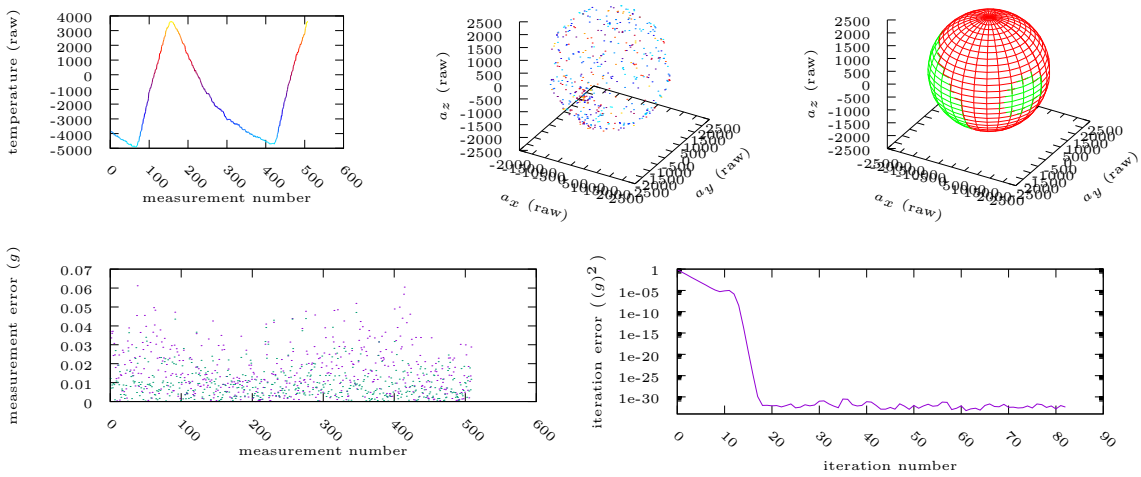


Figure B.3: Data representations: ID 11a_20190415174652 .

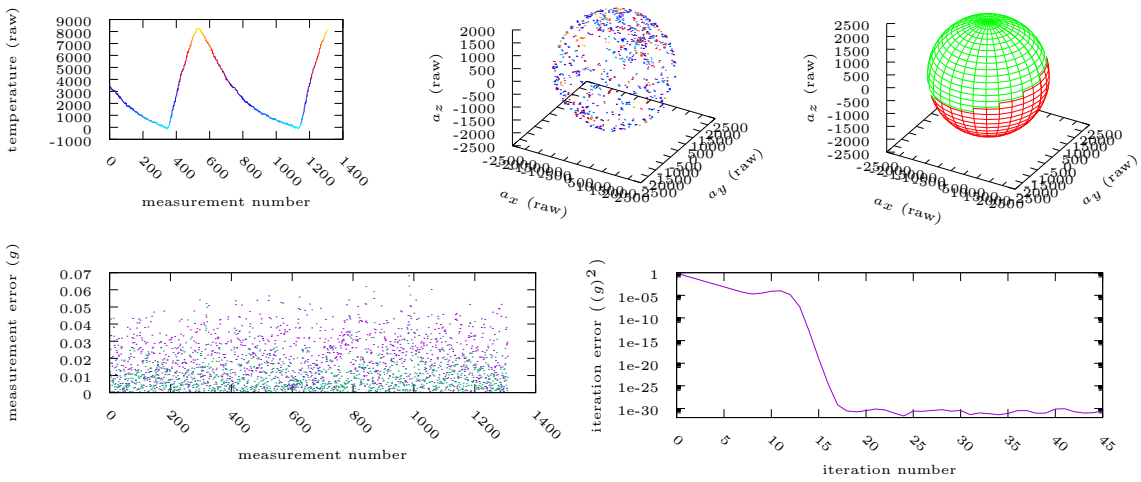


Figure B.4: Data representations: ID 12a_20190408191309 .

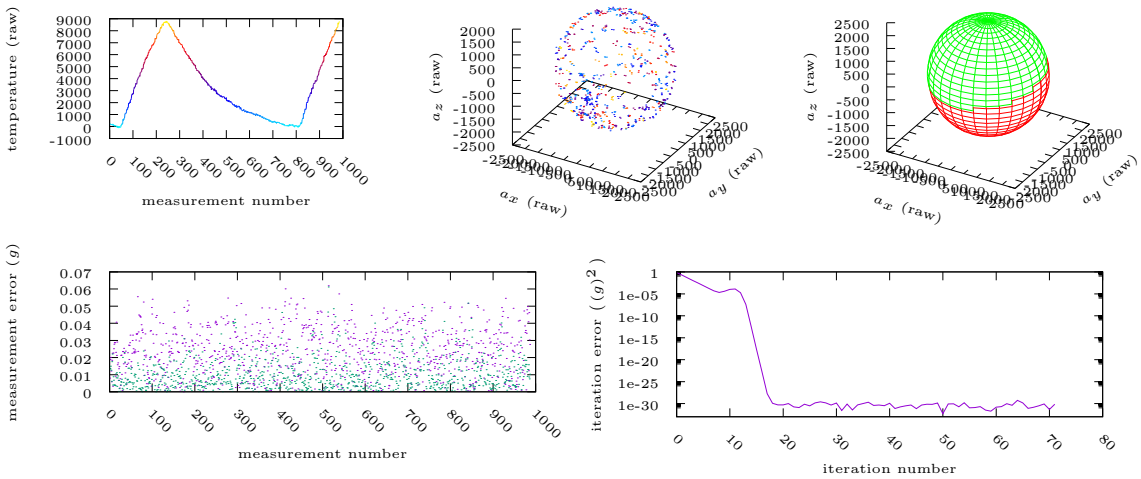


Figure B.5: Data representations: ID 12a_20190411171603 .

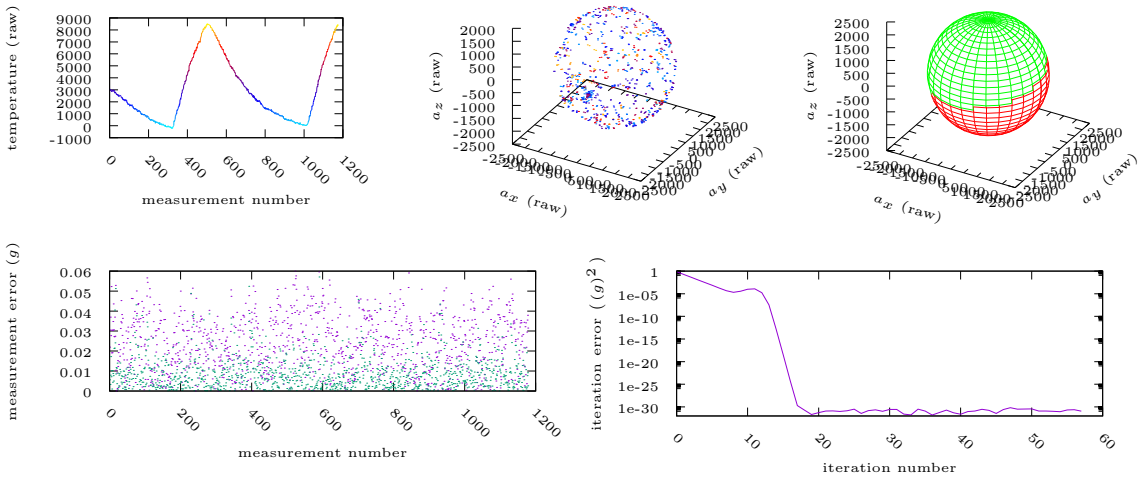


Figure B.6: Data representations: ID 12a_20190415202758 .

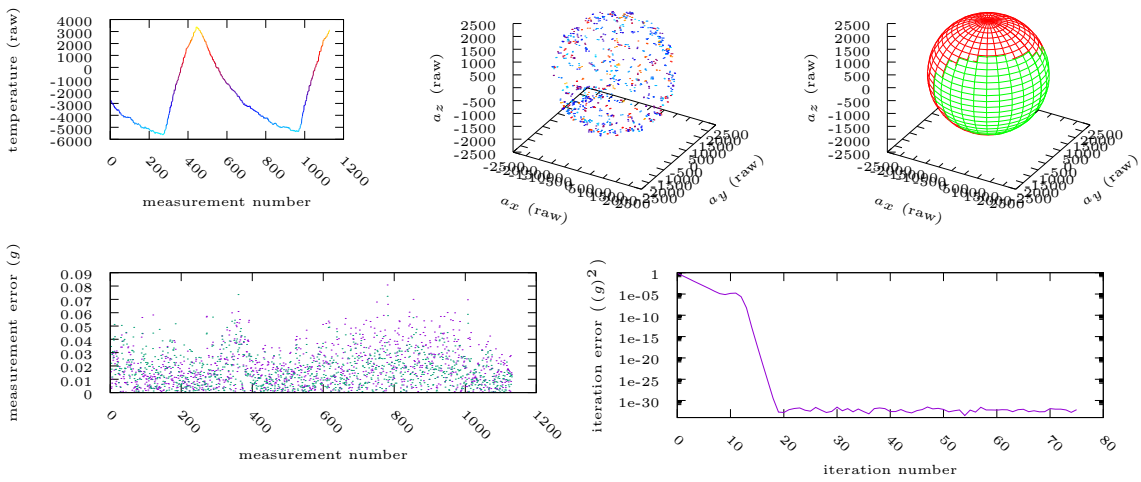


Figure B.7: Data representations: ID 13a_20190408202950 .

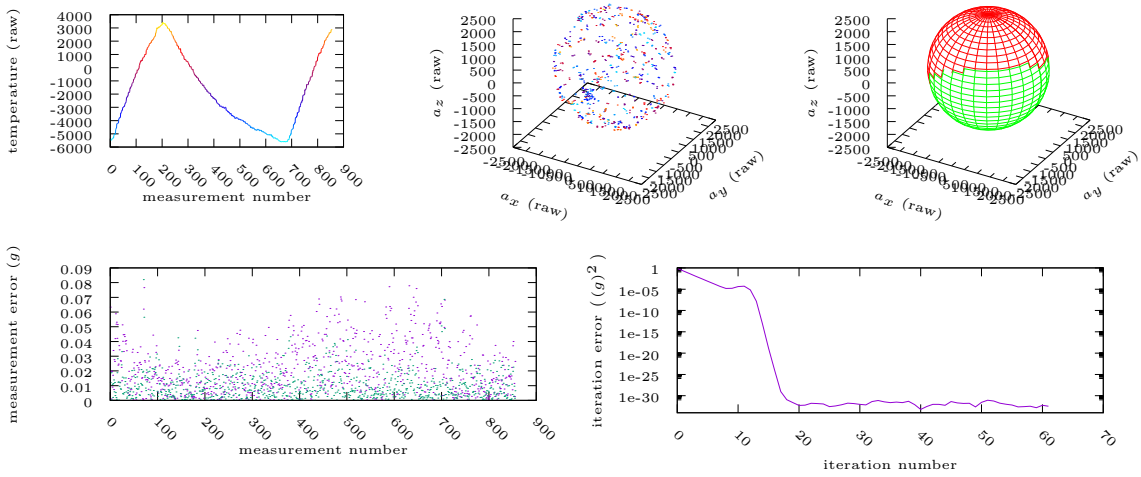


Figure B.8: Data representations: ID 13a_20190411205019 .

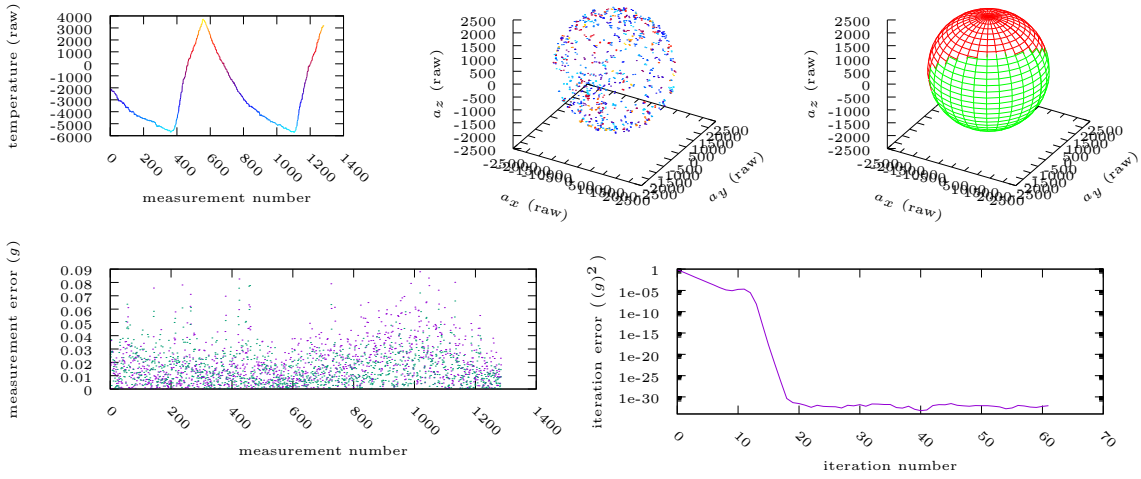


Figure B.9: Data representations: ID 13a_20190415190820 .

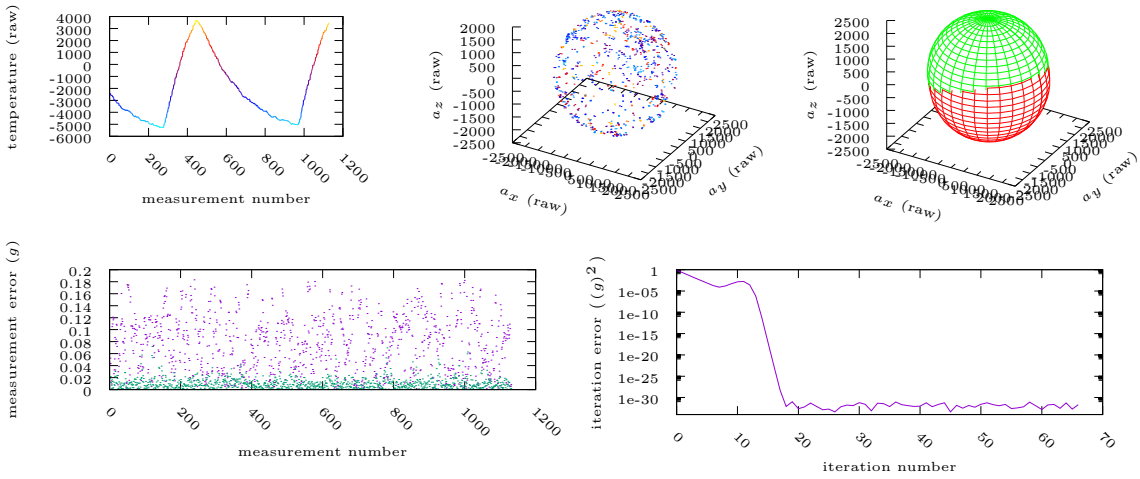


Figure B.10: Data representations: ID 14a_20190408202950 .

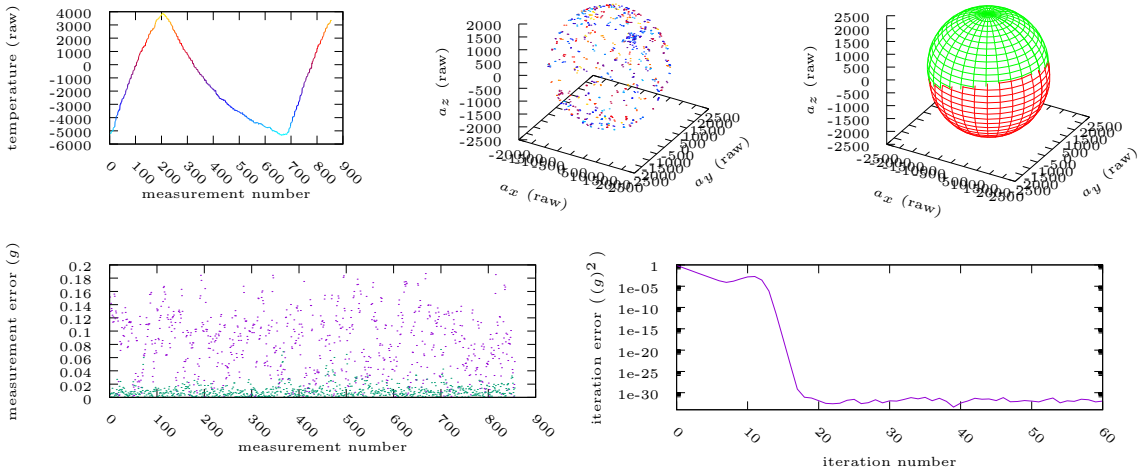


Figure B.11: Data representations: ID 14a_20190411205019 .

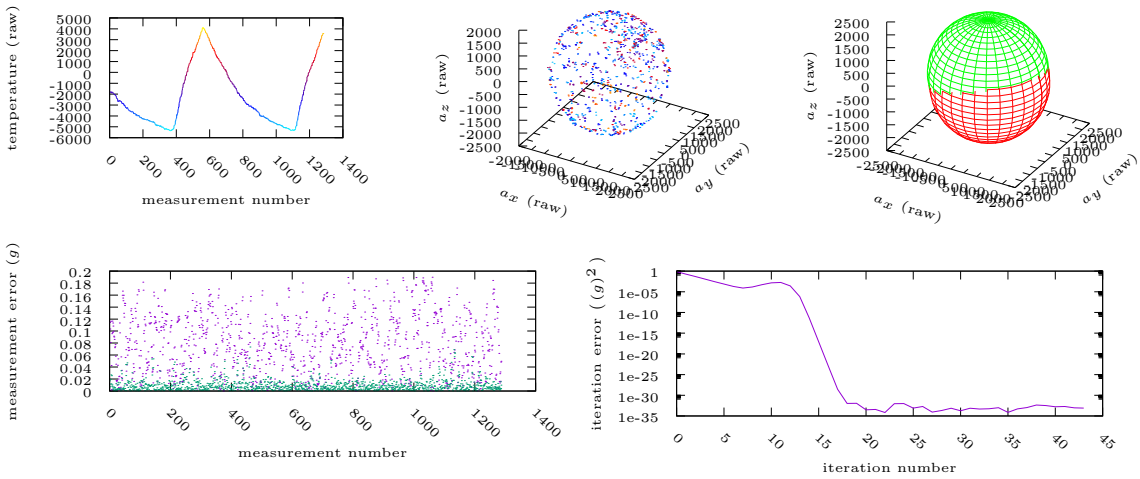


Figure B.12: Data representations: ID 14a_20190415190820 .

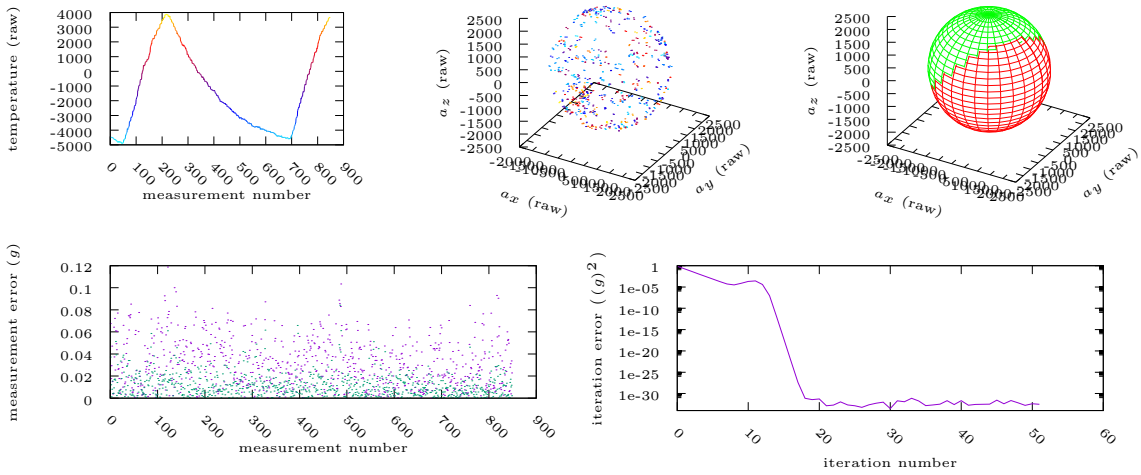


Figure B.13: Data representations: ID 15a_20190408174806 .

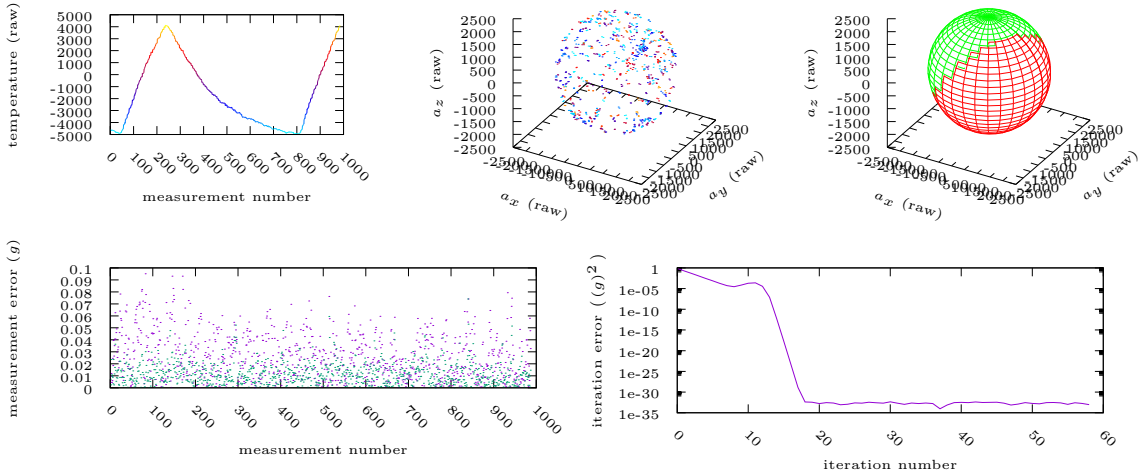


Figure B.14: Data representations: ID 15a_20190411171603 .

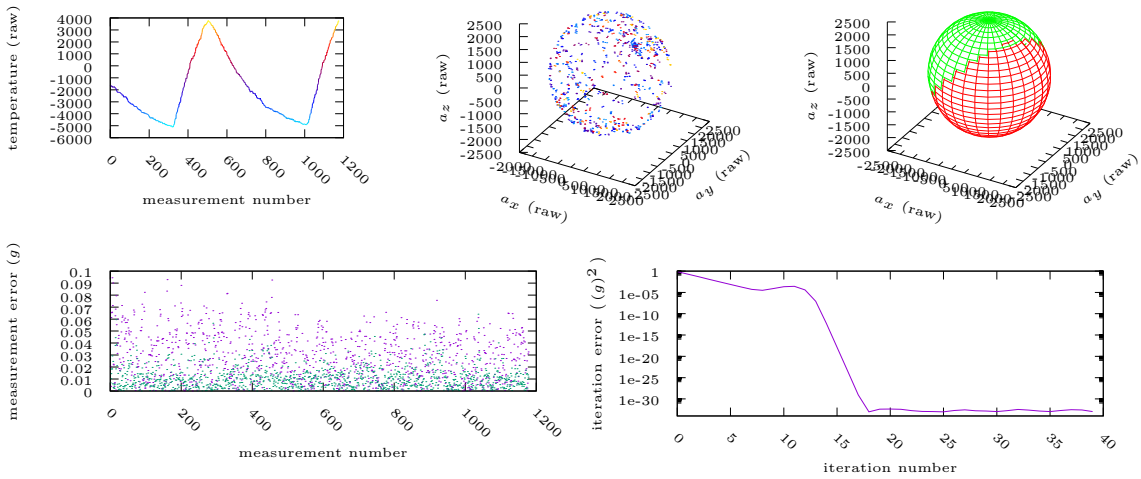


Figure B.15: Data representations: ID 15a_20190415202758 .

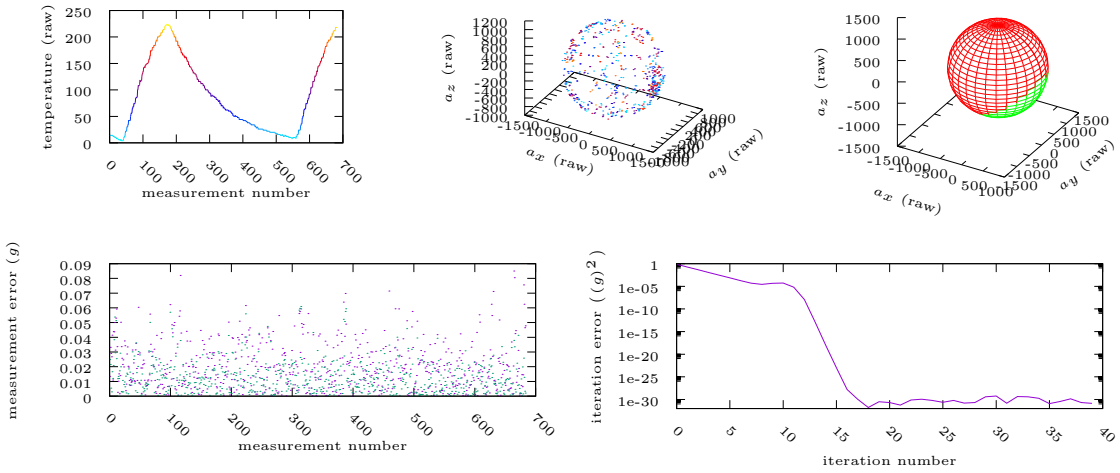


Figure B.16: Data representations: ID 16a_20190408174806 .

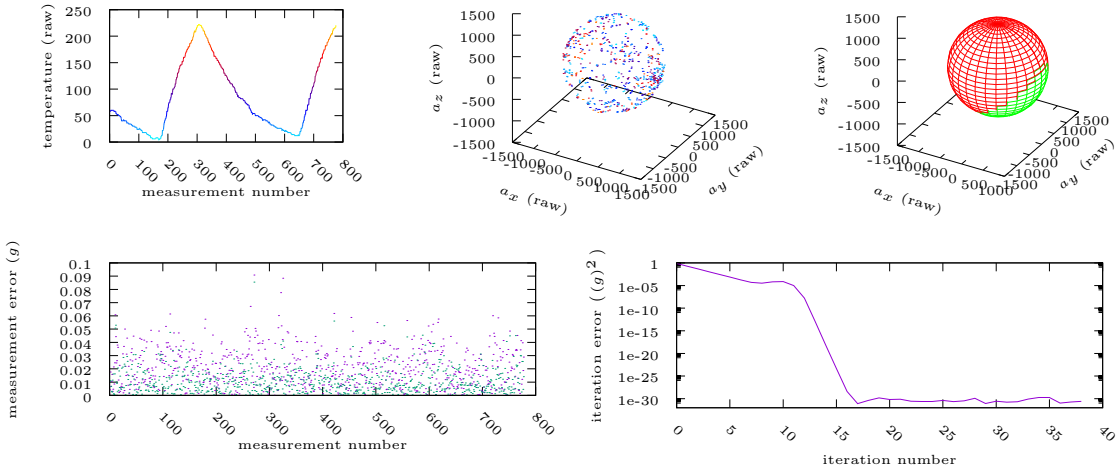


Figure B.17: Data representations: ID 16a_20190411220405 .

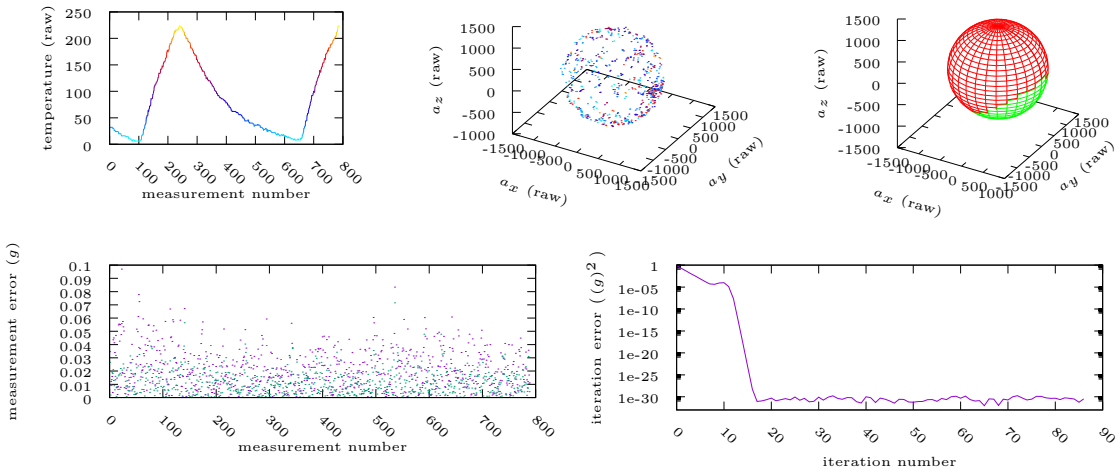


Figure B.18: Data representations: ID 16a_20190415174652 .

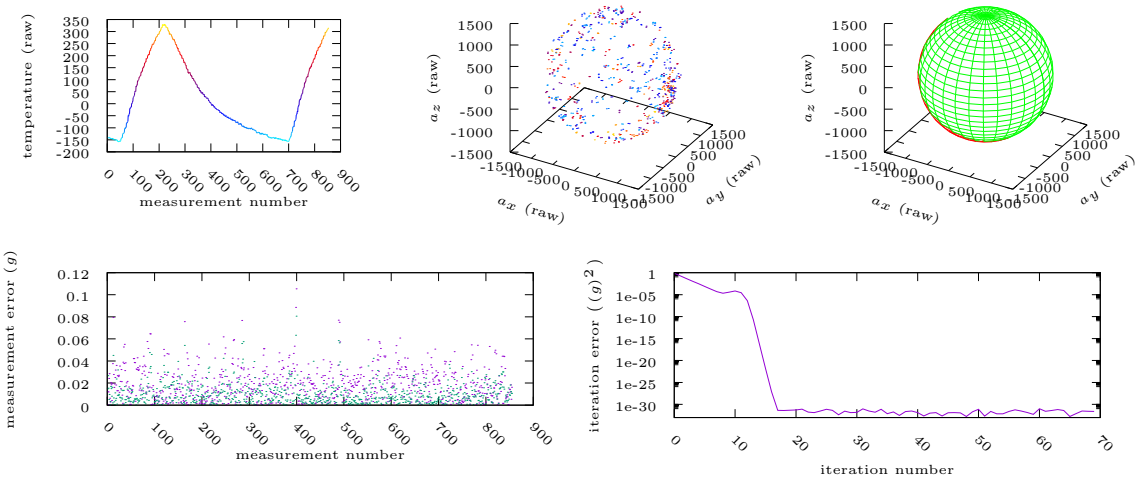


Figure B.19: Data representations: ID 17a_20190408174806 .

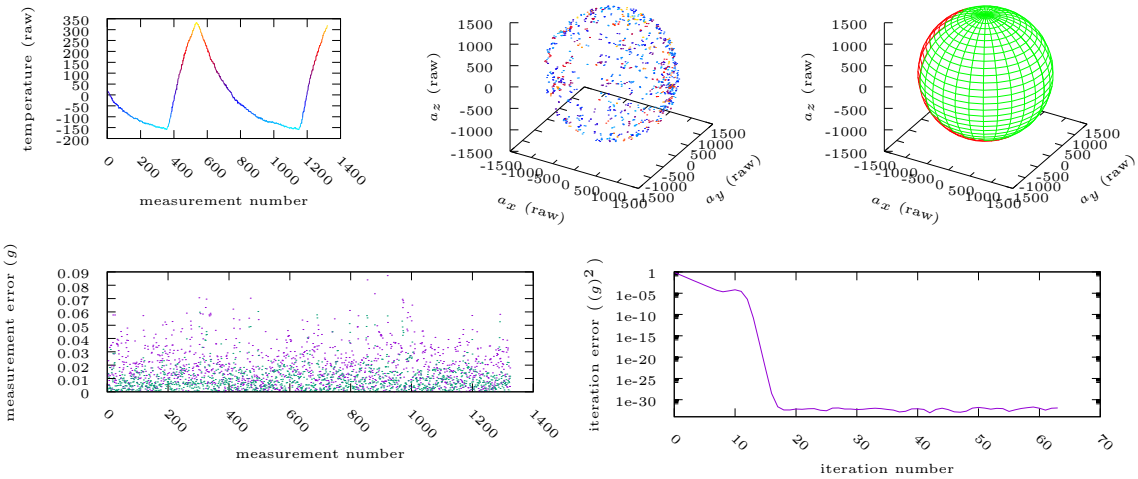


Figure B.20: Data representations: ID 17a_20190408191309 .

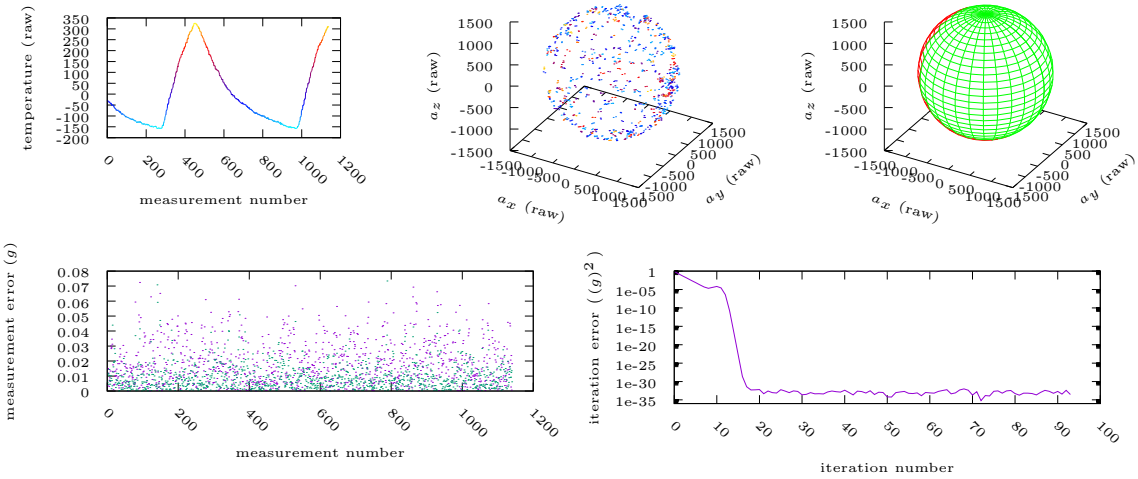


Figure B.21: Data representations: ID 17a_20190408202950 .

B.1.2. Gyroscopes

Figures are presented with the format of Table B.2

Table B.2: Arrangement of the figures of gyroscopes.

A		B	C
D	E	F	G

The details of each part are explained below:

A Temperature VS measurement number. Very cold temperatures are cyan. Cold temperatures are blue. Warm temperatures are red. Hot temperatures are yellow.

B Measurements taken with the triaxial sensor. The color of the point represents its temperature (see **A**).

C Measurement surfaces. The green sphere represents the surface where the measurements should appear (if the sensors where calibrated). The red ellipse represents the surface where the measurements actually appear (due to the miscalibration).

D Measurement error VS measurement number. Errors produced by the default calibration are drawn in purple. Errors produced by the optimal calibration are drawn in green.

E Algorithm convergence. $\|\boldsymbol{\delta}^{(k)}\|^2$ as a function of the iteration number.

F Offset VS temperature. Measurements of the x-axis are red. Measurements of the y-axis are green. Measurements of the z-axis are blue.

G Angular velocity magnitude measured with the triaxial sensor VS angular velocity magnitude established with the stepper motor. This representation is intended to verify the linearity of the angular velocity measurements.

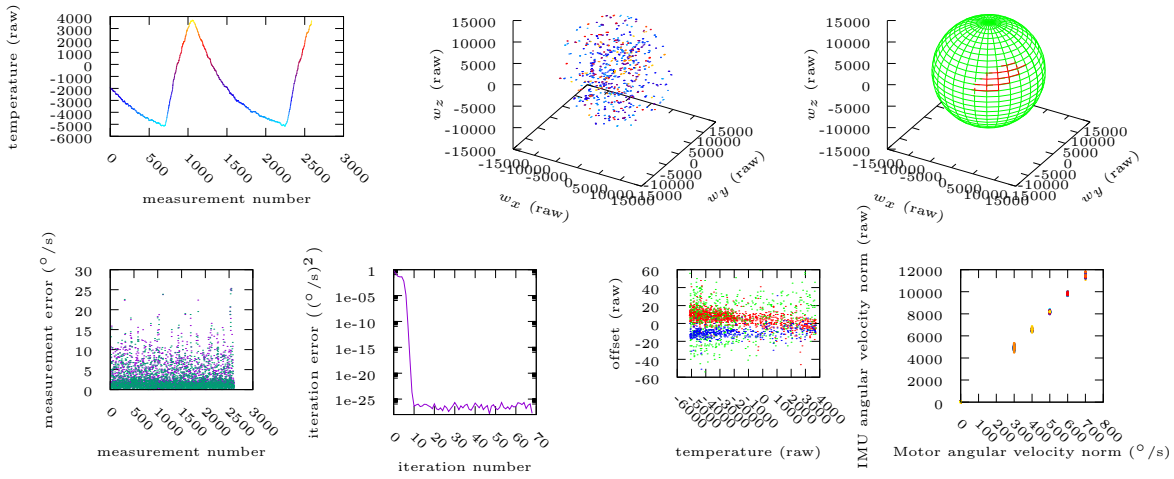


Figure B.22: Data representations: ID 11w_20190408191309 .

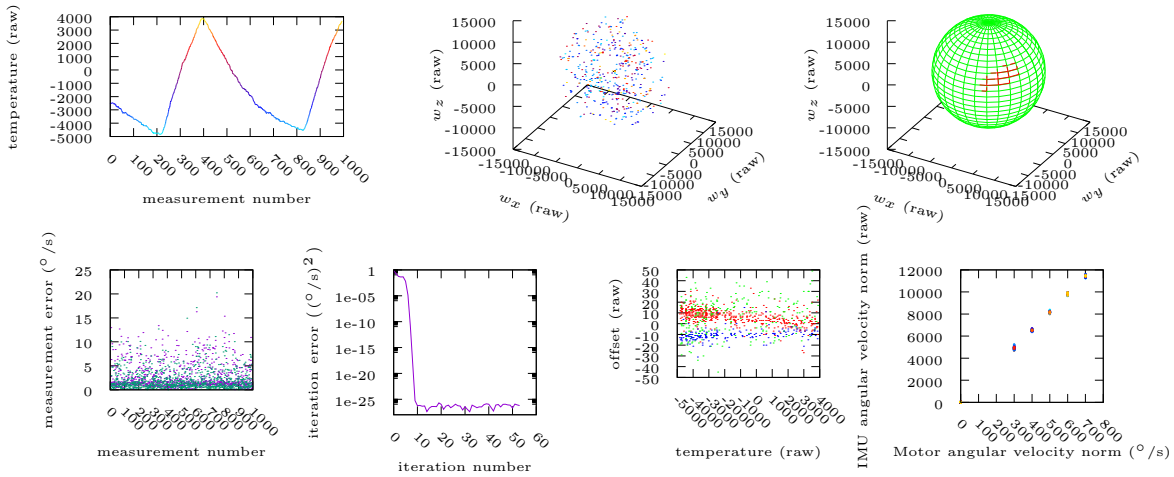


Figure B.23: Data representations: ID 11w_20190411220405 .

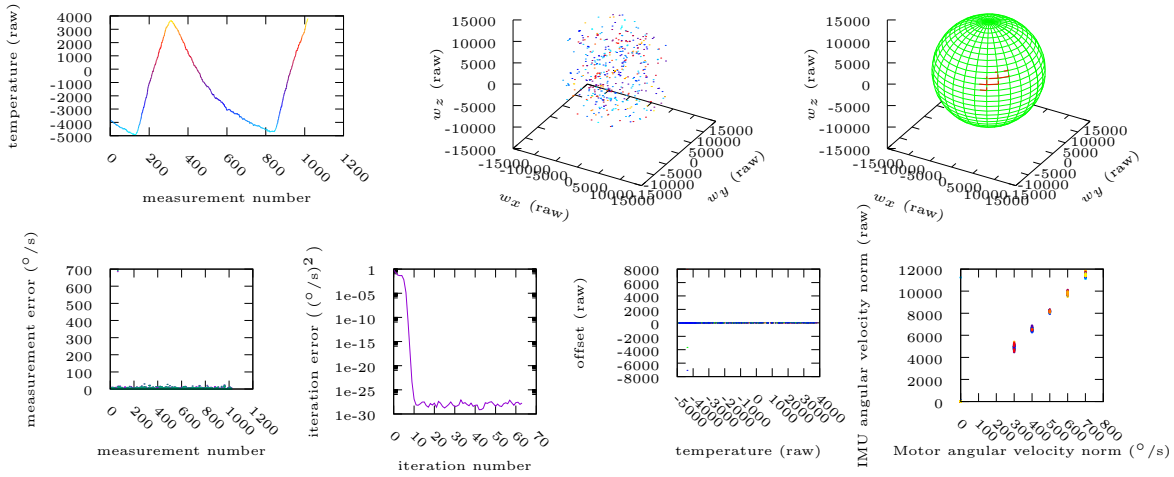


Figure B.24: Data representations: ID 11w_20190415174652 .

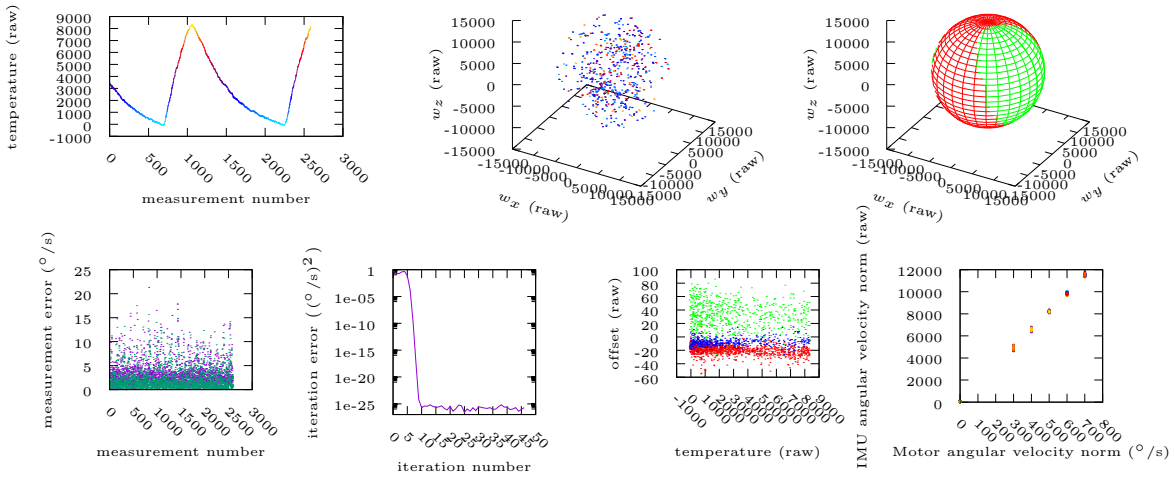


Figure B.25: Data representations: ID 12w_20190408191309 .

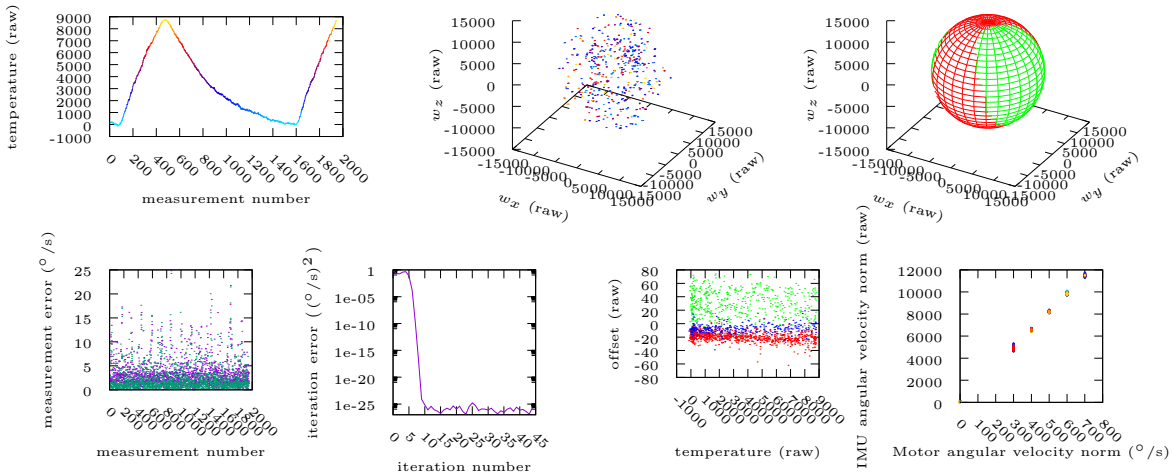


Figure B.26: Data representations: ID 12w_20190411171603 .

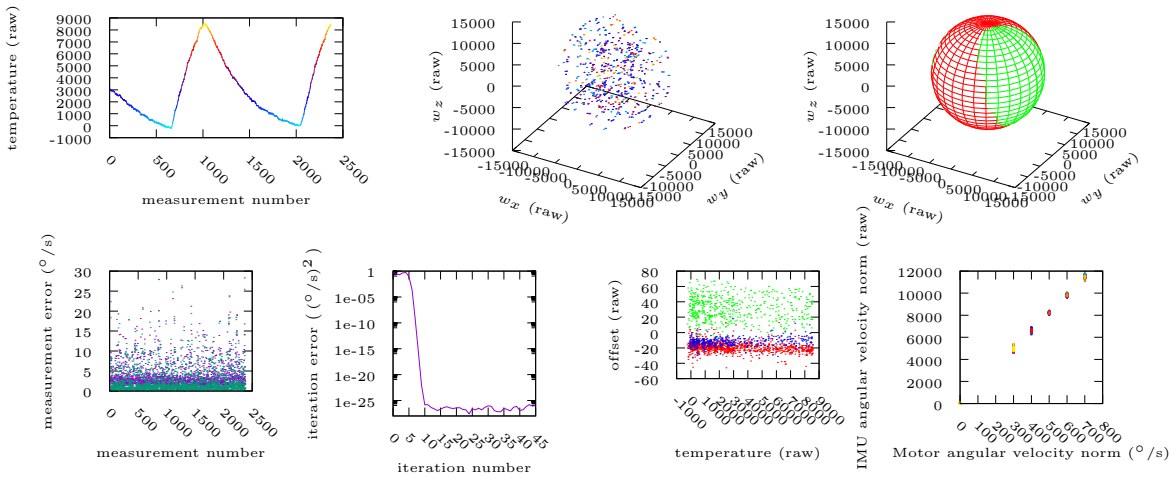


Figure B.27: Data representations: ID 12w_20190415202758 .

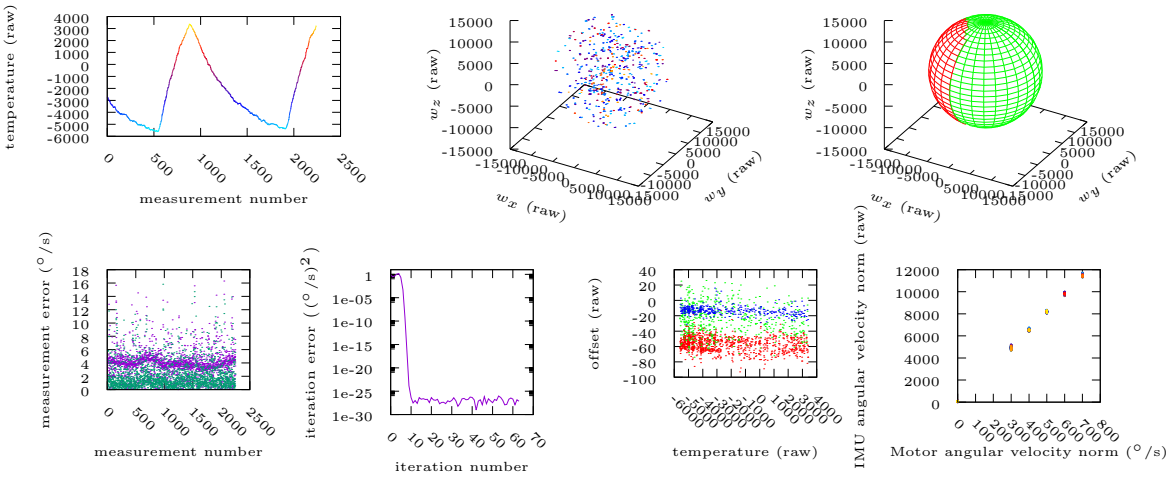


Figure B.28: Data representations: ID 13w_20190408202950 .

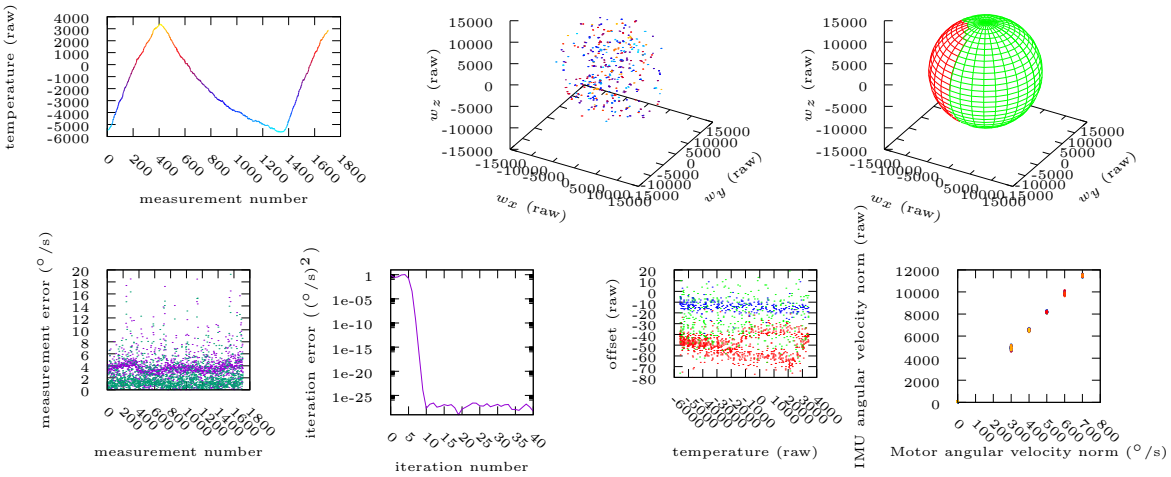


Figure B.29: Data representations: ID 13w_20190411205019 .

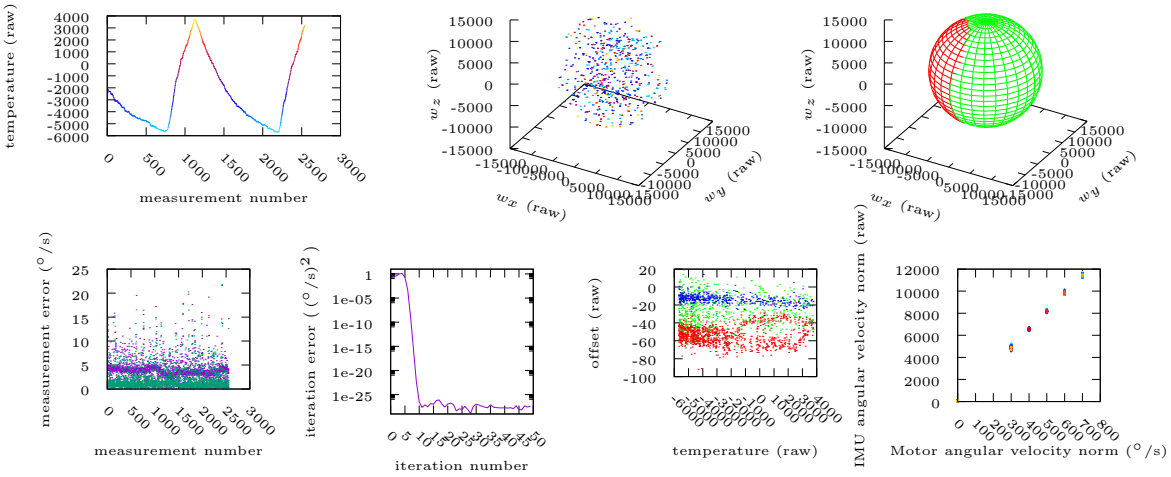


Figure B.30: Data representations: ID 13w_20190415190820 .

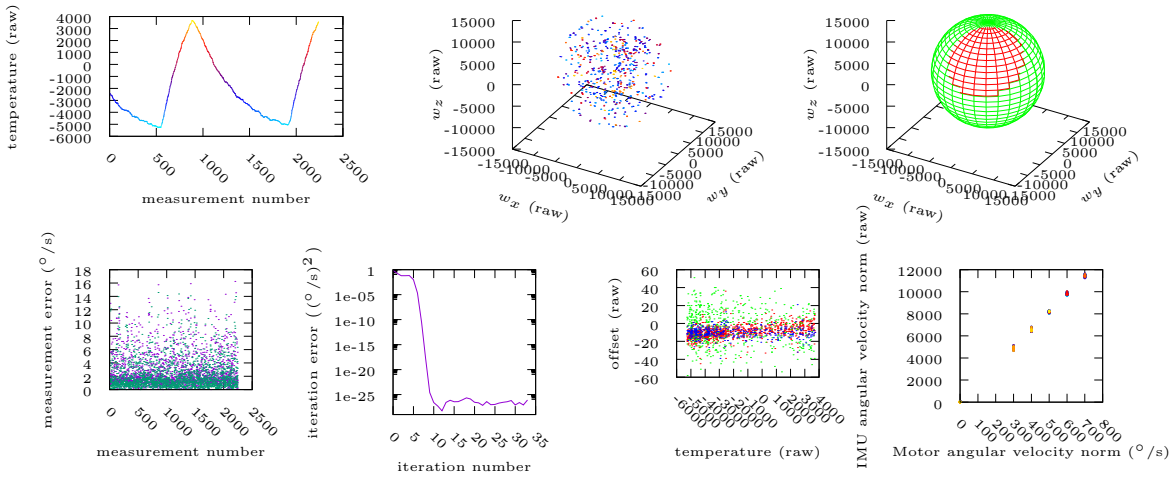


Figure B.31: Data representations: ID 14w_20190408202950 .

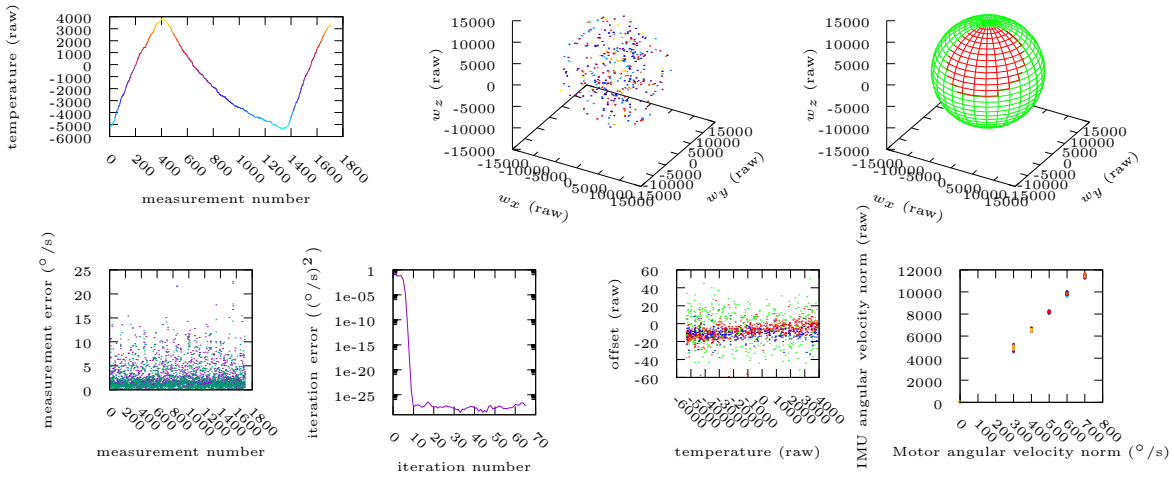


Figure B.32: Data representations: ID 14w_20190411205019 .

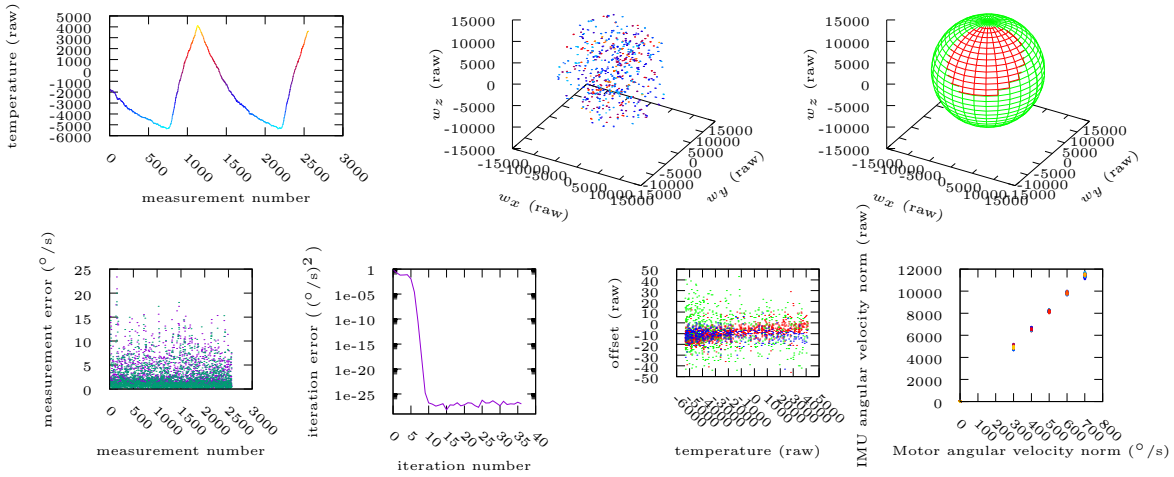


Figure B.33: Data representations: ID 14w_20190415190820 .

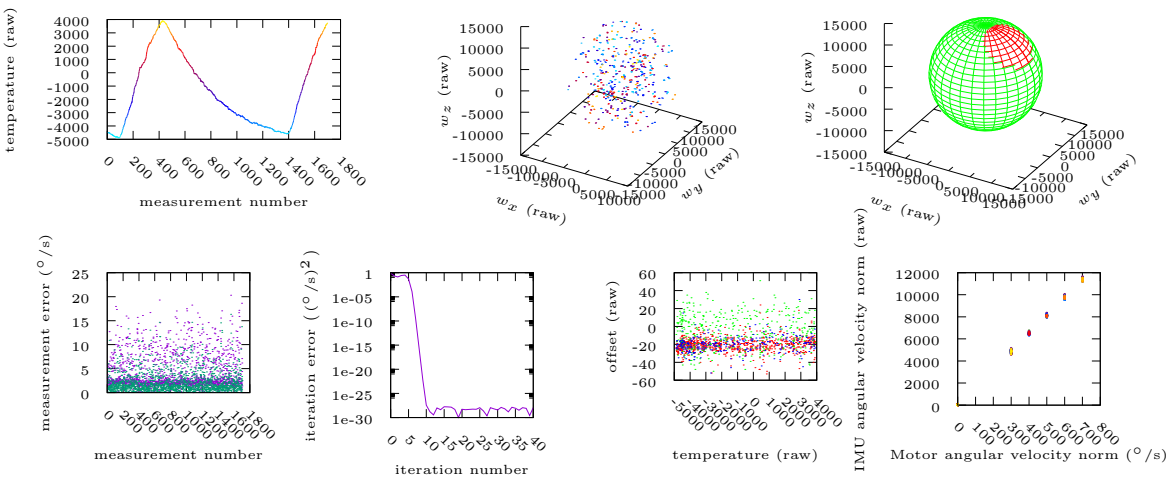


Figure B.34: Data representations: ID 15w_20190408174806 .

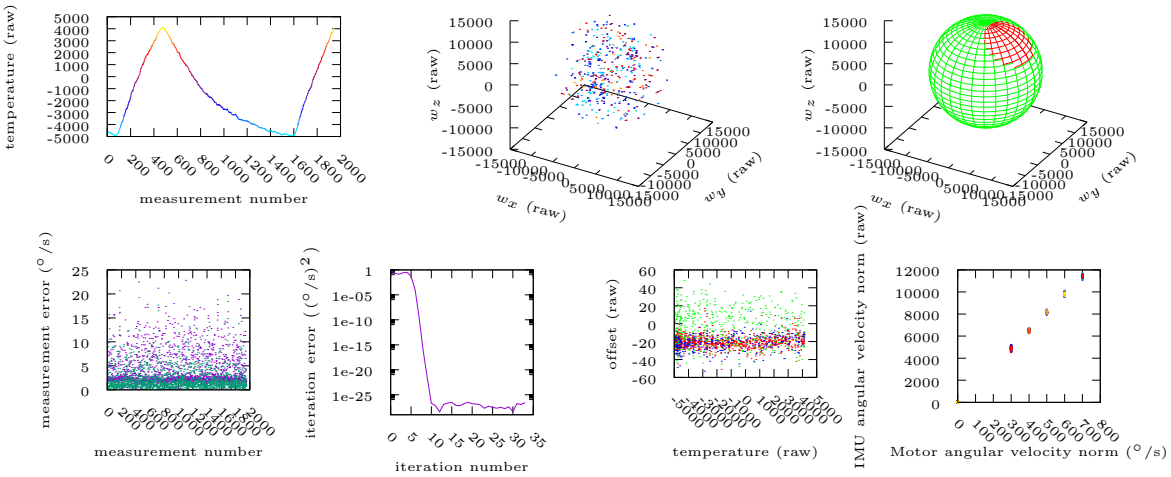


Figure B.35: Data representations: ID 15w_20190411171603 .

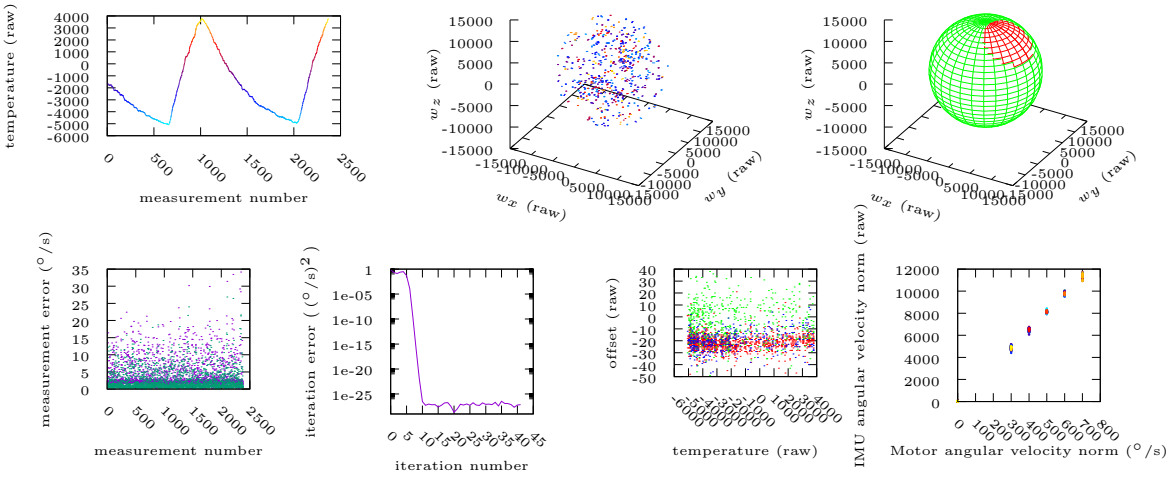


Figure B.36: Data representations: ID 15w_20190415202758 .

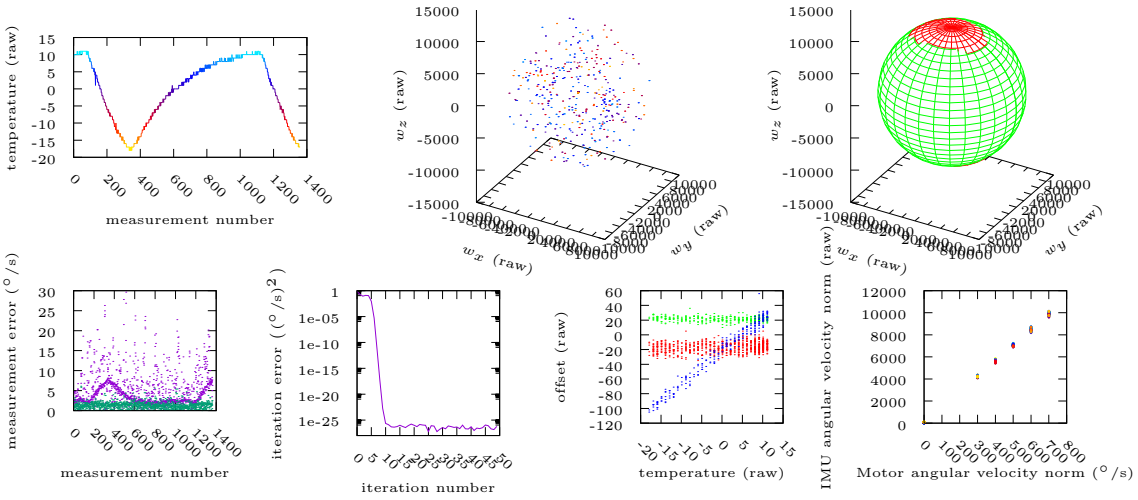


Figure B.37: Data representations: ID 16w_20190408174806 .

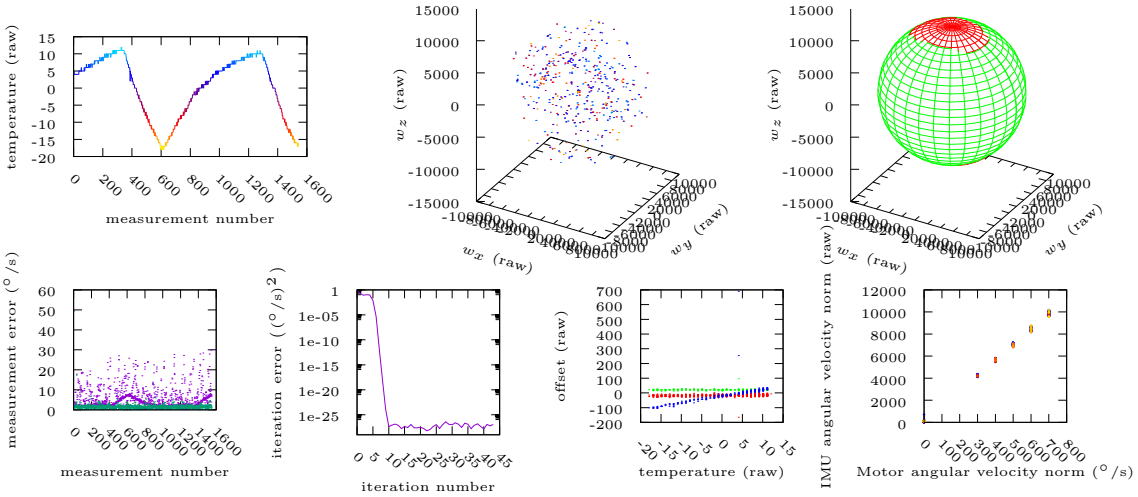


Figure B.38: Data representations: ID 16w_20190411220405 .

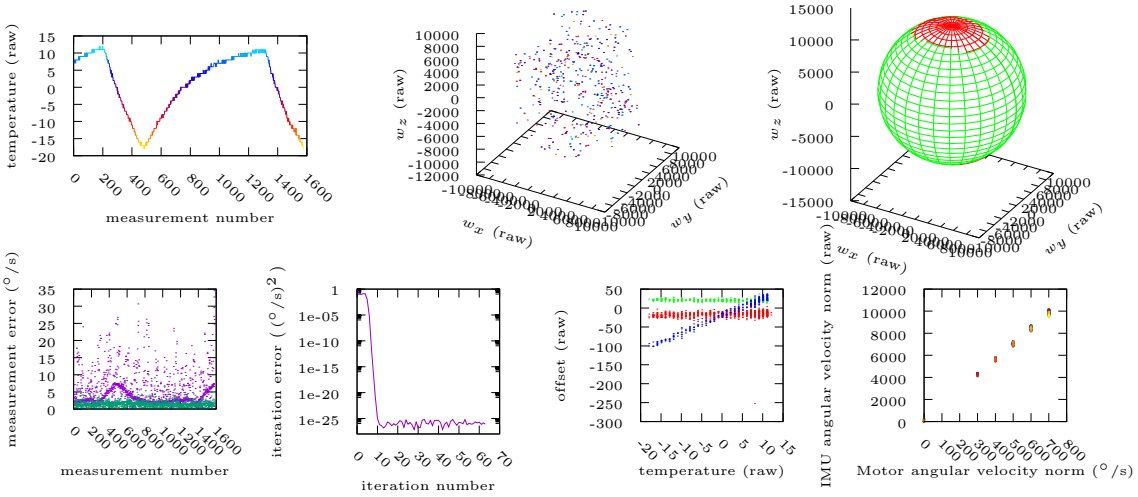


Figure B.39: Data representations: ID 16w_20190415174652 .

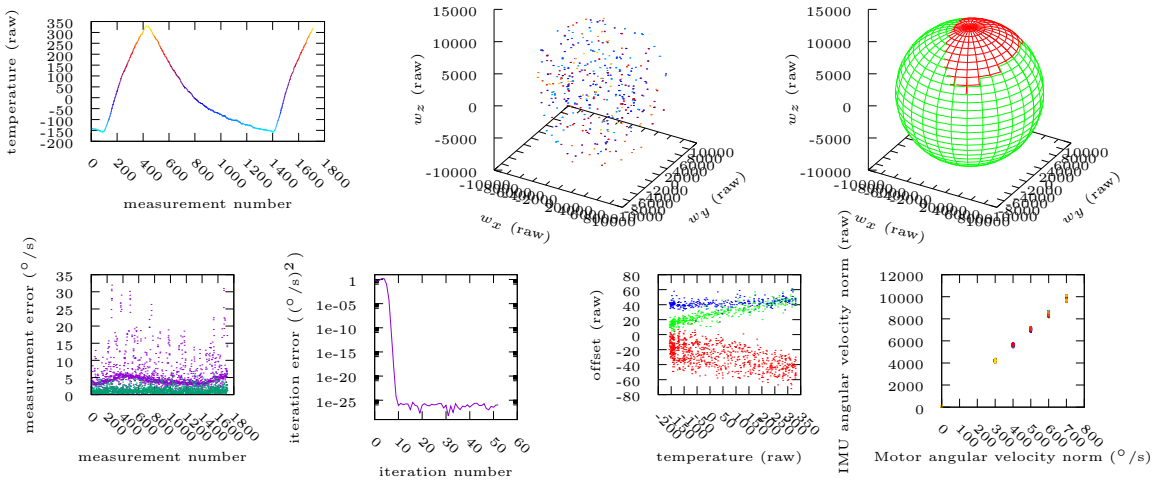


Figure B.40: Data representations: ID 17w_20190408174806 .

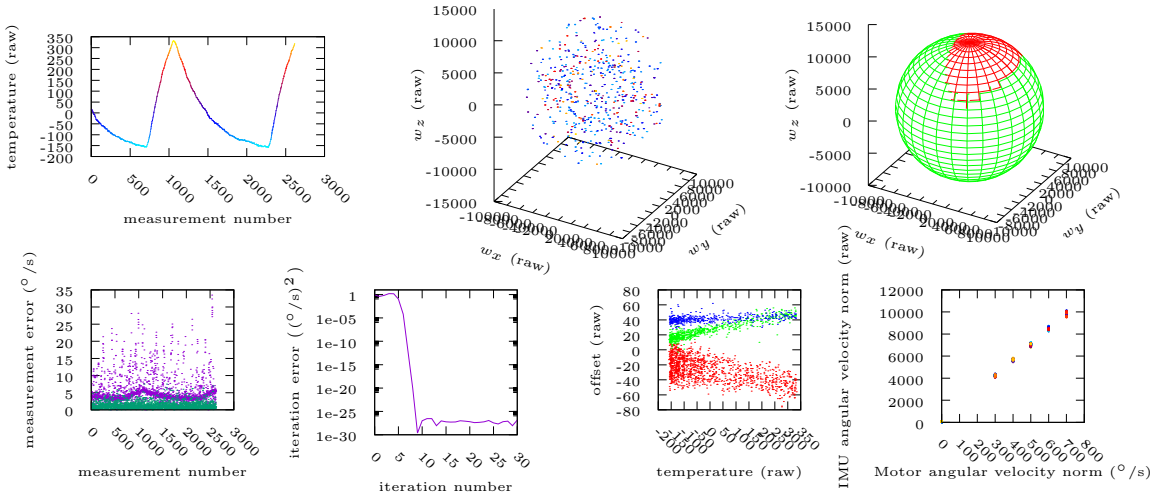


Figure B.41: Data representations: ID 17w_20190408191309 .

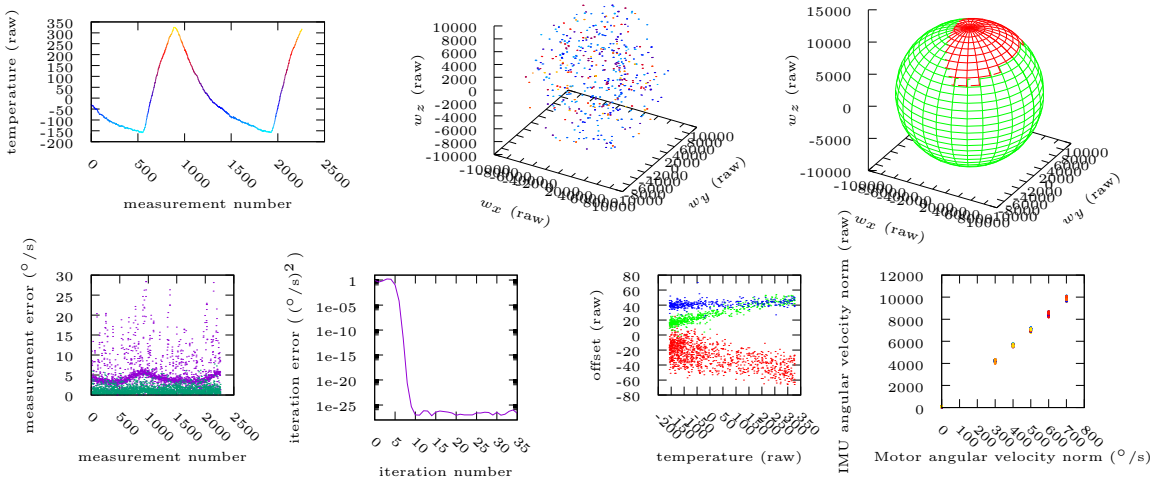


Figure B.42: Data representations: ID 17w_20190408202950 .

B.1.3. Magnetometers

Figures are presented with the format of Table B.3

Table B.3: Arrangement of the figures of magnetometers.

A	B	C
D	E	

The details of each part are explained below:

A Temperature VS measurement number. Very cold temperatures are cyan. Cold temperatures are blue. Warm temperatures are red. Hot temperatures are yellow.

B Measurements taken with the triaxial sensor. The color of the point represents its temperature (see **A**).

C Measurement surfaces. The green sphere represents the surface where the measurements should appear (if the sensors where calibrated). The red ellipse represents the surface where the measurements actually appear (due to the miscalibration).

D Measurement error VS measurement number. Errors produced by the default calibration are drawn in purple. Errors produced by the optimal calibration are drawn in green.

E Algorithm convergence. $\|\boldsymbol{\delta}^{(k)}\|^2$ as a function of the iteration number.

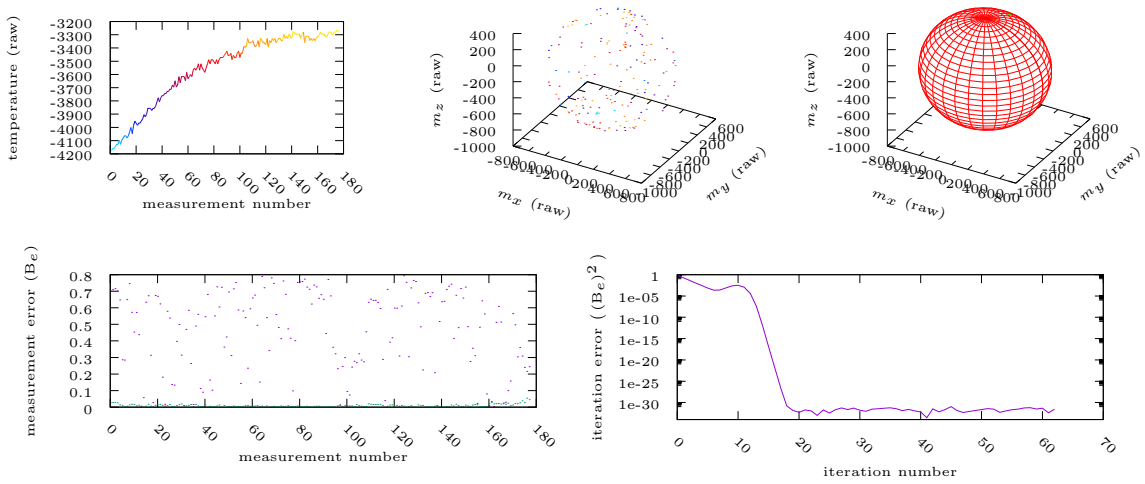


Figure B.43: Data representations: ID 11m_20190410123937 .

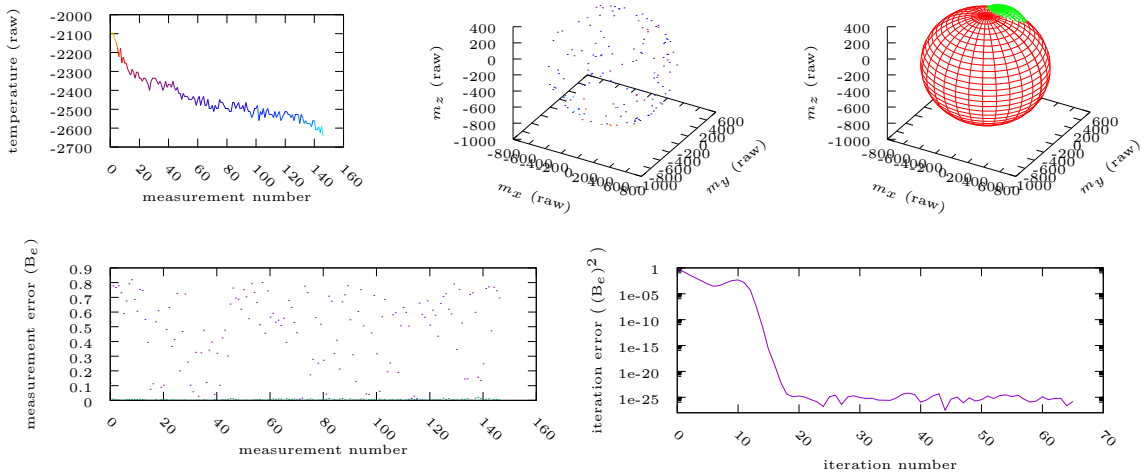


Figure B.44: Data representations: ID 11m_20190412110715 .

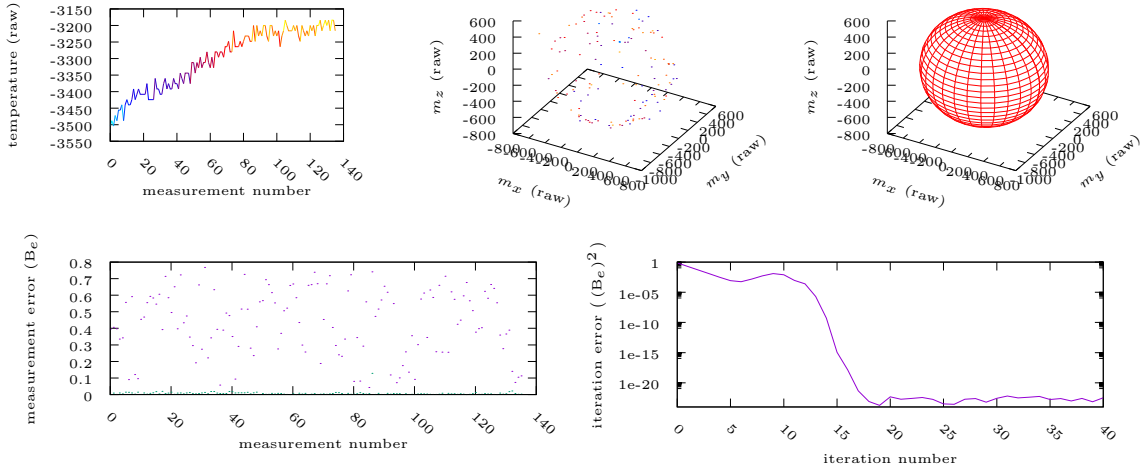


Figure B.45: Data representations: ID 11m_20190415220140 .

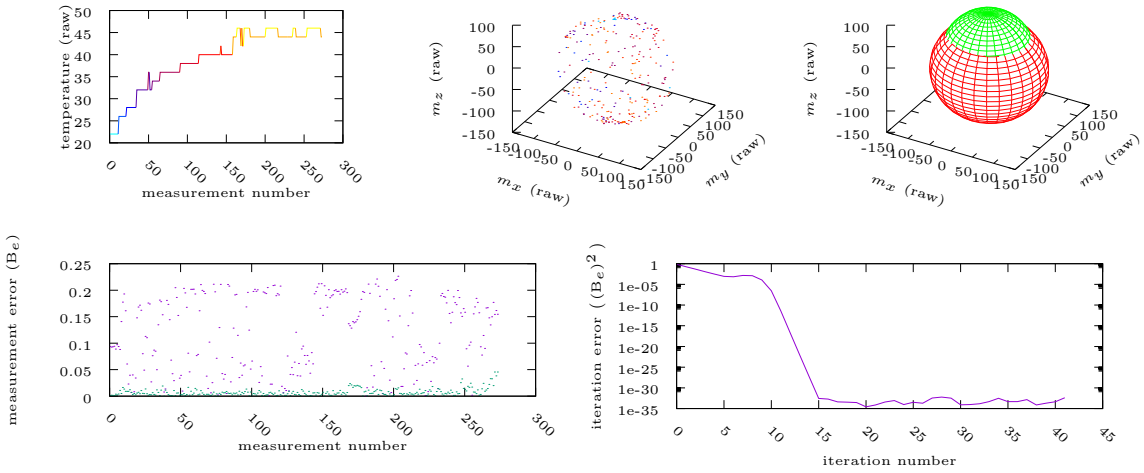


Figure B.46: Data representations: ID 16m_20190410123937 .

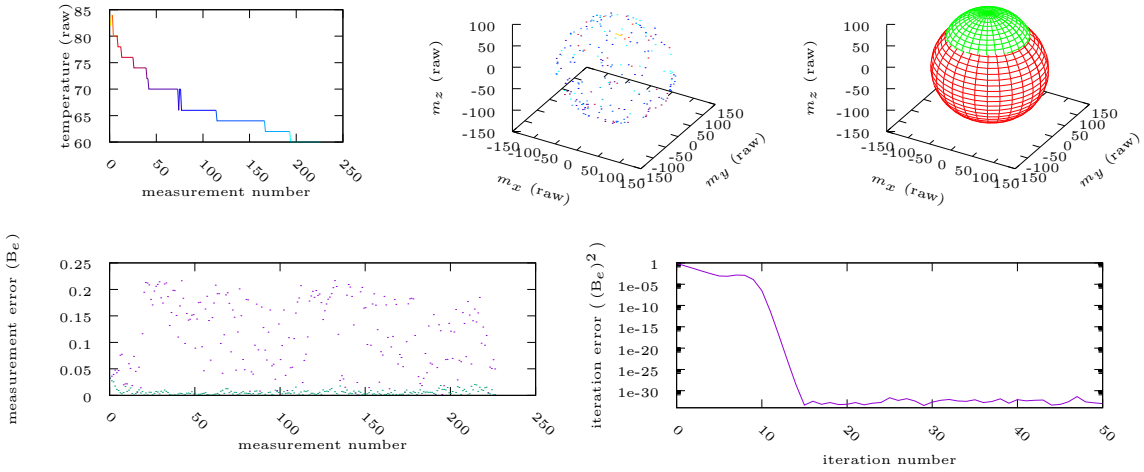


Figure B.47: Data representations: ID 16m_20190412110715 .

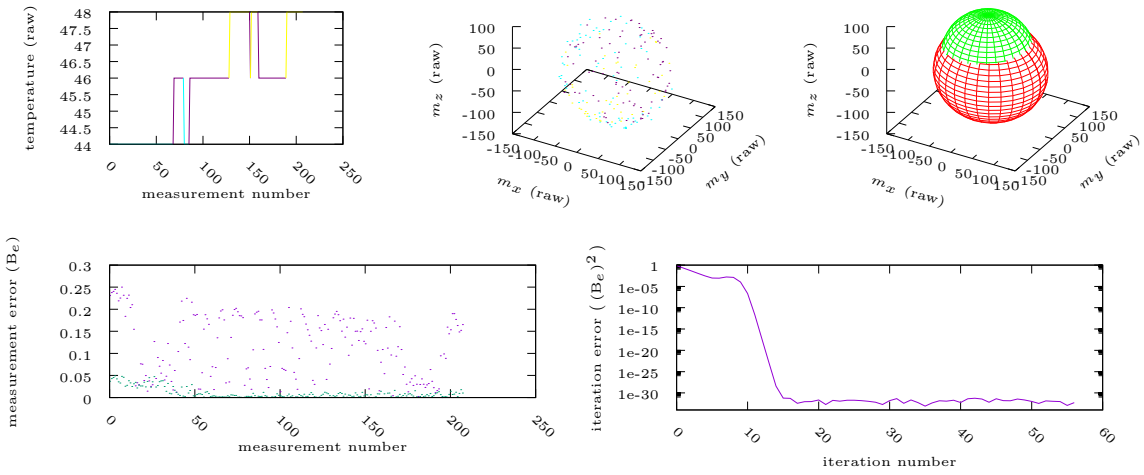


Figure B.48: Data representations: ID 16m_20190415220140 .

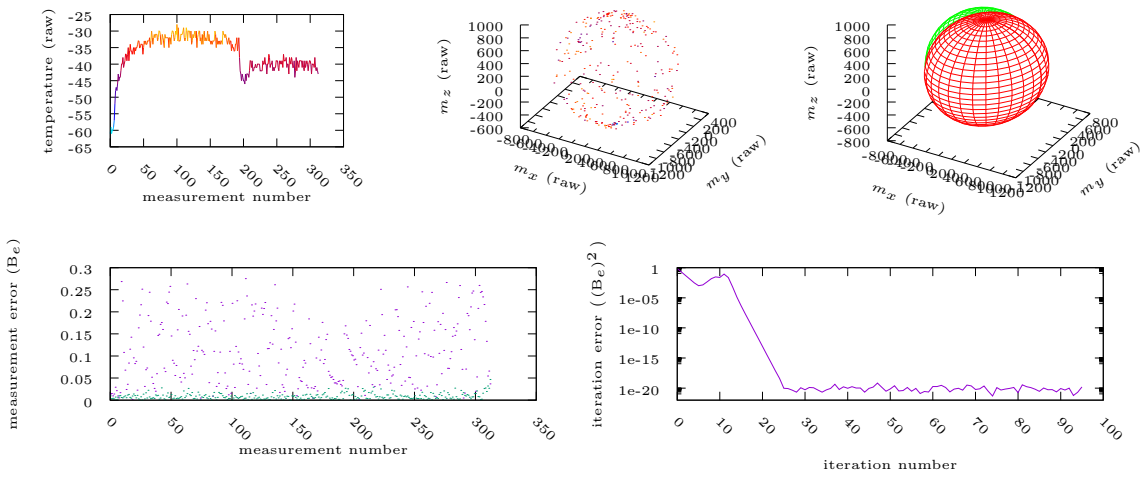


Figure B.49: Data representations: ID 17m_20190410123937 .

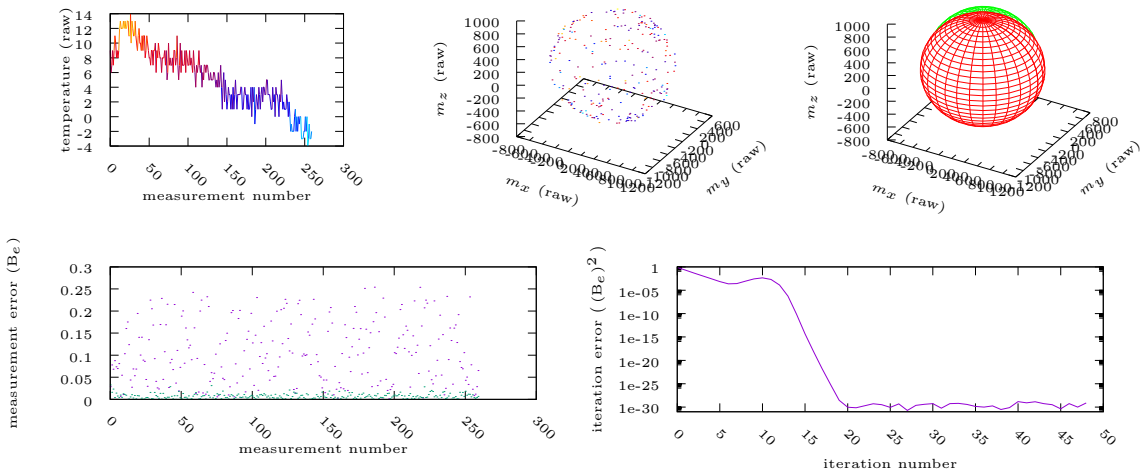


Figure B.50: Data representations: ID 17m_20190412110715 .

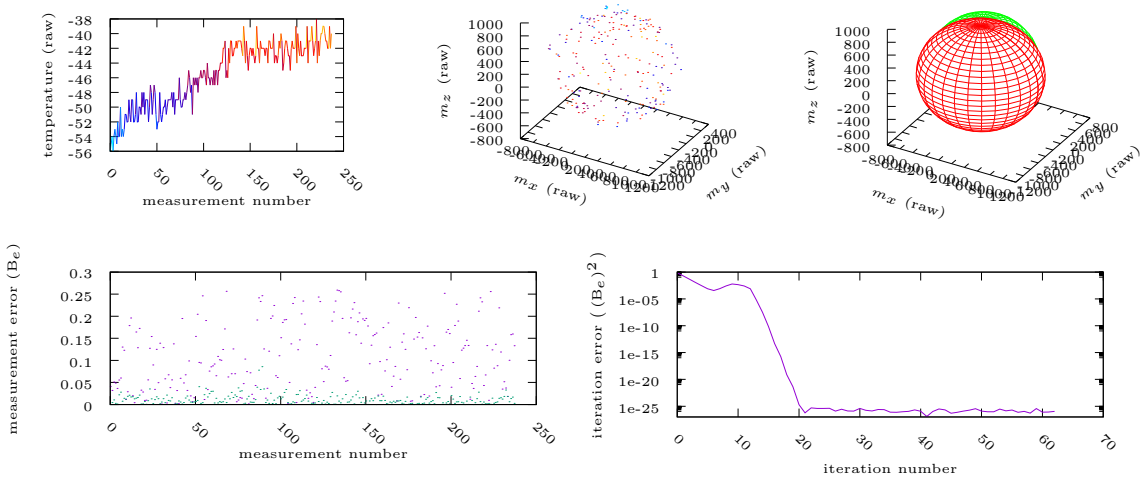


Figure B.51: Data representations: ID 17m_20190415220140 .

B.2. Calibration Error Graphics

B.2.1. Accelerometer Calibration Error Graphics

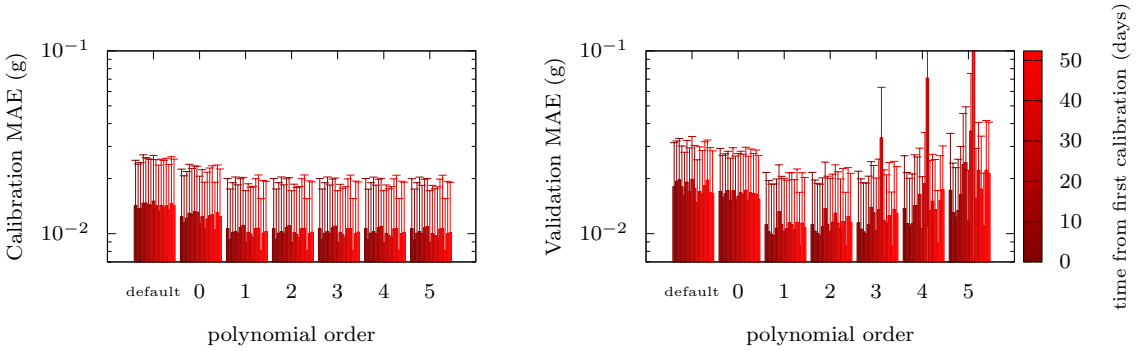


Figure B.52: Calibration errors vs polinomial order: ID 11a .

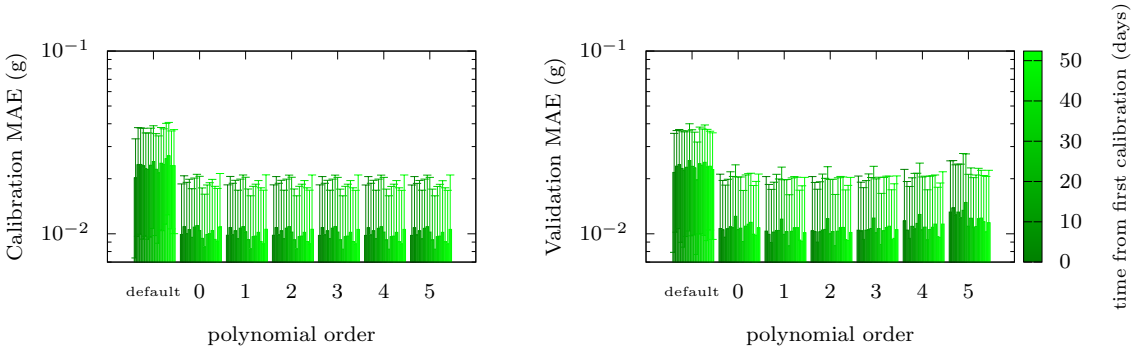


Figure B.53: Calibration errors vs polinomial order: ID 12a .

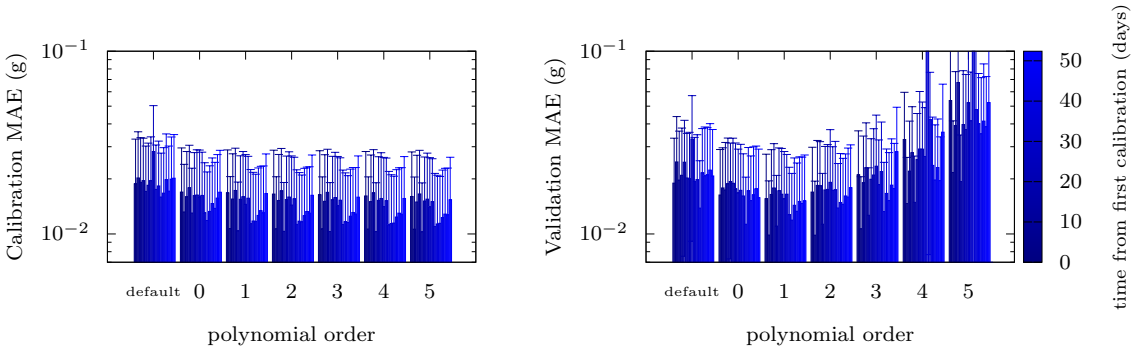


Figure B.54: Calibration errors vs polinomial order: ID 13a .

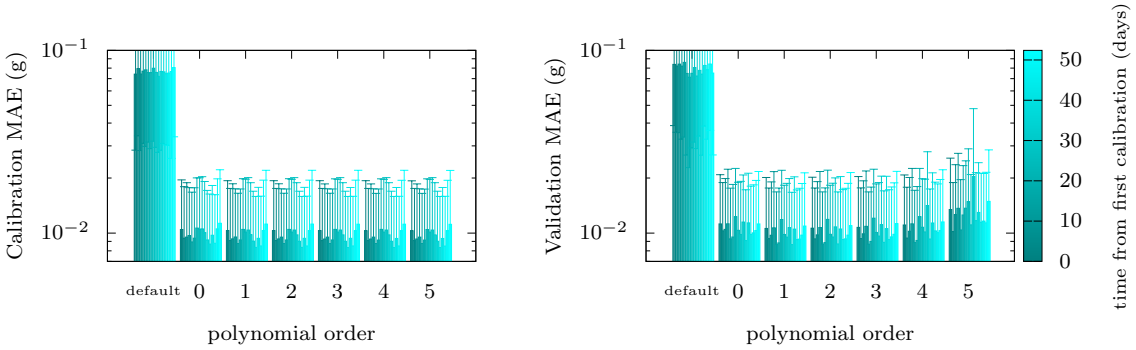


Figure B.55: Calibration errors vs polinomial order: ID 14a .

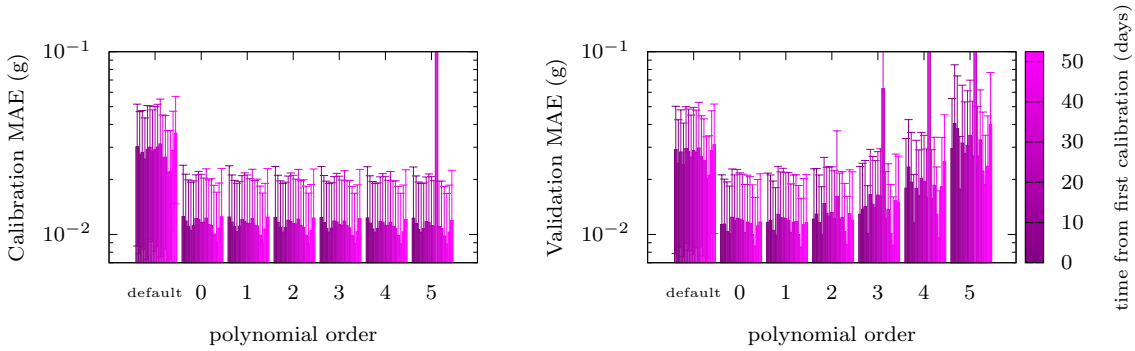


Figure B.56: Calibration errors vs polinomial order: ID 15a .

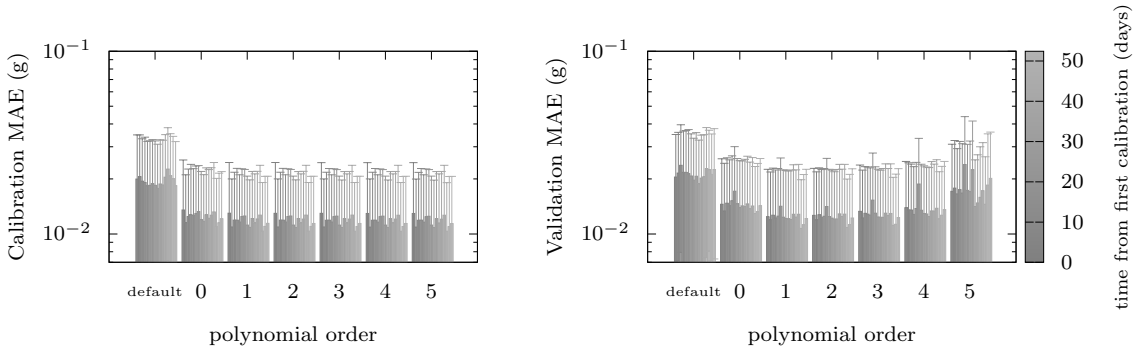


Figure B.57: Calibration errors vs polinomial order: ID 16a .

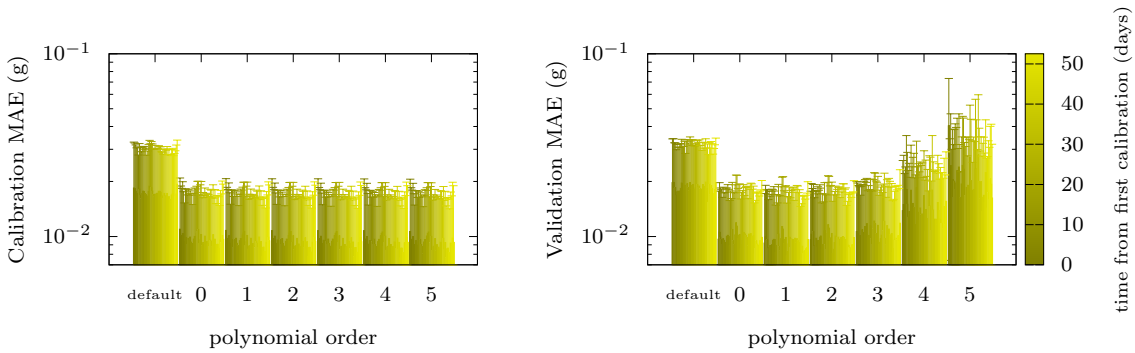


Figure B.58: Calibration errors vs polinomial order: ID 17a .

B.2.2. Gyroscope Calibration Error Graphics

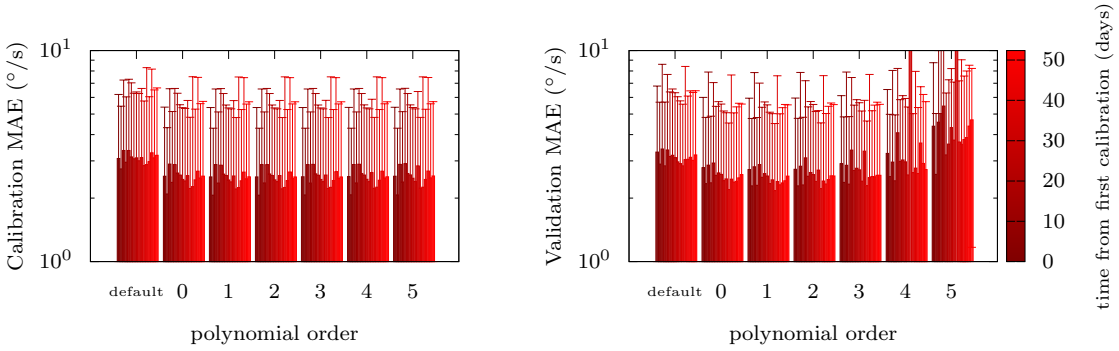


Figure B.59: Calibration errors vs polinomial order: ID 11w .

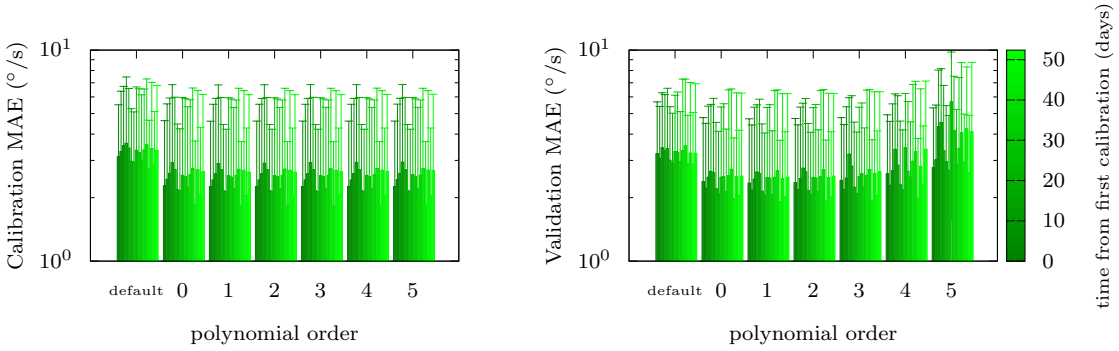


Figure B.60: Calibration errors vs polinomial order: ID 12w .

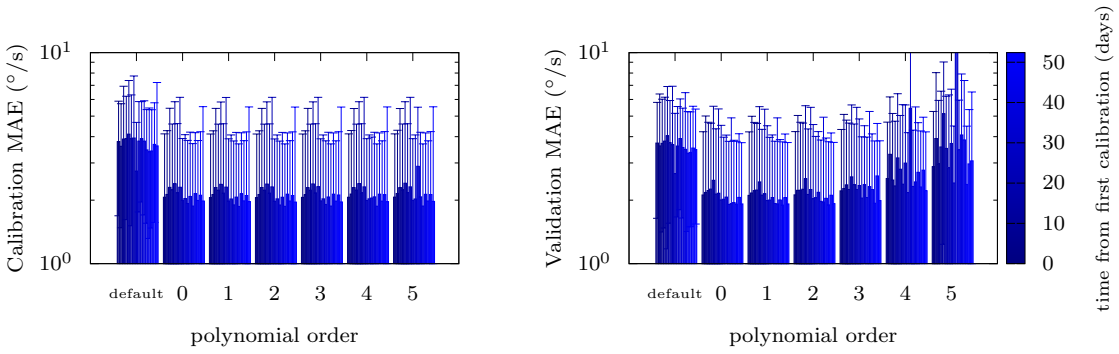


Figure B.61: Calibration errors vs polinomial order: ID 13w .

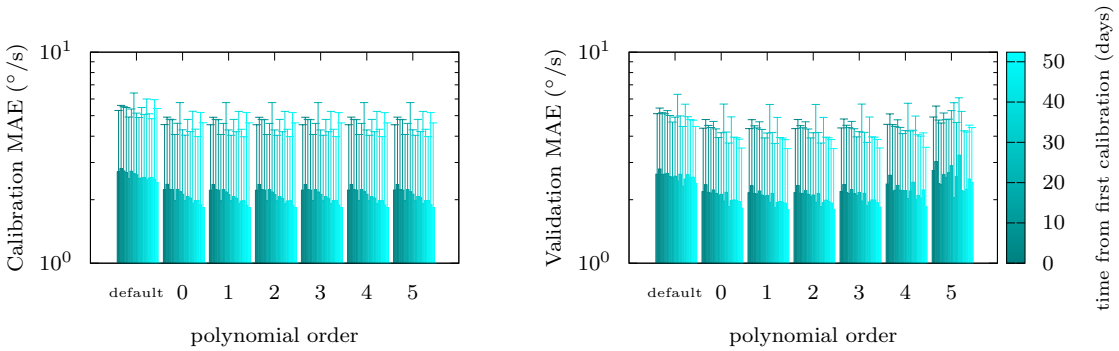


Figure B.62: Calibration errors vs polinomial order: ID 14w .

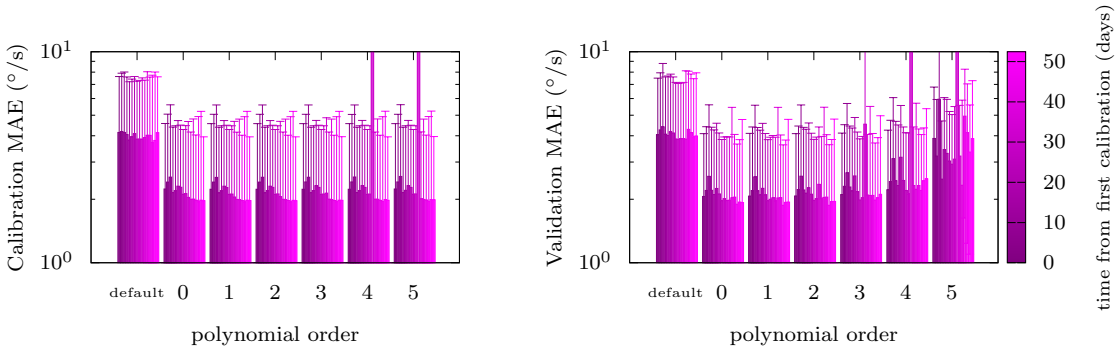


Figure B.63: Calibration errors vs polinomial order: ID 15w .

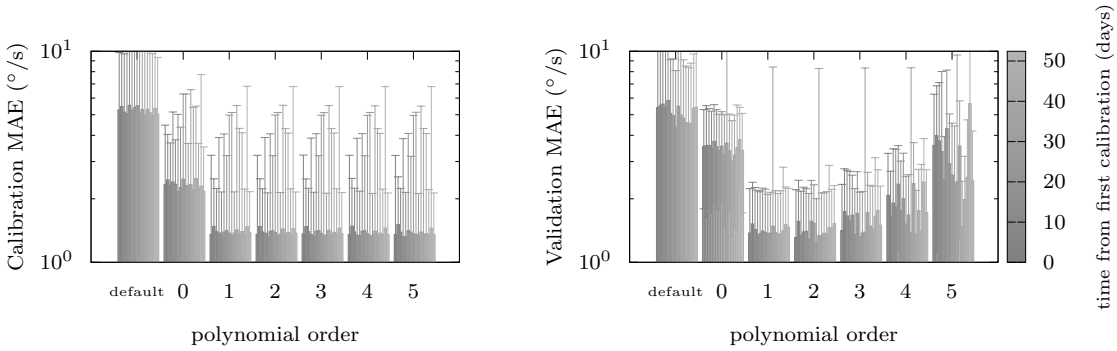


Figure B.64: Calibration errors vs polinomial order: ID 16w .

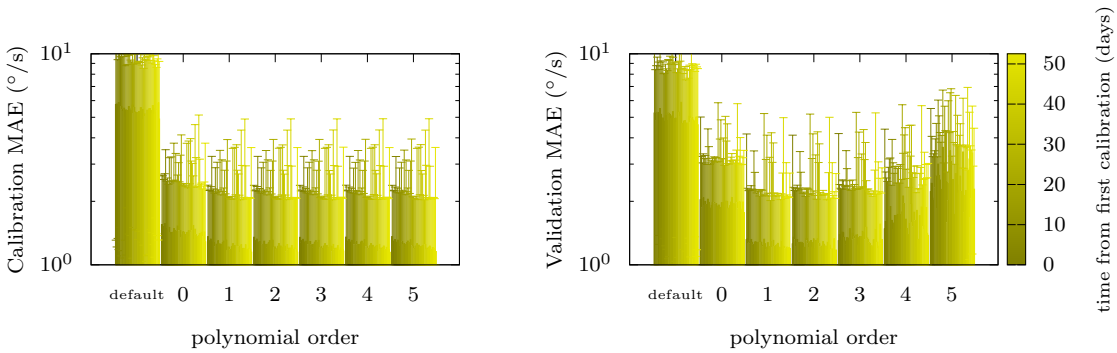


Figure B.65: Calibration errors vs polinomial order: ID 17w .

B.2.3. Magnetometer Calibration Error Graphics

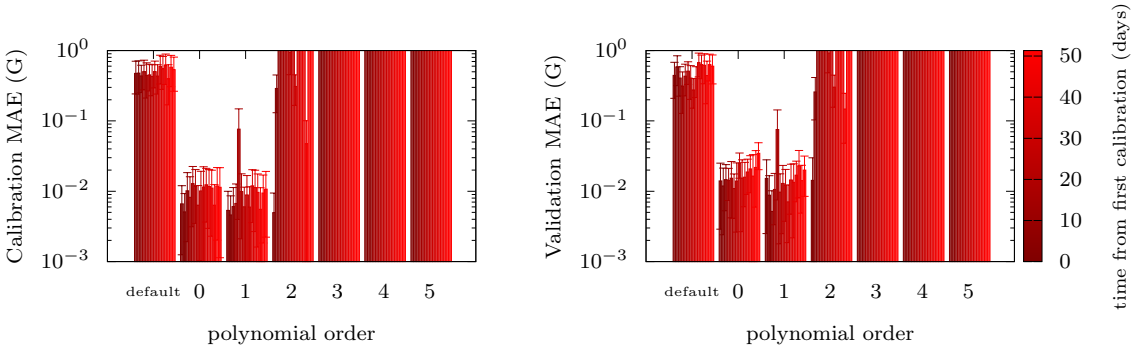


Figure B.66: Calibration errors vs polinomial order: ID 11m .

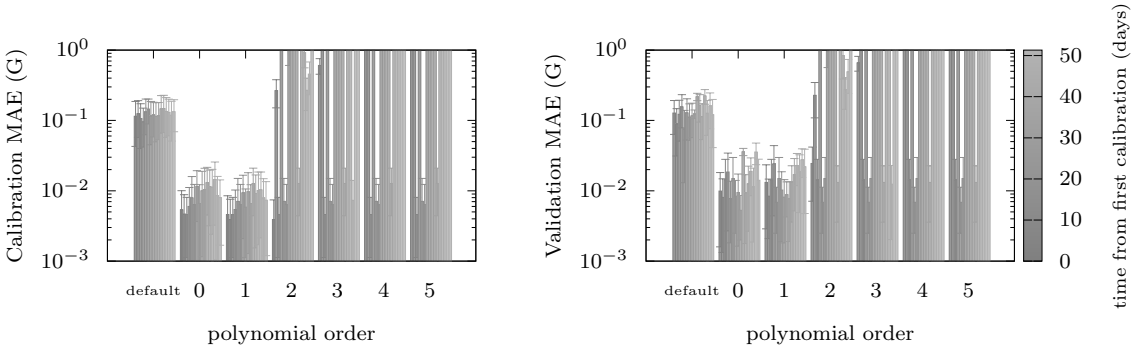


Figure B.67: Calibration errors vs polinomial order: ID 16m .

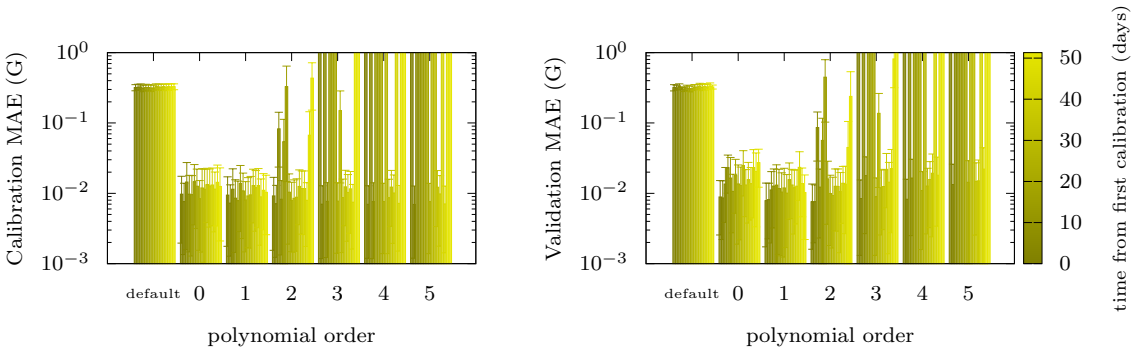


Figure B.68: Calibration errors vs polinomial order: ID 17m .

B.3. Calibration Parameters vs Time Graphics

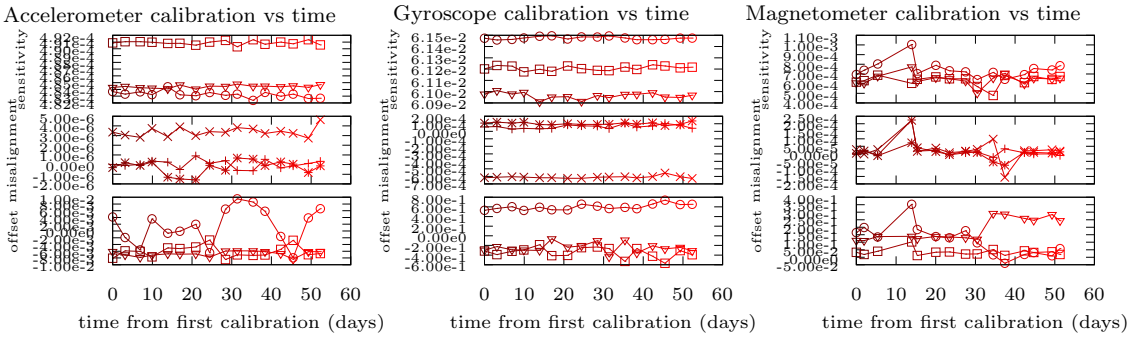


Figure B.69: Calibration parameters vs time: ID 11

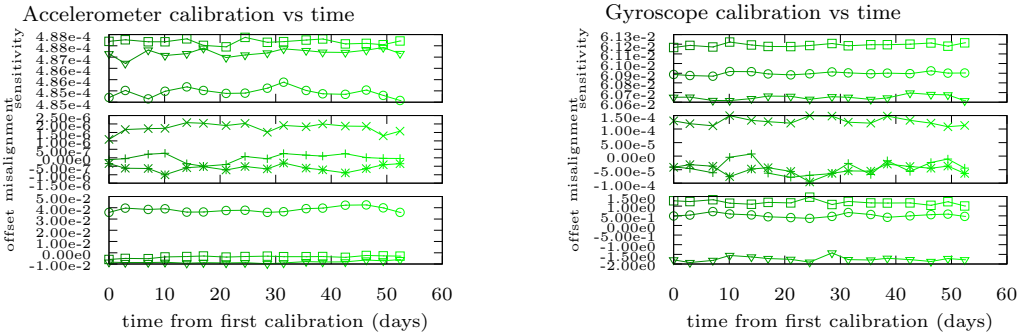


Figure B.70: Calibration parameters vs time: ID 12

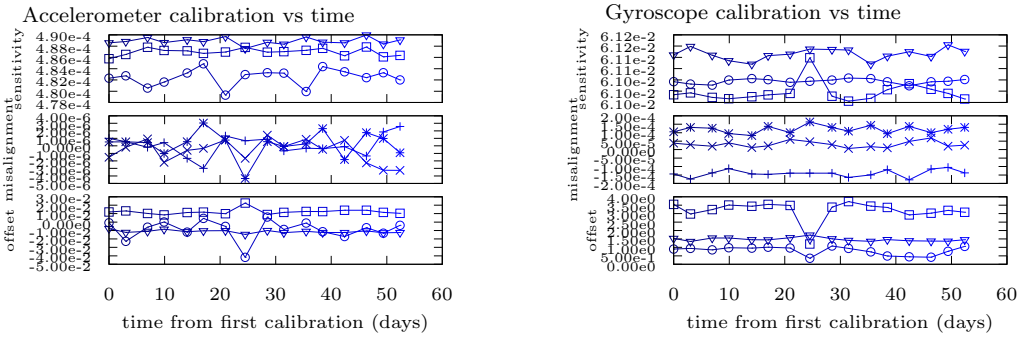


Figure B.71: Calibration parameters vs time: ID 13

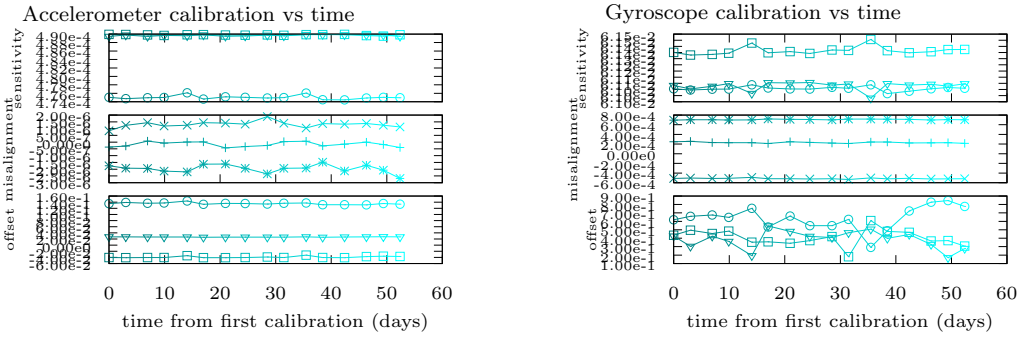


Figure B.72: Calibration parameters vs time: ID 14

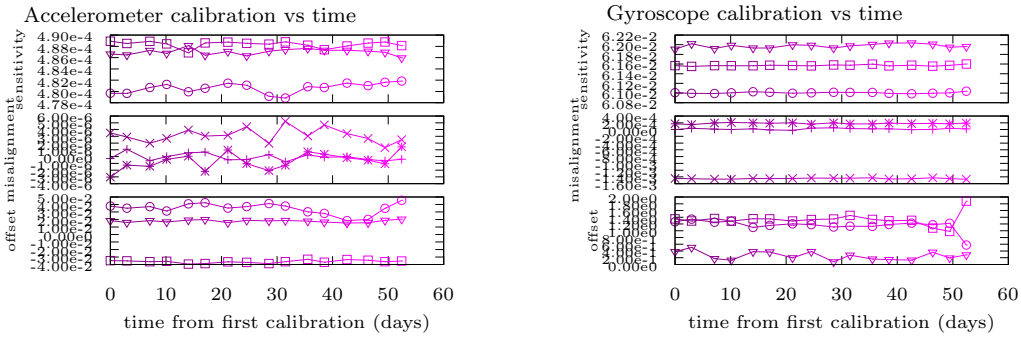


Figure B.73: Calibration parameters vs time: ID 15

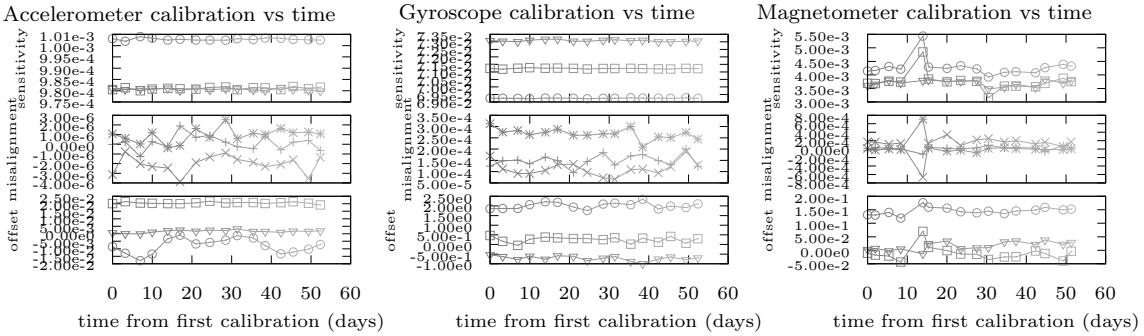


Figure B.74: Calibration parameters vs time: ID 16

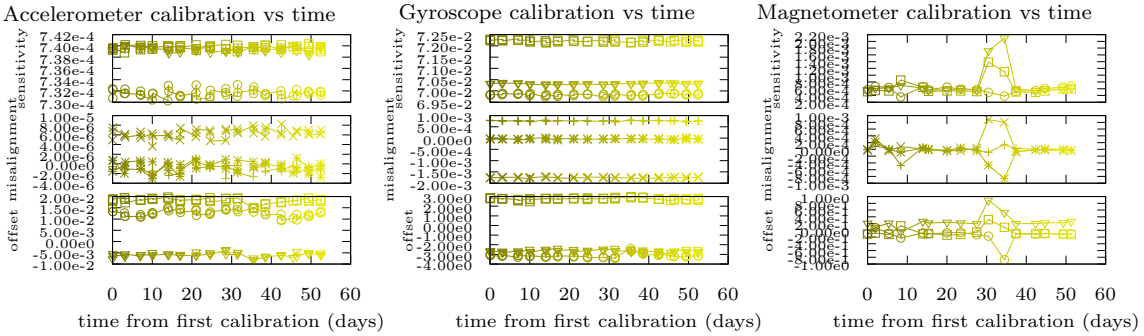


Figure B.75: Calibration parameters vs time: ID 17

Apéndice C

Resumen en Castellano

C.1. Motivación

En los últimos años, hemos sido testigos del nacimiento de tecnologías tales como la realidad virtual (VR en inglés), o los Vehículos Aéreos No-Tripulados (UAVs en inglés), coloquialmente conocidos como drones. Una de las razones principales por las que surgieron estas tecnologías fueron los avances en la fabricación de componentes miniaturizados, lo que resultó en los Sistemas Micro Electro Mecánicos (MEMS en inglés): son sistemas mecánicos en un chip. Un caso particular de MEMS son las Unidades de Medición Inercial (IMUs en inglés) que incluyen acelerómetros MEMS, que miden la aceleración propia, giróscopos MEMS, que miden la velocidad angular, y en algunos casos magnetómetros MEMS, que miden el campo magnético. Las IMUs clásicas eran dispositivos grandes, pesados, y caros. Pocos tenían acceso a este equipamiento, y solo podían ser instalados en sistemas especializados como submarinos, barcos, aviones, naves espaciales, satélites, y en algunos automóviles. Las IMUs MEMS surgieron como dispositivos pequeños, ligeros, y baratos. Esto hizo posible la instalación de IMUs en sistemas más pequeños, como las gafas de realidad virtual y drones mencionados anteriormente, y otros sistemas pequeños como robots, misiles, o dispositivos de captura de movimiento.

Las medidas de las IMUs contienen información sobre su orientación: las medidas del acelerómetro apuntan hacia abajo, y las medidas del giróscopo nos dan la velocidad de rotación. Conocer la orientación es importante en múltiples aplicaciones:

Control: algunos vehículos autónomos necesitan saber su orientación para calcular las salidas de control. Por ejemplo, un dron necesita saber su orientación para calcular la velocidad a la que las hélices deben rotar para acelerar en cierta dirección [1, 2]. Otro ejemplo es un cohete que necesita conocer su orientación para el Control de Empuje Vectorial (TVC en inglés): el sistema que controla la dirección del empuje desde su motor.

Navegación: hay dos razones principales por las que la orientación es importante para la navegación:

- Heading: la orientación determina la dirección de la parte delantera del vehículo, que tiende a ser la dirección de avance.
- Navegación inercial: las medidas del acelerómetro pueden usarse para estimar la posición de un vehículo [3]. Como las medidas se toman en un sistema de referencia no inercial (el vehículo), primero han de ser expresadas en un sistema de referencia inercial. Para esto necesitamos conocer la orientación del vehículo con respecto a ese sistema de referencia.

Realidad virtual: es necesario conocer la orientación de las gafas con respecto a un sistema de referencia estático para realizar la proyección de la escena virtual.

El creciente interés que suscitan estas aplicaciones nos lleva a investigar métodos para estimar la orientación de un sistema. Sin embargo, hay varios factores que dificultan la obtención de la orientación a través de medidas de IMU. Primero, las IMUs producen información parcial sobre el estado del sistema, lo que lo convierte en un sistema parcialmente observable. Entonces es necesario usar algún tipo de memoria para fusionar la información que portan las medidas a lo largo del tiempo. En segundo lugar, como cualquier otro sensor, las IMUs producen medidas imperfectas,

lo que degrada las estimaciones producidas por cualquier algoritmo de estimación alimentado con las mismas. Este problema se mitiga con una calibración. Sin embargo, dado que las IMUs son sensores triaxiales (miden vectores en lugar de cantidades escalares), el proceso de calibración presenta problemas adicionales.

Esta tesis aborda los dos problemas mencionados anteriormente. A saber,

- La estimación de la orientación de un sistema utilizando mediciones de IMU.
- La calibración de una IMU.

En la siguiente sección, pondremos en perspectiva estos problemas revisando la literatura científica relacionada.

C.2. Revisión de la Literatura

Esta sección presenta los dos problemas principales abordados en esta tesis. Revisamos la literatura científica que, de una forma u otra, ha influido en el desarrollo de este trabajo.

C.2.1. Estimación de la Orientación

Para estimar el estado dinámico de un sistema, generalmente usamos el modelo del sólido rígido. Un sólido rígido se describe usando cuatro objetos matemáticos: dos de ellos representan su posición y velocidad, y los otros dos representan su orientación y velocidad angular. Sin embargo, nuestros sensores no suelen medir estos objetos matemáticos de forma explícita; más bien, estos proporcionan medidas que contienen información sobre el estado dinámico de forma implícita. Por ejemplo, para las IMUs, la información sobre la velocidad angular se proporciona de forma explícita, pero la información sobre la orientación se proporciona "diciéndonos dónde es abajo". Para extraer de las medidas la información subyacente sobre el estado dinámico del sólido rígido, utilizamos algoritmos de estimación.

Aunque existen otras herramientas matemáticas que también se usan en estimación [4, 5], el filtro de Kalman [6] se ha convertido en el algoritmo por excelencia en este área. Debido a su simplicidad, el rigor y la elegancia en su derivación matemática, y su naturaleza recursiva, resulta ser muy atractivo para muchas aplicaciones prácticas. Sus versiones no lineales han sido ampliamente usadas en el campo de la estimación de la orientación: el Filtro de Kalman Extendido (EKF en inglés), y el Filtro de Kalman sin aroma (UKF en inglés. La "U" viene de "Unscented" que se traduce como "sin aroma", pero al parecer, el motivo de este nombre proviene de su creador, Jeffrey Uhlmann, el cual no quería llamarlo explícitamente "Uhlmann filter" y escogió la palabra "Unscented" de forma arbitraria) [7]. Sin embargo, existen problemas derivados de la parametrización utilizada para describir la orientación.

La orientación de un sistema es representada por la transformación de rotación que relaciona dos sistemas de referencia: un sistema de referencia anclado a nuestro sólido rígido, y un sistema de referencia externo. En [8] se proporciona una recopilación exhaustiva de representaciones de la orientación. La parametrización escogida para describir una transformación de rotación podría ser singular, o presentar discontinuidades, entros inconvenientes. La Tabla C.1 resume las características principales de las parametrizaciones más comunes.

Tabla C.1: Características principales de las parametrizaciones más utilizadas para representar una orientación.

Representación	**Parámetros**	**Continua**	**No singular**	**Ecuación de evolución lineal**
Ángulos de Euler	3	✗	✗	✗
Eje-ángulo	3–4	✗	✗	✗
Matriz de rotación	9	✓	✓	✓
Cuaternión unitario	4	✓	✓	✓

Sabiendo que el *grupo de rotación 3D* SO(3) tiene dimensión tres, idealmente buscaríamos una representación continua, y no singular descrita por 3 parámetros. Sin embargo, desde 1964 sabemos que es topológicamente imposible tener una parametrización tridimensional global sin

puntos singulares para el grupo de rotación 3D [9]. Sabiendo esto, no estaríamos equivocados al afirmar que los cuaterniones unitarios son la representación más conveniente que tenemos, y que tendremos, para orientaciones.

Los cuaterniones son entidades 4-dimensionales, pero solo aquellos con norma unidad representan una transformación de rotación. Este hecho conlleva un problema al aplicar el filtro de Kalman en su forma habitual, de modo que han surgido diferentes aproximaciones. Dado que un cuaternión tiene dimensión 4, lo primero en lo que uno piensa es en una matriz de covarianza de tamaño 4×4, y en la aplicación directa del filtro de Kalman [10]. Esta primera idea nos llevaría al fracaso. Dado que todas las predicciones están contenidas en la superficie definida por la restricción de la unitariedad de los cuaterniones, la distribución de cuaterniones se contraería en la dirección ortogonal a dicha superficie, lo que conduciría, después de varias actualizaciones del filtro, a una matriz de covarianza singular.

Otra perspectiva fue presentada inicialmente en [11], donde se revisa la literatura sobre estimación de la actitud hasta 1982, cuando otras parametrizaciones como los ángulos de Euler eran comunes. Ese trabajo establece las bases de la estimación moderna basada en cuaterniones, sobre las que se apoya este trabajo. La idea clave radica en definir un "cuaternión de error" que se transforma en un vector 3-dimensional. Tal vector se utiliza para construir la matriz de covarianza, y uno se refiere a la misma como la "representación 3×3 de la matriz de covarianza de cuaterniones". Tras ese trabajo, muchos otros han explorado este punto de vista, y han demostrado su superioridad. En [12], se aclara la relación existente entre el cuaternión de error y el vector 3-dimensional, y se bautiza el enfoque como "Filtro de Kalman Extendido Multiplicativo". En [13], se explora una transformación general de cuaternión de error al vector 3-dimensional bajo el marco del UKF. El algoritmo desarrollado se evalúa en simulaciones utilizando un modelo de nave espacial realista. En [14] se compara el "Filtro de Kalman Extendido Multiplicativo" con el "Filtro de Kalman Extendido Aditivo", y se concluye que no se ve una razón válida para preferir el segundo al primero. Del mismo modo, en [15], el "Filtro de Kalman Extendido Multiplicativo" se compara con varios enfoques en los que la actitud se estima a partir de estimaciones de cuaterniones sin restricciones, y se alcanzan las mismas conclusiones. En [16] se desarrolla un algoritmo basado en el UKF para estimar tanto la actitud del sistema como sus parámetros. El algoritmo se compara con otro basado en el EKF a través de simulaciones. En [17] se deriva un algoritmo equivalente al "Filtro de Kalman Extendido Multiplicativo" para estimar la orientación de una nave espacial utilizando mediciones de una IMU y un sensor solar. También revisa las derivaciones, propiedades y relaciones entre los objetos matemáticos que aparecen en el algoritmo. En [18] se presenta un algoritmo claramente inspirado en el "Filtro de Kalman Extendido Multiplicativo" para estimar la orientación de un Vehículo Aéreo Miniaturizado (MAV) usando una IMU y un GPS. En [19] se presenta un algoritmo que usa ideas comparables a las del "Filtro de Kalman Extendido Multiplicativo" para estimar la pose de un vehículo usando una IMU y una cámara. El algoritmo se prueba en simulaciones y en experimentos reales. El anterior fue el precursor de muchos otros trabajos exitosos basados en los mismos principios [20, 21, 22, 23, 24, 25, 26].

Aunque el "Filtro de Kalman Extendido Multiplicativo" ha sido usado, y sigue usándose ampliamente para estimar la orientación, todavía hay detalles en esta adaptación que se están desarrollando actualmente. A saber, el "paso de corrección de la matriz de covarianza" [27].

C.2.2. Calibración de IMUs

Se ha dedicado un esfuerzo sustancial al desarrollo de algoritmos de calibración para sensores triaxiales y, como resultado, han surgido varias estrategias. Cuando se trata de la calibración de un acelerómetro, es común explotar el hecho de que la magnitud del vector de gravedad debe ser independiente de la orientación del sensor [28, 29, 30, 31]. Otros trabajos acomodan esta idea para calibrar magnetómetros [32, 33, 34, 35]. La ventaja de este enfoque es que no necesitamos conocer la orientación real del sensor para realizar la calibración; solo se requiere la magnitud del vector medido. Naturalmente, hay trabajos en los que esta idea no se utiliza [36].

Para calibrar un sensor es necesario elegir un modelo. El modelo describe el comportamiento de nuestro sensor; cómo se relacionan las mediciones con las cantidades físicas que queremos medir. Es común usar un modelo lineal para el sensor [28, 29, 30, 32, 33, 34, 31, 35, 36]. Pero hay trabajos en los que se han propuesto modelos no lineales [37]. En [38] incluso se usa un modelo de red neuronal. Sin embargo, el uso de un modelo tan complejo como una red neuronal implicaría una sobrecarga computacional probablemente mayor que la de un algoritmo de estimación. Una sobrecarga

computacional de esa magnitud para obtener las mediciones calibradas sería desproporcionada.

Se han realizado calibraciones utilizando tanto algoritmos online como offline. Los algoritmos de calibración online tienen la ventaja de no tener que realizar una calibración antes de usar el sensor. En contra, implican una mayor sobrecarga computacional online y una mayor complejidad del algoritmo de estimación, lo que generalmente implica la simplificación del modelo del sensor. La referencia [12] es un ejemplo de algoritmo de calibración online en el que solo se calibran los offsets del giróscopo. Por otro lado, las calibraciones offline deben realizarse antes de usar el sensor, pero permiten la calibración para modelos de sensores complejos y, después de la calibración, las medidas corregidas se obtienen de manera muy eficiente. Por lo tanto, este segundo enfoque es la opción preferida.

Los algoritmos de calibración offline suelen ser métodos para encontrar la solución a un problema de optimización, que generalmente consiste en la minimización de una función de coste. Por supuesto, no siempre se usa la estrategia de minimizar una función de coste. En [36] usan un filtro de Kalman fuera de línea para encontrar los parámetros del modelo.

Hay sensores cuyas medidas dependen mucho de la temperatura, como es el caso que se muestra en la Figura C.1. Aún así, ninguno de los trabajos mencionados anteriormente considera una

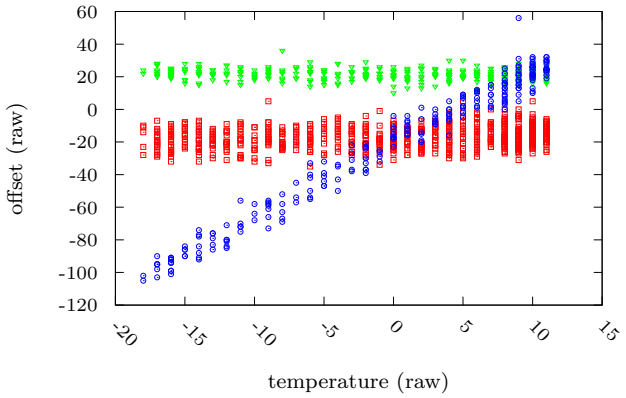


Figura C.1: Ejemplo de un sensor cuyas mediciones dependen en gran medida de la temperatura. Es el giróscopo contenido en el chip L3GD20H (Adafruit 9-DOF). Las medidas del eje x son cuadrados rojos. Las medidas del eje y son triángulos verdes. Las medidas del eje z son círculos azules. Las medidas se recolectaron variando la temperatura de un sensor que permaneció estático.

calibración dependiente de la temperatura. A saber por el autor, hay pocos trabajos que aborden este problema. Un buen ejemplo es [39], donde se realiza una calibración no lineal dependiente de la temperatura de un giróscopo de fibra óptica (FOG). Sin embargo, no se trata de una calibración de un sensor triaxial.

C.3. Resumen de Contenidos

En este trabajo se expone una nueva perspectiva sobre el problema de la estimación de la actitud usando filtros de Kalman cuando la orientación es representada por cuaterniones unitarios. Para ello es necesario reflexionar sobre lo que significa el "cuaternión de error" y la "representación 3×3 de la matriz de covarianza de cuaterniones". Con el fin de explorar estos conceptos es necesario repasar algunos otros conceptos básicos.

En el Capítulo 2 se repasan los conceptos básicos sobre el filtro de Kalman. Se comienza repasando las definiciones de *variable aleatoria*, *valor esperado*, o *momento* de una distribución. A partir de estos, se define el concepto de *media*, y de *matriz de covarianza* de una distribución, los cuales son esenciales a la hora de comprender el filtro de Kalman. También se repasa la forma en la que se transforman (de forma aproximada) la media y la matriz de covarianza de una distribución al aplicarles una función. Se continúa recordando las asunciones principales, y como las mismas conducen a la forma habitual del filtro de Kalman. También se repasan las versiones no lineales del filtro: el EKF, y el UKF. Para el EKF también se repasa la versión continua-discreta del filtro, la cual suele usarse en sistemas reales, donde las ecuaciones del sistema vienen descritas por ecuaciones diferenciales, y las ecuaciones de medida son descritas por ecuaciones discretas.

En el Capítulo 3 se repasa la definición y las propiedades básicas de los cuaterniones. También se incluye una explicación sobre cómo los cuaterniones ejecutan rotaciones en el espacio 4-dimensional $\mathbb{R}^4$, y de cómo estas entidades 4-dimensionales también logran describir rotaciones en el espacio 3-dimensional $\mathbb{R}^3$. Después, se recuerdan algunas definiciones básicas importadas de la teoría de variedades, que nos serán de utilidad a la hora de definir la media y la matriz de covarianza de una distribución de cuaterniones unitarios en ese mismo capítulo. En particular, el concepto de *carta* resulta ser de especial interés. Por último, se recuerda el concepto de mapa de transición, que nos será de utilidad a la hora de abordar el problema del "paso de corrección de la matriz de covarianza" en el Capítulo 5.

En el Capítulo 4 se presentan las ecuaciones que usamos para modelar el sistema. Se comienza presentando una nomenclatura que resulta conveniente para analizar cómo se relacionan distintos sistemas de referencia. Después, se usa dicha nomenclatura para encontrar las ecuaciones diferenciales para la orientación, y para la velocidad angular de un sólido rígido, habiendo derivado previamente la relación que existe entre la derivada temporal de un vector expresado en un sistema de referencia inercial, y la del mismo vector expresado en un sistema de referencia no inercial. Finalmente, se encuentran las ecuaciones que definen las relaciones existentes entre las medidas de una IMU (compuesta por un acelerómetro, un giróscopo, y en ocasiones un magnetómetro), y el estado del sólido rígido (orientación, representada por un cuaternion, y velocidad angular).

El Capítulo 5 representa la culminación de la tesis. Su propósito es múltiple. En primer lugar, se pretende diseñar, en un contexto simple, un filtro de Kalman que incluya cuaterniones unitarios (los que describen la orientación) como parte del estado; evitando modelos complicados que desvíen la atención. De esta manera, la extrapolación a modelos más complicados debería ser más simple. En segundo lugar, se pretende que este diseño simple sirva como base para el diseño de otros filtros de Kalman cuyo estado se defina en una variedad; otros Manifold Kalman filters. Teniendo en cuenta estos objetivos, se comienza proponiendo un modelo más sencillo que el desarrollado en el Capítulo 4. También se define el estado del sistema, compuesto por la orientación y la velocidad angular del sólido rígido. Tras esto, se presentan los diseños de los estimadores de la orientación basados en el EKF y en el UKF. El caso del estimador basado en el EKF es especialmente interesante, ya que se resuelve el problema del "paso de corrección de la matriz de covarianza". Una vez diseñados los estimadores, estos son testeados con IMUs reales, y también son examinados por medio de una simulación. Para ello se define una métrica que mide la precisión de las estimaciones de un algoritmo. Después, se genera un conjunto de trayectorias, cada una con un conjunto de medidas asociado. Cada algoritmo se alimenta con cada conjunto de medidas para generar una trayectoria estimada. Esa trayectoria estimada se compara con la trayectoria generada, por medio de la métrica antes definida, para obtener finalmente una medida de la calidad de las estimaciones. Este proceso se usa para comparar algoritmos (EKF vs UKF), cartas (O vs RP vs MRP vs RV), y para comparar los resultados de usar el "paso de corrección de la matriz de covarianza" o no usarlo. Las conclusiones que se extraen de las simulaciones son las siguientes:

- No hay ninguna carta que presente una ventaja clara sobre las otras, pero la carta RP tiene algunas características que nos motivan a preferirla.
- El filtro basado en el EKF (el MEKF) es preferible al filtro basado en el UKF (el MUKF) debido a su menor costo computacional, y a la mayor exactitud de sus estimaciones de orientación.
- El "paso de corrección de la matriz de covarianza" no es necesario en la práctica. No se observa ninguna mejoría al utilizarlo.

Entonces, el MEKF con la carta RP y desechando el "paso de corrección de la matriz de covarianza" es seleccionado como nuestro mejor estimador de la orientación de acuerdo a la métrica adoptada. Tal algoritmo es equiparable al algoritmo llamado "Multiplicative Extended Kalman Filter", pero en nuestro caso el MEKF se obtuvo sin necesidad de re-definir ningún aspecto del filtro de Kalman clásico: tan solo fue necesario definir los momentos de una distribución de cuaterniones unitarios.

Por último, en el Capítulo 6 se presenta un método para calibrar un sensor triaxial considerando su dependencia con la temperatura. Se comienza revisando el modelo del sensor. Esto nos conduce a proponer un modelo híbrido, con una dependencia lineal de las salidas del sensor, pero una dependencia no lineal de la temperatura. Se continúa definiendo el algoritmo de calibración, que es un método iterativo que se ejecuta offline. También se escribe sobre los prototipos que se

diseñaron para coleccionar datos de sensores triaxiales reales. Los datos de calibración coleccionados con la versión final del prototipo se usan para testear el algoritmo de calibración. Finalmente, se presentan algunos resultados. En concreto, se estudia la dependencia de la temperatura de los sensores, cómo las calibraciones cambian a lo largo del tiempo, y cómo la calibración retrasa el deterioro de las estimaciones de velocidad y posición.

En definitiva, este trabajo presenta una base sólida para el desarrollo de algoritmos usados para estimar el estado dinámico de un sistema en el que la orientación se considera parte del estado. También aborda satisfactoriamente el problema de la calibración de sensores triaxiales dependientes de la temperatura.

Bibliography

[1] A. Tayebi and S. McGilvray, "Attitude stabilization of a vtol quadrotor aircraft," *IEEE Transactions on control systems technology*, vol. 14, no. 3, pp. 562–571, 2006.

[2] E. Fresk and G. Nikolakopoulos, "Full quaternion based attitude control for a quadrotor," in *2013 European Control Conference (ECC)*, pp. 3864–3869, IEEE, 2013.

[3] O. J. Woodman, "An introduction to inertial navigation," tech. rep., University of Cambridge, Computer Laboratory, 2007.

[4] J. L. Crassidis, F. L. Markley, and Y. Cheng, "Survey of nonlinear attitude estimation methods," *Journal of guidance, control, and dynamics*, vol. 30, no. 1, pp. 12–28, 2007.

[5] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, "Estimation of imu and marg orientation using a gradient descent algorithm," in *2011 IEEE international conference on rehabilitation robotics*, pp. 1–7, IEEE, 2011.

[6] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[7] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *Signal processing, sensor fusion, and target recognition VI*, vol. 3068, pp. 182–194, International Society for Optics and Photonics, 1997.

[8] M. D. Shuster, "A survey of attitude representations," *Navigation*, vol. 8, no. 9, pp. 439–517, 1993.

[9] J. Stuelpnagel, "On the parametrization of the three-dimensional rotation group," *SIAM review*, vol. 6, no. 4, pp. 422–430, 1964.

[10] I. Bar-Itzhack and Y. Oshman, "Attitude determination from vector observations: Quaternion estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 21, no. 1, pp. 128–136, 1985.

[11] E. J. Lefferts, F. L. Markley, and M. D. Shuster, "Kalman filtering for spacecraft attitude estimation," *Journal of Guidance, Control, and Dynamics*, vol. 5, no. 5, pp. 417–429, 1982.

[12] F. L. Markley, "Attitude error representations for kalman filtering," *Journal of guidance, control, and dynamics*, vol. 26, no. 2, pp. 311–317, 2003.

[13] J. L. Crassidis and F. L. Markley, "Unscented filtering for spacecraft attitude estimation," *Journal of guidance, control, and dynamics*, vol. 26, no. 4, pp. 536–542, 2003.

[14] F. L. Markley, "Multiplicative vs. additive filtering for spacecraft attitude determination," *Dynamics and Control of Systems and Structures in Space*, pp. 467–474, 2004.

[15] F. L. Markley, "Attitude estimation or quaternion estimation?," *Journal of the Astronautical Sciences*, vol. 52, no. 1-2, pp. 221–238, 2004.

[16] M. C. VanDyke, J. L. Schwartz, C. D. Hall, *et al.*, "Unscented kalman filtering for spacecraft attitude state and parameter estimation," *Advances in the Astronautical Sciences*, vol. 118, no. 1, pp. 217–228, 2004.

[17] N. Trawny and S. I. Roumeliotis, "Indirect kalman filter for 3d attitude estimation," *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, vol. 2, p. 2005, 2005.

[18] J. K. Hall, N. B. Knoebel, and T. W. McLain, "Quaternion attitude estimation for miniature air vehicles using a multiplicative extended kalman filter," in *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pp. 1230–1237, IEEE, 2008.

[19] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 3565–3572, IEEE, 2007.

[20] A. I. Mourikis, N. Trawny, S. I. Roumeliotis, A. E. Johnson, and L. H. Matthies, "Vision-aided inertial navigation for precise planetary landing: Analysis and experiments.," in *Robotics: Science and Systems*, Atlanta, GA, 2007.

[21] N. Trawny, A. I. Mourikis, S. I. Roumeliotis, A. E. Johnson, and J. F. Montgomery, "Vision-aided inertial navigation for pin-point landing using observations of mapped landmarks," *Journal of Field Robotics*, vol. 24, no. 5, pp. 357–378, 2007.

[22] F. M. Mirzaei and S. I. Roumeliotis, "A kalman filter-based algorithm for imu-camera calibration: Observability analysis and performance evaluation," *IEEE transactions on robotics*, vol. 24, no. 5, pp. 1143–1156, 2008.

[23] A. I. Mourikis, N. Trawny, S. I. Roumeliotis, A. E. Johnson, A. Ansar, and L. Matthies, "Vision-aided inertial navigation for spacecraft entry, descent, and landing," *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 264–280, 2009.

[24] M. Li and A. I. Mourikis, "3-d motion estimation and online temporal calibration for camera-imu systems," in *2013 IEEE International Conference on Robotics and Automation*, pp. 5709–5716, IEEE, 2013.

[25] M. Li, H. Yu, X. Zheng, and A. I. Mourikis, "High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 409–416, IEEE, 2014.

[26] M. Li and A. I. Mourikis, "Online temporal calibration for camera–imu systems: Theory and algorithms," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 947–964, 2014.

[27] M. W. Mueller, M. Hehn, and R. D'Andrea, "Covariance correction step for kalman filtering with an attitude," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 9, pp. 2301–2306, 2016.

[28] I. Skog and P. Händel, "Calibration of a mems inertial measurement unit," in *XVII IMEKO World Congress*, pp. 1–6, 2006.

[29] W. Fong, S. Ong, and A. Nee, "Methods for in-field user calibration of an inertial measurement unit without external equipment," *Measurement Science and technology*, vol. 19, no. 8, p. 085202, 2008.

[30] H. Zhang, Y. Wu, W. Wu, M. Wu, and X. Hu, "Improved multi-position calibration for inertial measurement units," *Measurement Science and Technology*, vol. 21, no. 1, p. 015107, 2009.

[31] M. Sipos, P. Paces, J. Rohac, and P. Novacek, "Analyses of triaxial accelerometer calibration algorithms," *IEEE Sensors Journal*, vol. 12, no. 5, pp. 1157–1165, 2011.

[32] E. Dorveaux, D. Vissière, A.-P. Martin, and N. Petit, "Iterative calibration method for inertial and magnetic sensors," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 8296–8303, IEEE, 2009.

[33] S. Bonnet, C. Bassompierre, C. Godin, S. Lesecq, and A. Barraud, "Calibration methods for inertial and magnetic sensors," *Sensors and Actuators A: Physical*, vol. 156, no. 2, pp. 302–311, 2009.

[34] V. Renaudin, M. H. Afzal, and G. Lachapelle, "Complete triaxis magnetometer calibration in the magnetic domain," *Journal of sensors*, vol. 2010, 2010.

[35] J. F. Vasconcelos, G. Elkaim, C. Silvestre, P. Oliveira, and B. Cardeira, "Geometric approach to strapdown magnetometer calibration in sensor frame," *IEEE Transactions on Aerospace and Electronic systems*, vol. 47, no. 2, pp. 1293–1306, 2011.

[36] T. Beravs, S. Beguš, J. Podobnik, and M. Munih, "Magnetometer calibration using kalman filter covariance matrix for online estimation of magnetic field orientation," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 8, pp. 2013–2020, 2014.

[37] P. Batista, C. Silvestre, P. Oliveira, and B. Cardeira, "Accelerometer calibration and dynamic bias and gravity estimation: Analysis, design, and experimental evaluation," *IEEE transactions on control systems technology*, vol. 19, no. 5, pp. 1128–1137, 2010.

[38] J.-H. Wang and Y. Gao, "A new magnetic compass calibration algorithm using neural networks," *Measurement Science and Technology*, vol. 17, no. 1, p. 153, 2005.

[39] L. Ojeda, H. Chung, and J. Borenstein, "Precision calibration of fiber-optics gyroscopes for mobile robot navigation," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 3, pp. 2064–2069, IEEE, 2000.

[40] R. G. Reynolds, "Asymptotically optimal attitude filtering with guaranteed convergence," *Journal of guidance, control, and dynamics*, vol. 31, no. 1, pp. 114–122, 2008.

[41] F. L. Markley, "Lessons learned," *The Journal of the Astronautical Sciences*, vol. 57, no. 1-2, pp. 3–29, 2009.

[42] jrowberg, "i2cdevlib." `https://github.com/jrowberg/i2cdevlib`, 2019. [Online software; accessed January-2020].

[43] P. Bernal-Polo, "RPiModules." `https://github.com/PBernalPolo/RPiModules`, 2019. [Online software; accessed January-2020].

[44] P. Bernal-Polo and H. Martínez-Barberá, "Orientation estimation by means of extended kalman filter, quaternions, and charts," *Journal of Physical Agents*, vol. 8, no. 1, pp. 11–24, 2017.

[45] P. Bernal-Polo and H. Martínez-Barberá, "Triaxial sensor calibration: A prototype for accelerometer and gyroscope calibration," in *Iberian Robotics conference*, pp. 79–90, Springer, 2017.

[46] P. Bernal-Polo and H. Martínez-Barberá, "First steps towards a general algorithm to estimate the mechanical state of a vehicle," in *Workshop of Physical Agents*, pp. 319–332, Springer, 2018.

[47] P. Bernal-Polo and H. Martínez-Barberá, "Kalman filtering for attitude estimation with quaternions and concepts from manifold theory," *Sensors*, vol. 19, no. 1, p. 149, 2019.

[48] P. Bernal-Polo and H. Martínez-Barberá, "Temperature-dependent calibration of triaxial sensors: Algorithm, prototype, and some results," *IEEE Sensors Journal*, vol. 20, no. 2, pp. 876–884, 2019.

[49] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, pp. 153–158, Ieee, 2000.

[50] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[51] B. M. Bell and F. W. Cathey, "The iterated kalman filter update as a gauss-newton method," *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 294–297, 1993.

[52] W. R. Hamilton, "On a new species of imaginary quantities, connected with the theory of quaternions," *Proceedings of the Royal Irish Academy (1836-1869)*, vol. 2, pp. 424–434, 1843.

[53] R. P. Graves, *Life of Sir William Rowan Hamilton*, vol. 2, pp. 434–435. Hodges, 1885.

[54] H. Frobenius, "Über lineare substitutionen und bilineare formen," *Journal für die reine und angewandte Mathematik (Crelles Journal)*, vol. 1878, no. 84, pp. 1–63, 1878.

[55] R. E. Bradley and C. E. Sandifer, *Leonhard Euler: Life, Work and Legacy*, vol. 5, pp. 191–194. Elsevier, 2007.

[56] C. F. Gauss, *Carl Friedrich Gauss Werke: Arithmetik und Algebra*, vol. 8, pp. 357–362. Gedruckt in der Dieterichschen Universitäts-Buchdruckerei, 1900.

[57] C. Gramkow, "On averaging rotations," *Journal of Mathematical Imaging and Vision*, vol. 15, no. 1-2, pp. 7–16, 2001.

[58] J. J. LaViola, "A comparison of unscented and extended kalman filtering for estimating quaternion motion," in *American Control Conference, 2003. Proceedings of the 2003*, vol. 3, pp. 2435–2440, IEEE, 2003.