

# Orientation Estimation by Means of Extended Kalman Filter, Quaternions, and Charts

P. Bernal-Polo and H. Martínez-Barberá

**Abstract**—An orientation estimation algorithm is presented. This algorithm is based on the Extended Kalman Filter, and uses quaternions as the orientation descriptor. For the filter update, we use measurements from an Inertial Measurement Unit (IMU). The IMU consists in a triaxial angular rate sensor, and an also triaxial accelerometer.

Quaternions describing orientations live in the unit sphere of  $\mathbb{R}^4$ . Knowing that this space is a manifold, we can apply some basic concepts regarding these mathematical objects, and an algorithm that reminds the also called “Multiplicative Extended Kalman Filter” arises in a natural way.

The algorithm is tested in a simulated experiment, and in a real one.

**Index Terms**—Extended Kalman filter, quaternions, attitude, pose, orientation, estimation, IMU, manifold, charts.

## I. INTRODUCTION

**K**NOWLEDGE about the mechanical state of a system is necessary in many engineering fields. The orientation of the system is an important part of this mechanical state. Fields like robotics, virtual reality, or vehicle navigation among others, could require knowledge of the orientation of a system for tasks like:

- Controlling an *Unmanned Vehicle*.
- Knowing the orientation of a camera in a scenario.
- Knowing the heading of a vehicle in a navigation system.
- Transforming measurements taken in the vehicle reference frame to an extern reference frame.

The problem presents two main issues that need to be addressed:

- We need to choose an orientation descriptor.
- We need to choose an estimation approach.

The orientation of a system is understood as the rotation transformation that relates two reference frames: the one whose orientation we are interested in, and the reference frame of the system with respect to which we want to express such orientation. It only makes sense to speak about one orientation with respect to another system. Knowing that an orientation is a rotation transformation, our issue is to choose the most convenient parameterization for this rotation transformation. The most used parameterizations are the *Euler angles*, and their analogous, *Tait-Bryan angles*, *rotation vectors*, *rotation matrices*, and *unit quaternions*. A fairly complete survey of

orientation representations is given in [1]. Unit quaternions have properties that make them preferable against the other parameterizations. Namely:

- 1) There are no singularities (we avoid the “gimbal lock”, that is present in Euler angles).
- 2) They describe the orientation in a continuous way (unlike axis-angle representation).
- 3) Motion equations are linear with quaternions.
- 4) They are determined by 4 parameters (in contrast with a rotation matrix, that needs 9 parameters).

Because of these properties, unit quaternions have been the most widely used orientation representation since the early 1980s [2], and we also use them in this work.

The orientation estimation problem has been addressed using several approaches. In [3] it is provided a survey of methods for orientation estimation by far more complete than could be given in this work, and it would not make sense to repeat it here. The Kalman Filter with its nonlinear versions is the protagonist. But there is a major issue: unit quaternion do not live in the Euclidean space, where the Kalman Filter is defined. This fact leads to a variety of approaches in the application of this formalism. In particular, the known as “Multiplicative Extended Kalman Filter” is the method of choice because of its accuracy, its relative simplicity, its computational efficiency, and for being flexible to incorporate a great variety of measurements. However, there seem to be some aspects of it that are still not well understood.

The objective of this paper is to explore a new view point for the Extended Kalman Filter applied to the estimation of orientations represented by unit quaternions. Its final form is very similar to that of the “Multiplicative Extended Kalman Filter”, but it gets rid of the probably tricky definition of the “reset” operation, and it arises the introduction of a new update called “chart update”. Not being that different the structure of these two algorithms, it would not be unreasonable to rename this MEKF as “Manifold Extended Kalman Filter”.

The algorithm developed here is designed to take measurements from an Inertial Measurement Unit (IMU) which returns acceleration, and angular velocity measurements. Yet, this design is easily modifiable in order to adapt it to other type of sensors.

The MEKF has been tested in a simulated experiment, together with the known Madgwick algorithm [4]. It also has been implemented in a real system, and tested with a commercial IMU.

The remainder of the paper is organized as follows. In Section II, we introduce the main properties of unit quaternions. In Section III, we introduce the basic concepts of manifold theory that will be used in the algorithm development. In Section IV,

Pablo Bernal is with University of Murcia.  
E-mail: pablo.bernal.polo@gmail.com

Humberto Martínez is with University of Murcia.  
E-mail: humberto@um.es

Faculty of computer science. Applied engineering research group.

we review the motion equations and measurement equations. In Section V, we present the developed equations for the state prediction. In Section VI, we present the developed equations for the measurement prediction. In Section VII, we present the developed equations for the filter updates. Section VIII displays the experimental results. Finally, we expose the conclusions, and we picture the future work pathways in Section IX.

## II. QUATERNIONS

Quaternions are hypercomplex numbers with three different imaginary units  $\{i, j, k\}$ , and can be expressed as

$$\mathbf{q} = q_0 + q_1 i + q_2 j + q_3 k . \quad (1)$$

They can also be expressed in a vectorial form as

$$\mathbf{q} = \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} q_0 \\ \mathbf{q} \end{pmatrix} . \quad (2)$$

Quaternion product is defined by the Hamilton axiom

$$i^2 = j^2 = k^2 = i * j * k = -1 , \quad (3)$$

which produces the multiplication rule

$$\mathbf{p} * \mathbf{q} = \begin{pmatrix} p_0 q_0 - \mathbf{p} \cdot \mathbf{q} \\ p_0 \mathbf{q} + q_0 \mathbf{p} + \mathbf{p} \times \mathbf{q} \end{pmatrix} . \quad (4)$$

Quaternions describing rotations can be built with a unit vector that defines the rotation axis,  $\hat{\mathbf{q}}$ , and the angle of rotation,  $\theta$ , through

$$\mathbf{q} = \begin{pmatrix} \cos(\theta/2) \\ \hat{\mathbf{q}} \sin(\theta/2) \end{pmatrix} . \quad (5)$$

Having this form, they satisfy the restriction

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 . \quad (6)$$

This means that quaternions describing orientations live in the unit sphere of  $\mathbb{R}^4$ . This space has dimension 3, although its elements are determined using 4 parameters.

We will use basic concepts of manifold theory to handle this kind of space.

## III. BASICS OF MANIFOLD THEORY

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

$$\mathbf{P} = E \left[ (\mathbf{x} - \bar{\mathbf{x}}) (\mathbf{x} - \bar{\mathbf{x}})^T \right] . \quad (7)$$

This can be done when our random variables are elements of an Euclidean space. But when a random variable is an element of a manifold our covariance matrix definition could lose sense. This is our case, where the random variable  $\mathbf{q} - \bar{\mathbf{q}}$ , does not describe an orientation. Then we need to redefine our covariance matrix in a different way, but we can not change the form of the definition of the covariance matrix if we want to use the Kalman filter, because this precise form is used in its derivation. We will solve this problem by defining our covariance matrix in a different space. But first we will review some previous definitions:

a) *Definition. 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  $f : N \rightarrow B_n$ , with  $B_n$  the unit ball of  $\mathbb{R}^n$ .

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

$$\varphi : U \subset M \rightarrow V \subset \mathbb{R}^n ,$$

with  $\varphi$  a homeomorphism.

Traditionally a chart is expressed as the pair  $(U, \varphi)$ .

### A. The Set of Charts

Assume we know the expected value of our distribution of quaternions,  $\bar{\mathbf{q}}$ . In such case, we can express any unit quaternion as

$$\mathbf{q} = \bar{\mathbf{q}} * \boldsymbol{\delta} , \quad (8)$$

with  $\boldsymbol{\delta}$  another unit quaternion (unit quaternions together with their multiplication rule form a group). And then, we can define the set of charts

$$\varphi_{\bar{\mathbf{q}}}(\mathbf{q}) = 2 \begin{pmatrix} \frac{\delta_1}{\delta_0} \\ \frac{\delta_2}{\delta_0} \\ \frac{\delta_3}{\delta_0} \end{pmatrix} . \quad (9)$$

The set of charts (9), is used in [5], but this work does not talk about charts, and what we call ‘‘chart update’’ is not applied.

In each chart,  $\varphi_{\bar{\mathbf{q}}}$ , the quaternion  $\bar{\mathbf{q}}$  is mapped to the origin. As the space deformation produced in the neighborhood of the origin is small, being the variance small, the distribution in each chart will be similar to the distribution in the manifold.

The inverse transformations for these charts are given by

$$\varphi_{\bar{\mathbf{q}}}^{-1}(\mathbf{e}_q) = \bar{\mathbf{q}} * \frac{1}{\sqrt{4 + \|\mathbf{e}_q\|^2}} \begin{pmatrix} 2 \\ \mathbf{e}_q \end{pmatrix} . \quad (10)$$

### B. The Transition Map

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

$$\varphi_{\alpha\beta}(x) = \varphi_\beta(\varphi_\alpha^{-1}(x)) ,$$

with  $x \in \varphi_\alpha(U_{\alpha\beta})$ .

Having the set of charts defined by (9), and having two charts centered in quaternions  $\bar{\mathbf{p}}$  and  $\bar{\mathbf{q}}$ , and related by  $\bar{\mathbf{p}} = \bar{\mathbf{q}} * \bar{\boldsymbol{\delta}}$ , then our transition map takes the form

$$\begin{aligned} \mathbf{e}_p &= \varphi_{\bar{\mathbf{q}}, \bar{\mathbf{p}}}(\mathbf{e}_q) = \\ &= 2 \frac{\bar{\delta}_0 \mathbf{e}_q - 2 \bar{\boldsymbol{\delta}} - \bar{\boldsymbol{\delta}} \times \mathbf{e}_q}{2 \bar{\delta}_0 + \bar{\boldsymbol{\delta}} \cdot \mathbf{e}_q} . \end{aligned} \quad (11)$$

#### IV. MOTION EQUATIONS AND MEASUREMENT EQUATIONS

The state of the system is defined by an orientation, encoded by a unit quaternion  $\mathbf{q}$ , and by an angular velocity measured in the reference frame attached to our system, given by a vector  $\boldsymbol{\omega}'$ . The unit quaternion defines a rotation transformation that determines the orientation of the system. This transformation relates vectors (denoted as  $\mathbf{v}'$ ) expressed in a reference frame attached to the solid whose state we want to describe, with the same vectors (denoted as  $\mathbf{v}$ ) expressed in an inertial reference frame in which the gravity vector is expressed as  $\mathbf{g} = (0, 0, -1)$ . Thus, our rotation transformation will yield

$$\mathbf{v} = T_{\mathbf{R}}(\mathbf{v}') . \quad (12)$$

Using a rotation matrix,

$$\mathbf{v} = \mathbf{R} \mathbf{v}' . \quad (13)$$

And using our unit quaternion,

$$\mathbf{v} = \mathbf{q} * \mathbf{v}' * \mathbf{q}^* . \quad (14)$$

where this time  $\mathbf{v} = \begin{pmatrix} 0 \\ \mathbf{v} \end{pmatrix}$ , and  $\mathbf{q}^*$  is the complex conjugate quaternion, that being  $\mathbf{q}$  a unit quaternion, it is also its inverse.

##### A. Motion Equations

Knowing what our quaternion means, we can write the motion equations for the random variables that we use to describe the state of our system:

$$\dot{\mathbf{q}}(t) = \frac{1}{2} \mathbf{q}(t) * \begin{pmatrix} 0 \\ \boldsymbol{\omega}'(t) \end{pmatrix} , \quad (15)$$

$$\dot{\boldsymbol{\omega}}'(t) = \tilde{\boldsymbol{\tau}}'(t) , \quad (16)$$

where  $\tilde{\boldsymbol{\tau}}'$  is the process noise, associated with the torque acting on the system, and its inertia tensor.

##### B. Measurement Equations

This work uses an IMU as information source. We can write the measurement equations that relate the random variables describing the state of our system, with the random variables describing the measurements of our sensors as follows:

$$\mathbf{a}_t^m = \mathbf{R}_t^T (\mathbf{a}_t - \mathbf{g}_t) + \mathbf{r}_t^a , \quad (17)$$

$$\boldsymbol{\omega}_t^m = \boldsymbol{\omega}'_t + \mathbf{r}_t^\omega , \quad (18)$$

where  $\mathbf{r}_t^a$  is the noise in the accelerometer measurement,  $\mathbf{r}_t^\omega$  is the noise in the gyroscope measurement, and  $\mathbf{a}_t$  are non-gravitational accelerations.

#### V. STATE PREDICTION

In this section we expose the evolution equations used to perform the prediction of the expected value of the state, and of its covariance matrix.

##### A. Evolution of the Expected Value of the State

Taking the expected value in equations (15) and (16), assuming the random variables  $\mathbf{q}(t)$  and  $\mathbf{q}_t^\omega(t) = \int_{t-\Delta t}^t \tilde{\boldsymbol{\tau}}'(\tau) d\tau$  to be independent, and the expected value of the process noise,  $\bar{\mathbf{q}}_t^\omega(\tau)$ , to be constant when  $\tau \in [t - \Delta t, t)$ , our differential equations are transformed into other ones whose solutions are

$$\bar{\boldsymbol{\omega}}'_{t|t-\Delta t} = \bar{\boldsymbol{\omega}}'_{t-\Delta t|t-\Delta t} + \bar{\mathbf{q}}_t^\omega , \quad (19)$$

$$\begin{aligned} \bar{\mathbf{q}}_{t|t-\Delta t} &= \bar{\mathbf{q}}_{t-\Delta t|t-\Delta t} * \mathbf{q}_\omega = \\ &= \bar{\mathbf{q}}_{t-\Delta t|t-\Delta t} * \begin{pmatrix} \cos\left(\frac{\bar{\boldsymbol{\omega}}'_{t|t-\Delta t} \Delta t}{2}\right) \\ \frac{\bar{\boldsymbol{\omega}}'_{t|t-\Delta t}}{\bar{\boldsymbol{\omega}}'_{t|t-\Delta t}} \sin\left(\frac{\bar{\boldsymbol{\omega}}'_{t|t-\Delta t} \Delta t}{2}\right) \end{pmatrix} . \end{aligned} \quad (20)$$

##### B. Evolution of the State Covariance Matrix

Since we need a covariance matrix with a form like (7), we will define the covariance matrix of the state in the set of charts defined by (9). In particular, for each filter update we will have an expected value for the unit quaternion describing the orientation,  $\bar{\mathbf{q}}$ , and a covariance matrix defined in the Euclidean space,  $\mathbb{R}^3$ , whose points are related with those of the unit sphere of  $\mathbb{R}^4$  through the chart  $\varphi_{\bar{\mathbf{q}}}$ . The origin of  $\mathbb{R}^3$  is mapped with the  $\bar{\mathbf{q}}$  quaternion, and points around the origin represent quaternions in the neighborhood of  $\bar{\mathbf{q}}$ .

1) *Differential Equations for our Charts:* This result, and its derivation, is totally inspired and is almost equal to that which appears in [2]. Having the definition for our charts in (8) and (9), we can find the differential equations for the  $\delta$  quaternion using the differential equations for  $\mathbf{q}$ , (15), and for  $\bar{\mathbf{q}}$ . And having the differential equations for the  $\delta$  quaternion, we can find the differential equations for a point  $\mathbf{e} = 2 \frac{\delta}{\delta_0}$  on the charts:

$$\dot{\mathbf{e}} = \Delta \boldsymbol{\omega}' - [2\bar{\boldsymbol{\omega}}' + \Delta \boldsymbol{\omega}']_{\times} \frac{\mathbf{e}}{2} + \frac{\mathbf{e}}{2} \frac{\mathbf{e}^T}{2} \Delta \boldsymbol{\omega}' . \quad (21)$$

Note that, by the definition of the charts, the vector of random variables  $\mathbf{e}_t$  is expressed in the chart centered in the quaternion  $\bar{\mathbf{q}}_t$ . For each instant,  $t$ , we have a quaternion  $\bar{\mathbf{q}}_t$  defining the chart for that time. Then the differential equation (21) defines the evolution of the vector  $\mathbf{e}$  that “travels” between charts.

2) *Differential Equations for the Covariance Matrix:* We define the covariance matrix for the state of our system by

$$\begin{aligned} \mathbf{P}_t &= \begin{pmatrix} E[\mathbf{e}_t \mathbf{e}_t^T] & E[\mathbf{e}_t \Delta \boldsymbol{\omega}_t'^T] \\ E[\Delta \boldsymbol{\omega}_t' \mathbf{e}_t^T] & E[\Delta \boldsymbol{\omega}_t' \Delta \boldsymbol{\omega}_t'^T] \end{pmatrix} = \\ &= \begin{pmatrix} \mathbf{P}_t^{ee} & \mathbf{P}_t^{e\omega} \\ \mathbf{P}_t^{\omega e} & \mathbf{P}_t^{\omega\omega} \end{pmatrix} . \end{aligned} \quad (22)$$

Notice that we do not write “ $\Delta \mathbf{e}_t$ ”. By the new definition of the covariance matrix, the term  $\mathbf{e}_t$  can be interpreted as a displacement from the  $\bar{\mathbf{q}}_t$  quaternion, which is mapped to the origin of  $\mathbb{R}^3$  by the chart.

Note also that being the covariance matrix symmetric, we do not need to find the evolution of all its terms. We just

need to find the evolution of the terms  $\mathbf{P}^{ee}$ ,  $\mathbf{P}^{e\omega}$ , and  $\mathbf{P}^{\omega\omega}$ .

We are looking for an estimation of the covariance matrix in  $t$ , using the information in  $t - \Delta t$ . This is, we want to get  $\mathbf{P}_{t|t-\Delta t}$  from  $\mathbf{P}_{t-\Delta t|t-\Delta t}$ . For  $\mathbf{P}^{\omega\omega}$  it is easy to find this relation. Assuming the random variables  $\omega'_{t-\Delta t}$  and  $\mathbf{q}'_t$  to be independents,

$$\mathbf{P}_{t|t-\Delta t}^{\omega\omega} = \mathbf{P}_{t-\Delta t|t-\Delta t}^{\omega\omega} + \mathbf{Q}_t^\omega. \quad (23)$$

If we had a function  $\mathbf{e}_t = f(\mathbf{e}_{t-\Delta t})$ , we could replace in (22), and perhaps obtain a relation similar to (23). But we are not able to find a closed solution for (21). However, we can find a differential equation for  $\mathbf{P}_t$  using this differential equation for  $\mathbf{e}_t$ . Starting from (22),

$$\frac{d\mathbf{P}}{dt} = \begin{pmatrix} \dot{\mathbf{P}}^{ee} & \dot{\mathbf{P}}^{e\omega} \\ \dot{\mathbf{P}}^{e\omega} & \dot{\mathbf{P}}^{\omega\omega} \end{pmatrix}, \quad (24)$$

with

$$\dot{\mathbf{P}}^{ee} = E[\dot{\mathbf{e}} \mathbf{e}^T] + E[\mathbf{e} \dot{\mathbf{e}}^T], \quad (25)$$

$$\dot{\mathbf{P}}^{e\omega} = E[\dot{\mathbf{e}} \Delta \omega'^T] + E[\mathbf{e} \Delta \dot{\omega}'^T]. \quad (26)$$

After replacing (21), assuming that higher moments are negligible compared to second-order moments, and remembering our assumption of independence of the random variables  $\mathbf{q}$  and  $\mathbf{q}'_t$ , and therefore, of  $\mathbf{e}$  and  $\tilde{\boldsymbol{\tau}}'$ , (25) and (26) can be approximated by

$$\begin{aligned} \dot{\mathbf{P}}^{ee} &\approx (\mathbf{P}^{e\omega})^T - [\bar{\omega}']_{\times} \mathbf{P}^{ee} + \\ &+ \mathbf{P}^{e\omega} - \mathbf{P}^{ee} [\bar{\omega}']_{\times}^T. \end{aligned} \quad (27)$$

$$\dot{\mathbf{P}}^{e\omega} \approx \mathbf{P}^{\omega\omega} - [\bar{\omega}']_{\times} \mathbf{P}^{e\omega}. \quad (28)$$

3) *Evolution Equations for the Covariance Matrix:* We are dealing with a system of inhomogeneous linear matrix differential equations. Generally, a system of this type is untreatable, but in our case the equations are sufficiently decoupled to be able to find a solution.

Given a solution for  $\mathbf{P}^{\omega\omega}$ , we can find an approximate solution for  $\mathbf{P}^{e\omega}$ . And with this solution, we can find an approximate solution for  $\mathbf{P}^{ee}$ . Denoting  $\boldsymbol{\Omega} = [\bar{\omega}'_{t}]_{\times}$  we can write

$$\begin{aligned} \mathbf{P}_{t|t-\Delta t}^{e\omega} &\approx e^{-\boldsymbol{\Omega} \Delta t} \left[ \mathbf{P}_{t-\Delta t|t-\Delta t}^{e\omega} + \right. \\ &+ \Delta t \mathbf{P}_{t-\Delta t|t-\Delta t}^{e\omega} + \left. \frac{(\Delta t)^2}{2} \mathbf{Q}_0^\omega \right], \end{aligned} \quad (29)$$

$$\begin{aligned} \mathbf{P}_{t|t-\Delta t}^{ee} &\approx e^{-\boldsymbol{\Omega} \Delta t} \left[ \mathbf{P}_{t-\Delta t|t-\Delta t}^{ee} + \right. \\ &+ (\Delta t)^2 \mathbf{P}_{t-\Delta t|t-\Delta t}^{\omega\omega} + \frac{(\Delta t)^3}{3} \mathbf{Q}_0^\omega + \\ &+ \mathbf{P}_{t-\Delta t|t-\Delta t}^{e\omega} \Delta t + \\ &+ \left. (\mathbf{P}_{t-\Delta t|t-\Delta t}^{e\omega})^T \Delta t \right] e^{-\boldsymbol{\Omega}^T \Delta t}. \end{aligned} \quad (30)$$

with  $\mathbf{Q}_t^\omega = \mathbf{Q}_0^\omega t$ , being  $\mathbf{Q}_0^\omega$  a constant matrix representing the process noise covariance per time unit.

## VI. MEASUREMENT PREDICTION

In this section we expose the measurement equations used to perform the prediction of the expected value of the measurement, and of its covariance matrix.

### A. Expected Value of the Measurement

1) *Expected Value of the Gyroscope Measurement:* Taking the expected value on (18),

$$\bar{\omega}_t^m = \bar{\omega}_t' + \bar{\mathbf{r}}_t^\omega. \quad (31)$$

2) *Expected Value of the Accelerometer Measurement:* Taking the expected value on (17), knowing that the  $\mathbf{g}_t$  vector does not change, and assuming that the non-gravitational accelerations affecting our system,  $\mathbf{a}_t$ , does not depend on its orientation,

$$\bar{\mathbf{a}}_t^m = E[\mathbf{R}_t^T] (\bar{\mathbf{a}}_t - \bar{\mathbf{g}}_t) + \bar{\mathbf{r}}_t^a. \quad (32)$$

Using the unit quaternion describing the orientation of our system, this relation takes the form

$$\bar{\mathbf{a}}_t^m = \bar{\mathbf{q}}_t^* (\bar{\mathbf{a}}_t - \bar{\mathbf{g}}_t) * \bar{\mathbf{q}}_t + \bar{\mathbf{r}}_t^a. \quad (33)$$

And if we use the rotation matrix constructed from this unit quaternion,

$$\bar{\mathbf{a}}_t^m = \bar{\mathbf{R}}_t^T (\bar{\mathbf{a}}_t - \bar{\mathbf{g}}_t) + \bar{\mathbf{r}}_t^a. \quad (34)$$

### B. Covariance Matrix of the Measurement

The measurement is related to the state by means of the measurement equations:

$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t, \mathbf{r}_t).$$

We can approximate linearly the relationship around the expected values,  $\bar{\mathbf{x}}_t$ , and  $\bar{\mathbf{r}}_t$ , using a Taylor series

$$\mathbf{z}_t \approx \mathbf{h}(\bar{\mathbf{x}}_t, \bar{\mathbf{r}}_t) + \mathbf{H}_t (\mathbf{x}_t - \bar{\mathbf{x}}_t) + \mathbf{M}_t (\mathbf{r}_t - \bar{\mathbf{r}}_t),$$

being

$$\mathbf{H}_t = \left. \frac{\partial \mathbf{h}(\mathbf{x}_t, \mathbf{r}_t)}{\partial \mathbf{x}_t} \right|_{\bar{\mathbf{x}}_t, \bar{\mathbf{r}}_t}, \quad \mathbf{M}_t = \left. \frac{\partial \mathbf{h}(\mathbf{x}_t, \mathbf{r}_t)}{\partial \mathbf{r}_t} \right|_{\bar{\mathbf{x}}_t, \bar{\mathbf{r}}_t}, \quad (35)$$

the Jacobian matrices evaluated on the expected value of the random variables. Then, our prediction equation of the measurement covariance matrix takes the form

$$\mathbf{S}_t = \mathbf{H}_t \mathbf{P}_t \mathbf{H}_t^T + \mathbf{M}_t \mathbf{R}_t \mathbf{M}_t^T. \quad (36)$$

1) *Gyroscope Block:* The measurement equation for the gyroscope is linear. Using (18) and (35) we obtain

$$\mathbf{H}_t^\omega = (\mathbf{0} \quad \mathbf{1}). \quad (37)$$

2) *Accelerometer Block*: In order of being consistent with the Kalman filter formulation, the acceleration term,  $\mathbf{a}_t$ , should be part of the noise in the measurement, since if it were not so, it should be part of the state. Then, measurement noise in our Kalman filter has two components:

- $\mathbf{r}_t$ : the main measurement noise. This noise comes from the sensor.
- $\mathbf{a}_t$ : non-gravitational accelerations acting on the system. These accelerations obstruct the measurement of the  $\bar{\mathbf{g}}$  vector.

Recalling that we express the covariance matrix of the state in  $\mathbb{R}^3$ , and knowing that doing  $\mathbf{q}_t = \bar{\mathbf{q}}_t$  in the manifold, is equivalent to do  $\mathbf{e}_t = \mathbf{0}$  in this space, we will have for the accelerometer measurement equation:

$$\begin{aligned} \mathbf{a}_t^m &\approx \bar{\mathbf{a}}_t^m + \left. \frac{\partial \mathbf{h}_a(\mathbf{e}_t, \mathbf{a}_t)}{\partial \mathbf{e}_t} \right|_{\substack{\mathbf{e}_t=\mathbf{0} \\ \mathbf{a}_t=\bar{\mathbf{a}}_t}} \mathbf{e}_t + \\ &+ \left. \frac{\partial \mathbf{h}_a(\mathbf{e}_t, \mathbf{a}_t)}{\partial \mathbf{a}_t} \right|_{\substack{\mathbf{e}_t=\mathbf{0} \\ \mathbf{a}_t=\bar{\mathbf{a}}_t}} (\mathbf{a}_t - \bar{\mathbf{a}}_t) + \mathbf{r}_t^a - \bar{\mathbf{r}}_t^a . \end{aligned}$$

Denoting  $\bar{\mathbf{g}}_t^{\mathbf{R}} = \bar{\mathbf{R}}_t^T (\bar{\mathbf{a}}_t - \bar{\mathbf{g}}_t)$ , and after some calculus,

$$\begin{aligned} \left. \frac{\partial \mathbf{h}_a}{\partial \mathbf{a}_t} \right|_{\substack{\mathbf{e}_t=\mathbf{0} \\ \mathbf{a}_t=\bar{\mathbf{a}}_t}} &= \bar{\mathbf{R}}_t^T , \\ \left. \frac{\partial \mathbf{h}_a(\mathbf{e}_t)}{\partial \mathbf{e}_t} \right|_{\substack{\mathbf{e}_t=\mathbf{0} \\ \mathbf{a}_t=\bar{\mathbf{a}}_t}} &= \begin{pmatrix} 0 & -(\bar{\mathbf{g}}_t^{\mathbf{R}})_3 & (\bar{\mathbf{g}}_t^{\mathbf{R}})_2 \\ (\bar{\mathbf{g}}_t^{\mathbf{R}})_3 & 0 & -(\bar{\mathbf{g}}_t^{\mathbf{R}})_1 \\ -(\bar{\mathbf{g}}_t^{\mathbf{R}})_2 & (\bar{\mathbf{g}}_t^{\mathbf{R}})_1 & 0 \end{pmatrix} . \end{aligned}$$

3) *Measurement Covariance Matrix*: Assuming independence of all random variables involved in the measure, our prediction equation for the measurement is

$$\begin{aligned} \mathbf{S}_t &= \begin{pmatrix} [\bar{\mathbf{g}}_t^{\mathbf{R}}]_{\times} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{pmatrix} \mathbf{P}_t \begin{pmatrix} [\bar{\mathbf{g}}_t^{\mathbf{R}}]_{\times} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{pmatrix}^T + \\ &+ \begin{pmatrix} \bar{\mathbf{R}}_t^T \mathbf{Q}_t^a \bar{\mathbf{R}}_t + \mathbf{R}_t^a & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_t^{\omega} \end{pmatrix} , \quad (38) \end{aligned}$$

where

- $\mathbf{Q}_t^a$  is the covariance matrix of the random variable  $\mathbf{a}_t$ .
- $\mathbf{R}_t^a$  is the covariance matrix that describes the noise in the accelerometer measurement, which is modeled by the random variable  $\mathbf{r}_t^a$ .
- $\mathbf{R}_t^{\omega}$  is the covariance matrix describing the noise in the gyroscope measurement, which is modeled by the random variable  $\mathbf{r}_t^{\omega}$ .

## VII. UPDATE

Although the original Kalman filter algorithm just requires the Kalman update, the fact that our covariance matrix is expressed in a chart makes necessary the computation of a second update.

### A. Kalman Update

The Kalman update is performed in the space where the covariance matrix is defined. This is, we do not perform the Kalman update in the manifold, but in the chart. Given the estimate of the covariance matrix of the state,  $\mathbf{P}_{t|t-\Delta t}$ , and the estimate of the measurement covariance matrix,  $\mathbf{S}_{t|t-\Delta t}$ , the optimal Kalman gain is computed as

$$\mathbf{K}_t = \mathbf{P}_{t|t-\Delta t} \mathbf{H}_t^T \mathbf{S}_{t|t-\Delta t}^{-1} . \quad (39)$$

Given the gain, we can update the state distribution in the usual way:

$$\mathbf{x}_{t|t} = \mathbf{x}_{t|t-\Delta t} + \mathbf{K}_t (\mathbf{z}_t - \mathbf{z}_{t|t-\Delta t}) , \quad (40)$$

$$\mathbf{P}_{t|t} = (\mathbf{1} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_{t|t-\Delta t} . \quad (41)$$

The new state distribution will be expressed in the chart centered on  $\bar{\mathbf{x}}_{t|t-\Delta t}$ . In this chart,  $\bar{\mathbf{e}}_{t|t-\Delta t} = \mathbf{0}$ , but  $\bar{\mathbf{e}}_{t|t} \neq \mathbf{0}$ .

### B. Manifold Update

In order to find the quaternion corresponding to the updated vector  $\bar{\mathbf{e}}_{t|t}$  expressed in the chart, we must reverse the function  $\varphi_{\bar{\mathbf{q}}_{t|t-\Delta t}}(\bar{\mathbf{e}}_{t|t})$  making use of the equation (10):

$$\begin{aligned} \bar{\mathbf{q}}_{t|t} &= \bar{\mathbf{q}}_{t|t-\Delta t} * \delta(\bar{\mathbf{e}}_{t|t}) = \\ &= \bar{\mathbf{q}}_{t|t-\Delta t} * \frac{1}{\sqrt{4 + \|\bar{\mathbf{e}}_{t|t}\|^2}} \begin{pmatrix} 2 \\ \bar{\mathbf{e}}_{t|t} \end{pmatrix} . \quad (42) \end{aligned}$$

### C. Chart Update

After the Kalman update, the new state distribution is expressed in the chart centered on  $\bar{\mathbf{x}}_{t|t-\Delta t}$ . We must update the covariance matrix expressing it in the chart centered on the updated state,  $\bar{\mathbf{x}}_{t|t}$ , so that our information is expressed as at the beginning of the iteration. In order of achieve this objective, we must use the concept of transition maps, that for our charts take the form of (11). Being non-linear this relation, we need to find a linear approximation:

$$\mathbf{e}_p(\mathbf{e}_q) \approx \mathbf{e}_p(\bar{\mathbf{e}}_q) + \left. \frac{\partial \mathbf{e}_p}{\partial \mathbf{e}_q} \right|_{\mathbf{e}_q=\bar{\mathbf{e}}_q} (\mathbf{e}_q - \bar{\mathbf{e}}_q) . \quad (43)$$

After differentiating our transition map and evaluating in  $\bar{\mathbf{e}}_{t|t}$ , having identified the charts  $\varphi_{\bar{\mathbf{p}}} = \varphi_{\bar{\mathbf{q}}_{t|t}}$  and  $\varphi_{\bar{\mathbf{q}}} = \varphi_{\bar{\mathbf{q}}_{t|t-\Delta t}}$ , we find out

$$\begin{aligned} \mathbf{e}_p(\mathbf{e}_q) &\approx \bar{\delta}_0 \left( \bar{\delta}_0 \mathbf{1} - [\bar{\delta}]_{\times} \right) (\mathbf{e}_q - \bar{\mathbf{e}}_{t|t}) = \\ &= \mathbf{G}_t (\mathbf{e}_q - \bar{\mathbf{e}}_{t|t}) . \quad (44) \end{aligned}$$

Then, our update equations for the charts are

$$\mathbf{P}_{\bar{\mathbf{q}}_{t|t}}^{ee} = \mathbf{G}_t \mathbf{P}_{\bar{\mathbf{q}}_{t|t-\Delta t}}^{ee} \mathbf{G}_t^T , \quad (45)$$

$$\mathbf{P}_{\bar{\mathbf{q}}_{t|t}}^{e\omega} = \mathbf{G}_t \mathbf{P}_{\bar{\mathbf{q}}_{t|t-\Delta t}}^{e\omega} , \quad (46)$$

$$\mathbf{P}_{\bar{\mathbf{q}}_{t|t}}^{\omega\omega} = \mathbf{P}_{\bar{\mathbf{q}}_{t|t-\Delta t}}^{\omega\omega} . \quad (47)$$

## VIII. EXPERIMENTAL VALIDATION

In this section we present results obtained from a simulated experiment, and a real one.

### A. Simulated Experiment

Our simulated experiment consists on the definition of a path, the extraction of simulated measurements, the processing of this measurements, and the evaluation of the algorithm performance. Only knowing the real state, we are able to define some metrics to measure the performance of our algorithm. Finally, we display a comparison of the algorithm developed in this paper, which will be called Manifold Extended Kalman Filter (MEKF), and the currently popular algorithm developed by Madgwick [4], whose code can be found in [6].

1) *Experiment Setup*: For testing our algorithm, we can think in a simple and intuitive simulation. Let us imagine that we can freely roam the surface of a torus, which is a manifold whose space can be described in  $\mathbb{R}^3$  by

$$x(\theta, \phi) = (R + r \cos \theta) \cos \phi, \quad (48)$$

$$y(\theta, \phi) = (R + r \cos \theta) \sin \phi, \quad (49)$$

$$z(\theta, \phi) = r \sin \theta. \quad (50)$$

The torus of our simulation will have  $R = 0.2\text{m}$ , and  $r = 0.05\text{m}$ . We can define a path in the torus using a third parameter to set the other two:

$$\theta = v_\theta t, \quad (51)$$

$$\phi = v_\phi t. \quad (52)$$

We will use the path defined by  $v_\phi = 1\text{ rad/s}$  and  $v_\theta = 3\text{ rad/s}$ , and we will travel the path around the torus 3 times. This path can be seen in Figure 1.

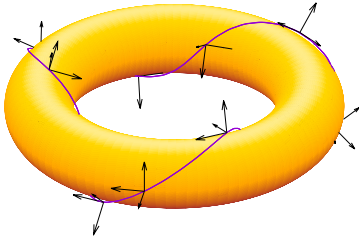


Fig. 1. Path followed on the torus in our simulation. Reference frames that define the orientation of the IMU can be observed.

Accelerations occurring in the IMU can be calculated by differentiating twice in (48)-(50) with respect to the  $t$  parameter, resulting

$$\begin{aligned} \ddot{x}(\theta, \phi) &= -(R + r \cos \theta) \cos \phi v_\phi^2 + \\ &+ 2r \sin \theta v_\theta \sin \phi v_\phi + \\ &- r \cos \theta v_\theta^2 \cos \phi, \end{aligned} \quad (53)$$

$$\begin{aligned} \ddot{y}(\theta, \phi) &= -(R + r \cos \theta) \sin \phi v_\phi^2 + \\ &- 2r \sin \theta v_\theta \cos \phi v_\phi + \\ &- r \cos \theta v_\theta^2 \sin \phi, \end{aligned} \quad (54)$$

$$\ddot{z}(\theta, \phi) = -r \sin \theta v_\theta^2. \quad (55)$$

Now, we can define a reference frame for each point of the path. The axis of the reference frame will have the directions of the vectors

$$\left\{ \frac{\partial \mathbf{x}}{\partial \theta}, \frac{\partial \mathbf{x}}{\partial \phi}, \frac{\partial \mathbf{x}}{\partial \theta} \times \frac{\partial \mathbf{x}}{\partial \phi} \right\}. \quad (56)$$

After making the derivatives in (56), and choosing the direction of the vectors so that  $\hat{\mathbf{z}}$  points outward the torus surface, our rotation matrix relating a vector measured in the IMU reference frame, with the same vector measured in the external reference frame will be

$$\mathbf{R} = \begin{pmatrix} | & | & | \\ \hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{z}} \\ | & | & | \end{pmatrix} = \begin{pmatrix} -\sin \theta \cos \phi & \sin \phi & \cos \theta \cos \phi \\ -\sin \theta \sin \phi & -\cos \phi & \cos \theta \sin \phi \\ \cos \theta & 0 & \sin \theta \end{pmatrix}. \quad (57)$$

Matrix (57) can be expressed as the product of 3 known rotation matrices:

$$\mathbf{R} = \begin{pmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{pmatrix}. \quad (58)$$

Recognizing these matrices in (58), and using (5), we can find out the quaternion describing the orientation of our reference frame:

$$\mathbf{q} = \begin{pmatrix} \cos(\phi/2) \\ 0 \\ 0 \\ \sin(\phi/2) \end{pmatrix} * \begin{pmatrix} \cos(\theta/2) \\ 0 \\ -\sin(\theta/2) \\ 0 \end{pmatrix} * \begin{pmatrix} 0 \\ 1/\sqrt{2} \\ 0 \\ 1/\sqrt{2} \end{pmatrix}. \quad (59)$$

Having (59) we can obtain the  $\dot{\mathbf{q}}$  quaternion:

$$\dot{\mathbf{q}} = (\dot{\mathbf{q}}_\phi * \mathbf{q}_\theta + \mathbf{q}_\phi * \dot{\mathbf{q}}_\theta) * \mathbf{q}_0, \quad (60)$$

where

$$\dot{\mathbf{q}}_\phi = \begin{pmatrix} -\sin(\phi/2) \\ 0 \\ 0 \\ \cos(\phi/2) \end{pmatrix} \frac{v_\phi}{2}, \quad (61)$$

$$\dot{\mathbf{q}}_\theta = \begin{pmatrix} -\sin(\theta/2) \\ 0 \\ -\cos(\theta/2) \\ 0 \end{pmatrix} \frac{v_\theta}{2}. \quad (62)$$

And with the  $\dot{\mathbf{q}}$  quaternion we can use (15) to get the angular velocity:

$$\boldsymbol{\omega} = 2 \mathbf{q}^* * \dot{\mathbf{q}}. \quad (63)$$

With all, we can generate a succession of states  $(\mathbf{q}_r^t)$ , and for each state simulate a measurement  $\begin{pmatrix} \mathbf{a}_t^m \\ \boldsymbol{\omega}_t^m \end{pmatrix}$ . After that, taking only the succession of measurements, we can make a succession of estimations about the state  $(\mathbf{q}_e^t)$  using the orientation algorithms, and then compare our estimation with the known real state  $(\mathbf{q}_r^t)$ .

2) *Error Definition*: We will evaluate the performance of the algorithm through the definition of two errors:

First, defining  $\mathbf{g}'_r = (\overline{\mathbf{R}}^r)^T(0, 0, -1)^T$  as the gravity vector measured in the real reference frame attached to our system, and  $\mathbf{g}'_e = (\overline{\mathbf{R}}^e)^T(0, 0, -1)^T$  as the gravity vector measured in the estimated reference frame, we define

$$e_g = \arccos \left( \frac{\mathbf{g}'_r \cdot \mathbf{g}'_e}{\|\mathbf{g}'_r\| \|\mathbf{g}'_e\|} \right). \quad (64)$$

The  $e_g$  error in (64) is defined as the angle between the vectors  $\mathbf{g}'_r$  and  $\mathbf{g}'_e$ . Being the lowest error the better, this gives us a measure of how well the algorithm estimates the direction of a vector for which we have directly related

measurements.

Second, we define  $\mathbf{q}_i^r$ , and  $\mathbf{q}_f^r$  as the initial and final quaternions describing the real orientation of the system in our simulation; and  $\mathbf{q}_i^e$ , and  $\mathbf{q}_f^e$  as the initial and final quaternions describing the estimated orientation of the system. Then, defining  $\Delta^r$ , and  $\Delta^e$  as the quaternions describing the rotation transformations that relates the initials and finals orientations by  $\mathbf{q}_f^r = \mathbf{q}_i^r * \Delta^r$ , and  $\mathbf{q}_f^e = \mathbf{q}_i^e * \Delta^e$ , we define

$$e_\theta = 2 \arccos \left( [(\Delta^r)^* * \Delta^e]_0 \right) . \quad (65)$$

The  $e_\theta$  error in (65) is defined as the angle of the rotation defined by the  $\delta_\theta$  quaternion, which satisfies  $\Delta^e = \Delta^r * \delta_\theta$ . Being the lowest error the better, this gives us a measure of how well the algorithm estimates the whole orientation, including heading, for which we do not have directly related measurements.

This second error definition seems unnecessarily complicated. We could think in something like  $e'_\theta = 2 \arccos \left( (\mathbf{q}_f^r)^* * \mathbf{q}_f^e \right)$ , but if we start the simulation with an unknown orientation for the algorithm, this definition would lead to a different quaternion from the starting one. The  $e'_\theta$  error would have a bias because of the ignorance of the initial heading. Our  $e_\theta$  error definition is independent of this initial heading ignorance.

### 3) Setting the Algorithm Values:

a) *Initialization:* The simulation starts with a known orientation state defined by

$$\bar{\mathbf{q}}_{0|0} = \left( 0, 1/\sqrt{2}, 0, 1/\sqrt{2} \right)^T , \quad (66)$$

$$\bar{\boldsymbol{\omega}}_{0|0} = \left( 1, 3, 0 \right)^T , \quad (67)$$

$$\mathbf{P}_{0|0} = 10^{-2} \mathbf{1} . \quad (68)$$

b) *Characterization of Process Noise:* The following values have been established:

$$\bar{\mathbf{q}}_t^\omega = \mathbf{0} , \quad \mathbf{Q}_t^\omega = 10^1 \mathbf{1} , \quad (69)$$

$$\bar{\mathbf{a}}_t = \mathbf{0} , \quad \mathbf{Q}_t^a = 10^1 \mathbf{1} . \quad (70)$$

Still, after some testing, we found that the algorithm behaves similarly with other configurations, provided that they are not disproportionate. A more worked up algorithm would introduce dynamical values for this variables.

c) *Characterization of Measurement Noise:* The Kalman filter requires

$$\bar{\mathbf{r}}_t^a = \mathbf{0} , \quad \bar{\mathbf{r}}_t^\omega = \mathbf{0} , \quad (71)$$

in order to produce an unbiased estimation.

For the covariance matrices we will set

$$\mathbf{R}_t^a = \mathbf{R}_t^\omega = \sigma^2 \mathbf{1} , \quad (72)$$

and we will compare how the error behaves as a function of the magnitude of the noise in the measurement.

Test	Sampling time (s)	$\sigma^2$
1	Ranging in $(10^{-3}, 10^{-1})$	Fixed to $10^{-3}$
2	Fixed to $10^{-1}$	Ranging in $(10^{-7}, 10^{-3})$
3	Ranging in $(10^{-3}, 10^{-1})$	Fixed to $10^{-7}$
4	Fixed to $10^{-3}$	Ranging in $(10^{-7}, 10^{-3})$

TABLE I  
TESTS CONDUCTED IN OUR SIMULATION.

4) *Simulation Results:* We will place our simulation in 4 different scenarios, whose details are displayed in table I, and that have been chosen according to the current possibilities.

Results of Test 1 and 2 are not shown since the errors produced are too large. This suggests that both a good processor (small sampling time) as a good sensor (small variance) are required.

For Test 3 and 4 the time evolution of error measurements are plotted in Figures 2 - 5. In Figures 3 and 5 we observe how the error becomes smaller as we improve our IMU (decreasing  $\sigma^2$ ), while having a good processor (small  $\Delta t$ ). In Figures 2 and 4 we observe how the error becomes smaller as we improve our processor (decreasing  $\Delta t$ ), while having a good IMU (small  $\sigma^2$ ).

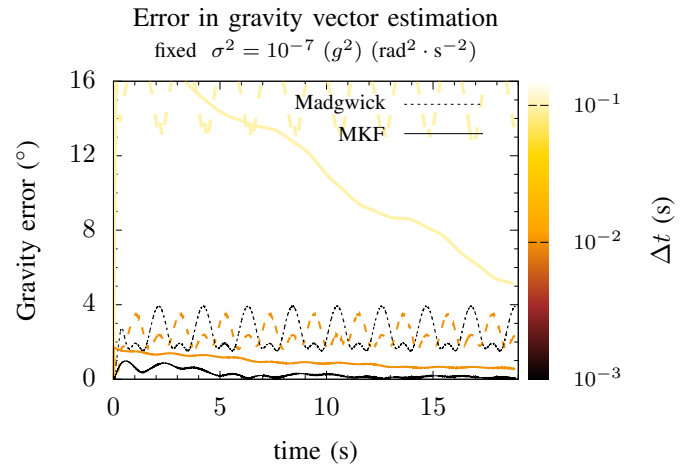
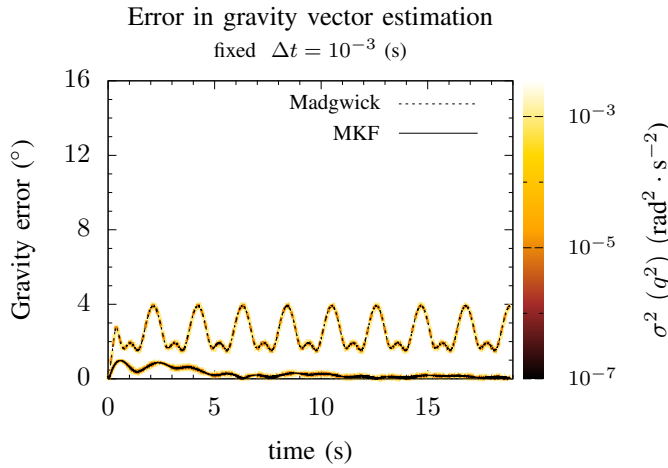
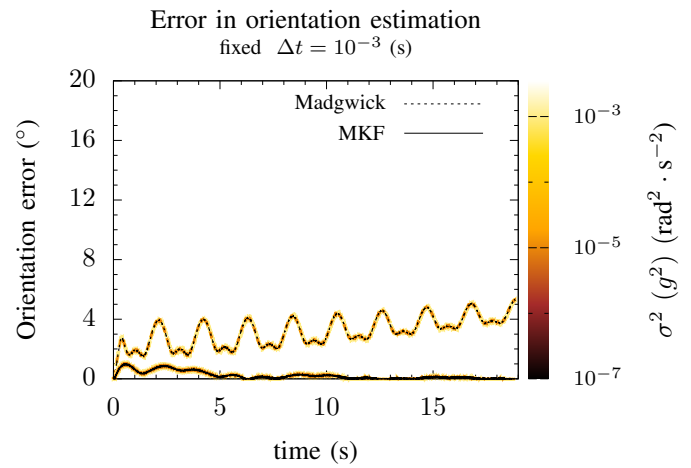
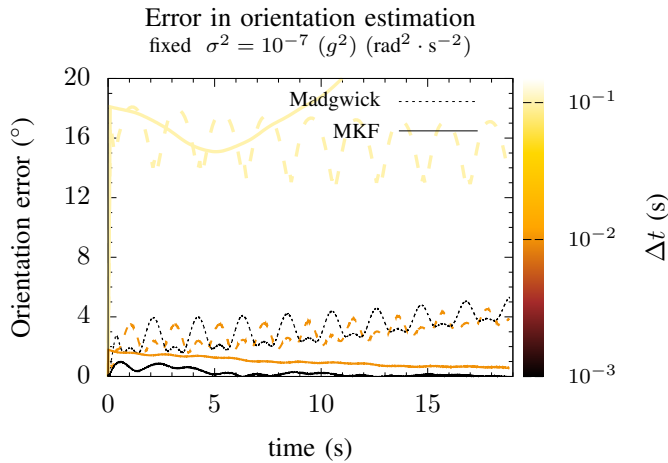


Fig. 2. Evolution of the  $e_g$  error. Test 3.

In Figure 2 we observe that the MEKF increases its accuracy in time, due to it adds information about the state in each update. The faster it updates (less  $\Delta t$ ) the faster it reaches convergence.

On the other hand the Madgwick algorithm does not adds information, which implies that it can not “learn” about the past, and it does not increases its accuracy in time. We also observe that after reaching a certain  $\Delta t$  no improvement in accuracy is seen.

In Figure 3 we can confirm the same observation made in the paragraph above. But we also note that it seems to be a limit in the algorithms accuracy as a function of the sensor noise. It could be an interesting appreciation because it could mean that beyond a certain sensor quality, there would not be

Fig. 3. Evolution of the  $e_g$  error. Test 4.Fig. 5. Evolution of the  $e_\theta$  error. Test 4.Fig. 4. Evolution of the  $e_\theta$  error. Test 3.

an appreciable improvement in the estimation.

In Figure 4 we can notice that the errors tends to increase over time. It is best appreciated for the Madgwick algorithm, and for low update frequencies with the MEKF. Probably it is not appreciated for higher update frequencies because we have not waited enough.

This happens because we have no reference for orientation in the plane perpendicular to the gravity vector. If we want to have a complete non-biased estimation of the orientation we should add measurements from additional sensors as a magnetometer, or a camera.

In Figure 5 we again see the same behavior noticed in the previous paragraph. We also repeat our observation about the limit of the estimation accuracy as a function of the sensor noise made two paragraphs above.

### B. Real Experiment

Our real experiment consists on the visual inspection of the returned information by our algorithm, and the one returned by the algorithm implemented in a commercial IMU.

1) *Test Bed:* The algorithm has been implemented in a real system. It has been used a board containing a MPU6050 sensor. The processing is performed in the ATmega328P chip contained in an Arduino board. In this system the sampling time turns out to be about

$$\Delta t \approx 0.04 \text{ s} ,$$

what means that the algorithm runs about 25 times per second. The sensor variance is approximately

$$\sigma_a^2 \approx 10^{-4} (g)^2 , \quad (73)$$

$$\sigma_\omega^2 \approx 10^{-4} (\text{rad/s})^2 . \quad (74)$$

Figure 6 shows the assembled system, consisting in the MPU6050 sensor, the Arduino UNO, and a MTi sensor of Xsens.

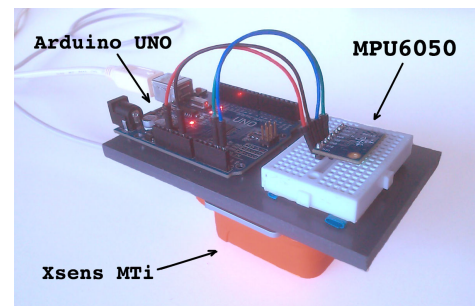


Fig. 6. Real system composed of an Arduino Uno, a MPU6050 chip, and a MTi sensor of Xsens.

2) *Experiment Results:* We have described a series of movements with the system. The movements have been carried out in four phases. The dynamics of each phase has been more aggressive than that of the previous phase. We have tried to finish with the same orientation with which the system began. Both have been saved the sensors measurements and the estimated states which are returned by the algorithms. In Figures 7-9 these data are shown.



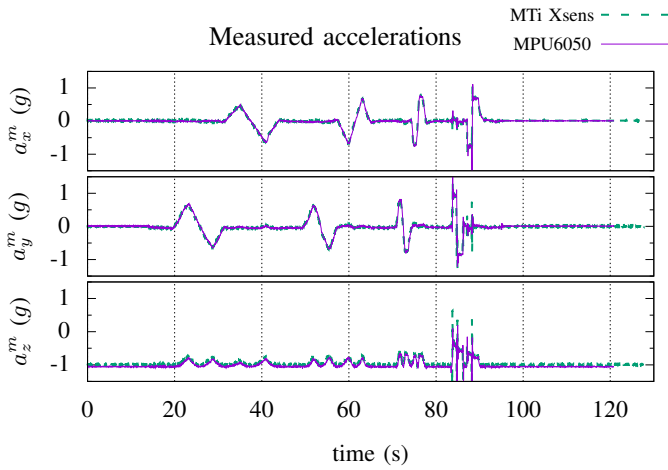


Fig. 7. Acceleration measurement during the real experiment.

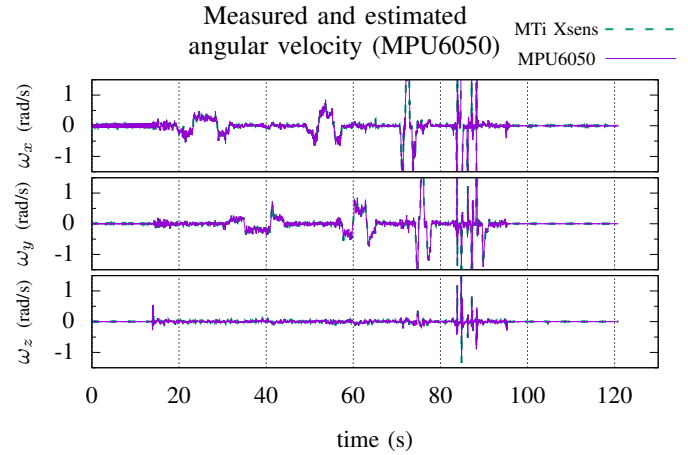


Fig. 9. Estimated and measured angular velocity during the real experiment.

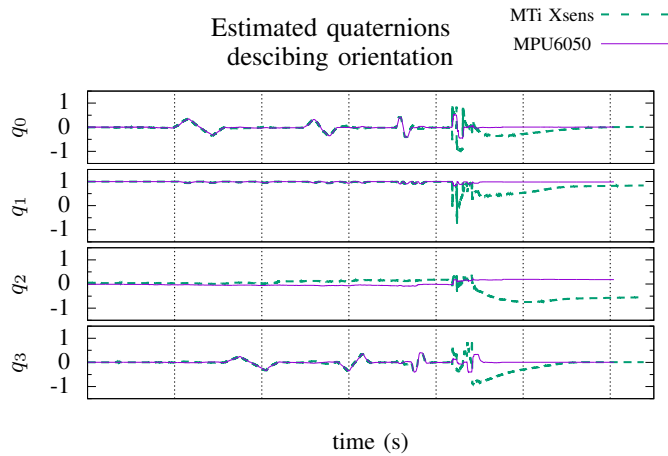


Fig. 8. Estimated quaternion describing orientation during the real experiment. The real initial and final orientations are tried to match.

In Figure 7 we can see that both sensor acceleration measurements are very similar. This makes us think that the misalignment between the two sensors is small.

In Figure 8 we note that both systems return a similar estimation of the orientation when the dynamics is not too aggressive. However, after some aggressive moves, the algorithm presented in this paper has a fast convergence. We also note the bias in the heading estimation of both algorithms when we look at  $q_2$  quaternion component. The initial and final orientation should be the same, but we have no reference for orientation in the plane perpendicular to the gravity vector.

In Figure 9 we note that the measured angular velocity is very similar to the estimated angular velocity. Perhaps we could accept the gyroscope measurement as the real angular velocity of our system. Maybe then we could get some advantage in processing speed, and therefore greater accuracy of our algorithm. But this is left for future research.

## IX. CONCLUSIONS AND FUTURE WORK

We have successfully used basic concepts of manifold theory for estimating orientations using quaternions as descriptors. A similar algorithm to the known as “Multiplicative Extended Kalman Filter” naturally arises in applying these concepts without having to redefine any aspect of the Extended Kalman Filter.

The orientation has been estimated using measurements from an IMU, but the basic idea introduced in this work is applicable to any other type of sensor intended to estimate the orientation.

We have tested the algorithm in a real experiment and we have compared our estimation with the one given by a commercial IMU, finding that both orientation estimations are similar. This tells us that our algorithm works as expected.

We also have tested the algorithm in a simulation. We have compared the performance of our algorithm with the algorithm developed by Madgwick. The results suggest that the algorithm developed in this paper could achieve a better accuracy than the one achieved by the Madgwick algorithm. However we dare not say so, as there may be various sources of error that we have not considered:

- The update frequency depends on the processor. In the ATmega328P chip, contained in an Arduino board, the Madgwick algorithm is 16 times faster than the MEKF.
- The chosen path could lead to pathological behaviors because of its symmetry.
- We have seen the result of just a path.

These considerations lead us to the following issues that will be addressed in future work:

- We will design a simulation with results based on the averaging of multiple trajectories, and free of pathological behaviors.
- We will test the algorithm for various chart definitions.
- We will test its Unscented Kalman Filter version with the various chart definitions.
- We will compare the MEKF with the MUKF, and with the Madgwick algorithm.

- We will study our sighting about the limit in the algorithm accuracy as a function of the sensor noise.

## REFERENCES

- [1] M. D. Shuster, "A survey of attitude representations," *Navigation*, vol. 8, no. 9, pp. 439–517, 1993.
- [2] 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.
- [3] 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.
- [4] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, "Estimation of imu and marg orientation using a gradient descent algorithm," in *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–7.
- [5] F. L. Markley, "Attitude error representations for kalman filtering," *Journal of guidance, control, and dynamics*, vol. 26, no. 2, pp. 311–317, 2003.
- [6] S. Madgwick. (2016) Madgwickahrs. [Online]. Available: <https://github.com/arduino-libraries/MadgwickAHRS>

APPENDIX A  
STATE PREDICTION

In this appendix we present in greater detail the developments concerning the state prediction.

## A. Evolution of the Expected Value of the State

1) Evolution in the Expected Value of the Angular Velocity: Taking the expected value in equation (16),

$$\begin{aligned} \dot{\bar{\omega}}'(\tau) &= \bar{\tilde{\tau}}'(\tau) \implies \\ \implies \bar{\omega}'_{t|t-\Delta t} &= \bar{\omega}'_{t-\Delta t|t-\Delta t} + \bar{\mathbf{q}}_t^\omega, \end{aligned}$$

with  $\mathbf{q}_t^\omega = \int_{t-\Delta t}^t \tilde{\tau}'(\tau) d\tau$ .

2) Evolution in the Expected Value of the Orientation: Taking the expected value in equation (15),

$$\begin{aligned} E \left[ \frac{d\mathbf{q}(\tau)}{d\tau} \right] &= \frac{1}{2} E [\mathbf{q}(\tau) * \omega'(\tau)] = \\ &= \frac{1}{2} E \left[ \mathbf{q}(\tau) * \left( \bar{\omega}'_{t-\Delta t|t-\Delta t} + \mathbf{q}_t^\omega(\tau) \right) \right]. \end{aligned} \quad (75)$$

Assuming the random variables  $\mathbf{q}(\tau)$  and  $\mathbf{q}_t^\omega(\tau)$  to be independent,<sup>1</sup>

$$\begin{aligned} \frac{d\bar{\mathbf{q}}(\tau)}{d\tau} &= \frac{1}{2} \bar{\mathbf{q}}(\tau) * \bar{\omega}'_{t-\Delta t|t-\Delta t} + \frac{1}{2} \bar{\mathbf{q}}(\tau) * \bar{\mathbf{q}}_t^\omega(\tau) = \\ &= \frac{1}{2} \bar{\mathbf{q}}(\tau) * \bar{\omega}'_{\tau|t-\Delta t}. \end{aligned} \quad (76)$$

This differential equation has no general closed solution. But if we assume that the expected value of the process noise,  $\bar{\mathbf{q}}_t^\omega(\tau)$ , is constant when  $\tau \in [t - \Delta t, t)$ , then we will have the matrix differential equation

$$\dot{\bar{\mathbf{q}}}(\tau) = \tilde{\Omega} \bar{\mathbf{q}}(\tau),$$

with

$$\tilde{\Omega} = \frac{1}{2} \begin{pmatrix} 0 & -\bar{\omega}'_1 & -\bar{\omega}'_2 & -\bar{\omega}'_3 \\ \bar{\omega}'_1 & 0 & \bar{\omega}'_3 & -\bar{\omega}'_2 \\ \bar{\omega}'_2 & -\bar{\omega}'_3 & 0 & \bar{\omega}'_1 \\ \bar{\omega}'_3 & \bar{\omega}'_2 & -\bar{\omega}'_1 & 0 \end{pmatrix}_{t|t-\Delta t}.$$

This differential equation has the solution

$$\bar{\mathbf{q}}(t) = e^{\tilde{\Omega} \Delta t} \bar{\mathbf{q}}(t - \Delta t).$$

We can express this solution using the quaternion product, as

$$\begin{aligned} \bar{\mathbf{q}}_{t|t-\Delta t} &= \bar{\mathbf{q}}_{t-\Delta t|t-\Delta t} * \mathbf{q}_\omega = \\ &= \bar{\mathbf{q}}_{t-\Delta t|t-\Delta t} * \begin{pmatrix} \cos \left( \frac{\bar{\omega}'_{t|t-\Delta t} \Delta t}{2} \right) \\ \frac{\bar{\omega}'_{t|t-\Delta t}}{\bar{\omega}'_{t|t-\Delta t}} \sin \left( \frac{\bar{\omega}'_{t|t-\Delta t} \Delta t}{2} \right) \end{pmatrix}. \end{aligned} \quad (77)$$

<sup>1</sup>This is, torques acting on the system does not depend on its orientation.

## B. Evolution of the State Covariance Matrix

### 1) Differential Equations for our Chart:

a) *Differential Equation for  $\delta$* : Using (8), (15), and (76),

$$\begin{aligned} \mathbf{q} &= \bar{\mathbf{q}} * \delta \quad \implies \\ \implies \dot{\mathbf{q}} &= \dot{\bar{\mathbf{q}}} * \delta + \bar{\mathbf{q}} * \dot{\delta} \quad \equiv \\ \equiv \frac{1}{2} \mathbf{q} * \boldsymbol{\omega}' &= \frac{1}{2} \bar{\mathbf{q}} * \bar{\boldsymbol{\omega}}' * \delta + \bar{\mathbf{q}} * \dot{\delta} \quad \implies \end{aligned}$$

Isolating the  $\dot{\delta}$  quaternion,

$$\begin{aligned} \dot{\delta} &= \frac{1}{2} \underbrace{\bar{\mathbf{q}}^* * \mathbf{q}}_{\delta} * \boldsymbol{\omega}' - \frac{1}{2} \underbrace{\bar{\mathbf{q}}^* * \bar{\mathbf{q}}}_{\mathbf{1}} * \bar{\boldsymbol{\omega}}' * \delta = \\ &= \frac{1}{2} [\delta * \boldsymbol{\omega}' - \bar{\boldsymbol{\omega}}' * \delta] = \\ &= \frac{1}{2} \left[ \begin{pmatrix} \delta_0 \\ \boldsymbol{\delta} \end{pmatrix} * \begin{pmatrix} 0 \\ \boldsymbol{\omega}' \end{pmatrix} - \begin{pmatrix} 0 \\ \bar{\boldsymbol{\omega}}' \end{pmatrix} * \begin{pmatrix} \delta_0 \\ \boldsymbol{\delta} \end{pmatrix} \right] = \\ &= \frac{1}{2} \begin{pmatrix} -(\boldsymbol{\omega}' - \bar{\boldsymbol{\omega}}') \cdot \boldsymbol{\delta} \\ \delta_0 (\boldsymbol{\omega}' - \bar{\boldsymbol{\omega}}') - (\boldsymbol{\omega}' + \bar{\boldsymbol{\omega}}') \times \boldsymbol{\delta} \end{pmatrix} = \\ &= \frac{1}{2} \begin{pmatrix} -\Delta\boldsymbol{\omega}' \cdot \boldsymbol{\delta} \\ \delta_0 \Delta\boldsymbol{\omega}' - (2\bar{\boldsymbol{\omega}}' + \Delta\boldsymbol{\omega}') \times \boldsymbol{\delta} \end{pmatrix}. \quad (78) \end{aligned}$$

b) *Differential Equations for  $\mathbf{e}$  on the Chart*: Using (9) and (78),

$$\begin{aligned} \dot{\mathbf{e}} &= 2 \frac{\dot{\delta} \delta_0 - \dot{\delta}_0 \delta}{\delta_0^2} = \\ &= \Delta\boldsymbol{\omega}' - (2\bar{\boldsymbol{\omega}}' + \Delta\boldsymbol{\omega}') \times \frac{\mathbf{e}}{2} + \left[ \Delta\boldsymbol{\omega}' \cdot \frac{\mathbf{e}}{2} \right] \frac{\mathbf{e}}{2}, \end{aligned}$$

or in matrix form,

$$\dot{\mathbf{e}} = \Delta\boldsymbol{\omega}' - [2\bar{\boldsymbol{\omega}}' + \Delta\boldsymbol{\omega}']_{\times} \frac{\mathbf{e}}{2} + \frac{\mathbf{e}}{2} \frac{\mathbf{e}^T}{2} \Delta\boldsymbol{\omega}' \quad (79)$$

### 2) Differential Equations for the Covariance Matrix:

#### a) Evolution Equation for $\mathbf{P}^{\omega\omega}$ :

$$\begin{aligned} \mathbf{P}_t^{\omega\omega} &= E[\Delta\boldsymbol{\omega}'_t \Delta\boldsymbol{\omega}'_t{}^T] = \\ &= E[(\boldsymbol{\omega}'_t - \bar{\boldsymbol{\omega}}'_t)(\boldsymbol{\omega}'_t - \bar{\boldsymbol{\omega}}'_t)^T] = \\ &= E[(\Delta\boldsymbol{\omega}'_{t-\Delta t} + \Delta\mathbf{q}_t^\omega)(\Delta\boldsymbol{\omega}'_{t-\Delta t} + \Delta\mathbf{q}_t^\omega)^T] = \\ &= \underbrace{E[\Delta\boldsymbol{\omega}'_{t-\Delta t} \Delta\boldsymbol{\omega}'_{t-\Delta t}{}^T]}_{\mathbf{P}_{t-\Delta t}^{\omega\omega}} + E[\Delta\boldsymbol{\omega}'_{t-\Delta t} \Delta\mathbf{q}_t^{\omega T}] + \\ &\quad + E[\Delta\mathbf{q}_t^\omega \Delta\boldsymbol{\omega}'_{t-\Delta t}{}^T] + \underbrace{E[\Delta\mathbf{q}_t^\omega \Delta\mathbf{q}_t^{\omega T}]}_{\mathbf{Q}_t^\omega}. \end{aligned}$$

Assuming the random variables  $\boldsymbol{\omega}'_{t-\Delta t}$  and  $\mathbf{q}_t^\omega$  to be independents, their covariance is null. In such case,

$$\mathbf{P}_t^{\omega\omega} = \mathbf{P}_{t-\Delta t}^{\omega\omega} + \mathbf{Q}_t^\omega \quad (80)$$

b) *Differential Equation for  $\mathbf{P}^{ee}$* : Replacing (21) in (25) we obtain,

$$\begin{aligned} \dot{\mathbf{P}}^{ee} &= E[\Delta\boldsymbol{\omega}' \mathbf{e}^T] + \frac{1}{4} E[\mathbf{e} \mathbf{e}^T \Delta\boldsymbol{\omega}' \mathbf{e}^T] + \\ &\quad - [\bar{\boldsymbol{\omega}}']_{\times} E[\mathbf{e} \mathbf{e}^T] - \frac{1}{2} E[\Delta\boldsymbol{\omega}'_{\times} \mathbf{e} \mathbf{e}^T] + \\ &\quad + E[\mathbf{e} \Delta\boldsymbol{\omega}'^T] + \frac{1}{4} E[\mathbf{e} \Delta\boldsymbol{\omega}'^T \mathbf{e} \mathbf{e}^T] + \\ &\quad - E[\mathbf{e} \mathbf{e}^T] [\bar{\boldsymbol{\omega}}']_{\times}^T - \frac{1}{2} E[\mathbf{e} \mathbf{e}^T [\Delta\boldsymbol{\omega}'_{\times}^T]]. \end{aligned}$$

Here we can see the consequences of treating a nonlinear system. The evolution in the covariance matrix  $\mathbf{P}^{ee}$ , which is composed by moments of second order, is affected by the higher moments of the distribution. To find the evolution equations of the covariance matrix we would need information about the moments of order 3 and 4. These may depend of moments of order higher than them. Knowing all the moments of a distribution would mean to know all statistical information. What we can assume and expect is that higher moments to be negligible compared to second-order moments. In that case we can write,

$$\begin{aligned} \dot{\mathbf{P}}^{ee} &\approx E[\Delta\boldsymbol{\omega}' \mathbf{e}^T] - [\bar{\boldsymbol{\omega}}']_{\times} E[\mathbf{e} \mathbf{e}^T] + \\ &\quad + E[\mathbf{e} \Delta\boldsymbol{\omega}'^T] - E[\mathbf{e} \mathbf{e}^T] [\bar{\boldsymbol{\omega}}']_{\times}^T = \\ &= (\mathbf{P}^{e\omega})^T - [\bar{\boldsymbol{\omega}}']_{\times} \mathbf{P}^{ee} + \\ &\quad + \mathbf{P}^{e\omega} - \mathbf{P}^{ee} [\bar{\boldsymbol{\omega}}']_{\times}^T. \quad (81) \end{aligned}$$

c) *Differential Equation for  $\mathbf{P}^{e\omega}$* : Replacing (21) in (26) we obtain,

$$\begin{aligned} \dot{\mathbf{P}}^{e\omega} &= E[\Delta\boldsymbol{\omega}' \Delta\boldsymbol{\omega}'^T] + \frac{1}{4} E[\mathbf{e} \mathbf{e}^T \Delta\boldsymbol{\omega}' \Delta\boldsymbol{\omega}'^T] + \\ &\quad - [\bar{\boldsymbol{\omega}}']_{\times} E[\mathbf{e} \Delta\boldsymbol{\omega}'^T] + \\ &\quad - \frac{1}{2} E[\Delta\boldsymbol{\omega}'_{\times} \mathbf{e} \Delta\boldsymbol{\omega}'^T] + E[\mathbf{e} \Delta\tilde{\boldsymbol{\tau}}'^T]. \end{aligned}$$

Having assumed the independence of  $\mathbf{q}$  and  $\mathbf{q}_t^\omega$ , then  $\mathbf{e}$  and  $\Delta\tilde{\boldsymbol{\tau}}'$ , are also independent. With that in mind, and neglecting higher order moments,

$$\begin{aligned} \dot{\mathbf{P}}^{e\omega} &\approx E[\Delta\boldsymbol{\omega}' \Delta\boldsymbol{\omega}'^T] - [\bar{\boldsymbol{\omega}}']_{\times} E[\mathbf{e} \Delta\boldsymbol{\omega}'^T] = \\ &= \mathbf{P}^{\omega\omega} - [\bar{\boldsymbol{\omega}}']_{\times} \mathbf{P}^{e\omega}. \quad (82) \end{aligned}$$

3) *Evolution Equations for the Covariance Matrix*: Denoting  $\boldsymbol{\Omega} = [\bar{\boldsymbol{\omega}}']_{\times}$  our differential equations take the form:

$$\mathbf{P}_t^{\omega\omega} = \mathbf{P}_{t-\Delta t}^{\omega\omega} + \mathbf{Q}_t^\omega, \quad (83)$$

$$\dot{\mathbf{P}}^{e\omega}(\tau) \approx \mathbf{P}^{\omega\omega}(\tau) - \boldsymbol{\Omega} \mathbf{P}^{e\omega}(\tau), \quad (84)$$

$$\dot{\mathbf{P}}^{ee}(\tau) \approx (\mathbf{P}^{e\omega}(\tau))^T - \boldsymbol{\Omega} \mathbf{P}^{ee}(\tau) + \quad (85)$$

$$+ \mathbf{P}^{e\omega}(\tau) - \mathbf{P}^{ee}(\tau) \boldsymbol{\Omega}^T. \quad (86)$$

a) *Evolution Equation for  $\mathbf{P}^{e\omega}$ :*

First of all, let us consider the homogeneous equation of (84),

$$\dot{\mathbf{P}}^{e\omega}(\tau) = -\Omega \mathbf{P}^{e\omega}(\tau) . \quad (87)$$

The solution for (87) is

$$\mathbf{P}^{e\omega}(\tau) = e^{-\Omega\tau} \mathbf{P}^{e\omega}(0) .$$

To find the solution to the inhomogeneous differential equation, we use the *variation of constants method*:

$$\begin{aligned} \mathbf{P}^{e\omega}(\tau) &= e^{-\Omega\tau} \mathbf{C}(\tau) \implies \\ \dot{\mathbf{P}}^{e\omega}(\tau) &= -\Omega e^{-\Omega\tau} \mathbf{C}(\tau) + e^{-\Omega\tau} \dot{\mathbf{C}}(\tau) = \\ &= -\Omega \mathbf{P}^{e\omega}(\tau) + e^{-\Omega\tau} \dot{\mathbf{C}}(\tau) . \end{aligned}$$

Identifying the terms in the differential equation we obtain the following relation:

$$\begin{aligned} e^{-\Omega\tau} \dot{\mathbf{C}}(\tau) &= \mathbf{P}^{\omega\omega}(\tau) \implies \\ \implies \dot{\mathbf{C}}(\tau) &= e^{\Omega\tau} \mathbf{P}^{\omega\omega}(\tau) . \quad (88) \end{aligned}$$

To solve this last differential equation, it is necessary to propose a continuous evolution equation for  $\mathbf{P}^{\omega\omega}$ . The simplest option is the linear function

$$\mathbf{P}^{\omega\omega}(\tau) = \mathbf{P}^{\omega\omega}(0) + \mathbf{Q}_0^\omega \tau , \quad (89)$$

with  $\mathbf{Q}_t^\omega = \mathbf{Q}_0^\omega t$ , being  $\mathbf{Q}_0^\omega$  a constant matrix representing the process noise covariance per unit of time.

Having defined this continuous evolution equation, (88) is transformed into

$$\begin{aligned} \dot{\mathbf{C}}(\tau) &= e^{\Omega\tau} [\mathbf{P}^{\omega\omega}(0) + \mathbf{Q}_0^\omega \tau] = \\ &= \left( \sum_{n=0}^{\infty} \frac{\Omega^n \tau^n}{n!} \right) [\mathbf{P}^{\omega\omega}(0) + \mathbf{Q}_0^\omega \tau] . \quad (90) \end{aligned}$$

Integrating (90),

$$\begin{aligned} \mathbf{C}(\tau) &= \left( \sum_{n=0}^{\infty} \frac{\Omega^n \tau^{n+1}}{(n+1)!} \right) \mathbf{P}^{\omega\omega}(0) + \\ &+ \left( \sum_{n=0}^{\infty} \frac{\Omega^n \tau^{n+2}}{n!(n+2)} \right) \mathbf{Q}_0^\omega + \mathbf{C}' . \end{aligned}$$

The constant  $\mathbf{C}'$  is determined by the initial conditions  $\mathbf{P}^{e\omega}(0) = \mathbf{C}(0)$ . With this in mind,

$$\begin{aligned} \mathbf{P}^{e\omega}(\tau) &= e^{-\Omega\tau} \left[ \left( \sum_{n=0}^{\infty} \frac{\Omega^n \tau^{n+1}}{(n+1)!} \right) \mathbf{P}^{\omega\omega}(0) + \right. \\ &\left. + \left( \sum_{n=0}^{\infty} \frac{\Omega^n \tau^{n+2}}{n!(n+2)} \right) \mathbf{Q}_0^\omega + \mathbf{P}^{e\omega}(0) \right] . \quad (91) \end{aligned}$$

Knowing that we have the information at  $t - \Delta t$ , and we want to update the information at  $t$ , our equation becomes

$$\begin{aligned} \mathbf{P}_{t|t-\Delta t}^{e\omega} &= e^{-\Omega\Delta t} \left[ \left( \sum_{n=0}^{\infty} \frac{\Omega^n (\Delta t)^{n+1}}{(n+1)!} \right) \mathbf{P}_{t-\Delta t|t-\Delta t}^{\omega\omega} + \right. \\ &\left. + \left( \sum_{n=0}^{\infty} \frac{\Omega^n (\Delta t)^{n+2}}{n!(n+2)} \right) \mathbf{Q}_0^\omega + \mathbf{P}_{t-\Delta t|t-\Delta t}^{e\omega} \right] . \quad (92) \end{aligned}$$

Finally, knowing that calculating infinite sums would take a lot, we can truncate in the first term, and write

$$\begin{aligned} \mathbf{P}_{t|t-\Delta t}^{e\omega} &\approx e^{-\Omega\Delta t} \left[ \mathbf{P}_{t-\Delta t|t-\Delta t}^{e\omega} + \right. \\ &\left. + \Delta t \mathbf{P}_{t-\Delta t|t-\Delta t}^{\omega\omega} + \frac{(\Delta t)^2}{2} \mathbf{Q}_0^\omega \right] . \quad (93) \end{aligned}$$

b) *Evolution Equation for  $\mathbf{P}^{ee}$ :*

In this case we have the homogeneous equation

$$\dot{\mathbf{P}}^{ee}(\tau) \approx -\Omega \mathbf{P}^{ee}(\tau) - \mathbf{P}^{ee}(\tau) \Omega^T ,$$

whose solution is

$$\mathbf{P}^{ee}(\tau) = e^{-\Omega\tau} \mathbf{P}^{ee}(0) e^{-\Omega^T\tau} .$$

Using the *variation of constants method*,

$$\begin{aligned} \mathbf{P}^{ee}(\tau) &= e^{-\Omega\tau} \mathbf{C}(\tau) e^{-\Omega^T\tau} \implies \\ \implies \dot{\mathbf{P}}^{ee}(\tau) &= -\Omega \mathbf{P}^{ee}(\tau) + \\ &+ e^{-\Omega\tau} \dot{\mathbf{C}}(\tau) e^{-\Omega^T\tau} + \\ &- \mathbf{P}^{ee}(\tau) \Omega^T . \end{aligned}$$

Identifying terms, we deduce the relation

$$\dot{\mathbf{C}}(\tau) = e^{\Omega\tau} \left[ \mathbf{P}^{e\omega}(\tau) + (\mathbf{P}^{e\omega}(\tau))^T \right] e^{\Omega^T\tau} .$$

After substituting the expression for  $\mathbf{P}^{e\omega}$ , integrate with respect to time, and truncating in the first term of the infinite sums, the solution we want is given by

$$\begin{aligned} \mathbf{P}_{t|t-\Delta t}^{ee} &= e^{-\Omega\Delta t} \left[ \mathbf{P}_{t-\Delta t|t-\Delta t}^{ee} + \right. \\ &+ (\Delta t)^2 \mathbf{P}_{t-\Delta t|t-\Delta t}^{\omega\omega} + \frac{(\Delta t)^3}{3} \mathbf{Q}_0^\omega + \\ &+ \mathbf{P}_{t-\Delta t|t-\Delta t}^{e\omega} \Delta t + \\ &\left. + (\mathbf{P}_{t-\Delta t|t-\Delta t}^{e\omega})^T \Delta t \right] e^{-\Omega^T\Delta t} . \end{aligned}$$

## APPENDIX B MEASUREMENT PREDICTION

In this appendix we present in greater detail the developments concerning the measurement prediction.

### A. Expected Value of the Measurement

#### 1) Expected Value of the Accelerometer Measurement:

Taking the expected value on (17),

$$\bar{\mathbf{a}}_t^m = E[\mathbf{R}_t^T (\mathbf{a}_t - \mathbf{g}_t)] + \bar{\mathbf{r}}_t^a .$$

The  $\mathbf{g}_t$  vector does not change. If we also assume that the accelerations affecting our system,  $\mathbf{a}_t$ , does not depend on its orientation,

$$\bar{\mathbf{a}}_t^m = E[\mathbf{R}_t^T] (\bar{\mathbf{a}}_t - \bar{\mathbf{g}}_t) + \bar{\mathbf{r}}_t^a .$$

One might be tempted to try to find the expected value of the matrix  $\mathbf{R}_t$  written as a function of the  $\mathbf{q}_t$  quaternion, but then we would run into the problem of the computation of expected values such as  $E[q_1^2]$  or  $E[q_1 q_2]$ . These expected values are defined in the manifold, and are what we try to avoid defining covariance matrices in the charts.

What we seek is not the expected value of the rotation matrix, but something like the ‘‘expected transformation’’. Then, using the quaternion describing the orientation of our system, this expression must be equivalent to

$$\bar{\mathbf{a}}_t^m = \bar{\mathbf{q}}_t^* * (\bar{\mathbf{a}}_t - \bar{\mathbf{g}}_t) * \bar{\mathbf{q}}_t + \bar{\mathbf{r}}_t^a .$$

### B. Covariance Matrix of the Measurement

#### 1) Accelerometer Block: Knowing that

$$\begin{aligned} \mathbf{h}_a(\mathbf{e}_t, \mathbf{a}_t) &= \mathbf{q}_t^* * (\mathbf{a}_t - \bar{\mathbf{g}}_t) * \mathbf{q}_t = \\ &= \boldsymbol{\delta}_t^* * \bar{\mathbf{q}}_t^* * (\mathbf{a}_t - \bar{\mathbf{g}}_t) * \bar{\mathbf{q}}_t * \boldsymbol{\delta}_t = \\ &= (\mathbf{R}_t^\delta)^T \bar{\mathbf{R}}_t^T (\mathbf{a}_t - \bar{\mathbf{g}}_t) , \end{aligned}$$

we will have,

$$\left. \frac{\partial \mathbf{h}_a}{\partial \mathbf{a}_t} \right|_{\substack{\mathbf{e}_t=\mathbf{0} \\ \mathbf{a}_t=\bar{\mathbf{a}}_t}} = \bar{\mathbf{R}}_t^T ,$$

$$\left. \frac{\partial \mathbf{h}_a}{\partial \mathbf{e}_t} \right|_{\substack{\mathbf{e}_t=\mathbf{0} \\ \mathbf{a}_t=\bar{\mathbf{a}}_t}} = \left. \frac{\partial (\mathbf{R}_t^\delta)^T}{\partial \boldsymbol{\delta}_t} \right|_{\boldsymbol{\delta}=\mathbf{1}} \left. \frac{\partial \boldsymbol{\delta}_t}{\partial \mathbf{e}_t} \right|_{\mathbf{e}_t=\mathbf{0}} \bar{\mathbf{R}}_t^T (\bar{\mathbf{a}}_t - \bar{\mathbf{g}}_t) .$$

Let us note that  $\frac{\partial \boldsymbol{\delta}_t}{\partial \mathbf{e}_t} \in \mathbb{R}^{4 \times 3}$ . However, the term  $\frac{\partial (\mathbf{R}_t^\delta)^T}{\partial \boldsymbol{\delta}_t}$  is a  $1 \times 4$  matrix, whose elements are again  $3 \times 3$  matrices.

#### a) First term:

Given the expression for the rotation matrix corresponding to quaternion  $\boldsymbol{\delta}$ ,

$$(\mathbf{R}^\delta)^T = \begin{pmatrix} 1 - 2\delta_2^2 - 2\delta_3^2 & 2(\delta_1\delta_2 + \delta_3\delta_0) & 2(\delta_1\delta_3 - \delta_2\delta_0) \\ 2(\delta_1\delta_2 - \delta_3\delta_0) & 1 - 2\delta_1^2 - 2\delta_3^2 & 2(\delta_2\delta_3 + \delta_1\delta_0) \\ 2(\delta_1\delta_3 + \delta_2\delta_0) & 2(\delta_2\delta_3 - \delta_1\delta_0) & 1 - 2\delta_1^2 - 2\delta_2^2 \end{pmatrix} ,$$

we can derive and evaluate at  $\boldsymbol{\delta} = \mathbf{1} = (1, 0, 0, 0)$ , resulting

$$\left. \frac{\partial (\mathbf{R}^\delta)^T}{\partial \delta_0} \right|_{\boldsymbol{\delta}=\mathbf{1}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} ,$$

$$\left. \frac{\partial (\mathbf{R}^\delta)^T}{\partial \delta_1} \right|_{\boldsymbol{\delta}=\mathbf{1}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & -2 & 0 \end{pmatrix} ,$$

$$\left. \frac{\partial (\mathbf{R}^\delta)^T}{\partial \delta_2} \right|_{\boldsymbol{\delta}=\mathbf{1}} = \begin{pmatrix} 0 & 0 & -2 \\ 0 & 0 & 0 \\ 2 & 0 & 0 \end{pmatrix} ,$$

$$\left. \frac{\partial (\mathbf{R}^\delta)^T}{\partial \delta_3} \right|_{\boldsymbol{\delta}=\mathbf{1}} = \begin{pmatrix} 0 & 2 & 0 \\ -2 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} .$$

#### b) Second term:

For the charts defined by (9), we can recover the quaternion by (10). Then,

$$\frac{\partial \boldsymbol{\delta}}{\partial \mathbf{e}} = \frac{1}{(4 + \|\mathbf{e}\|^2)^{3/2}} \begin{pmatrix} -2e_1 & -2e_2 & -2e_3 \\ 4+e_2^2+e_3^2 & -e_1e_2 & -e_1e_3 \\ -e_2e_1 & 4+e_1^2+e_3^2 & -e_2e_3 \\ -e_3e_1 & -e_3e_2 & 4+e_1^2+e_2^2 \end{pmatrix} .$$

And evaluating at  $\mathbf{e} = \mathbf{0}$ ,

$$\left. \frac{\partial \boldsymbol{\delta}}{\partial \mathbf{e}} \right|_{\mathbf{e}=\mathbf{0}} = \frac{1}{2} \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} .$$

#### c) The matrix:

Denoting  $\bar{\mathbf{g}}_t^{\mathbf{R}} = \bar{\mathbf{R}}_t^T (\bar{\mathbf{a}} - \bar{\mathbf{g}})$ , and computing the matrix products,

$$\left. \frac{\partial \mathbf{h}_a(\mathbf{e}_t)}{\partial \mathbf{e}_t} \right|_{\substack{\mathbf{e}_t=\mathbf{0} \\ \mathbf{a}_t=\bar{\mathbf{a}}_t}} = \begin{pmatrix} 0 & -(\bar{\mathbf{g}}_t^{\mathbf{R}})_3 & (\bar{\mathbf{g}}_t^{\mathbf{R}})_2 \\ (\bar{\mathbf{g}}_t^{\mathbf{R}})_3 & 0 & -(\bar{\mathbf{g}}_t^{\mathbf{R}})_1 \\ -(\bar{\mathbf{g}}_t^{\mathbf{R}})_2 & (\bar{\mathbf{g}}_t^{\mathbf{R}})_1 & 0 \end{pmatrix} .$$

### APPENDIX C UPDATE

In this appendix we present in greater detail the developments concerning the filter update.

### A. Chart Update

Differentiating our transition map,

$$\begin{aligned} \frac{\partial \mathbf{e}_p}{\partial \mathbf{e}_q} &= 2 \frac{\bar{\delta}_0 \mathbf{1} - [\bar{\boldsymbol{\delta}}]_{\times}}{2\bar{\delta}_0 + \bar{\boldsymbol{\delta}} \cdot \mathbf{e}_q} + \\ &- 2 \frac{\bar{\delta}_0 \mathbf{e}_q - 2\bar{\boldsymbol{\delta}} - \bar{\boldsymbol{\delta}} \times \mathbf{e}_q}{(2\bar{\delta}_0 + \bar{\boldsymbol{\delta}} \cdot \mathbf{e}_q)^2} \bar{\boldsymbol{\delta}}^T . \end{aligned}$$

Now, if  $\varphi_{\bar{p}} = \varphi_{\bar{q}|t}$ , and if  $\varphi_{\bar{q}} = \varphi_{\bar{q}|t-\Delta t}$ , and we are interested in expressing the covariance matrix in the chart  $\varphi_{\bar{q}|t}$ , from the covariance matrix expressed in the chart  $\varphi_{\bar{q}|t-\Delta t}$ , where  $\bar{\mathbf{e}}_q = \bar{\mathbf{e}}_{q|t}$ , then we must keep in mind that

$$\begin{aligned} \mathbf{e}_p(\bar{\mathbf{e}}_{q|t}) &= 2 \frac{\bar{\delta}_0 \bar{\mathbf{e}}_{q|t} - 2\bar{\boldsymbol{\delta}} - \bar{\boldsymbol{\delta}} \times \bar{\mathbf{e}}_{q|t}}{2\bar{\delta}_0 + \bar{\boldsymbol{\delta}} \cdot \bar{\mathbf{e}}_{q|t}} = \\ &= \varphi_1(\underbrace{\bar{\boldsymbol{\delta}}^* * \boldsymbol{\delta}(\bar{\mathbf{e}}_{q|t})}_{\bar{\boldsymbol{\delta}}}) = \mathbf{0} . \end{aligned}$$

Then replacing in (43),

$$\mathbf{e}_p(\mathbf{e}_q) \approx 2 \frac{\bar{\delta}_0 \mathbf{1} - [\bar{\boldsymbol{\delta}}]_{\times}}{2\bar{\delta}_0 + \bar{\boldsymbol{\delta}} \cdot \bar{\mathbf{e}}_{t|t}} (\mathbf{e}_q - \bar{\mathbf{e}}_{t|t}) .$$

Finally, knowing that

$$2\bar{\delta}_0 + \bar{\boldsymbol{\delta}} \cdot \bar{\mathbf{e}}_{t|t} = \underbrace{(\bar{\boldsymbol{\delta}}^* * \boldsymbol{\delta}(\bar{\mathbf{e}}_{t|t}))_0}_1 \sqrt{4 + \|\bar{\mathbf{e}}_{t|t}\|^2} ,$$

we have

$$\begin{aligned} \mathbf{e}_p(\mathbf{e}_q) &\approx \frac{2}{\sqrt{4 + \|\bar{\mathbf{e}}_{t|t}\|^2}} (\bar{\delta}_0 \mathbf{1} - [\bar{\boldsymbol{\delta}}]_{\times}) (\mathbf{e}_q - \bar{\mathbf{e}}_{t|t}) = \\ &= \bar{\delta}_0 (\bar{\delta}_0 \mathbf{1} - [\bar{\boldsymbol{\delta}}]_{\times}) (\mathbf{e}_q - \bar{\mathbf{e}}_{t|t}) . \end{aligned}$$