A Comparison of Bayesian Filters for Orientation Estimation

Claudio J. Paz, Gonzalo F. Perez Paina, and Julio H. Toloza

National University of Technology, Cordoba Regional Faculty, Argentina.
{cpaz, gperez, jtoloza}@scdt.frc.utn.edu.ar

Abstract— **This work presents an empirical study that compares the performance of Bayesian filters for orientation estimation using data provided by an Inertial Measurement Unit. A two-stages measurement update was implemented for different variants of Kalman and particles filters using quaternions as orientation representation method. Empirical results show that all tested algorithms converge to the correct orientation even if the initial orientation is unknown.**

Keywords— **Quaternions, Bayesian Filter, Extended Kalman Filter, Unscented Kalman Filter, Particle Filter, Gaussian Particle Filter, Inertial Measurement Units**

1 INTRODUCTION

An important topic in the research field of mobile robotics is autonomous navigation, in which it is required to determine the robot position and orientation with high precision. The problem of position and orientation (also known as pose) estimation depends on the kind of robot locomotion, being either terrestrial or aerial robots. Recently, there is an increasing interest in using Unmanned Aerial Vehicles (UAV) for autonomous tasks for different applications. Pose estimation for aerial vehicles has to be performed in tree dimensional space, which is composed of 6 variables (3 for position and 3 for orientation). Moreover, this also applies for Autonomous Ground Vehicles (AGV) operating in an outdoor environment in rough terrain.

In mobile robotics there exist different types of sensor that can be used for position and orientation estimation, and generally a combination of the data provided by several of them provides better results. This combination is known as sensor fusion or integration, and Bayesian filtering is widely used approach for doing it. Generally inertial sensors like accelerometers, gyroscopes, and magnetometers are combined for pose estimation in mobile robotics; and recently the development of MEMS (Micro Electro Mechanical Systems) [1] applied to inertial sensoring has made possible to reduce their production cost.

Bayesian filtering is a recursive method used for state estimation based on measurements affected by noise with known statistical description, together with the stochastic state space system model [2]. Moreover, as previously mentioned, Bayesian filtering can be used to perform information fusion using different sensors. Numerous implementations of Bayesian filters exist, being either Gaussian or non-Gaussian approaches. Gaussian approaches include the generally used Extended Kalman Filter (EKF)[3], and the Unscented Kalman Filter (UKF)[4]; and non-Gaussian approach are based on Monte Carlo method known as Particle Filters (PF)[5].

Extended Kalman Filter (EKF) is an extension of the Kalman Filter (KF), where the main difference is that while the latter is applied for linear system the former is for non-linear cases. The linearization in EKF is performed using a first order Taylor series expansion centered in the estimated value, which is used to propagate the mean and covariance of a Gaussian random variable. In a different way, the UKF avoid the use of a linearizing version of non-linear model because is based on the Unscented Transform (UT). The UT allows to propagate a Gaussian random variable through a non-linear function, and differs from the Taylor expansion in the sense that the UT directly approximates the mean and covariance of the target distribution instead of trying to approximate the non-linear function. The Unscented Transform works by choosing sample points (sigma points) deterministically which capture the mean and covariance of the original distribution. This points are transformed by the non-linear function and then statistics of the transformed samples are computed. On the other hand, Particle Filter (PF) works similarly to the UKF in the sense that the target distribution of a given random variable is represented by samples or particles, even thought it is not restricted to Gaussian or even unimodal distribution. An special case of PF is the Gaussian Particle Filter (GPF)[6] which is based on the propagation of the mean and covariance matrix of the variables. The implementation of a PF have an important problem known as sample degeneracy, which can be partially overcome using resample techniques. However, in GPF sample degeneracy can be avoided by drawing new samples in every step using the estimated mean and covariance matrix.

In this work, a comparative study of different Bayesian filtering methods will be performed applied to the problem of orientation estimation using inertial measurements from MEMS sensors. In this respect we extend the work done in [7] and [8] in comparing other filters (PF and GPF) capable of being implemented in parallelizable platforms. The orientation estimation is based on quaternion representation which presents some advantages against other orientation representation. The stochastic state space model for quaternion estimation will be described in details. A comparative evaluation will be shown using EKF, UKF as well as PF and GPF, all based on measurements from real sensors.

This work is organized as follows. In Section 2, an introduction to recursive Bayesian state estimation techniques including EKF, UKF, PF and GPF are given. Details concerning the implementations of the different filters is discussed in section 3. Finally, theirs results are shown in section 4.

2 BAYESIAN FILTERS

Given a stochastic state-space model, with

$$\mathbf{x}_{k+1} = f_k(\mathbf{x}_k, \mathbf{w}_k), \tag{1}$$
$$\mathbf{y}_k = h_k(\mathbf{x}_k, \mathbf{v}_k), \tag{2}$$

as the process and observation equations respectively where $\mathbf{x}_k \in \mathbb{R}^n$ is the state vector at time k, with $f_k : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ as a state transition function and where $\mathbf{y}_k \in \mathbb{R}^p$ is the observation vector at time k, with $h_k : \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}^p$ as a measurement function relating the state vector $\mathbf{x}_k$ with the observation $\mathbf{y}_k$. Additionally $\mathbf{w}_k \in \mathbb{R}^n$ and $\mathbf{v}_k \in \mathbb{R}^r$ are zero mean white noise.

Considering a time step k, if $p(\mathbf{x}_{k-1}|D_{k-1})$ is available at $k-1$, with $D_{k-1} = \{\mathbf{y}_i : i = 1, \ldots, k-1\}$ the set of all measurements until k, it is possible to use the system model to find the prior state probability density function (pdf)

$$p(\mathbf{x}_k|D_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|D_{k-1}) d\mathbf{x}_{k-1}, \tag{3}$$

where $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is the state transition pdf from step $k - 1$ to step k.

Later, when observation $\mathbf{y}_k$ is available, the prior state can be updated via the Bayes rule

$$p(\mathbf{x}_k|D_k) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|D_{k-1})}{p(\mathbf{y}_k|D_{k-1})}, \tag{4}$$

where

$$p(\mathbf{y}_k|\mathbf{x}_k) = \int \delta(\mathbf{y}_k - h_k(\mathbf{x}_k, \mathbf{v}_k)) p(\mathbf{v}_k) d\mathbf{v}_k \tag{5}$$

is known as likelihood and can be obtained by the system measurement model and parameters of the noise $\mathbf{v}_k$.

The denominator of (4) is a normalization factor called evidence and it is given by

$$p(\mathbf{y}_k|D_{k-1}) = \int p(\mathbf{y}_k|\mathbf{x}_k) p(\mathbf{x}_k|D_{k-1}) d\mathbf{x}_k. \tag{6}$$

The goal of the recursive Bayesian filter is to find the current state pdf $p(\mathbf{x}_k|D_k)$ given all the information available at the discrete time k with the successive application of prediction and update, (3) and (4) respectively.

When $f(\cdot)$ and $h(\cdot)$ are linear, it is possible to find a closed-form solution and the KF algorithms are suitable. Otherwise, alternative methods must be used.

2.1 Extended Kalman Filter

The Extended Kalman Filter is a variant of the KF in which $f_k(\cdot)$ and $h_k(\cdot)$ are linearaized around the estimated state $\hat{\mathbf{x}}_k$. Formally, given the discrete time non-linear system in (1) and (2) the state estimation $\hat{\mathbf{x}}_k$ is performed following the next steps.

- *Prediction*. Compute the process equation Jacobian around previous state estimation $\hat{\mathbf{x}}_{k-1}$ and predict the state $\hat{\mathbf{x}}_k^-$ and covariance error matrix P_k^-

$$F_{k-1} = \left.\frac{\partial f_{k-1}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k-1}}, \tag{7}$$
$$Q_{k-1} = E[\mathbf{w}_{k-1}\mathbf{w}_{k-1}^T], \tag{8}$$
$$\hat{\mathbf{x}}_k^- = f_{k-1}(\hat{\mathbf{x}}_{k-1}), \tag{9}$$
$$P_k^- = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1}. \tag{10}$$

- *Update*. Obtain the measurement equation Jacobian around the predicted state of the actual step $\hat{\mathbf{x}}_k^-$ then update the state prediction and covariance error matrix

$$H_k = \left.\frac{\partial h_k}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_k^-}, \tag{11}$$
$$R_k = E[\mathbf{v}_k\mathbf{v}_k^T], \tag{12}$$
$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}, \tag{13}$$
$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k(\mathbf{y}_k - h_k(\hat{\mathbf{x}}_k^-)), \tag{14}$$
$$P_k = (I - K_k H_k)P_k^-. \tag{15}$$

2.2 Unscented Kalman Filter

Another way of dealing with non-linear functions is using the unscented transform (UT). The UT is a numerical method used for estimate the mean and covariance of a random variable after a non-linear transformation.

2.2.1 *Unscented transform*

Given a random variable $\mathbf{x}$ with mean $\bar{\mathbf{x}}$ and covariance $P_{\mathbf{x}}$ and a non-linear transformation $\Phi(\cdot)$, UT seeks to obtain a new random variable $\mathbf{y} = \Phi(\mathbf{x})$. It is possible to recover the statistic parameters of $\mathbf{y}$ via the unscented transform following the next steps.

- Calculate sigma points $\mathbf{X} = \{\mathcal{X}_i : i = 0, \ldots, 2n\}$ with n the state space dimension

$$\mathcal{X}_0 = \bar{\mathbf{x}}, \tag{16}$$
$$\mathcal{X}_i = \bar{\mathbf{x}} + \sqrt{l+\lambda} \cdot S_i, \quad i = 1, \ldots, l, \tag{17}$$
$$\mathcal{X}_i = \bar{\mathbf{x}} - \sqrt{l+\lambda} \cdot S_i, \quad i = l+1, \ldots, 2l, \tag{18}$$

with $\lambda = \alpha^2(l + \kappa) - l$ and l the dimension of the state space. S_i is the i-th column of a Cholesky decomposition of matrix $P_{\mathbf{x}}$ to get $P_{\mathbf{x}} = SS^T$. The parameter α determines the spread of sigma points around the mean, κ is a secondary scaling parameter which is usually set to 0 or $3 - l$[9].

- Calculate each sigma point weight $\mathcal{W}$

$$\mathcal{W}_0^m = \frac{\lambda}{l+\lambda}, \quad (19)$$

$$\mathcal{W}_0^c = \frac{\lambda}{l+\lambda} + 1 - \alpha^2 + \beta, \quad (20)$$

$$\mathcal{W}_i^m = \mathcal{W}_i^c = \frac{\lambda}{2(l+\lambda)}, \quad i = 1, \ldots, 2l. \quad (21)$$

  The parameter β is used to incorporate prior knowledge of the distribution of **x** (for Gaussian distributions $\beta = 2$ is optimal)[9].

- Apply the non-linear function $f(\cdot)$ to sigma points, estimate new mean $\bar{\mathbf{y}}$ and covariance $P_{\mathbf{y}}$ and calculate cross covariance between input and output $P_{\mathbf{xy}}$

$$\mathcal{Y}_i = \Phi(\mathcal{X}_i), \quad i = 1, \ldots, 2l, \quad (22)$$

$$\bar{\mathbf{y}} = \sum_{i=0}^{2l} \mathcal{W}_i^m \mathcal{Y}_i, \quad (23)$$

$$P_{\mathbf{y}} = \sum_{i=0}^{2l} \mathcal{W}_i^c (\mathcal{Y}_i - \bar{\mathbf{y}}) \cdot (\mathcal{Y}_i - \bar{\mathbf{y}})^T, \quad (24)$$

$$P_{\mathbf{xy}} = \sum_{i=0}^{2l} \mathcal{W}_i^c (\mathcal{X}_i - \bar{\mathbf{x}}) \cdot (\mathcal{Y}_i - \bar{\mathbf{y}})^T. \quad (25)$$

2.2.2 UKF implementation

Given the non-linear system modeled by (1) and (2) the state estimation $\hat{\mathbf{x}}_k$ is performed as follows:

- Form the augmented state vector $\hat{\mathbf{x}}^a$ and augmented error covariance matrix P^a with mean and covariance from previous step and the process noise covariance matrix Q_{k-1} from (8)

$$\mathbf{x}_{k-1}^a = \begin{pmatrix} \hat{\mathbf{x}}_{k-1} \\ \mathbf{0} \end{pmatrix}, \quad (26)$$

$$P_{k-1}^a = \begin{pmatrix} P_{k-1} & \mathbf{0} \\ \mathbf{0} & Q_{k-1} \end{pmatrix}. \quad (27)$$

- Use the UT to obtain predicted state $\hat{\mathbf{x}}_k^-$ and predicted covariance error matrix P_k^- using $\Phi(\cdot)$ and non-linear function $f(\cdot)$

- Prepare a new augmented vector using predicted state $\hat{\mathbf{x}}_k^-$, predicted covariance error matrix P_k^- and measurement noise covariance matrix R_k like (12)

$$\mathbf{x}_k^{-a} = \begin{pmatrix} \hat{\mathbf{x}}_k^- \\ \mathbf{0} \end{pmatrix}, \quad (28)$$

$$P_k^{-a} = \begin{pmatrix} P_k^- & \mathbf{0} \\ \mathbf{0} & R_k \end{pmatrix}. \quad (29)$$

- Obtain the measurement prediction $\hat{\mathbf{y}}^-$ using UT with the augmented elements obtained in the previous step and with non-linear function $h(\cdot)$ and calculate Kalman gain matrix

$$K_k = P_{\mathbf{xy}} P_{\mathbf{y}}^{-1}, \quad (30)$$

- With Kalman gain update the state and covariance error matrix

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k(\mathbf{y} - \hat{\mathbf{y}}^-), \quad (31)$$

$$P_k = P_k^- - K_k P_{\mathbf{y}} K_k^t. \quad (32)$$

2.3 Particle Filter

This algorithm is so called because consists of a set of N elements or particles which are system state hypothesis. Each of these particles have a paired weight proportional to the likelihood of this state. Thus, a discrete pdf is obtained without need of its parameters and it is possible to deal with high nonlinearities in measurement and process functions [5].

Formally, given a set of N particles or samples $\mathbf{X}_{k-1} = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$ and associated weights $\mathbf{W}_{k-1} = \{w^1, \ldots, w^N\}$ distributed according to the pdf $p(\mathbf{x}_{k-1}|D_{k-1})$, the particles filter performs the propagation of samples through the non-linear function $f(\cdot)$ and updates this samples via Bayes rule and non-linear function $h(\cdot)$ to obtain the target pdf $p(\mathbf{x}_k|D_k)$. This is accomplished with recursive approximations of (3) and (4) as follows:

- In prediction step, pass each particle through the non-linear function

$$\mathbf{x}_k^{i-} = f(\hat{\mathbf{x}}_{k-1}^i, \mathbf{w}_{k-1}^i), \quad (33)$$

  where $\mathbf{w}_{k-1}^i$ are noise samples distributed according to $p(\mathbf{w}_{k-1})$.

- In update step, assign a weight value to each particle depending of their likelihood [10]

$$w_k^i \propto w_{k-1}^i \, p(\mathbf{y}_k|\mathbf{x}_k^{i-}), \quad (34)$$

- Normalize particles weight.

$$w_k^i = \frac{w_k^i}{\sum\limits_{j=1}^{N} w_k^j}, \quad (35)$$

This implementation has the so called degeneracy problem, in which after a few steps, all but one particle will have negligible weights. It is possible to reduce the effects of degeneracy with a method called resampling. In this method, particles with higher weight are selected and copied to replace those with smaller weights [10]. This step is the bottleneck of speed performance in PF. Different methods are known to perform the resampling [11]. In the simplest [5], N samples according to uniform distribution $\mathcal{U}$ on the interval $(0, 1]$ are drawn, and the particle $\mathbf{x}_k^m$ is selected for the copy as long as it meets with

$$\sum_{j=0}^{m-1} w_k^j < u_k^i \le \sum_{j=0}^{m} w_k^j. \quad (36)$$

After resampling, all weights must be set to $1/N$.

Assuming unimodality, the system state can be calculated using the particles and their updated weights, perform a weighted mean to obtain the estimated state $\hat{\mathbf{x}}_k$

$$\hat{\mathbf{x}}_k = \sum_{i=1}^{N} w_k^i \, \mathbf{x}_k^{i-}. \tag{37}$$

2.4 Gaussian Particle Filter

Gaussian particle filter is a variant of PF in which the samples are drawn according to previous mean and covariance estimation in each step. Hence, the resampling stage is not needed. The procedure is simpler than PF and it is described in the following steps.

- Draw samples from previous mean and covariance and denote them as $\{\mathbf{x}_k^j : j = 0, \dots, M\}$
- Pass this samples through the non-linear process function $f(\cdot)$ and compute the predicted mean $\hat{\mathbf{x}}_k^-$ and covariance P_k^-

$$\hat{\mathbf{x}}_k^- = \frac{1}{M} \sum_{j=0}^{N} \mathbf{x}_k^j, \tag{38}$$

$$P_k^- = \frac{1}{M} \sum_{j=0}^{N} \mathbf{x}_k^j. \tag{39}$$

- Draw samples according to predicted mean $\hat{\mathbf{x}}_k^-$ and covariance P_k^-
- Obtain the likelihood of each particle using the multivariate normal distribution as in (34)
- Normalize particles weight as in (35)
- Compute the mean $\hat{\mathbf{x}}_k$ and covariance P_k using the particles and their weights

$$\hat{\mathbf{x}}_k = \sum_{j=0}^{N} w_k^j \, \mathbf{x}_k^j, \tag{40}$$

$$P_k = \sum_{j=0}^{N} w_k^j \, \mathbf{x}_k^j. \tag{41}$$

Since at each step the algorithm draws new samples, the GPF does not suffer the degeneracy phenomenon and does not need resampling step.

3 FILTER IMPLEMENTATIONS

The goal of the filter is the state estimation using inertial sensor readings. Three axes accelerometers, gyroscopes and magnetometers are available.

Orientation of a rigid body is defined by a rotation from a reference frame to the current body frame. Generally, the reference frame is a fixed coordinate system and the body frame is attached to the body, consequently it rotates and moves with it. For notation matters a left superscript letter n is used to refer the navigation or reference frame and b is used to refer the body frame.

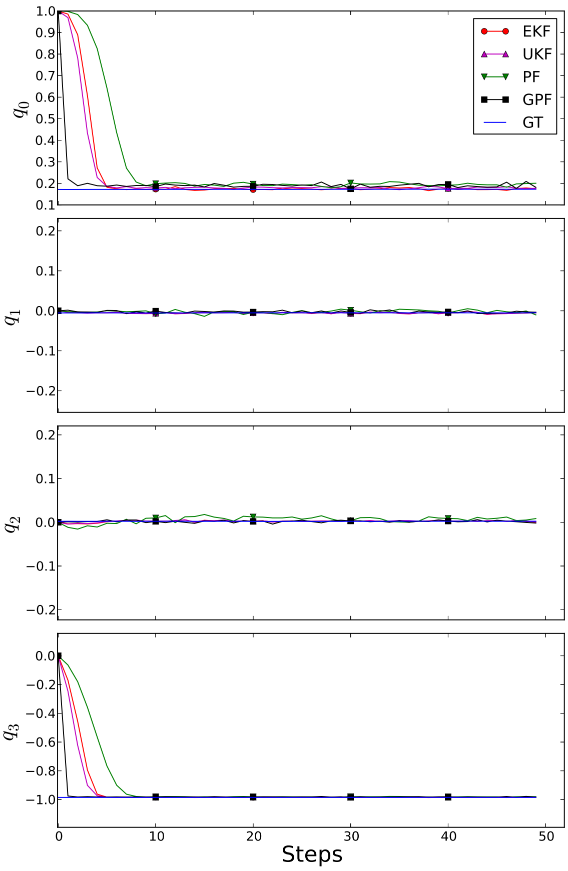

Figure 1: Estimated quaternions vs. GT

3.1 System State

The system state is composed of the orientation quaternion

$$\mathbf{x}_k = \mathbf{q}_k = \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix}, \tag{42}$$

which must always have unity norm.

3.2 Process equation

Given the time derivative of quaternion $\mathbf{q}$[12]

$$\dot{\mathbf{q}} = \frac{d\mathbf{q}}{dt} = \frac{1}{2}\mathbf{q} * \boldsymbol{\omega} \tag{43}$$

with the skew symmetric matrix of $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{pmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{pmatrix} \tag{44}$$

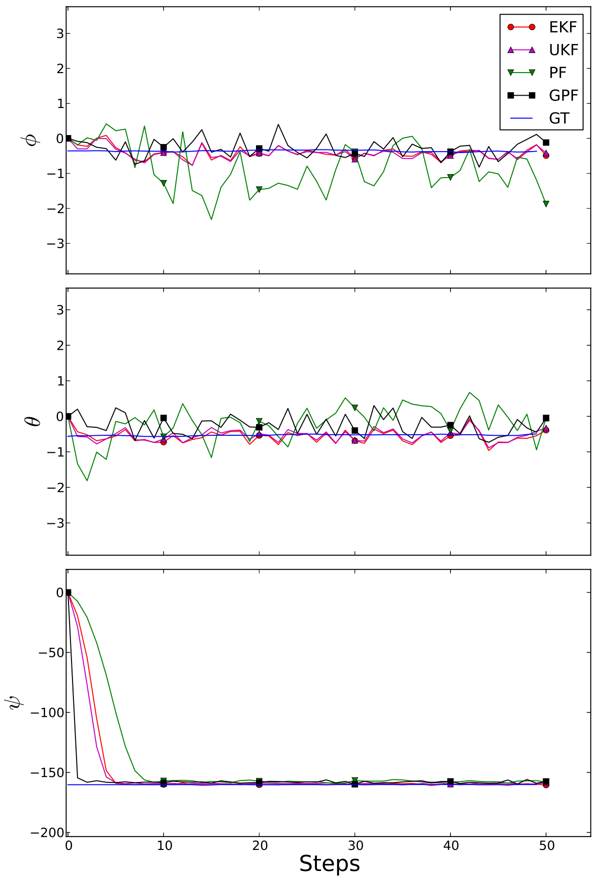

Figure 2: Euler angles from estimated quaternions

it is possible to operate as follows to find the process equation

$$\frac{\mathbf{q}_k - \mathbf{q}_{k-1}}{\Delta t} = \frac{1}{2}\mathbf{\Omega}(\boldsymbol{\omega})\mathbf{q}_{k-1} \tag{45}$$

$$\mathbf{q}_k = \frac{1}{2}\mathbf{\Omega}(\boldsymbol{\omega})\mathbf{q}_{k-1}\Delta t + \mathbf{q}_{k-1} \tag{46}$$

$$\mathbf{q}_k = \left(\frac{1}{2}\mathbf{\Omega}(\boldsymbol{\omega})\Delta t + I\right)\mathbf{q}_{k-1}, \tag{47}$$

then, using

$$F_k = \frac{1}{2}\mathbf{\Omega}(\boldsymbol{\omega})\Delta t + I, \tag{48}$$

the process equation becomes

$$\mathbf{x}_k = F_k \mathbf{x}_{k-1} + \mathbf{w}_{k-1}, \tag{49}$$

which is only valid if x_{k-1} is normalized. It can be seen than process equation is already linear.

3.3 Measurement equation

Measurement is performed in two stages, where the predicted state is updated first with the accelerometer readings and next with the magnetometer. The accelerometer is used as tilt sensor to obtain pitch and roll angles given that accelerometer detects gravity. The magnetometer is used to obtain yaw angle since it can measure the magnetic North. The two update stages allows the use of different rate sensors to correct yaw angle.

Measurement equations can be obtained using sensor models. Considering a static body, accelerometers can be modeled by

$${}^b\mathbf{a}_k = R^{bn}\,{}^n\mathbf{g} + \mathbf{b} + \mathbf{v}_{ak}, \tag{50}$$

where $\mathbf{b}$ is the bias and $\mathbf{v}_{ak}$ is zero mean Gaussian noise. Bias values can be neglected. Vector ${}^n\mathbf{g}$ is the acceleration due gravity, which varies with latitude but it can be considered constant in a reduced work space. Therefore ${}^n\mathbf{g} = [0\ 0\ g]^T$ can be supposed.

Hence, the rotation matrix from navigation to body frame according to [12] becomes

$$h_{ak}(\mathbf{x}_k) = g\begin{pmatrix} 2(q_1q_3 - q_2q_0) \\ 2(q_2q_3 + q_1q_0) \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}. \tag{51}$$

At the magnetometer update stage a similar reduction can be made. Magnetometer vector ${}^n\mathbf{m} = [m_N\ 0\ m_D]^T$ has two nonzero components, the North component and the component pointing to the center of the Earth, this third component can be eliminated given that it does not provide information about the yaw angle. Once this component is eliminated, normalization of the vector is needed to get ${}^n\tilde{\mathbf{m}} = [1\ 0\ 0]^T$. Hence, with the rotation matrix from navigation to body frame the measurement function becomes

$$h_{mk}(\mathbf{x}_k) = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 \\ 2(q_1q_2 - q_3q_0) \\ 2(q_1q_3 + q_2q_0) \end{pmatrix}. \tag{52}$$

In (51) and (52) nonlinearities are observed and it can deal with these using the described methods.

4 RESULTS

The evaluation test was made using data provided for a commercial inertial measurement unit (IMU). This IMU is able to provide Gyro-Stabilized quaternion and Euler angles, as well as raw data from inertial sensor at 50Hz. These additional features allows to compare the evaluated algorithm with a ground truth (GT) provided by the IMU itself.

All filters were set with the same parameters using the known sensor noise statistics. Besides, both PF and GPF are used with 30000 particles.

Figure 1 shows quaternion estimation obtained by each evaluated filter where it can be seen that all filters converge in few iterations. The fastest to converge is the GPF and the slowest is the PF. The comparison was made against the Gyro-Stabilized quaternion provided by IMU. Figure 2 shows roll, pitch and yaw angles calculated from quaternions of each filter [12]. Low variances are observed in all the implementations. EKF, UKF and GPF have a similar variance of $\pm 0.5°$, while the PF shows a poorer performance.

Figure 3 shows that all implementations have low estimation error. The dashed lines show the 1σ bounds.

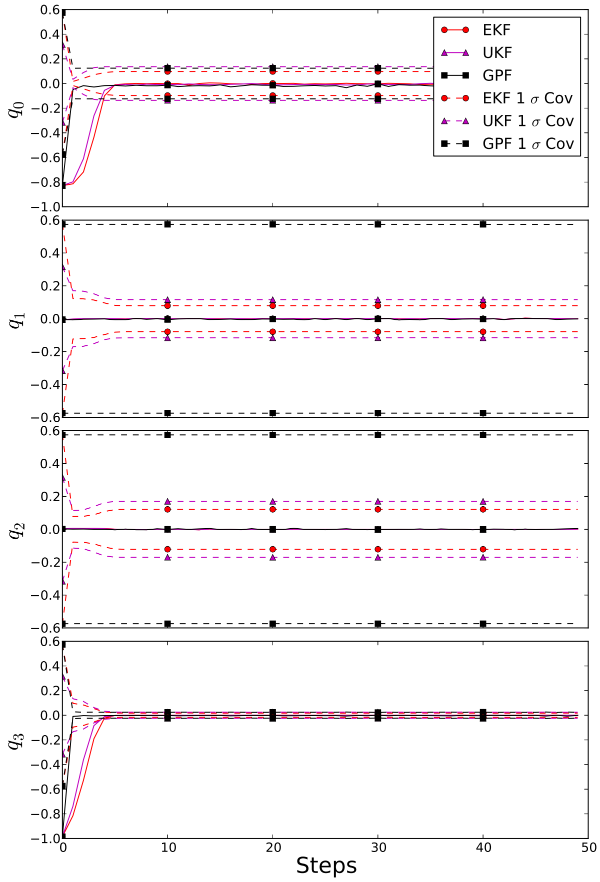

Figure 3: Estimation errors of each filter and 1σ covariance bound

5 CONCLUSIONS

In this work, an evaluation of different probabilistic algorithms for the rigid body orientation estimation was presented. Four known Bayesian filters were described and implemented, Extended Kalman Filter, Unscented Kalman Filter, Particle Filter and Gaussian Particle Filter.

A two-stages measurement algorithm was tested. This algorithm has the advantage of working with different sampling rate sensors like accelerometers and gyroscopes with digital compass or magnetometers or even cameras.

Due to its simplicity, GPF algorithm seemed to be a good candidate for orientation estimation applications. However, as it needs a rather large number of particles, it turned out to be not so suitable for implementation on simple architectures for real-time applications.

All filters show a similar performance for the quaternion based orientation estimation problem, but the GPF overcomes to the PF in terms of the number of particles required to achieve similar precision.

Future work will include parallelization of GPF for real-time applications using GPU and data provided by AGV or UAV sensors including inertial measurement units and cameras, taking advantage of the two stage measurement algorithm.

ACKNOWLEGMENTS

The authors thanks the anonymous referees for their comments and suggestions that led to an improvement of the present work.

This work was supported in part by ANPCyT grant PICT–PRH 2009/136.

REFERENCES

[1] K. Maenaka, "Mems inertial sensors and their applications," in *Networked Sensing Systems, 2008. INSS 2008. 5th International Conference on*, 2008, pp. 71–73.

[2] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, 2006.

[3] B. Anderson and J. More, *Optimal filtering*. Prentice-Hall, 2005.

[4] S. Julier, J. Uhlmann, and H. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *American Control Conference, Proceedings of the 1995*, vol. 3, 1995, pp. 1628–1632 vol.3.

[5] N. Gordon, D. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," *Radar and Signal Processing, IEE Proceedings F*, vol. 140, no. 2, pp. 107–113, 1993.

[6] J. H. Kotecha and P. Djuric, "Gaussian particle filtering," *Signal Processing, IEEE Transactions on*, vol. 51, no. 10, pp. 2592–2601, 2003.

[7] 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, 2003, pp. 2435–2440 vol.3.

[8] S. Sabatelli, M. Galgani, L. Fanucci, and A. Rocchi, "A double stage kalman filter for sensor fusion and orientation tracking in 9d imu," in *Sensors Applications Symposium (SAS), 2012 IEEE*, 2012, pp. 1–5.

[9] S. Julier, "The scaled unscented transformation," in *American Control Conference, 2002. Proceedings of the 2002*, vol. 6, 2002, pp. 4555–4559 vol.6.

[10] M. Sanjeev Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *Signal Processing, IEEE Transactions on*, vol. 50, no. 2, pp. 174–188, 2002.

[11] R. Douc and O. Cappe, "Comparison of resampling schemes for particle filtering," in *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, 2005, pp. 64–69.

[12] W. Phillips and C. Hailey, "Review of attitude representations used for aircraft kinematics," *Journal of Aircraft*, vol. 38, no. 4, pp. 718–737, 2001.