OpenMotion: An open source library for attitude estimation.

Nizar Ouarti
IPAL (Sorbonne UPMC, CNRS,
Astar, NUS, UJF, IMT)
Singapore
Email: nizar.ouarti@ipal.cnrs.fr

Thomas Braud
IPAL (Sorbonne UPMC, CNRS,
Astar, NUS, UJF, IMT)
Singapore
Email: braud.thomas@ipal.cnrs.fr

Vivien Billaud
IPAL (Sorbonne UPMC, CNRS,
Astar, NUS, UJF, IMT)
Singapore

***Abstract*—The abstract goes here**

I. Introduction

II. Global Presentation of the Library

- The North East Down (NED) frame $\{a\}$ system has its origin fixed at the (moving) object center of gravity. The z-axis points upward perpendicularly to the tangent plane of the ellipsoid, and the x-axis points towards true north (and not the magnetic north). The y-axis point towards east.
- The object-fixed reference frame $\{c\}$ is a moving and rotating coordinate frame that is fixed to the device. The x-axis points in the forward direction, the y-axis to the right side and the z-axis upward.

We admit that all the components of the IMU belong to the same reference frame $\{c\}$ (ref figure 1). It is a cyclopean approximation [**?**] .

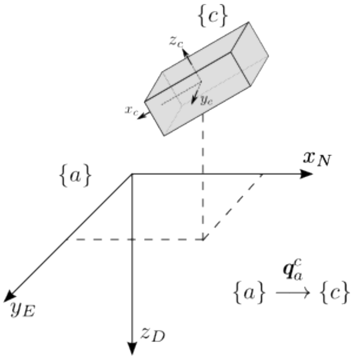


Fig. 1: Definition of the scene with 2 frame: a fixed frame NED (North East Down) noted $\{a\}$ and a moving frame object noted $\{c\}$

III. Calibration and initialization

IV. State of the art

A. Kinematic and quaternion representation

A quaternion is a four-dimensional vector, defined as $\boldsymbol{q} = \begin{pmatrix} \mathbf{q} \\ q_w \end{pmatrix}$ with $\mathbf{q} = (q_x, q_y, q_z)^T$. For attitude representation, the quaternion must satisfies a single constraint given by $||\boldsymbol{q}|| = 1$.The quaternion kinematics equation in discrete-time is given by :

$$\boldsymbol{q}_{k+1} = \Phi_{k+1|k}(\boldsymbol{\omega})\boldsymbol{q}_k \tag{1}$$

where $\boldsymbol{\omega}$ is the angular velocity and $\Phi_{k+4|k}$ is the transition matrix from time $k-1$ to k such as:

$$\Phi_{k+1|k}(\boldsymbol{\omega}) = \begin{pmatrix} \cos(0.5||\boldsymbol{\omega}||\Delta T)I_3 - [\boldsymbol{\psi}\times] & \boldsymbol{\psi} \\ -\boldsymbol{\psi}^T & \cos(0.5||\boldsymbol{\omega}||\Delta T) \end{pmatrix} \tag{2}$$

with,

$$\boldsymbol{\psi} = \sin(0.5||\boldsymbol{\omega}||\Delta T)\frac{\boldsymbol{\omega}}{||\boldsymbol{\omega}||} \tag{3}$$

And $[\mathbf{v}\times] \in \mathbb{R}^{3\times3}$ is the skew-symetric matrix of the vector $\mathbf{v}$ such as.

$$[\mathbf{v}\times] = \begin{pmatrix} 0 & v_z & -v_y \\ -v_z & 0 & v_x \\ v_y & -v_x & 0 \end{pmatrix} \tag{4}$$

Then, the attitude matrix $R \in SO(3)$ related to the quaternion $\boldsymbol{q}$ is given by :

$$R(\boldsymbol{q}) = (q_w^2 - |\mathbf{q}|^2)I + 2\mathbf{q}\mathbf{q}^T - 2q_w[\mathbf{q}\times] \tag{5}$$

Where $SO(3)$ is the special orthogonal group defined by:

$$SO(3) = \{R \in \mathbb{R}^{3\times3} | RR^T = R^T R = I, det(R) = 1\}$$

B. Most Common Fusion Methods

1) Kalman Filtering Methods: Introduced in 1960 by Kalman [**?**], the Kalman filter is an algorithm that uses a series of measurements observed over time, containing noise, in order to estimate unknown variables (state vector). Based on a model it is possible to obtain a more robust estimation of the state vector. Consider the following linear system, described by the difference equation and the observation model with additive noise:

$$\begin{cases} \mathbf{x}_k = F_k\mathbf{x}_{k-1} + B_k\mathbf{u}_k + G_k\mathbf{v}_k \\ \mathbf{z}_k = H_k\mathbf{x}_k + D_k\mathbf{w}_k \end{cases} \tag{6}$$

with,

$\mathbf{x}_k$	State vector at time k
$\mathbf{z}_k$	Measurement vector at time k
$\mathbf{u}_k$	Input vector (e.i.: from sensor) at time k
$\mathbf{v}_k$	State noise vector disturbing the system
$\mathbf{w}_k$	Observation noise disturbing the measurement

The figure **??** represents the dynamical system described in 6 . A brief presentation of the algorithm is provided here, see [**?**] for more details. The algorithm has 3 important steps :

- The first step is the **prediction** of state $\mathbf{x}_k^-$ at time k and covariance P_k^- is given by :

$$\begin{cases} \mathbf{x}_k^- = & F_k\mathbf{x}_{k-1}^+ + B_k\mathbf{u}_k \\ P_k^- = & F_kP_{k-1}^+F_k^T + G_kQ_kG_k^T \end{cases} \tag{7}$$

- The **innovation** is the difference between predicted measurement and real measurement. It measure up the deviation used by the Kalman gain.

$$\begin{cases} \boldsymbol{\nu}_k = & \mathbf{z}_k - H_k\mathbf{x}_k^- \\ S_k = & R_k + H_kP_k^-H_k^T \end{cases} \tag{8}$$

- And finally, An **update** is processed to correct the estimate state $\mathbf{x}_k^+$ and his covariance P_k^+ with the innovation.

$$\begin{cases} \mathbf{x}_k^+ = & \mathbf{x}_k^- + K_k\boldsymbol{\nu}_k \\ P_k^+ = & P_k^- - K_kS_kK_k^T \end{cases} \tag{9}$$

Where K_k is the Kalman gain representing relative importance of innovation ν_k.

$$K_k = P_k^-H_k^TS^{-1} \tag{10}$$

2) Nonlinear Observer: The theory of the state observer was first introduced by Kalman and Bucy for a linear system in a stochastic environment. Then Luenberger [**?**] made a general theory of observers for deterministic linear systems, introducing the notions of observer and reduced minimum observer [**?**]. Mahony et al. [**?**] have proposed an nonlinear observer called Constant Gain Observer (CGO), termed the explicit complementary filter, that provides attitude estimates as well as gyro bias estimates.

- Prediction

First, we propagate the estimate attitude and the gyro bias with equation 11.

$$\begin{cases} \boldsymbol{q}_k^- = \Phi_{k|k-1}(\boldsymbol{\omega}_{gyro} - \boldsymbol{\beta}_{k-1}^+)\boldsymbol{q}_{k-1}^+ \\ \boldsymbol{\beta}_k^- = \boldsymbol{\beta}_{k-1}^+ \end{cases} \tag{11}$$

- Correction

The idea is to compute a constraint given by a comparison between an estimate value of a sensor and his real data at a time k.

$$\begin{cases} \mathbf{v}_{acc}^- = R_{(}\boldsymbol{q}_k^-)\mathbf{a}^a + \boldsymbol{\beta}_{acc} \\ \mathbf{v}_{mag}^- = R_{(}\boldsymbol{q}_k^-)\mathbf{m}^a + \boldsymbol{\beta}_{mag} \end{cases} \tag{12}$$

All these vectors are normalized.

The " gain " is given by,

$$\boldsymbol{\omega}_{mes} = -vex(\sum_{i=1}^{n}\frac{k_i}{2}(\mathbf{v}_i(\mathbf{v}_i^-)^T - \mathbf{v}_i^-(\mathbf{v}_i)^T)) \tag{13}$$

where the vex operator define by the inverse of the cross product.

$$\begin{cases} \mathbf{a}(\mathbf{b})^T - \mathbf{b}(\mathbf{a})^T = [(\mathbf{a}\times\mathbf{b})\times] \\ vex([\mathbf{a}\times]) = \mathbf{a} \end{cases} \tag{14}$$

Then, we the gain is used to correct the predicted attitude $\boldsymbol{q}_k^-$ by the following equation:

$$\begin{cases} \boldsymbol{q}_k^+ = \boldsymbol{q}_k^- + \frac{\Delta T}{2}\Xi(\boldsymbol{q}_k^-)\boldsymbol{\omega}_{mes}k_p \\ \boldsymbol{\beta}_k^+ = \boldsymbol{\beta}_k^- - \boldsymbol{\omega}_{mes}k_i \end{cases} \tag{15}$$

One problem of Kalman Filters is to find the initial values in the covariance (matrix Q) because it is difficult to relate it to real physical quantities. The covariance matrix is not involve in nonlinear observer theory. This is one of the reasons there has been an increasing interest for nonlinear observer design the last decades. Another reason is their low computational need.

3) Methods Based on Solutions of the Wahba's Problem: The attitude determination problem for vector observation was first formulated as a least squares estimation problem by Wahba [**?**] in 1965 :

Given the two sets of n vectors $\{\mathbf{r}_1, ..., \mathbf{r}_n\}$ and $\{\mathbf{b}_1, ..., \mathbf{b}_n\}$, $n \geqslant 2$, where each pair $(\mathbf{r}_i, \mathbf{b}_i)$ corresponds to a generalised vector, $\mathbf{x}_i$, find the proper orthogonal matrix, A, which bring the first into the best least squares coincidence with the second. That is, find A which minimises the function J defined as:

$$J(A) = \frac{1}{2}\sum_{i=1}^{n}a_i||b_i - Ar_i||^2 \tag{16}$$

subject to the constraint $A^TA = I$ and $det(A) = 1$. The equation (16) can be written with a quaternion representation which gives:

$$J(\boldsymbol{q}) = \frac{1}{2}\sum_{i=1}^{n}a_i||b_i - R(\boldsymbol{q})r_i||^2 \tag{17}$$

The matrix $R(\boldsymbol{q})$, is the rotation matrix that corresponds to the quaternion $\boldsymbol{q}$ (equation 5). The QUEST algorithm are conducted in the following manner. Let

$$B = \sum_{i=1}^{N}a_i\mathbf{b}_i\mathbf{r}_i^T \tag{18}$$

Then, we can rewrite the equation (17) as follow

$$J(\boldsymbol{q}) = \frac{1}{2}\sum_{i=1}^{n}a_i - tr(R(\boldsymbol{q})B^T) \tag{19}$$

Due to the fact that the representation of the attitude matrix is a homogenous quadratic function of $\boldsymbol{q}$, we can say that

$$tr(R(\boldsymbol{q})B^T) = \boldsymbol{q}^T K \boldsymbol{q} \tag{20}$$

where K is the symmetric traceless matrix such as

$$K = \begin{pmatrix} S - tr(B)I_3 & \mathbf{z} \\ \mathbf{z}^T & tr(B) \end{pmatrix} \tag{21}$$

with

$$\begin{cases} S = B + B^T \\ \mathbf{z} = \sum_i a_i \mathbf{b}_i \times \mathbf{r}_i \end{cases} \tag{22}$$

The minimisation problem from Equation (19) is now a maximisation problem. thus, the solution $\boldsymbol{q}_{opt}$ is the eigen vector correspond to the largest eigen value of K [?]. The main problem of QUEST is that it does not take into account of all the previous observations performed since the beginning. A recursive QUEST algorithm (REQUEST) proposed by Bar-Itzhack manages this problem. According to the REQUEST algorithm, the propagation of K_k is given by

$$K_k^- = \Phi_{k|k-1}(\boldsymbol{\omega}_{gyro}) K_{k-1}^+ \Phi_{k|k-1}^T(\boldsymbol{\omega}_{gyro}) \tag{23}$$

We compute δK_k as

$$\delta K_k = \frac{1}{\sum_i a_i} \begin{pmatrix} \delta S_k - \delta\sigma_k I_3 & \delta\mathbf{z}_k \\ \delta\mathbf{z}_k^T & \delta\sigma_k \end{pmatrix} \tag{24}$$

with,

$$\begin{matrix} \delta B_k = \sum_i a_i \mathbf{b}_i \mathbf{r}_i^T & \delta S_k = B_k + B_k^T \\ \delta\mathbf{z}_k = \sum_i a_i \mathbf{b}_i \times \mathbf{r}_i & \delta\sigma_k = tr(B_k) \end{matrix} \tag{25}$$

Where $\mathbf{b}_i$ is the observation vector at time k. Then we update the K-matrix.

$$K_k^+ = \frac{\rho_k m_{k-1}}{m_k} K_k^- + \frac{1}{m_k} \delta K_k \tag{26}$$

A drawback is that the value of ρ_k comes from heuristics. Choukroun revisit REQUEST provides an improvement with his Optimal REQUEST [?] where an optimal ρ_k^* is computed which gives better results. Here we choose to implement this version of REQUEST. Contrary to other methods, the QUEST algorithms do not estimate the gyroscope bias. Which can make it less robust in some cases. There are also other methods based on Wahba's problem like the Davenport's q-method [?] proposed in 1971 considered more robust than QUEST. Or the method ESOQ proposed by Mortari [?] in 1997. But in this study we limit our investigation to REQUEST.

4) Particle Filtering Method: Particle filtering method is based on Monte-Carlo method[?]. The main idea is to use a large number of samples called particles to estimate a nonlinear function [?] and [?]. The set of N particle with associated weight at time k are denoted by $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}$. Initially, the particle are drawn from a proposal distribution. The PF is divided into three step, prediction, update, resampling, that constitute a filter cycle:

- Prediction

The first step is to propagate all particles $\{\mathbf{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\}$ to $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}$ with the nonlinear function f while the weight remain unchanged :

$$\mathbf{x}_k^{(i)} = f(\mathbf{x}_{k-1}^{(i)}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1}^{(i)} \tag{27}$$

where N sample $\mathbf{w}_{k-1}^{(i)}$ of the process noise (supposed gaussian in our case) are drawn.

- Update/Correction

At the update step, the weights associated with each particle is updated based on the likelihood function $p(\mathbf{z}_k|\mathbf{x}_k^{(i)})$.

$$\begin{cases} w_k^{(i)} = w_{k-1}^{(i)} p(\mathbf{z}_k|\mathbf{x}_k^{(i)}) \\ \tilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_i w_k^{(i)}} \end{cases} \tag{28}$$

The weight are normalised such as we have $\sum_i \tilde{w}_k^{(i)} = 1$. To finish, the mean is computed with the following equation:

$$\mathbf{x}_k^+ = \sum_{i=1}^{N} \tilde{w}_k^{(i)} \mathbf{x}_k^{(i)} \tag{29}$$

- Resampling/Regularisation

A common problem with PF is the degeneracy phenomenon. After some iterations, more and more particles will have a weak weight, which make them negligible and useless. To overcome this problem, a regularisation step called resampling is processed. It is not needed to perform PF but it allows to maintain a good performance relative to degeneracy problem. The idea of resampling is to eliminate particles having feeble weight and focus on strong weight particles. In order to measure degeneracy, the number of effective samples N_{eff} is computed (equation 30). If N_{eff} is lower than a threshold, the resampling step is processed.

$$N_{eff} = \sum_{i=1}^{N} \frac{1}{(\tilde{w}_k^{(i)})^2} \tag{30}$$

The choice of the threshold is important. Indeed, a high threshold gives $N_{eff} \approx N$, but that does not mean necessarily we have a lot of efficient particles. On the contrary, it means that most of the particle are identical making large computations unnecessary. Moreover, with a low threshold, many particles are neglected making computations less efficient. In general, the threshold is fixed at $\frac{N}{2}$ or $\frac{N}{3}$.

There are several types of resampling, a comparison is made in [**?**]. We took the resampling algorithm described in [**?**] because it is the most commun. Cheng, Yang and Crassidis have propose an attitute estimation with a particle filter method [**?**]. We have also considered the revisited algorithm of Cheng [**?**] based on bootstrap filter[**?**]. We have chosen to implement this one because they use the same approach as USQUE [**?**] and MEKF[**?**]. We decided to explore the performance of this three algorithms with the same model.

The particle filter provides an estimate of the nonlinear system without assuming that the noise is Gaussian. So it is more robust to any situation, which makes it more performant in the context of a real applications. But an important computation time is required and that is a limitation for a real-time application.

V. Performance Evaluation

We have access through our designed ground truth to the true attitude $\boldsymbol{q}_k^{real}$ at time k of the reference frame $\{c\}$ related to $\{a\}$. By computation, each method returns an estimate attitude $\boldsymbol{q}_k^{est}$ at time k (see figure 2).

$$\delta\boldsymbol{q}_k = \boldsymbol{q}_k^{real} \otimes (\boldsymbol{q}_k^{est})^{-1} = \begin{pmatrix} \delta\mathbf{q}_k \\ \delta q_{w_k} \end{pmatrix} \tag{31}$$

The error-MRP vector related to $\delta\boldsymbol{q}_k$ is given by

$$\delta\mathbf{p}_k = 4[\frac{\delta\mathbf{q}_k}{(1+\delta q_{w_k})}] \tag{32}$$

Thus, we can calculate the attitude error given by the root mean square (RMS) of $\delta\mathbf{p}_k$:

$$\epsilon_k = RMS(\delta\mathbf{p}_k) = \sqrt{\frac{1}{3}(\delta p_{k_x}^2 + \delta p_{k_y}^2 + \delta p_{k_z}^2)} \tag{33}$$

The mean of the error is given by :

$$\bar{\epsilon} = \frac{1}{n}\sum_{i=1}^{n} \epsilon_i \tag{34}$$

With n the number of data at our disposal. We use this information (mean error) for our performance study.

To differentiate the methods, we assigned some ”scores” in order to highlight the behaviour of these methods in different configurations. We defined these ”scores” as:

$$s_1 = \bar{\epsilon} \tag{35}$$

$$s_2 = \frac{1}{\bar{\epsilon} \times \bar{\tau}} \tag{36}$$

With $\bar{\epsilon}$ average attitude error (voir 33) et $\bar{\tau}$ the average computation time.

VI. Conclusion

notre library est trop bien...

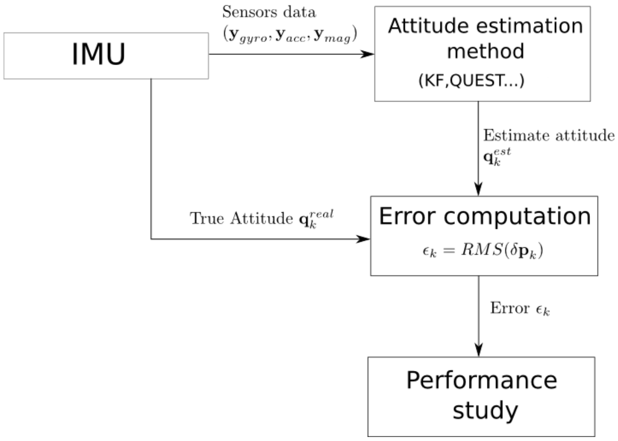
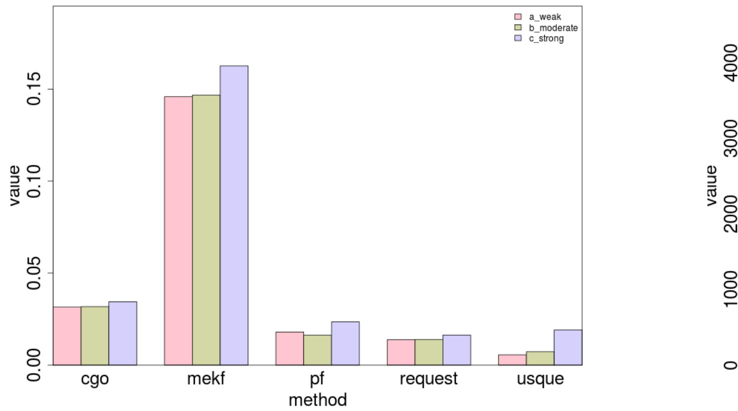

IMU
Sensors data
$(\mathbf{y}_{gyro}, \mathbf{y}_{acc}, \mathbf{y}_{mag})$
Attitude estimation method
(KF,QUEST...)
Estimate attitude
$\mathbf{q}_k^{est}$
True Attitude $\mathbf{q}_k^{real}$
Error computation
$\epsilon_k = RMS(\delta\mathbf{p}_k)$
Error ϵ_k
Performance study

Fig. 2: Algorithm designed to compute attitude estimation methods’ performances



(a) Mean Attitude Error (score s_1)

Fig. 3: Scores of methods where noise is AWGN with 3 strength