22nd International Congress of Mechanical Engineering (COBEM 2013)
November 3-7, 2013, Ribeirão Preto, SP, Brazil

GREEDY CALIBRATION OF THE ROTATION BETWEEN SENSOR AND BODY FRAME WITHOUT EXTERNAL REFERENCES

Conrado Silva Miranda
Janito Vaqueiro Ferreira
SCHOOL OF MECHANICAL ENGINEERING - UNICAMP, CAMPINAS - SP - BRASIL
{cmiranda,janito}@fem.unicamp.br

Abstract. *This work presents an easy to use algorithm for estimation of the rotation between sensor and body frame that places little restriction on the data capture methodology. Sensors aren't always precisely positioned in the system, and misalignment between the sensor and body frame may provide wrong measurements during state estimation. Using attitude estimates from the sensor frame, the algorithm iteratively optimizes a series of rotation estimates for each estimated attitude. The results show that the algorithm's performance scales linearly with attitude estimate errors and, with the least number of samples, provides an estimate as accurate as the attitude estimates.*

Keywords: *sensor calibration, alignment calibration, sensor frame, body frame*

1. INTRODUCTION

The sensor reading represents a fundamental step in the control loop of a robot, but these readings may be performed in a different frame than the one used by other algorithms present in the robot. In aerial vehicles, for instance, we care about the robot's attitude but not about the sensors' orientation by themselves. Most algorithms consider that the sensor readings occur on a desired frame of reference, named body frame, when trying to estimate some of the robot's states (Elkaim *et al.*, 2012; Gebre-Egziabher *et al.*, 2000; Thrun *et al.*, 2005; Schopp *et al.*, 2010), so the use of a different frame may cause incorrect estimates and their consequences, such as crash.

Elkaim (2013) presents a solution to the misalignment between two three-axis sensors whose unit vectors are known in the inertial frame and can be measured in body coordinates. Based on a solution to the Wahba's Problem (Shuster, 2006), the algorithm provides a rotation matrix such that the second sensor readings can be rotated to the coordinate frame that the first sensor uses for its readings. Kelly and Sukhatme (2009) describe an algorithm to calibrate the relative pose between a camera and a IMU using a filter. Although techniques like these allow every sensor reading to be represented in the same frame of reference, they don't provide the tools needed to compute the rotation between the sensor frame, in which the measurements are made, and the body frame, in which other algorithms such as attitude estimators operate.

As noted by Alonso and Shuster (2002), the polar decomposition theorem forces the magnetometer calibration to be made in a specific frame of reference. Due to the similarities between accelerometers and magnetometers, both suffer from this problem and must be calibrated in a sensor frame. Vasconcelos *et al.* (2008) presents a solution to this problem by using another sensor to measure the magnetic field in the desired body frame and finding the rotation matrix between the measurements performed in the body and sensor frame. While this method correctly solves the problem, it relies on the presence of another calibrated and aligned sensor that isn't always present.

The algorithm presented in this paper provides a solution to the problem of finding the rotation between the sensor and body frame without additional sensors. The algorithm uses a series of attitude estimates provided by other algorithms (Sabatini, 2006; Gebre-Egziabher *et al.*, 2000; Madgwick *et al.*, 2011) that are computed at rotations following the methodology described in this paper. To the best of our knowledge, this is the only algorithm that estimates the rotation between the sensor and body frame without an external reference or world knowledge.

The paper is organized as follows. In Section 2 we describe how the sensor frame is positioned for the attitude estimation to be performed. Using the attitudes, we develop a cost function and describe its optimization in Section 3, followed by the algorithm associated with the optimization in Section 4. We present the experimental study performed in this paper and its analysis in Section 5. Finally, we provide concluding remarks and future research directions in Section 6.

2. DATA CAPTURE

To describe how the data is captured, we must first establish how the sensor, body and inertial frames are related. The inertial frame $\mathcal{I}$ has its z direction perpendicular to Earth's surface pointing outwards, with the x direction pointing to an arbitrary direction that the attitude estimation algorithm may choose. The sensor frame $\mathcal{S}$ is an arbitrary rotation of the body frame, so that the calibration algorithm is free to choose the one it finds most appropriate.

The body frame $\mathcal{B}$ is just a rotation of the inertial frame $\mathcal{I}$ around its z axis. This assumption isn't restrictive because, based on this body frame, we can apply a geometrically known rotation matrix so that the new body frame is oriented as we desire. For instance, if we want the body frame to have its x direction parallel to a plane's body and going from tail to nose, we can perform the algorithm described in this paper and then rotate $\mathcal{B}$ around its y direction so that the body frame is as desired, or setup the reference orientation, which will be described latter, in a way that the plane's body is already parallel to the Earth's surface and no further calibration is needed.

It's important to highlight that the restrictions imposed to $\mathcal{B}$ and $\mathcal{I}$ aren't inherent to the algorithm. We've decided to enforce them so that the algorithm is as simple as possible, but it can be expanded to accept other orientations as long as we have enough information to build intermediary frames with known rotations.

Instead of describing how the data may be captured directly, we found it easier to first describe how the estimated attitudes are decomposed, so that the captured data must follow the imposed restriction. We'll present a method for achieving rotations that satisfy the restrictions in the end of the section.

Let $\{ {}^{i}\mathbf{q}_{\mathcal{I}}^{\mathcal{S}} \}$ be a set of known quaternions (Chou, 1992) collected, where ${}^{i}\mathbf{q}_{B}^{A}$ represents the ith quaternion that rotates from the frame B to A. For these quaternions to be valid inputs of our algorithm, in the absence of noise we must be able to decompose them in the following way:

$$ {}^{i}\mathbf{q}_{\mathcal{I}}^{\mathcal{S}} = \mathbf{q}_{\mathcal{B}}^{\mathcal{S}} \otimes {}^{i}\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}} \otimes {}^{i}\mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'} \otimes \mathbf{q}_{\mathcal{I}}^{\mathcal{I}'} \tag{1} $$

where $\otimes$ is the quaternion product operator and

$\mathbf{q}_{\mathcal{I}}^{\mathcal{I}'}$ is a quaternion that rotates an angle $\theta_{\mathcal{I}}$ around the z axis. This rotation constructs a new frame $\mathcal{I}'$ so that $\mathcal{I}$ may be chosen by the attitude algorithm, establishing $\mathcal{I}'$ as the reference orientation.

${}^{i}\mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'}$ is a quaternion that rotates an angle $\theta_{\mathcal{I}'}^{i} \in (-\frac{pi}{2}, \frac{pi}{2})$ around the z axis. This rotation allows each attitude to have its own deviation from $\mathcal{I}'$ by defining a new auxiliary frame $\mathcal{B}'$, thus making the data capture less restrictive to the user. However, to add this freedom we require the user to know whether $\theta_{\mathcal{I}'}^{i}$ is positive, negative or zero.

${}^{i}\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}}$ is a quaternion that rotates an angle $\theta_{\mathcal{B}'}^{i} \in (-\frac{pi}{2}, \frac{pi}{2})$ around the local x or y axis. This rotation finally defines the frame $\mathcal{B}$ at which the ith attitude was estimated. Like the previous quaternion, we require the user to inform whether $\theta_{\mathcal{B}'}^{i}$ is positive, negative or zero, in addition to around which axis the rotation was performed.

$\mathbf{q}_{\mathcal{B}}^{\mathcal{S}}$ is a quaternion that performs an arbitrary rotation. This is the rotation that we want to know so that measurements made in $\mathcal{S}$ may be represented in $\mathcal{B}$.

With this decomposition restriction, we suggest the following technique to position the system and estimate the attitudes:

1. Place the system in a plane parallel to the Earth's surface in the reference orientation. Estimate current attitude and make a mark on the ground indicating the current orientation, making it easier to note whether $\theta_{\mathcal{I}'}^{i}$ will be positive or negative.

2. Repeat steps 2.1 and 2.2 until enough attitudes are estimated.

2.1 Rotate the system around its z axis, respecting the restriction on $\theta_{\mathcal{I}'}^{i}$. Estimate current attitude and note whether the angle was positive or negative.

2.2 Rotate the system around its z axis and then around its local x or y axis, keeping in mind the restriction on $\theta_{\mathcal{I}'}^{i}$ and $\theta_{\mathcal{B}'}^{i}$. Estimate current attitude, and note the angles' signs and the axis used by the rotation.

We find this capture algorithm straightforward to be used as little information besides the attitudes is needed and the restrictions are mild, allowing it to be used in almost any system. As will be shown, the user may impose more restrictions to the method, such as more than one attitude estimate with the same $\theta_{\mathcal{I}'}^{i}$, to improve performance. Steps 1 and 2.1 are important because they are more restrictive than step 2.2 as at least one of the rotations is null, which helps the algorithm to find the other rotations involved more easily. However, the algorithm presents some quirks that may not be obvious and we must elucidate.

The need to known the angles' signs comes from the fact that the problem has multiple possible solutions. Consider, for instance, that a certain attitude was obtained by angles $\theta_{\mathcal{I}}$, $\theta_{\mathcal{I}'}^{i}$ and $\theta_{\mathcal{B}'}^{i}$. Without any information about the last angle's sign, the solution $\theta_{\mathcal{I}}' = \theta_{\mathcal{I}} + \pi$, $\theta_{\mathcal{I}'}^{i}$ and $\theta_{\mathcal{B}'}^{i'} = -\theta_{\mathcal{B}'}^{i}$ also provides the same attitude. In the absence of the sign's knowledge, there would be no way to ensure that $\mathcal{B}$ matches the desired frame, and the rotation between $\mathcal{B}$ and $\mathcal{S}$ may become incorrect.

However, the estimates for angles $\theta_{\mathcal{I}'}^{i}$ and $\theta_{\mathcal{B}'}^{i}$ may become inconsistent with the data provided if a certain attitude presents itself close to the border between negative and positive rotations at $\pm\pi$ or 0. This occurs because, due to noise, an angle that is positive may become more similar to a negative one, misleading the algorithm. Therefore, although the signs' knowledge is important to allow more freedom for the attitudes, it becomes a burden if noise makes it inconsistent with the data.

3. COST FUNCTION OPTIMIZATION

To estimate each quaternion in Eq. (1), we'll optimize a cost function. Let $\{{}^{i}\mathbf{q}_{\mathcal{I}}^{\mathcal{S}}\}$ be a set of N attitudes collected as specified in the previous section, then we can define a cost function given by

$$J\left(\mathbf{q}_{\mathcal{B}}^{\mathcal{S}}, \{{}^{i}\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}}\}, \{{}^{i}\mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'}\}, \mathbf{q}_{\mathcal{I}}^{\mathcal{I}'}; \{{}^{i}\mathbf{q}_{\mathcal{I}}^{\mathcal{S}}\}\right) = \sum_{i=1}^{N} \left\| \mathbf{q}_0 - \mathbf{q}_{\mathcal{B}}^{\mathcal{S}} \otimes {}^{i}\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}} \otimes {}^{i}\mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'} \otimes \mathbf{q}_{\mathcal{I}}^{\mathcal{I}'} \otimes \left({}^{i}\mathbf{q}_{\mathcal{I}}^{\mathcal{S}}\right)^{-1} \right\|^2 \tag{2}$$

where $\mathbf{q}_0$ is a quaternion without any rotation.

Note that this formulation doesn't take into account the quaternions' double mapping, which may lead to low quality local minima. To avoid this behaviour and still keep the method simple, all quaternions must use the same side of the double mapping. Therefore, we must ensure that all quaternions have their scalar components positive, negating them otherwise.

By using the formulation given in Eq. (2) and considering that all other quaternions are known, we can estimate each quaternion by solving the least squares (LS) problem (Kariya and Kurata, 2004). The LS problem may be stated as follows: given the observed values $\mathbf{X}$ and $\mathbf{y}$, compute β to minimize the cost function

$$J(\beta; \mathbf{X}, \mathbf{y}) = \|\mathbf{X}\beta - \mathbf{y}\|^2. \tag{3}$$

This problem's solution is given by

$$\beta^* = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{y} \tag{4}$$

To solve the LS problem for each quaternion in Eq. (2), we'll write $\mathbf{X}$ and $\mathbf{y}$ as

$$\mathbf{X} = \begin{bmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \\ \vdots \\ \mathbf{M}_N \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{q}_0 \\ \mathbf{q}_0 \\ \vdots \\ \mathbf{q}_0 \end{bmatrix} \tag{5}$$

where the matrices $\mathbf{M}_i$ and the vector β depend upon which quaternion is being optimized.

As a product of quaternions will be turn into a matrix to build $\mathbf{M}_i$, we've decided to make a quick note on how it is done and introduce the notation used in this paper. Let $\mathbf{q}_1 = [q_{11}, q_{12}, q_{13}, q_{14}]^\top$ and $\mathbf{q}_2 = [q_{21}, q_{22}, q_{23}, q_{24}]^\top$ be two quaternions where the first term is the scalar. Their product may be written in vector form as

$$\mathbf{q}_1 \otimes \mathbf{q}_2 = \underbrace{\begin{bmatrix} q_{11} & -q_{12} & -q_{13} & -q_{14} \\ q_{12} & q_{11} & -q_{14} & q_{13} \\ q_{13} & q_{14} & q_{11} & -q_{12} \\ q_{14} & -q_{13} & q_{12} & q_{11} \end{bmatrix}}_{[\mathbf{q}_1 \otimes]} \mathbf{q}_2 = \underbrace{\begin{bmatrix} q_{21} & -q_{22} & -q_{23} & -q_{24} \\ q_{22} & q_{21} & q_{24} & -q_{23} \\ q_{23} & -q_{24} & q_{21} & q_{22} \\ q_{24} & q_{23} & -q_{22} & q_{21} \end{bmatrix}}_{[\otimes \mathbf{q}_2]} \mathbf{q}_1$$

3.1 $\mathbf{q}_{\mathcal{B}}^{\mathcal{S}}$ estimation

Since $\mathbf{q}_{\mathcal{B}}^{\mathcal{S}}$ represents a rotation around any axis, we'll consider the whole vector as the parameters to be determined by setting $\beta = \mathbf{q}_{\mathcal{B}}^{\mathcal{S}}$. Comparing Eq. (2) to Eq. (3), it's clear that each $\mathbf{M}_i$ is written as

$$\mathbf{M}_i = \left[\otimes \left({}^{i}\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}} \otimes {}^{i}\mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'} \otimes \mathbf{q}_{\mathcal{I}}^{\mathcal{I}'} \otimes \left({}^{i}\mathbf{q}_{\mathcal{I}}^{\mathcal{S}}\right)^{-1}\right)\right]$$

Substituting these $\mathbf{M}_i$ into Eq. (5) and solving the problem with Eq. (4), we get our estimate for $\mathbf{q}_{\mathcal{B}}^{\mathcal{S}}$. However, this estimate may not have unitary norm, making it a valid quaternion but not a rotation. To solve this issue, we normalize it and ensure its scalar term is positive, as explained previously.

3.2 ${}^{i}\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}}$ estimation

As previously noted, the quaternions ${}^{i}\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}}$ only have one vector component as they rotate around one of the reference axis. Comparing again Eq. (2) to Eq. (3), we have

$$\mathbf{M}_1 = \left[\mathbf{q}_{\mathcal{B}}^{\mathcal{S}} \otimes\right] \left[\otimes \left({}^{i}\mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'} \otimes \mathbf{q}_{\mathcal{I}}^{\mathcal{I}'} \otimes \left({}^{i}\mathbf{q}_{\mathcal{I}}^{\mathcal{S}}\right)^{-1}\right)\right],$$

as its value depends on only one attitude estimate, and $\mathbf{M}_i = \mathbf{0}, i \in \{2, 3, \ldots, N\}$, which allows us to build $\mathbf{X}$ from Eq. (5).

Supposing that ${}^i\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}}$ is a rotation around the x axis, we can write it as ${}^i\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}} = [q_1, q_2, 0, 0]^\top$. As we can see, the matrix $\mathbf{X}$ isn't fully used during this estimation. By setting $\beta = [q_1, q_2]^\top$, we have to build a compatible matrix $\mathbf{X}'$ to be used during the LS solution. This matrix is given by extracting the relevant columns from $\mathbf{X}$, leaving

$$\mathbf{X}' = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 \end{bmatrix}$$

where $\mathbf{X}_i$ is the ith column from the original matrix $\mathbf{X}$ built from $\mathbf{M}_i$.

Using $\mathbf{X}'$ instead of $\mathbf{X}$ in (4), we have our estimates for q_1 and q_2. As we know whether q_2 should be positive, negative or null, we should have used this information to perform a constrained optimization. However, we found out experimentally that just changing its sign when it's wrong performed much better during the first few iterations when all quaternions aren't near their correct values yet. After we correct q_2 as needed, we rebuild ${}^i\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}}$, normalize it and make sure q_1 is positive, as previously explained.

The procedure in case the rotation was performed around the y axis is similar to the one presented, with different columns from $\mathbf{X}$ being extracted and the null elements from ${}^i\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}}$ being changed.

3.3 ${}^i\mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'}$ estimation

This estimation is similar to the one presented for ${}^i\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}}$, with the only difference being in the matrix $\mathbf{M}_1$ and the rotation occurring around the z axis. Comparing Eq. (2) to Eq. (3), we have

$$\mathbf{M}_1 = \left[\left(\mathbf{q}_{\mathcal{B}}^{\mathcal{S}} \otimes {}^i\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}}\right) \otimes\right] \left[\otimes \left(\mathbf{q}_{\mathcal{I}}^{\mathcal{I}'} \otimes \left({}^i\mathbf{q}_{\mathcal{I}}^{\mathcal{S}}\right)^{-1}\right)\right]$$

3.4 $\mathbf{q}_{\mathcal{I}}^{\mathcal{I}'}$ estimation

This quaternion's solution is a mix of two solutions. First, we have all matrices $\mathbf{M}_i$ like $\mathbf{q}_{\mathcal{B}}^{\mathcal{S}}$ as there's only one estimation for all attitudes. These matrices, according to Eq. (2) and Eq. (3), are given by

$$\mathbf{M}_i = \left[\left(\mathbf{q}_{\mathcal{B}}^{\mathcal{S}} \otimes {}^i\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}} \otimes {}^i\mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'}\right) \otimes\right] \left[\otimes \left({}^i\mathbf{q}_{\mathcal{I}}^{\mathcal{S}}\right)^{-1}\right]$$

However, the rotation $\mathbf{q}_{\mathcal{I}}^{\mathcal{I}'}$ represents a rotation around the z axis, making two elements of its vector form null, like ${}^i\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}}$. Once the matrix $\mathbf{X}$ is determined from $\mathbf{M}_i$, we build $\mathbf{X}'$ by extracting its first and fourth columns and solve the LS problem using Eq. (4). As we don't have any information about the sign of the angle rotated in this case, we just reconstruct the quaternion $\mathbf{q}_{\mathcal{I}}^{\mathcal{I}'}$ from its elements, normalize and correct the scalar sign if it's negative.

4. CALIBRATION ALGORITHM

Using the estimators developed in the last section, we can now describe a greedy algorithm to perform the calibration. Note that each step doesn't always minimizes the cost given by Eq. (2) as the signs of some quaternion elements must be kept consistent. The algorithm is:

1. Collect attitude estimates as described in Sec. 2.
2. Create initial estimates by setting all rotations to the identity.
3. Estimate $\mathbf{q}_{\mathcal{B}}^{\mathcal{S}}$ as described in Sec. 3.1.
4. Estimate ${}^i\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}}$ as described in Sec. 3.2.
5. Estimate ${}^i\mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'}$ as described in Sec. 3.3.
6. Estimate $\mathbf{q}_{\mathcal{I}}^{\mathcal{I}'}$ as described in Sec. 3.4.
7. If the algorithm didn't converge and a maximum number of steps haven't been performed, go to step 3. Otherwise, end calibration.

5. EXPERIMENTS

5.1 Experimental setup

The experiment was run in a AMD Phenom II x4 955 with 6GB of RAM memory available. The algorithm was implemented using SciPy (Jones *et al.*, 2001–) and NumPy. As the computing time was very small, we didn't measure it.

Table 1: Experiment parameters.

Parameter	Description	Value
M	Number of attitudes measured by rotating around each axis	
$\mathbf{q}_{\mathcal{B}}^{\mathcal{S}}$	Rotation between sensor and robot frame	See (Kuffner, 2004)
$\theta_{\mathcal{I}}$	Rotation angle for $\mathbf{q}_{\mathcal{I}}^{\mathcal{I}'}$	$\mathcal{U}\left([0, 2\pi)\right)$
$\theta_{\mathcal{I}'}^{i}$	Rotation angle for ${}^{i}\mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'}$	$\mathcal{U}\left((-\frac{\pi}{2}, \frac{\pi}{2})\right)$
$\theta_{\mathcal{B}'}^{i}$	Rotation angle for ${}^{i}\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}}$	$\mathcal{U}\left((-\frac{\pi}{2}, \frac{\pi}{2})\right)$
γ	Estimates' noise level	
	Total of Monte Carlo runs	100

5.2 Synthetic data

For each run, we create a random quaternion $\mathbf{q}_{\mathcal{B}}^{\mathcal{S}}$ as described in (Kuffner, 2004). For $\mathbf{q}_{\mathcal{I}}^{\mathcal{I}'}$, we choose an angle $\theta_{\mathcal{I}}$ from the interval $[0, 2\pi)$ and create the quaternion through the equation

$$\mathbf{q}_{\mathcal{I}}^{\mathcal{I}'} = \cos\left(\frac{\theta_{\mathcal{I}}}{2}\right) + \mathbf{v}\sin\left(\frac{\theta_{\mathcal{I}}}{2}\right)$$

where $\mathbf{v}$ is the unitary vector for the rotation axis, which in this case is $\vec{\mathbf{z}}$. The quaternions ${}^{i}\mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'}$ are created by a random rotation in the interval $(-\frac{pi}{2}, \frac{pi}{2})$ around the same axis $\vec{\mathbf{z}}$, but we make it so that the number of positive and negative rotations is the same, which experimentally provided better results.

We consider that M attitude estimations were made for each axis, that is, M executions of step 2.1 and $2M$ executions of step 2.2, M for each of x and y, as described in Sec. 2. The quaternions ${}^{i}\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}}$ are given by random rotations in the interval $(-\frac{pi}{2}, \frac{pi}{2})$ around their designed axis also keeping the number of positive and negative rotations the same.

To simulate error by the attitude estimation algorithm, each attitude estimate suffers from a random uniform noise, which is created from a random quaternion (Kuffner, 2004) whose angle is multiplied by γ to set the noise's range. We performed a Monte Carlo simulation with a total of 100 runs and new quaternions for each run. Table 1 presents a summary of the parameters used in the experiment.

5.3 Experimental results

After the calibration, we can compare the rotation given by $\mathbf{q}_{\mathcal{B}}^{\mathcal{S}}$ and its estimated value $\hat{\mathbf{q}}_{\mathcal{B}}^{\mathcal{S}}$ through the rotation between them, which is given by

$$\theta = 2\arccos\left(q_1^{\epsilon}\right) \tag{6}$$

where q_1^{ϵ} is the positive scalar term of $\mathbf{q}^{\epsilon}$, which in turn is defined by

$$\mathbf{q}^{\epsilon} = \hat{\mathbf{q}}_{\mathcal{B}}^{\mathcal{S}} \otimes \left(\mathbf{q}_{\mathcal{B}}^{\mathcal{S}}\right)^{-1}$$

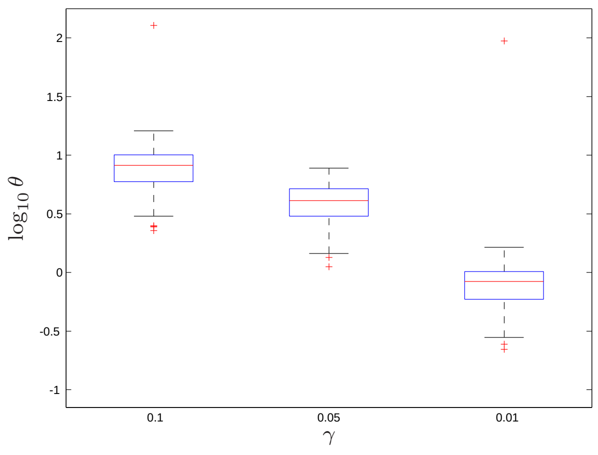
The first thing noteworthy in Fig. 1 is that the algorithm doesn't always converge to a value close to the true one. Moreover, an increase in the number of samples caused the algorithm to get stuck in a low quality local minima for $\gamma = 0.05$. This may be worrisome at first glance as the obtained estimate may become unreliable, but the algorithm's performance isn't always bad. We noticed that different subsets of the collected data provide different results, with the good ones being more common and closer to each other. Our suggestion is to perform the calibration on these subsets and discard outliers, which probably are caused by poor minima.

Once pass that observation, we note that the error level for $M = 2$ is generally similar to the noise level γ. For instance, for $\gamma = 0.1$ we have a noise level of $[-18, 18]$ degrees. As the error measure, given by, Eq. (6) considers only positive angles due to the rotation symmetry, we have a median on 9, which is approximately the value obtained in this case according to Fig. 1a. The same pattern is observed for other values of γ, so that with $M = 2$ we have a calibration as good as the algorithm used to estimate the attitudes.

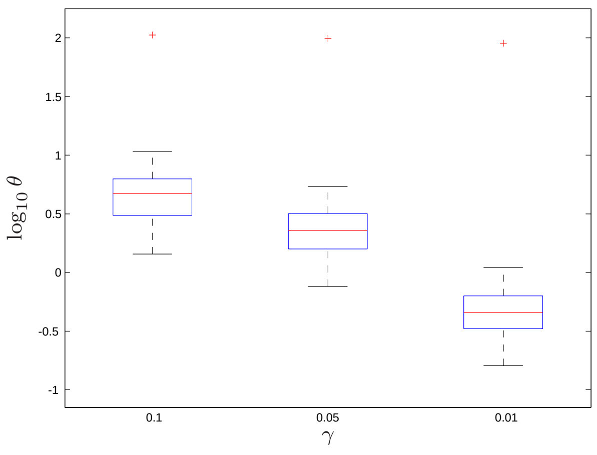
As can also be observed in Fig. 1 and disregarding the outliers, estimate errors decrease linearly with the noise level, which is clear when we compare $\gamma = 0.1$ and $\gamma = 0.01$ for all values of M. The error also generally decreases with increasing number of samples as one would expect.

6. CONCLUSION

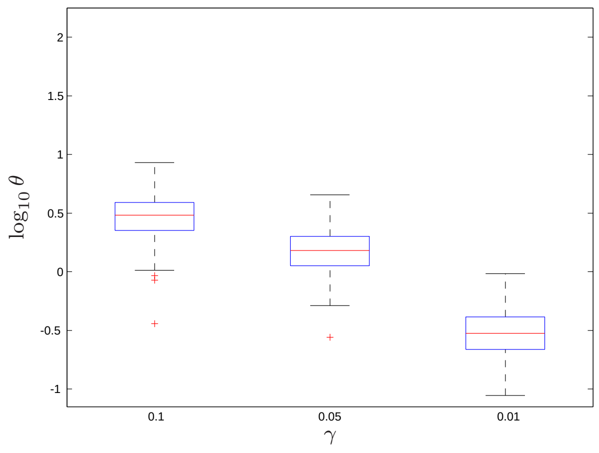
In this paper, we presented a calibration process to estimate the rotation between the sensor and robot frame using the least squares problem. To the best of the authors' knowledge, this is the only algorithm that tries to perform this



(a) $M = 2$



(b) $M = 5$



(c) $M = 10$

Figure 1: Estimate error varying noise level and number of attitudes measured for each axis.

estimation without any external auxiliary devices. The proposed algorithm is also compatible with any other attitude estimation algorithm, allowing it to be used in almost any setting where such estimations are already performed.

Using artificial data, we showed that usually the algorithm is able to achieve good quality local minima for the optimization process, with eventual spurious estimates. Furthermore, when we collect only 2 samples for each axis, which we consider the minimum to have a reliable estimation, the rotation estimate error is as accurate as the attitude estimation error for the data provided.

The quaternion fitting problem is posed as a least squares problem, which isn't the most adequate solution as it doesn't take into account the known sign of some terms and the norm's restriction. We are currently pursuing other ways to formulate the optimization problem so that its solution has the desired properties. We believe that this will eliminate the spurious minima present in the solutions found, leading to a more reliable calibration.

As discussed in the end of Sec. 2, there's a trade-off regarding the knowledge about angles' signs provided to the algorithm. Without it, the problem has at least one incorrect solution, because the angles' negated values fit a frame of reference different from the correct one. However, this knowledge also allows the data to become inconsistent due to noise in the attitude estimates. We think that an algorithm that is able to cope with these inconsistencies by allowing a few attitude estimations, defined by the user, to become inconsistent or defining a certain error threshold, below which the angles that are inconsistent aren't corrected, may improve the estimate between the sensor and body frame.

7. ACKNOWLEDGEMENTS

The authors would like to thank FAPESP for its support through the process 2012/01511-6.

8. REFERENCES

Alonso, R. and Shuster, M.D., 2002. "Complete linear attitude-independent magnetometer calibration". *Journal of the Astronautical Sciences*, Vol. 50, No. 4, pp. 477–490.

Chou, J.C.K., 1992. "Quaternion kinematic and dynamic differential equations". *Robotics and Automation, IEEE Trans-*

actions on, Vol. 8, No. 1, pp. 53–64. ISSN 1042296X. doi:10.1109/70.127239.

Elkaim, G.H., 2013. "Misalignment Calibration Using Body Frame Measurements". In *American Control Conference, ACC13*.

Elkaim, G.H., Lie, F.A.P. and Gebre-Egziabher, D., 2012. "Principles of Guidance, Navigation and Control of UAVs". Technical report.

Gebre-Egziabher, D., Elkaim, G.H., Powell, J.D. and Parkinson, B.W., 2000. "A gyro-free quaternion-based attitude determination system suitable for implementation using low cost sensors". In *Position Location and Navigation Symposium, IEEE 2000*. Ieee, pp. 185–192. ISBN 0-7803-5872-4. doi:10.1109/PLANS.2000.838301.

Jones, E., Oliphant, T., Peterson, P. *et al.*, 2001–. "SciPy: Open source scientific tools for Python". URL `"http://www.scipy.org/"`.

Kariya, T. and Kurata, H., 2004. *Generalized least squares*. Wiley.

Kelly, J. and Sukhatme, G.S., 2009. "Fast relative pose calibration for visual and inertial sensors". In *Experimental Robotics*. pp. 515–524.

Kuffner, J.J., 2004. "Effective sampling and distance metrics for 3D rigid body path planning". In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*. April, pp. 3993–3998, vol. 4. ISBN 0780382323.

Madgwick, S.O.H., Harrison, A.J.L. and Vaidyanathan, R., 2011. "Estimation of IMU and MARG orientation using a gradient descent algorithm". In *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*. Vol. 2011, pp. 1–7. ISBN 9781424498628. ISSN 1945-7901. doi:10.1109/ICORR.2011.5975346.

Sabatini, A.M., 2006. "Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing". *Biomedical Engineering, IEEE Transactions on*, Vol. 53, No. 7, pp. 1346–1356. ISSN 0018-9294. doi: 10.1109/TBME.2006.875664.

Schopp, P., Klingbeil, L., Peters, C. and Manoli, Y., 2010. "Design, geometry evaluation, and calibration of a gyroscope-free inertial measurement unit". *Sensors and Actuators A: Physical*, Vol. 162, No. 2, pp. 379–387. ISSN 09244247. doi:10.1016/j.sna.2010.01.019.

Shuster, M., 2006. "The generalized Wahba problem". *Journal of the Astronautical Sciences*, Vol. 54, No. 2, pp. 245–259.

Thrun, S., Burgard, W. and Fox, D., 2005. *Probabilistic Robotics*. Intelligent Robotics and Autonomous Agents Series. Mit Press. ISBN 9780262201629.

Vasconcelos, J.F., Elkaim, G.H., Silvestre, C., Oliveira, P. and Cardeira, B., 2008. "A geometric approach to strapdown magnetometer calibration in sensor frame". In *Navigation, Guidance and Control of Underwater Vehicles*. pp. 172–177, vol. 2.

9. RESPONSIBILITY NOTICE