

## SELF-CALIBRATION OF MEMS SENSORS ARRANGED IN TRIADS USING KALMAN FILTER ESTIMATOR

Francisco Granziera Júnior, MsC., [francisco.granzierajr@gmail.com](mailto:francisco.granzierajr@gmail.com)

Rolf H. Vargas V., Dr., [rolf.vargas@gmail.com](mailto:rolf.vargas@gmail.com)

National Institute for Space Research – INPE, São José dos Campos, SP, Brazil

Marcelo Carvalho Tosin, Dr., [mctosin@uel.br](mailto:mctosin@uel.br)

Osmar Tormena Júnior, MsC., [juniortormena@gmail.com](mailto:juniortormena@gmail.com)

State University of Londrina – UEL, Londrina, PR, Brazil

**Abstract.** MEMS sensors have gained more space in the industry over the last decade. How are compact, lightweight and energy efficient they are being applied to vehicles as cars, planes and satellites, and also in robotic manipulators, joysticks, laptops, cameras and other gadgets. MEMS sensors such as accelerometers, gyros, and magnetometers (MARG sensors) are widely used in navigation applications requiring linear motion estimation and / or angular. Although constantly evolving, these sensors require the maintenance of calibration, especially in regard to the biases and scale factors which are very sensitive to temperature variation but also the on-off effect. Generally, sensors used in navigation are arranged in triads, presents also alignment errors between the axes to be measured after assembly. The calibration procedure for swing often requires equipment such as positioning tables and robotic arms, which are expensive and can only be made in pre-use, or in the laboratory. This work is progressing on the issue of self-calibration of these types of sensors, showing the equation, simulation and testing of a Kalman Estimator for determining the parameters of the sensors that make up a triad. The algorithm runs in two steps. At first, positioning data of the triad are collected and then applied the Kalman recursive algorithm for determining the constants of the quadratic form designing the readings of the triad in an irregular ellipsoid. Then the algebraic equations that relate the constants of the ellipsoid with the parameters of the sensors are resolved (biases, scale factor and misalignment). Also discussed is a heuristic way to collect data and identify the right time to run the calibration algorithm based on a merit factor observed. The advantage of this algorithm is low computational cost, which allows its implementation and the possibility of onboard equipment to self-recalibrate. Also are presented and discussed practical results of this method applied to triads of accelerometers and magnetometers.

**Keywords:** MEMS Sensors, Self-calibration

### 1. INTRODUCTION

MEMS sensors are becoming increasingly required in embedded applications. These sensors are usually very small, lightweight and consume little power. MEMS sensors such as accelerometers, magnetometers and gyros, initially used in aeronautical, automotive and robotics, are now present in household applications like in smartphones, joysticks video games, autonomous toys among others.

These types of sensors are used in tasks ranging from simply motion detection, inclination angle identifying to even orientation and attitude of a body estimation. The more complex is the task, more sensors are used, and generally the same type of sensors are arranged in three orthogonal axes, denominated triads. A triad of sensors allows the reading of a complete vector, therefore, it is possible to measure the intensity and direction of a vector in the coordinate system fixed to the triad. Note that the contrary is not true; in other words, with the reading of the intensity and direction of only one vector it is not possible to fully know the orientation of a triad.

Sensors such as magnetometers can be used in reading of the geomagnetic field in order to identifying the direction of magnetic north. A Y axis sensor would suffice to identify the angle to the magnetic north since this sensor was aligned with the local plan and knowing the value of the magnetic field in the plane for comparison. The ratio of the field read by the sensor and the local field is the cosine of the angle that the sensor is to the north. If used two orthogonal sensors is no longer necessary to know the value of the local field or be parallel to the local level, because the value of the projection of the magnetic field in the plane can be calculated as a result with the values read by sensors X and Y. The resulting ratio between Y and X and Y results in the cosine of the angle to the north, but there will be a singularity when the XY plane is orthogonal to the vector, as there will zero readings for both sensors. Another possibility, more complete, is to use three orthogonal sensors. With the readings of X, Y and Z it is possible to determine the direction and intensity of the vector in question without singularities.

The same happens with accelerometers and gyros. The vector should read whenever possible made using triads or some other arrangement with three or more sensors.

Whatever the measured quantity, the measurement accuracy depends on all elements involved in the process so far as it has become digital. All kinds of electrical measurements (or not) are subject to several types of errors. In addition to the common errors to all sensors that are scale factors and offsets, we also have in this case the misalignment

between axes. In the case of sensors with analog amplifiers, filters and converters also include errors in measurement, therefore not enough simply to calibrate the sensor separately and later connect it to the hardware. This fact implies that using only the parameters of the factory calibration is not enough to ensure that the sensor will operate as expected.

For triads, it is possible to use the properties of this configuration to develop and implement algorithms that allow a self-recalibration of the system constantly.

An ideal triad consists of sensors that do not have offsets, scale factors are unitary and the arrangement is perfectly orthogonal. If an ideal triad is immersed in a constant vector field it will perform readings of projections in XYZ, regardless of their attitude that will generate a result set. This means that any readings in attitudes would be always plotted on the surface of a perfect sphere, of radius equal to the vector module read and centered on the origin of the axes.

A triad that has sensors with nonzero offsets, with scale factors different from each other and with misalignment between axes when immersed in a constant vector field reads projections that would be plotted on the surface of an ellipsoid distorted and shifted the axis origin.

Based on the reading of data from an uncalibrated triad the goal of calibration is to identify precisely which quadratic equation represents the ellipsoid read by the triad, and also at run time, sensor readings are selected by heuristic filters and then processed in a recursive Kalman estimator to update the parameters and thus maintain the triad always calibrated.

In section 2 of the paper presents the sensor measurements model and a method of calibration in two steps, including the estimation by batch and recursive Kalman estimator. In Section 3 the simulation method is briefly described. In Section 4 are used actual data from triads of sensors as accelerometers and magnetometers to test the method. The calibration quality evaluation is done by comparing data in attitude before and after calibration of triads. In section 6 are described possible self-calibration algorithms for implementation on a host PC or for implementing embedded in the application itself, and finally, general conclusions are made.

## 2. MEASUREMENT MODELS

### 2.1. Parameter Definition and Modeling

For a tri-axis sensor, integrated or assembled, the sensor model can be written as follow:

$$\begin{aligned}\hat{u}_x &= au_x + x_0 + v_x & \hat{u}_y &= b(u_y \cos(\rho) + u_x \sin(\rho)) + y_0 + v_y \\ \hat{u}_z &= c(u_z \cos(\phi) \cos(\lambda) + u_y \sin(\lambda) \cos(\phi) + u_x \sin(\phi) \cos(\lambda)) + z_0 + v_z\end{aligned}\quad (1)$$

where the subscripts x, y and z represent the sensing axes,  $\hat{u}_x$ ,  $\hat{u}_y$  and  $\hat{u}_z$  are the sensor outputs and  $u_x$ ,  $u_y$  and  $u_z$  are the components of the physical quantity. The values  $a$ ,  $b$  and  $c$  are the sensibility of the x, y and z axes, respectively,  $x_0$ ,  $y_0$  and  $z_0$  represent the offset of each axis. The angle  $\rho$  represents the y'-axis misalignment, inside the xy plane, the angles  $\phi$  and  $\lambda$  represent deviations of the z'-axis from the yz and xz planes, respectively. In Figure 1, x, y and z represent the proper orthogonal triad axes while x', y' and z' are the actual misaligned triad.

Additionally, there is a noise added in each measure. These noises are considered Gaussian with zero mean, constant variance and are uncorrelated.

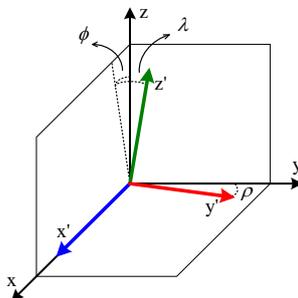


Figure 1. Graphical illustration of the orthogonality deviation angles.

If a Triad is immersed at a constant vector field, the absolute value of the physical quantity is used as a boundary condition that must be satisfied:

$$|\mathbf{u}|^2 = u_x^2 + u_y^2 + u_z^2 \quad (2)$$

Rewriting Eq. (1), solving for the physical quantity actual value:

$$u_x = \frac{(\hat{u}_x - x_0)}{a} \quad u_y = \frac{a(\hat{u}_y - y_0) - b \sin(\rho)(\hat{u}_x - x_0)}{ab \cos(\rho)} \quad (3)$$

$$u_z = \frac{(ab \cos(\rho)(\hat{u}_z - z_0) - ac \cos(\phi) \sin(\lambda)(\hat{u}_y - y_0) + bc(\sin(\rho) \cos(\phi) \sin(\lambda) - \cos(\rho) \sin(\phi) \cos(\lambda))(\hat{u}_x - x_0))}{ab \cos(\rho) \cos(\phi) \cos(\lambda)}$$

Substituting  $u_x$ ,  $u_y$  and  $u_z$  from Eq. (3) into Eq. (2), one gets an expression in the quadratic form:

$$A\hat{u}_x^2 + B\hat{u}_x\hat{u}_y + C\hat{u}_x\hat{u}_z + D\hat{u}_y^2 + E\hat{u}_y\hat{u}_z + F\hat{u}_z^2 + G\hat{u}_x + H\hat{u}_y + I\hat{u}_z + J = 0 \quad (4)$$

where the intermediate variables  $A$ ,  $B$ ,  $C$ ,  $D$ ,  $E$ ,  $F$ ,  $G$ ,  $H$ ,  $I$ , and  $J$  are non-linear functions of the actual sensor parameters. This way, Eq. (4) is non-linear in terms of the sensor parameters, but it is linear in terms of the intermediate variables. Equation (4) can be written in a more convenient form as follows:

$$\frac{A\hat{u}_x^2}{F\hat{u}_z^2} + \frac{B\hat{u}_x\hat{u}_y}{F\hat{u}_z^2} + \frac{C\hat{u}_x\hat{u}_z}{F\hat{u}_z^2} + \frac{D\hat{u}_y^2}{F\hat{u}_z^2} + \frac{E\hat{u}_y\hat{u}_z}{F\hat{u}_z^2} + \frac{G\hat{u}_x}{F\hat{u}_z^2} + \frac{H\hat{u}_y}{F\hat{u}_z^2} + \frac{I\hat{u}_z}{F\hat{u}_z^2} + \frac{J}{F\hat{u}_z^2} = -1 \quad (5)$$

In matrix notation:

$$\underbrace{\begin{bmatrix} \hat{u}_{x_1}^2 / \hat{u}_{z_1}^2 & \hat{u}_{x_1}\hat{u}_{y_1} / \hat{u}_{z_1}^2 & \dots & 1 / \hat{u}_{z_1}^2 \\ \vdots & \ddots & & \vdots \\ \hat{u}_{x_n}^2 / \hat{u}_{z_n}^2 & \hat{u}_{x_n}\hat{u}_{y_n} / \hat{u}_{z_n}^2 & \dots & 1 / \hat{u}_{z_n}^2 \end{bmatrix}}_{\mathbf{X}} \underbrace{\begin{bmatrix} A/F \\ B/F \\ \vdots \\ J/F \end{bmatrix}}_{\mathbf{k}} = \underbrace{\begin{bmatrix} -1 \\ \vdots \\ -1 \end{bmatrix}}_{\mathbf{w}} \quad \text{or} \quad \mathbf{X}_{n \times 9} \mathbf{k}_{9 \times 1} = \mathbf{w}_{n \times 1} \quad (6)$$

where  $n$  is the number of measurements used for the estimation,  $\mathbf{X}$  is a measurement or observation matrix,  $\mathbf{k}$  is the intermediate variables vector and  $\mathbf{w}$  is a negative ones vector. The choice of putting  $F\hat{u}_z^2$  as a common factor in Eq. (5) is due to the fact that  $F$  is the simplest of the non-linear functions from  $A$  to  $J$ . This way, the algebraic work solving the sensor parameters from  $A/F$ ,  $B/F$ , ...,  $J/F$  was minimized.

After solving Eq. (6), the vector  $\hat{\mathbf{k}}$  holds the intermediate variables vector estimated. Parting from this variable is possible to find the nine parameters  $\hat{a}$ ,  $\hat{b}$ ,  $\hat{c}$ ,  $\hat{x}_0$ ,  $\hat{y}_0$ ,  $\hat{z}_0$ ,  $\rho$ ,  $\phi$  e  $\lambda$ .

The orthogonal deviation angles are expected to be very small (less than five degrees for assembled triads and less than one degree in integrated tri-axis sensors), so the small angles approximation ( $\sin \theta \approx \theta$  and  $\cos \theta \approx 1$ ) can be used to minimize the algebraic work solving the sensor parameters in terms of the intermediate variables, without appreciable loss of precision. Applying some algebra, it is possible to reach the following solution for the sensor parameters (the "hat" (^) means estimated value):

$$\hat{x}_0 = \frac{4\chi\delta + \chi\varepsilon^2 - 2\eta\beta - \eta\gamma\varepsilon - \iota\beta\varepsilon + 2\gamma\delta}{2\gamma^2\delta - 8\alpha\delta - 2\alpha\varepsilon^2 + 2\beta^2 + 2\beta\gamma\varepsilon} \quad \hat{y}_0 = \frac{\gamma^2\eta - 4\alpha\eta + 2\beta\chi + \beta\gamma\iota - 2\alpha\varepsilon\iota + \gamma\chi\varepsilon}{-2\gamma^2\delta + 8\alpha\delta + 2\alpha\varepsilon^2 - 2\beta^2 - 2\beta\gamma\varepsilon}$$

$$\hat{z}_0 = \frac{\beta^2\iota - \beta\gamma\eta - 4\alpha\delta\iota + 2\alpha\varepsilon\eta - \beta\varepsilon\chi + 2\gamma\delta\chi}{2\gamma^2\delta - 8\alpha\delta - 2\alpha\varepsilon^2 + 2\beta^2 + 2\beta\gamma\varepsilon} \quad \hat{c} = \frac{1}{|\mathbf{u}|} \sqrt{\kappa + \alpha\hat{x}_0^2 - \beta\hat{x}_0\hat{y}_0 - \gamma\hat{x}_0\hat{z}_0 - \delta\hat{y}_0^2 - \varepsilon\hat{y}_0\hat{z}_0 + \hat{z}_0^2} \quad (7)$$

$$\hat{b} = \frac{\hat{c}}{\sqrt{-\delta}} \quad \hat{a} = \sqrt{\frac{-2\hat{c}^4(1-\hat{\lambda}^2)^2}{2\alpha\hat{c}^2(1-\hat{\lambda}^2)^2 - (\beta\hat{b}\hat{\lambda} + \gamma\hat{c}\hat{\lambda}^2)(\beta\hat{b}\hat{\lambda} + \gamma\hat{c})}} \quad \hat{\rho} = \frac{\hat{a}(\beta\hat{b} + \gamma\hat{c}\hat{\lambda})}{2\hat{c}^2(1-\hat{\lambda}^2)} \quad \hat{\phi} = \frac{\hat{a}(\beta\hat{b}\hat{\lambda} + \gamma\hat{c})}{2\hat{c}^2(1-\hat{\lambda}^2)} \quad \hat{\lambda} = \frac{\varepsilon}{2\sqrt{-\delta}}$$

Assuming:

$$\alpha = -A/F \quad \beta = -B/F \quad \gamma = -C/F \quad \delta = -D/F \quad \varepsilon = -E/F \quad \chi = -G/F \quad \eta = -H/F \quad \iota = -I/F \quad \kappa = -J/F \quad (8)$$

Although finding this solution involved a long and complex algebraic work, once it was done, one can easily apply it very easily, reaching the desired sensor parameters with much less computational load than a direct approach of the non-linear estimation.

If there would not be noises in the sensors measures, the solution of the Eq. (6) would be exact with only nine distinct measures. However, as it is not viable to solve deterministically this equation, we can use several forms. Two of these are shown below.

## 2.2. Estimation using Least Square

Solving Eq. (6) for  $\mathbf{k}$ , using the Moore-Penrose Pseudo Inverse (also called Generalized Inverse), one reaches Eq. (9), which is mathematically equivalent to a Least Squares Estimator.

$$\hat{\mathbf{k}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{w} \quad (9)$$

Where  $\hat{\mathbf{k}}$  indicates the estimated intermediate variables vector. The Eq. (9) is a critical point in the estimation problem, this is mostly due to two factors: first, although it is possible to solve the Moore-Penrose Pseudo Inverse the way it is stated in Eq. (9), this approach lacks numerical robustness and is error prone. The best way to compute the pseudo inverse is applying Eq. (9) not directly over  $\mathbf{X}$ , but on its QR decomposition (where  $\mathbf{X}$  is factored in a orthogonal matrix  $\mathbf{Q}$  and a upper triangular matrix  $\mathbf{R}$ ) or SVD (Singular Value Decomposition), where  $\mathbf{X}$  is factored in a positive diagonal matrix and two unitary matrices. Second, the product  $\mathbf{X}^T \mathbf{X}$  has the characteristic of amplifying the noise present in  $\mathbf{X}$  so, in order to form a well-conditioned system, the collinearity of  $\mathbf{X}$  must be the smallest possible (i.e. the samples used in the calibration procedure must be well spread over the ellipsoid). A more detailed analysis of the generalized inverse can be found in Rao and Mitra (1970).

Although there are numerical instabilities, the biggest problem of the form of Eq. (9) is the computational cost to implement these products in a matrix embedded application that has more modest computational resources than a PC.

## 2.3. Estimation using Kalman Filter form

An equivalent implementation intensively used to solve equations with the form of Eq. (6) is the recursive Kalman. The advantage of this algorithm, applied to parameters estimation, lies in the fact avoid reversals matrix and also because its recursion characteristic makes it very suitable for programming embedded systems, since the arrays are smaller and therefore less memory is utilized in implementing the algorithm. Another advantage is that processing can be done after each sample without the need to collect all samples to run the estimator.

The Kalman estimator form is shown in the Eq. (10):

$$\begin{aligned} \mathbf{K}_i &= \mathbf{P}_{i-1} \mathbf{H}_i^T (\mathbf{H}_i \mathbf{P}_{i-1} \mathbf{H}_i^T + 1)^{-1} \\ \mathbf{P}_i &= (\mathbf{I} - \mathbf{K}_i \mathbf{H}_i) \mathbf{P}_{i-1} \\ \hat{\mathbf{x}}_i &= \hat{\mathbf{x}}_{i-1} + \mathbf{K}_i (\mathbf{y}_i - \mathbf{H}_i \hat{\mathbf{x}}_{i-1}) \end{aligned} \quad (10)$$

Where the  $\mathbf{H}$  vectors are lines of  $\mathbf{X}$  matrix as can be seen in the Eq. (11); the  $\mathbf{y}$  is the measures vector that is ever constant and equal to 1. The  $\mathbf{K}$  matrix is the Kalman gain. The  $\mathbf{P}$  matrix is the estimated covariance and the  $\mathbf{x}$  vector is the estimated state.

$$\underbrace{\begin{bmatrix} \hat{u}_{x_1}^2 / \hat{u}_{z_1}^2 & \hat{u}_{x_1} \hat{u}_{y_1} / \hat{u}_{z_1}^2 & \dots & 1 / \hat{u}_{z_1}^2 \\ \vdots & \ddots & & \vdots \\ \hat{u}_{x_n}^2 / \hat{u}_{z_n}^2 & \hat{u}_{x_n} \hat{u}_{y_n} / \hat{u}_{z_n}^2 & \dots & 1 / \hat{u}_{z_n}^2 \end{bmatrix}}_{\mathbf{X}} = \begin{bmatrix} \mathbf{H}_1 \\ \vdots \\ \mathbf{H}_n \end{bmatrix} \quad (11)$$

It is necessary that the initial values to process the algorithm recursively. An initial estimate for  $\hat{\mathbf{P}}_0$  and  $\hat{\mathbf{x}}_0$  must be made as shown in Eq. (12):

$$\hat{\mathbf{P}}_0 = \left( \begin{bmatrix} \mathbf{H}_1^T & \dots & \mathbf{H}_9^T \end{bmatrix} \begin{bmatrix} \mathbf{H}_1 \\ \vdots \\ \mathbf{H}_9 \end{bmatrix} \right)^{-1} \quad \hat{\mathbf{x}}_0 = \hat{\mathbf{P}}_0 \begin{bmatrix} \mathbf{H}_1^T & \dots & \mathbf{H}_9^T \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \quad (12)$$

This method requires only nine initial values to start. Note that  $\hat{\mathbf{P}}_0$  inverse of Eq. (12) is very simple because the vector product results in a scalar value.

Although this method processes one sample by time, the result obtained after to process all the samples must be exactly the same that one using Least Square batch.

### 3. SIMULATIONS

Some simulations were performed to verify the efficiency of algorithms for estimating the parameters of the sensors. The simulation can be divided into four steps: data generation, heuristic filtering, parameter estimation and performance evaluation.

Following each subsection provides a description of each of these steps.

#### 3.1. Data generation

The algorithms described here are applied to a collection of data, where the selection of these data is also one of the tasks of the algorithm. Raffle pointing directions would be the best case for the estimator, but it is known that in practice this assumption is quite unrealistic if we consider a short time of use. For example, a triad of magnetometers installed in a robot, car, plane or other object, hardly noticed the inclination and declination data very distant from each other in a few minutes. A more realistic proposal is raffling inclination and declination with means and variances that do represent a motion close to a real application. Figure (2a) illustrates how the generation of data with this profile. Each point on the unit sphere represents an inclination and declination randomly selected. The average position, the variances and the number of points are the arguments of the function that draws angles. After generating the points, they are decomposed in  $u_x$ ,  $u_y$ , and  $u_z$  and then Eq. (1) is applied to generate the distorted ellipsoid and outside the home.

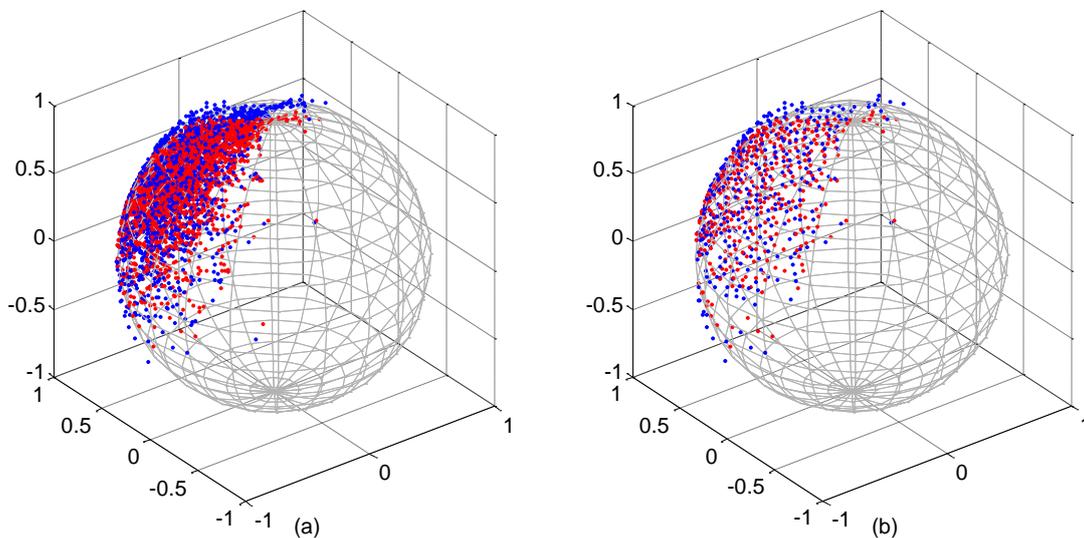


Figure 2. (a) Example of generated points; (b) Generated points after heuristic filtering.

#### 3.2. Heuristic filtering

After the data has been generated and Eq. (1) has been applied, we must make a check to ensure that there will be no numerical problems during estimation. Two filtrations are made: (a) The vectors whose term  $\hat{u}_z$  is less than a certain  $\mu$  value are discarded; and (b) any vector which makes an angular distance of less than between any other will also be discarded. The first justification for filtering is to prevent the division of the terms  $\hat{u}_x$  and  $\hat{u}_y$  by  $\hat{u}_z$ , in the formation of a row of the matrix  $\mathbf{X}$ , resulting in very high or non-numeric value. The second goal is to avoid the filtering that are collected almost-coincident points that generate singularities in the matrix  $\mathbf{X}$ , when applying the method of least squares. Figure (2b) illustrates the points selected by filtering.

After the selection, application of the Eq. (1) and heuristic filtering, the *generated data* become the *observed data* and these are the data to be used as input to the estimator.

### 3.3. Parameters estimation

The estimation is the reverse process of generation. Using only the observed data the goal is to find the parameters that were applied to the data generated. The observed data make it possible to construct the matrix  $\mathbf{X}$  as in Eq. (6) and consequently, we can apply the forms of Eq. (9) or (10) to estimate the constants of the quadratic equation. From these constants estimated by applying Eq. (7), we finally find the parameters of the triad. The implementation of these parameters in Eq. (3) results in measures with errors of bias, scale factor and misalignment minimized.

### 3.4. Performance evaluation

The estimated parameters are not exactly the same as those applied to generate the observed data due to Gaussian noise added. The error in estimating each parameter is not a good reference to evaluate the quality of the estimation performed. To evaluate the calibration of simulation can evaluate the error of the angles and the plane perpendicular to the reference vector before and after parameter estimation of the triad (Frosio, 2009).

These angles are calculated as shown in Eq. (13). Note that the angles must be calculated first with the values of  $\hat{u}_x$ ,  $\hat{u}_y$  and  $\hat{u}_z$  with no corrections and later with  $u_x$ ,  $u_y$  e  $u_z$ , after making corrections.

$$\begin{aligned} \theta &= \arctan\left(u_y / \sqrt{u_x^2 + u_z^2}\right) & \tilde{\theta} &= \arctan\left(\hat{u}_y / \sqrt{\hat{u}_x^2 + \hat{u}_z^2}\right) & \Delta\theta &= \tilde{\theta} - \theta \\ \vartheta &= \arctan\left(u_x / \sqrt{u_y^2 + u_z^2}\right) & \tilde{\vartheta} &= \arctan\left(\hat{u}_x / \sqrt{\hat{u}_y^2 + \hat{u}_z^2}\right) & \Delta\vartheta &= \tilde{\vartheta} - \vartheta \end{aligned} \quad (13)$$

A Fig. (3) shows these angular errors plotted before the calibration and the after the calibration.

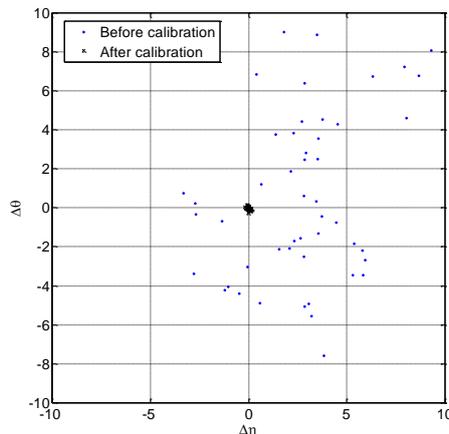


Figure 3 – Angular errors before and after calibration

## 4. TESTS USING REAL DATA

### 4.1. Description of the hardware used

To test the algorithm with real data were used MEMS sensors, specifically a triad of magnetometers, the HMC2003 Honeywell®, and an accelerometer triad, formed by three ADXL202 from Analog Devices®. The Tab. (1) summarizes the parameters and errors of the sensors used.

Table 1. Typical parameters of MEMS accelerometer and magnetometer used in this test

| Parameters                  | ADXL202JE     | HMC2003          |
|-----------------------------|---------------|------------------|
| Scale Factor range          | 312 ± 50 mV/g | 1 ± 0.02 V/Gauss |
| Offset range                | 2,5 ± 0,5 V   | 2.5 ± 0.2 V      |
| Temperature Drift           | 2 mg/°C       | 100 ppm/°C       |
| Misalignment among the axes | < 5°          | < 5°             |

All sensors have analog output. The analog signals are amplified, filtered, and finally digitized by an ADS8364 with a 16 bit resolution and sample simultaneously converts the signals from all sensors. A Freescale HCS12 microcontroller ® reads the data of the ADC, puts in ASCII format and sends the data via the serial interface (SCI). On a PC, the data can then be read and written in a text file for analysis. The sensors and other circuits are mounted on a single PCB, inserted in a cubic non-magnetic support. Fig (4) presents a partial picture of the hardware that is described in more detail in (Granziera, 2006).



Figure 4. Hardware with triads of MEMS sensors used in calibrations tests

#### 4.2. Data Collecting and Handling

The study of this paper is precisely to avoid the need for special conditions such as rotation tables with programmed movements to collect data which can be applied the method of calibration (estimation of parameters of the sensors). Therefore, to produce test data, the sensors are randomly moved slowly in any environment with few restrictions. The sudden movements must be avoided for accelerometers read only the acceleration of gravity. For the magnetometers, the only care is avoiding the proximity of the sensor cables and ferromagnetic materials.

The hardware that contains the triads of sensors acquires and sends data to a PC via serial interface at a rate of 25 Hz by using a terminal program like HyperTerminal®, for example, and the data is stored in a text file.

Subsequent data processing is done in Matlab® or Octave®. The functions used to estimate the parameters are the same used in the simulation, including the heuristic filter. The only function is disabled that generates random data. Once calibrated, both the magnetometer data when the data from the accelerometers must be mostly contained in the surface of a unit sphere.

Fig. (5a) presents data from both the triads and uncalibrated. Fig. (5b) the same data after calibration.

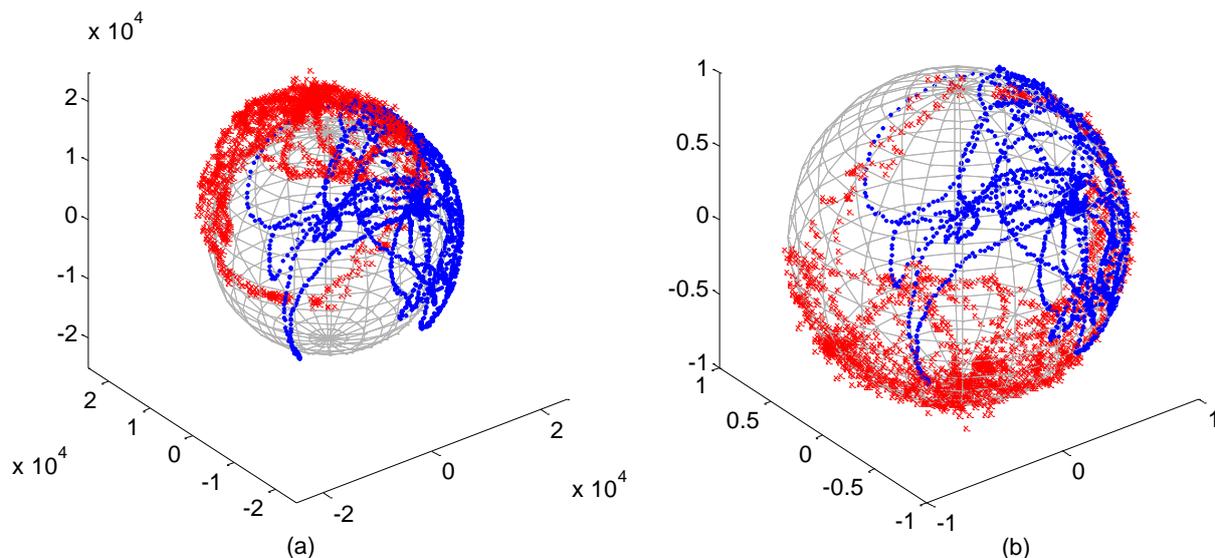


Figure 5. (a) Data from accelerometers (x) and magnetometers (•) triads before calibration. (b) The same data after calibration.

### 4.3. Performance check

The verification of the performance of the algorithm calibration was performed by comparing the attitude quaternion calculated using information from both the triads before and after adjusting the parameters, such as the quaternion from the AHRS Innalabs®. The comparison results are shown in Fig. (6).

In Fig (6a) can be noted that with no calibration errors in the quaternion components reach 0.25. In Figure (6b) the same data were reprocessed after the calibration algorithm in order to estimate the parameters of the triad. The differences of the components reach 0.05 at the latest.

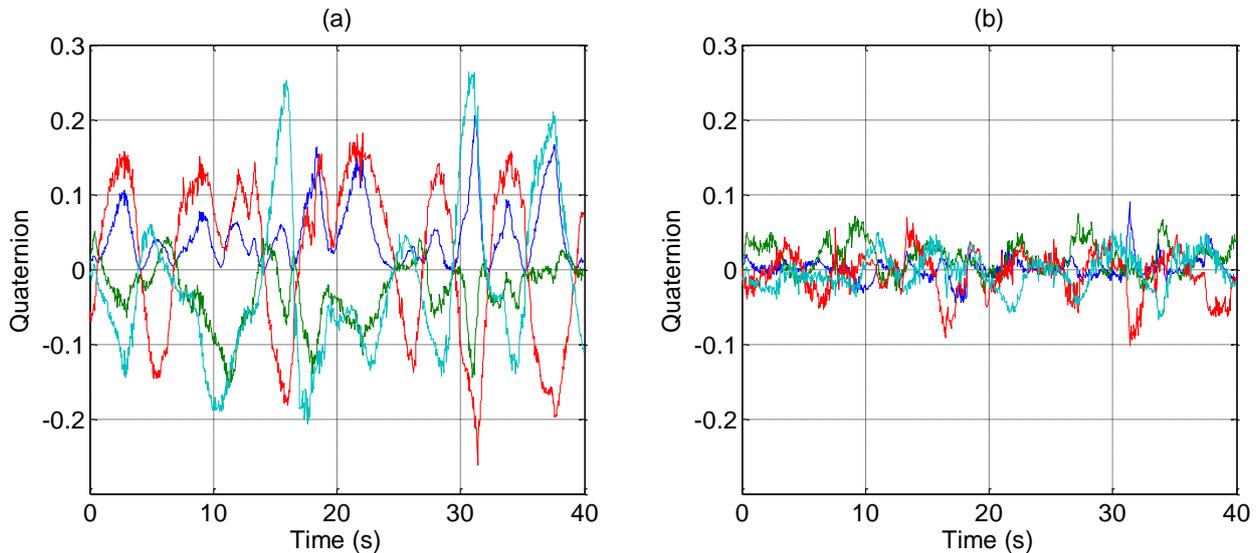


Figure 6. (a) Quaternion error calculated with triads before calibration. (b) The same attitude error after calibration.

## 5. SELF-CALIBRATION ALGORITHM

### 5.1. Intermediary Method

To implement a system for self-calibration is necessary to carry on the same CPU of the application, an algorithm that collects and processes the data to estimate the parameters of the triad. The algorithm described in Section 2.3 provides this implementation of way little expensive compared to the form of Section 2.2 that processes all data simultaneously.

Before board the algorithm in an application CPU it is possible implement an intermediate form where the parameters are calculated on a host PC and then sent to the embedded system. This method is perfectly applicable to perform a factory calibration.

### 5.2. Autonomous Method

To implement a fully autonomous method that maintains the sensors working properly is necessary to identify factors that cause the need of making a new calibration and also there must be a efficient method of identifying whether a new calibration will result in better parameters.

Temperatures above a threshold, the on-off of the sensors, or the aging of electronic components, are cases that can provoke a new calibration. These conditions can be easily verified using temperature sensors, flags to indicate the on-off and a real-time clock to count the time since the last calibration. The biggest challenge for run the calibration is to have a fully autonomous reliable method to collect data that can be used for calibration. An important result that was obtained by simulation is discussed in (Tormena, 2010) and reproduced at Fig. (7). Observing this graph we can say that at accuracy of the readings after the calibration procedure depends on the ratio between at variance of the sample group for the variance of noise samples. The higher this ratio, the better the quality of the estimation.

The implementation of this method can be done computationally by forming an orderly queue of fixed size  $n$ . This queue has two important properties: the mean and the variance of the inclination and declination angles of the vectors that compose it. At the head of the queue is stored the data further from the mean, or which contributes most to the variance, and on the tail, the closer to the mean, that is, which contributes less to at variance. Whenever a new data is inserted into the queue, at properties of mean and variance shall be updated. The test to collect data is very simple: if the new vector contributes to the increase in the queue total variance then it must take the place of that contributes less and

then the queue is reordered once again. When the variance reaches a predefined threshold, the calibration routine is called and the parameters of the sensors are recalculated.

The embedded implementation of this method is the ultimate goal of this research and will be described in future work.

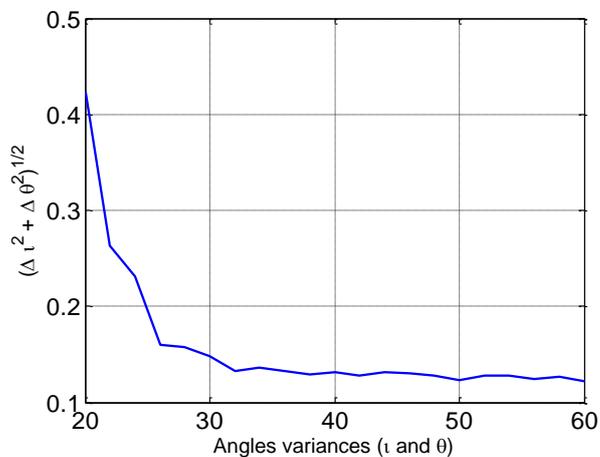


Figure 7.

## 6. CONCLUSIONS

This paper goes beyond the others that already were realized about calibration like the works of (Foster, 2008) and (Tormena, 2010) because was started the use of a hardware with data sensors to put on proof the early simulations.

This work presents a calibration algorithm that avoid a large number of operational matrix multiplications and inversions. This was obtained using a recursive Kalman estimator to avoid the use of Moore-Penrose pseudo inverse.

Tests with real data showed that the algorithm works a little below expectations, but the test showed that performance after calibration in an application to attitude determination, the quaternion error calculated was considerably reduced.

For an implementation of a functional algorithm in an embedded system, more needs to be analyzed and tested, but the initial results encourage the continuation of this development.

## 7. ACKNOWLEDGEMENTS

The authors are grateful to the Brazilian Space Agency (AEB), which through the Uniespaço II Program sponsored this research. The authors also are thankful to Londrina State University (UEL) and Labsim at National Institute for Space Research (INPE) that provided its physical and human resources that made this research work possible.

## 8. REFERENCES

- Alonso, R. and Shuster, M. D., 2002, "TWOSTEP: A Fast Robust Algorithm for Attitude-Independent Magnetometer-Bias Determination", *The Journal of Astronautical Sciences*, Vol. 50, No. 4, pp. 433-451.
- Foster, C.C. and Elkaim, G.H., 2008, "Extension of a Two-Step Calibration Methodology to Include Nonorthogonal Sensor Axes", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 44, No. 3, pp. 1070-1078.
- Frosio, I., Pedersini, F. and Borghese, N.A., 2009, "Autocalibration of MEMS Accelerometers", *IEEE Transactions on Instrumentation and Measurement*, Vol. 58, No. 6, pp. 2034-2041.
- Gebre-Egziabher, D. et al., 2001, "A Non-Linear, Two-Step Estimation Algorithm for Calibrating Solid-State Strapdown Magnetometers", In: 8th International St. Petersburg Conference on Navigational Systems. St. Petersburg, Russia.
- Granziera Jr., F., "Simulation and Implementation of a real-time attitude determination system using MEMS sensors", Master Thesis, Electrical Department Engineering of Londrina State University, 2006 (in Portuguese)
- Lerner, G. M. and Shuster, M. D., 1979, "Magnetometer Bias Determination and Attitude Determination for Near-Earth Spacecraft", In: AIAA Guidance and Control Conference. Boulder, Colorado, USA.
- Tormena Jr, O., Granziera Jr., Tosin, M. C., "Development and Performance Analysis of an Autocalibration Method for Tri-axis Sensors in Attitude Estimation Systems", *Proceedings of the XIV International Symposium on Dynamic Problems of Mechanics, DINAME*, 2011.

Tormena Jr, O., Granziera Jr., Tosin, M. C., “Self-calibration Method applied to MEMS Sensors used in Attitude Estimators Systems”, In: Brazilian Symposium on Inertial Engineering (SBEIN), 2010, Brazil (in Portuguese)  
Tormena Jr, O., “Self-Calibration Method for Tri-axes Sensors Used in Applications Attitude Estimation”, Master Thesis, Electrical Department Engineering of Londrina State University, 2010 (in Portuguese)

## **9. RESPONSIBILITY NOTICE**

The authors are the only responsible for the printed material included in this paper.