Journal of Engineering and Development, Vol.19, No.4. July. 2015, ISSN 1813- 7822

STATIC CALIBRATION OF TRIAXIAL ACCELEROMETER USING NONLINEAR NUMERICAL METHOD

Ass. Prof. Dr. Emad Natiq Abdulwahab
Mechanical Engineering Department
University of Technology

Ass. Prof. Dr. Mohammed Idrees Mohsin
Mechanical Engineering Department
University of Technology

Ass. Lec. Akeel Ali Wannas
Mechanical Engineering Department
University of Technology

Abstract

Triaxial accelerometers are used in various applications, such as inertial navigation systems (INSs) and inclinometers. Such accelerometers must be calibrated as accurately as possible because accelerometers with even small biases could result in a very fast position drift when they are used for INS applications and could result in inaccurate tilt angle measurements. Since the calibration parameters of a triaxial accelerometer vary under different environmental conditions, including temperature change, the gain factors and biases change whenever the sensor is switched on. Thus, an accelerometer must be calibrated before use or when the temperature is significantly changed. This paper presents a triaxial accelerometer calibration method using a mathematical model of nine calibration parameters: three gain factors, three biases and three non orthogonality factors. The fundamental principle of the proposed calibration method is that the sum of the triaxial accelerometer outputs is equal to the gravity vector when the accelerometer is stationary. The proposed method requires the triaxial accelerometer to be placed in forty eight tilt angles to estimate the nine calibration parameters. Since the mathematical model of the calibration parameters is nonlinear, an iterative method is used Levenberg-marquardt algorithm (LMA). The results are verified by comparing the estimated data and corrected data with how far or close from surface of sphere that have radius 1g and center (0,0,0) by calculating the standard deviation.

Keyword: MEMS*, Accelerometer, calibration, LMA.*

معايرة مقياس التعجيل ثلاثي المحاور في الوضع الثابت بأستخدام طريقة عددية غير خطية

أ.م.د. عماد ناطق عبد الوهاب
قسم الهندسة الميكانيكية
الجامعة التكنولوجية

أ.م.د. محمد ادريس محسن
قسم الهندسة الميكانيكية
الجامعة التكنولوجية

م.م. عقيل علي وناس
قسم الهندسة الميكانيكية
الجامعة التكنولوجية

الخلاصة

تستخدم اجهزة قياس التعجيل ثلاثي المحاور في مختلف التطبيقات، مثل نظم الملاحة بالقصور الذاتي وحساب الميل . قبل استخدام هذه الاجهزه يجب معايرتها بسبب وجود التحيز، حتى القيم الصغيرة يمكن أن تؤدي إلى الانجراف بشكل سريع جدا عندما يتم استخدامها لتطبيقات الملاحة يمكن أن يؤدي إلى قياسات غير دقيقة في زاوية الميل(كما في اجهزة المساحة الالكترونية). معاملات المعايرة لمقياس التعجيل الثلاثي المحاور تتغير بسبب تغير الظرو ف، بما في ذلك تغير درجة الحرارة، ووضعية الخزن لمده طويله حيث تتغير معاملات المعايرة كلما تم تشغيل جهاز استشعار . وبالتالي، يجب معايرة الاجهزة قبل الاستخدام أو عند تغيير كبير في درجة الحرارة . يقدم هذا المقال طريقة معايرة باستخدام نموذج رياضي من تسعة معاملات معايرة: ثلاثة عوامل للكسب و ثلاثة للتحيز وثلاثة عوامل لعدم التعامد . المبدأ الأساسي لطريقة المعايرة المقترح هو أن مجموع المخرجات لمقياس التعجيل يساوي متجه الجاذبية في حالة الوضعية الساكنة. تتطلب الطريقة المقترحة ان يوضع جهاز قياس التعجيل في ثمان وأربعين ز اوية مختلفة لحساب معاملات المعايرة التسعة. بما ان النموذج الرياضي لمعاملات المعايرة هو غير خطي، يتم استخدام أسلوب التكرار (LMA). بعد ذلك يتم التحقق من النتائج بمقارنة البيانات المقاسة من جهاز الاستشعار مع البيانات المصححة و مدى قربها من سطح الكرة التي نصف قطرها (g1) ومركزها (0,0,0) باحتساب الانحراف المعياري.

1 INTRODUCTION:

An accelerometer is a device that measures proper acceleration (g-force). Proper acceleration is not the same as coordinate acceleration (rate of change of velocity). For example, an accelerometer at rest on the surface of the Earth will measure an acceleration $g= 9.81\ m/s^2$ straight upwards. By contrast, accelerometers in free fall orbiting and accelerating due to the gravity of Earth will measure zero.

Accelerometers have multiple applications in industry and science. Highly sensitive accelerometers are components of inertial navigation systems for aircraft, missiles, UAV’s and robots [1]. Accelerometers are used to detect and monitor vibration in rotating machinery. Accelerometers are used in tablet computers and digital cameras so that images on screens are always displayed upright [2]. Accelerometers are used in drones for flight stabilisation. Pairs of accelerometers extended over a region of space can be used to detect differences (gradients) in the proper accelerations of frames of references associated with those points. These devices are called gravity gradiometers, as they measure gradients in the gravitational field. Such pairs of accelerometers in theory may also be able to detect gravitational waves [3]

Single and multi-axis models of accelerometer are available to detect magnitude and direction of the proper acceleration (or g-force), as a vector quantity, and can be used to sense orientation (because direction of weight changes), coordinate acceleration (so long as it produces g-force or a change in g-force), vibration, shock, and falling in a resistive medium (a case where the proper acceleration changes [4], since it starts at zero, then increases). Micromachined accelerometers are increasingly present in portable electronic devices and video game controllers, to detect the position of the device or provide for game input [4].

Triaxial accelerometers supplied for the consumer market are typically calibrated by using a six-element linear model(ellipsoid formula with arbitrary center) comprising a gain and offset or a nine-element nonlinear model comprising a gain, non-orthogonally and offset each of the three axes [5]. This factory calibration will change slightly as a result of the thermal stresses during soldering of the accelerometer to the circuit board. Additional small errors, external to the accelerometer, including rotation of the accelerometer package relative to the circuit board and misalignment of the circuit board to the final product, will also be introduced during the soldering and final assembly process, therefore we looking to obtain improved accuracy [6].

The accelerometer provides information about acceleration, speed and position of the Vehicle, by using statistical filtering. The development of such a system requires the calibration of sensor, which is a challenging task [7] , [8], [9], because, the data output depends on:

1. Local acceleration.
2. The sensor attitude.
3. The real gain and bias parameters.

In a 3D sensor each axis *i* have a different gain noted as α_i and *a* bias noted as β_i. Figure 1 represents a 3D sensor with offset $\vec{\beta} = (\beta_x, \beta_y, \beta_z)^T$, gains $\alpha_x, \alpha_y, \alpha_z$ and orthogonality errors S_{xy}, S_{xz}, S_{yz} for accelerometer; the model of the sensor output is given by [4] see Figure 2.

$$\overbrace{\begin{bmatrix} a_x(t) \\ a_y(t) \\ a_z(t) \end{bmatrix}}^{\vec{a}} = \overbrace{\begin{bmatrix} \alpha_x & S_{xy} & S_{xz} \\ S_{xy} & \alpha_x & S_{yz} \\ S_{xz} & S_{yz} & \alpha_x \end{bmatrix}}^{\alpha} \overbrace{\begin{bmatrix} a'_x(t) \\ a'_y(t) \\ a'_z(t) \end{bmatrix}}^{\vec{A}} + \overbrace{\begin{bmatrix} \beta_x \\ \beta_y \\ \beta_z \end{bmatrix}}^{\vec{\beta}} + \overbrace{\begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \end{bmatrix}}^{\varepsilon} \quad (1)$$

,

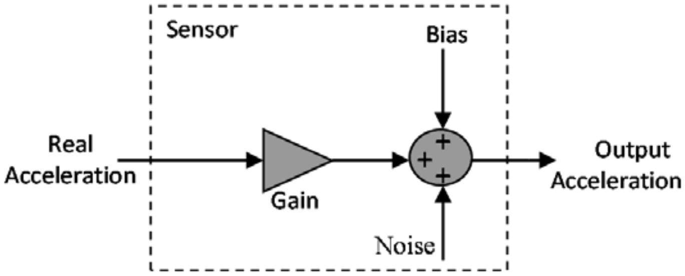


Figure 1 Sensor behavior

where ε is a residual error or noise, S_{ij} represents the orthogonality errors between sensor axes(cosine of angles between axes of sensor *x-y,y-z ,z-x where* $S_{xy}= S_{yx}$, $S_{yz}= S_{zy}$, *and* $S_{zx}= S_{xz}$). The matrix α is considered symmetrical [4].

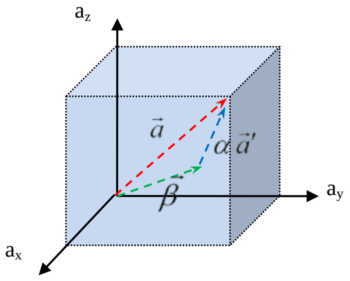


Figure 2 Acceleration vector $\vec{a}$ measured as $\vec{a}'$

2 CALIBRATION PROCEDURE

2.1 Acquisition of a set of points

A simple test rig (see Figure 3) consisting of an adjustable platform, a cube mounted with a triaxial accelerometer and a V-block was built to perform these forty eight positions [6]. Set of measurements with various attitudes are needed, so the parameters in equation (1) can be estimated. In the case of an accelerometer, the magnitude of the local force (or acceleration) must be precisely known; it equals the magnitude of the Earth gravity force (1g).

The calibration procedure stops when the measurements cover most of the surface of equation (1) [10]. The minimum procedure is to rotate the sensor in the reference *XYZ* coordinate system to measure a set of points which describes a closed solid.

2.2 2D projection of measurements

The calibration can be facilitated by a display of measured acceleration magnitude in spherical coordinates r(t) ,ϕ(t) and θ(t) (see Figure 4). This transformation requires an estimation of parameters that is obtained from the first measured points [11]:

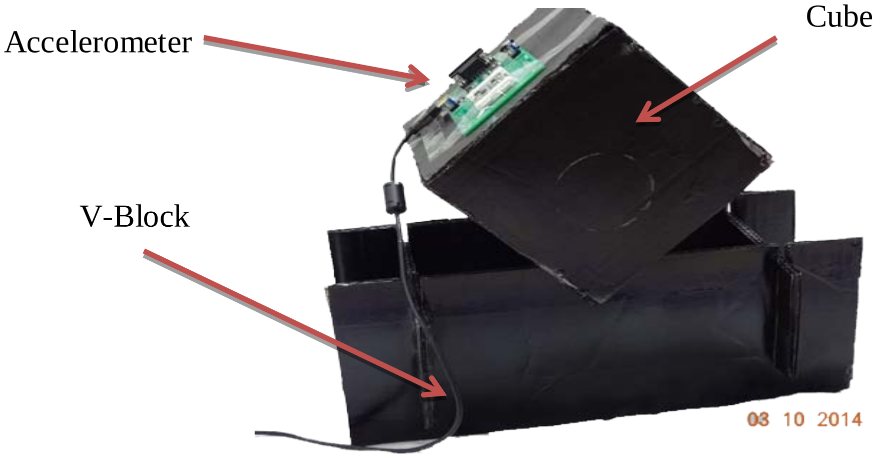


Figure 3 Experiment platform setup

$$r(t)=\sqrt{a_x(t)^2+a_y(t)^2+a_z(t)^2}\;, \tag{2}$$

$$\theta(t)=\tan^{-1}\left(\frac{\sqrt{a_x(t)^2+a_y(t)^2}}{a_z(t)^2}\right), \tag{3}$$

$$\varphi(t)=\tan^{-1}\left(\frac{a_y(t)^2}{a_x(t)^2}\right), \tag{4}$$

2.3 Estimation of parameters

To estimate the matrix α and the vector $\vec{\beta}$ in equation (1), we can minimize the following error function [12]:

$$E(p) = \sum_{t} (e_p(t))^2 , \qquad (5)$$

with:

$$e_p(t) = \|\vec{a}\|^2 - (\vec{a}'(t) - \vec{\beta})^T (\alpha^{-1})^2 (\vec{a}'(t) - \vec{\beta}) , \qquad (6)$$

where

$$\|\vec{a}\|^2 = a_x(t)^2 + a_y(t)^2 + a_z(t)^2 = 1g , \qquad (7)$$

$p = [\alpha_x, \alpha_y, \alpha_z, S_{xy}, S_{xz}, S_{yz}, \beta_x, \beta_y, \beta_z]^T$, the parameters vector to be estimated using LMA (Levenberg–Marquardt Algorithm)

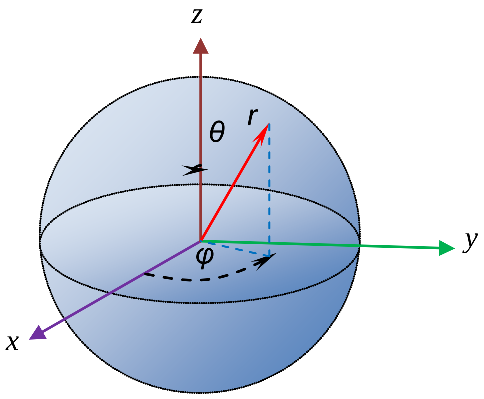


Figure 4 Sphere coordinates

2.4 The levenberg-marquardt method for nonlinear least squares

The Levenberg-Marquardt method is a standard technique used to solve nonlinear least squares problems. Least squares problems arise when fitting a parameterized function to a set of measured data points by minimizing the sum of the squares of the errors between the data points and the function. Nonlinear least squares problems arise when the function is not linear in the parameters. Nonlinear least squares methods involve an iterative improvement to parameter values in order to reduce the sum of the squares of the errors between the function and the measured data points. The Levenberg-Marquardt curve-fitting method is actually a combination of two minimization methods: the gradient descent method and the Gauss-Newton method. In the gradient descent method, the sum of the squared errors is reduced by updating the parameters in the direction of the greatest reduction of the least squares objective. In the Gauss-Newton method, the sum of the squared errors is reduced by assuming the least squares function is locally quadratic, and finding the minimum of the quadratic. The Levenberg-Marquardt method acts more like a gradient-descent method when the parameters are far from their optimal value, and act more like the Gauss-Newton method when the parameters are close to their optimal value.

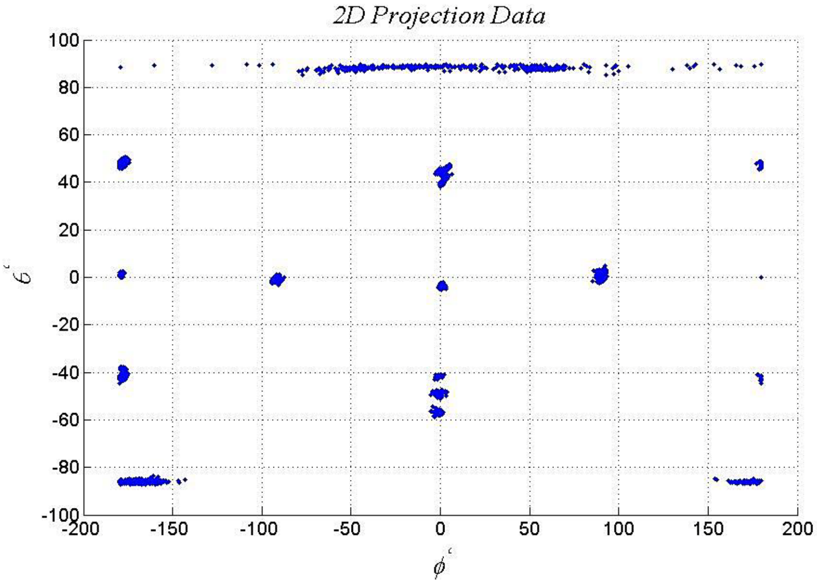


Figure 5 Measurement data of acceleration in 2d projection

In fitting a function $\hat{y}(t;p)$ of an independent variable t and a vector of n parameters p to a set of m data points (t_i, y_i), it is customary and convenient to minimize the sum of the weighted squares of the errors (or weighted residuals) between the measured data $y(t_i)$ and the curve-fit function $\hat{y}(t_i;p)$. This scalar-valued goodness-of-fit measure is called the chi-squared error criterion [10].

$$\chi^2(p) = \sum_{i=1}^{m}\left[\frac{y(t_i) - \hat{y}(t_i;p))}{w_i}\right]^2, \qquad (8)$$

$$\chi^2(p) = (y - \hat{y}(p))^T W(y - \hat{y}(p)), \qquad (9)$$

$$\chi^2(p) = y^T Wy - 2y^T W\hat{y} + \hat{y}^T W\hat{y}, \qquad (10)$$

The value w_i is a measure of the error in measurement $y(t_i)$. The weighting matrix W is diagonal with $W_{ii} = 1/w_i$. If the function $\hat{y}$ is nonlinear in the model parameters *p*, then the minimization of χ^2 with respect to the parameters must be carried out iteratively. The goal of each iteration is to find a perturbation *h* to the parameters *p* that reduces χ^2.

The Levenberg-Marquardt algorithm adaptively varies the parameter updates between the gradient descent update and the Gauss-Newton update [10],

$$\left[J^T WJ + \lambda I\right]h = J^T W(y - \hat{y}), \qquad (11)$$

where $m \times n$ Jacobian matrix $\mathrm{J} = \partial\hat{\mathrm{y}}/\partial\mathrm{p}$ and small values of the algorithmic parameter λ result in a Gauss-Newton update and large values of λ result in a gradient descent update. The parameter λ is initialized to be large. If iteration happens to result in a worse approximation, λ is increased. As the solution approaches the minimum, λ is decreased, the Levenberg-Marquardt method approaches the Gauss-Newton method, and the solution typically converges rapidly to the local minimum [6]. Marquardt's suggested update relationship [10],

$$\left[J^T WJ + \lambda\, diag(J^T WJ)\right]h = J^T W(y - \hat{y}), \qquad (12)$$

The alg orith m adjusts λ acco rd ing to whether χ^2 is increasing or decreasing as follows:

Given an initial guess for the set of fitted parameters p

1. Compute χ^2(p)
2. Choose a value for λ, for instance λ = 0:001
3. Calculate *h* and evaluate $\chi^2\ (p+h)$
4. If $\chi^2(p+h) \geq \chi^2(p)$ increase λ by a factor and go to (3) and try an update again.
5. If $\chi^2(p+h) < \chi^2(p)$ decrease λ by a factor, accept the updated trial solution $p \leftarrow p+h$ and go to (3) and try an update again.

The reasoning of the method is that if the error is increasing, the quadratic approximation in the Gauss-Newton method is not working well and we are likely not near a minimum, so λ should be increased in order to blend more towards steepest descent.

On the other hand,

if the error is decreasing, the approximation is working well, and we expect that we are getting closer to a minimum so λ is decreased to blend more towards Gauss-Newton. Levenberg-Marquardt's algorithm has the disadvantage that if the value of damping factor, λ, is large, inverting *($J^TWJ + \lambda I$)* is not used at all. Marquardt [12] realized that each component of the gradient can be scaled according to the curvature so that there is larger movement along the directions where the gradient is smaller. Thereby avoiding slow convergence in the direction of small gradient. Marquardt utilized this by replacing the identity matrix, *I* in Levenberg-Marquardt's original equations Equation (11) with the diagonal matrix consisting of the diagonal elements of J^TWJ (the Hessian matrix), resulting in the Levenberg-Marquardt algorithm, Equation (12) which then includes an estimation of the local curvature information and uses this to move further in the directions in which the gradient is smaller.

2.5 Numerical implementation

This work including the enhancement of a rank-1 Jacobian update. In iteration i, the step h is evaluated by comparing $\chi^2(p) to\, \chi^2(p+h)$. the step is accepted if the metric ρ_i [10] is greater than a user-specified value, ϵ_4,

$$\rho i(h) = \frac{\chi^2(p) - \chi^2(p+h)}{2h^T(\lambda_i + J^TW(y - \hat{y}(p)))}, \tag{13}$$

If in an iteration ρi(h) > ϵ4 then p + h is sufficiently better than p, p is replaced by p + h, and λ is reduced by a factor. Otherwise λ is increased by a factor, and the algorithm proceeds to the next iteration

2.5.1. Initialization and update of the L-M parameter, λ, and the parameters p [10]

1. $\lambda_0 = \lambda_0$; λ_0 is user-specified .
   $[J^TWJ + \lambda i\, diag[J^TWJ]]h = JTW(y - \hat{y}(p))$;
   $if\ \rho_i(h) > \epsilon_4 : p \leftarrow p + h; \lambda_{i+1} = max[\lambda_i / L_{\downarrow}, 10^{-7}]$;
   otherwise: $\lambda_{i+1} = min[\lambda_i / L_{\uparrow}, 10^{7}]$;];

2. $\lambda_0 = \lambda_0$ max$\left[diag[J^TWJ]\right] h$; λ_0 is user-specified .
   $[J^TWJ + \lambda i\, I]]h = J^TW(y - \hat{y}(p))$;
   $$= \left(\left(J^TW\left(y - \hat{y}(p)\right)\right)^T h\right) / \left(\left(\lambda^2(p+h) - \lambda^2(p)\right)/2 + / \left(J^TW\left(y - \hat{y}(p)\right)\right)^T h\right)$$

$if\ \rho_i(h) > \epsilon_4 : p \leftarrow p\ + \alpha h; \lambda_{i+1}\ =\ max[\lambda_i\,/(1+\alpha), 10^{-7}];$
otherwise: $\lambda_{i+1}\ =\ \lambda_i\ +\ |\lambda^2(p+\alpha h) - \lambda^2(p)|/(2\alpha);];$

3. $\lambda_0\ =\ \lambda_0$ max$\left[diag[J^TWJ]\right] h;$ λ_0 is user-specified .
$[J^TWJ\ +\ \lambda i\, I]]h\ =\ J^TW(y\ -\ \hat{y}(p));$
$if\ \rho_i(h) > \epsilon_4 : p \leftarrow p\ + \alpha h; \lambda_{i+1}\ = \lambda_i\, max[1/3\,, 1-(2\rho_i - 1)^3\,];$
otherwise: $\lambda_{i+1}\ =\ \lambda_i\ +\ |\lambda^2(p+\alpha h) - \lambda^2(p)|/(2\alpha);];$

where ϵ_4 is determines acceptance of a L-M step

2.5.2. Computation and rank-1 update of the Jacobian, $[\boldsymbol{\partial y/\partial p}]$

In the first iteration, in every $2n$ iterations, and in iterations where $\chi^2(p+h) > \chi^2(p)$, the Jacobian ($\boldsymbol{J}\ \in\ \mathbb{R}^{\boldsymbol{m\times n}}$) is numerically approximated using forward differences,

$$J_{ij} = \frac{\partial \hat{y}_i}{\partial p_j} = \frac{\hat{y}(t_i; p+\delta p_j) - \hat{y}(t_i; p)}{\left\|\delta p_j\right\|}, \tag{14}$$

or central differences (default)

$$J_{ij} = \frac{\partial \hat{y}_i}{\partial p_j} = \frac{\hat{y}(t_i; p+\delta p_j) - \hat{y}(t_i; p-\delta p_j)}{2\left\|\delta p_j\right\|}, \tag{15}$$

where the *j-th* element of δp_i is the only non-zero element and is set to $\Delta_j\left(1+\left|p_j\right|\right)$.n all other iterations, the Jacobian is updated using the Broyden rank-1 update formula,

$$J = J + \frac{(\hat{y}(p+h) - \hat{y}(p) - Jh)h^T}{h^T h}, \tag{16}$$

For problems with many parameters, a finite differences Jacobian is computationally expensive. Convergence can be achieved with fewer function evaluations if the Jacobian is re-computed using finite differences only occasionally. The rank-1 Jacobian updates equation (16) requires no additional function evaluations.

2.5.3. Convergence criteria

Convergence is achieved when one of the following three criteria is satisfied,

1. Convergence in the gradient, $max|J^TW(y\ -\ \hat{y})| < \epsilon_1;$
2. Convergence in parameters, $max\ |h_i/p_i|\ < \epsilon_2$; or
3. Convergence inχ^2, $\chi^2/(m\ -\ n\ +\ 1)\ <\ \epsilon_3$.

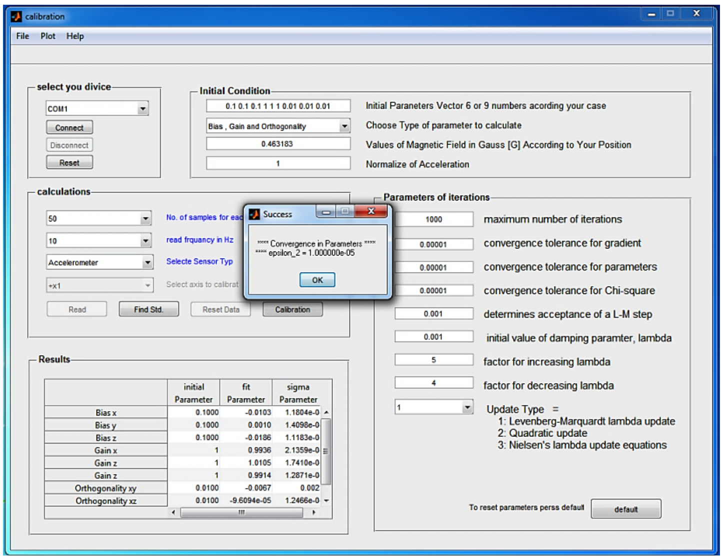
where ϵ_1, ϵ_2 and ϵ_3 are convergence tolerance for gradient, parameters and Chi-square respectively.

Otherwise, iterations terminate when the iteration count exceeds a pre-specified limit

To get data from sensor and solve equation (1) numerically for estimate nine parameters, Figure 6 illustrates the graphical user interface of the developed calibration software, interfaced with a Matlab

3 RESULTS AND DISCUSSION

This section gives some calibration results for a 3D accelerometer specifications in [13] The calibration procedure has been performed three times successively, according to section (II). Each time, 4800 data points are used for the estimation of parameters. The obtained gains, bias and orthogonality are then compared in the following tables. The estimation algorithm LMA fit performs the alpha, beta and S_{ij} computation instantly, with about less than 100 iterations [4].



select you divice

COM1

Initial Condition

0.1 0.1 0.1 1 1 1 0.01 0.01 0.01 Initial Paraneters Vector 6 or 9 numbers acording your case

Bias , Gain and Orthogonality Choose Type of parameter to calculate

0.463183 Values of Magnetic Field in Gauss [G] According to Your Position

1 Normalize of Acceleration

calculations

50 No. of samples for eac

10 read frquancy in Hz

Accelerometer Selecte Sensor Typ

+x1 Select axis to calibrat

**** Convergence in Parameters ****
**** epsilon_2 = 1.000000e-05

Parameters of iterations

1000 maximum number of iterations

0.00001 convergence tolerance for gradient

0.00001 convergence tolerance for parameters

0.00001 convergence tolerance for Chi-square

0.001 determines acceptance of a L-M step

0.001 initial value of damping paramter, lambda

5 factor for increasing lambda

4 factor for decreasing lambda

1 Update Type =
1: Levenberg-Marquardt lambda update
2: Quadratic update
3: Nielsen's lambda update equations

To reset parameters perss defaul

Results

	initial Parameter	fit Parameter	sigma Parameter
Bias x	0.1000	-0.0103	1.1804e-0
Bias y	0.1000	0.0010	1.4098e-0
Bias z	0.1000	-0.0186	1.1183e-0
Gain x	1	0.9936	2.1359e-0
Gain z	1	1.0105	1.7410e-0
Gain z	1	0.9914	1.2871e-0
Orthogonality xy	0.0100	-0.0067	0.002
Orthogonality xz	0.0100	-9.6094e-05	1.2466e-0

Figure 6 Matlab Gui program to Estimate Calibration Parameters

In Table 1 the calibration results for a 3D accelerometer sensor. The sensor must only measure the static acceleration of gravity, for which the norm is constant. We need to exclude measurements where the sensor is not static, for which dynamic acceleration is present, by observing the temporal variance of the data signal. In static position the accelerometer has a low variance.

Figure 7 shows that the estimated parameters define a sphere according to the model equation (2) that fits the measurement [3]. The calibration results in Table 1 shows that the gains, bias and the orthogonality of sensors gives very good stability. In some applications the orthogonality errors cannot be taken into account because of the low impact on the measurements [3].

Table 1 Measured Parameters

Parameter Name	Test 1	Test 2	Test 3
β_x	-0.0103	-0.0099	-0.00101
β_y	0.0010	-0.0008525	-0.0094
β_z	-0.0186	-0.0186	-0.0186
α_x	0.9936	0.9930	0.9938
α_y	1.0105	1.0098	1.0102
α_z	0.9914	0.9917	0.9920
s_{xy}	-0.0067	0.0011	0.00118
s_{xz}	-0.000096	0.000766	0.000806
S_{yz}	0.0072	0.000528	0.00055
standard deviation for x axis	0.001419	0.000956	0.0015438
standard deviation for y axis	0.00156	0.00194	0.00167
standard deviation for z axis	0.00115	0.00123	0.00145

4 CONCLUSIONS

The numerical calibration of 3D accelerometers sensors is not an expensive solution and can be implemented with no mechanical means. The calibration procedure can be performed after the integration phase of electronics to take into account the drift and uncertainty of the component itself, all the gains, bias and geometry. The calibration parameters of accelerometers are accurate at 1%.

This method of calibration has been successfully used for the preparation of **M**icro**E**lectro**M**echanical **S**ystems (MEMS) inertial units, in a pedestrian tracking system, robot and UAVs. It is worth noting the possibility of using this method in calibration of magnetometer.

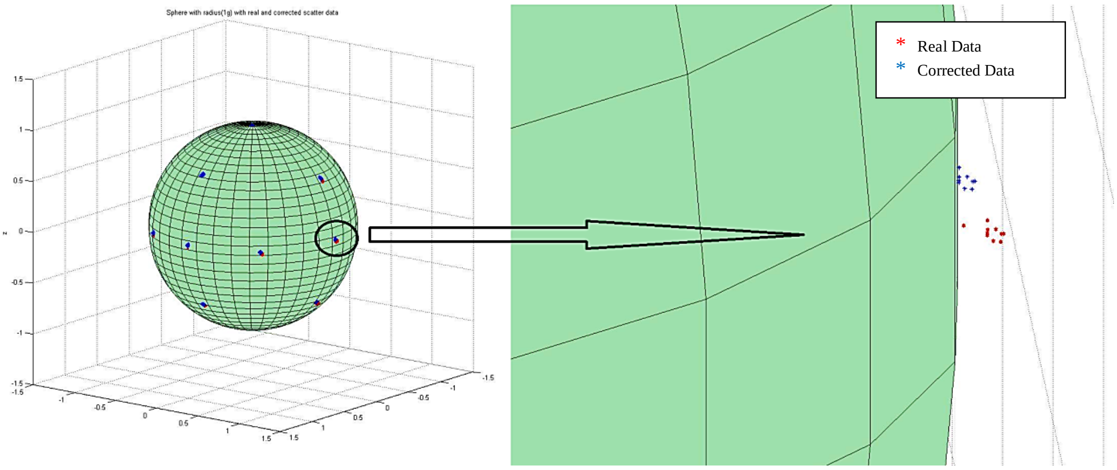


Figure 7 **Sphere of 3D accelerometer sensor**

5 REFERENCES

[1] Jiancheng Fang, Beijing, and Zhanchao Liu, "A New Inclination Error Calibration Method of Motion Table Based on Accelerometers," *Instrumentation and Measurement, IEEE*, vol. 64, no. 2, pp. 487-493, August 2014.

[2] Enrico Calore and Iuri Frosio, "Accelerometer-based correction of skewed horizon and keystone distortion in digital photography," *Vision Computing, scienceDirect*, vol. 32, no. 9, pp. 606–615, 2014.

[3] P. Batista, C. Silvestre, P. Oliveira, and B. Cardeira, "Accelerometer Calibration and Dynamic Bias and Gravity Estimation: Analysis, Design, and Experimental Evaluation," *Control Systems Technology*, vol. 19, no. 5, pp. 1128 - 1137, Sept 2011.

[4] F. Camps, S. Harasse, and A. Monin, "Numerical calibration for 3-axis accelerometers and magnetometers," *in Electro/Information Technology, 2009. eit '09. IEEE International Conference on, Windsor*, pp. 217 - 221, June 2009.

[5] Tadej Beravs, Janez Podobnik, and Munih Marko, "Three-Axial Accelerometer Calibration Using Kalman Filter Covariance Matrix for Online Estimation of Optimal Sensor Orientation," *Instrumentation and Measurement, IEEE*, pp. 2501 - 2511, Sept 2012.

[6] Mohammed Awad Kian Sek Tee, Abbas Dehghani, David Moser, and Saeed Zahedi, "Triaxial Accelerometer Static Calibration," in *Proceedings of the World Congress on Engineering 2011 Vol III* , vol. 1, London, U.K., January July 6 - 8, 2011, p. 1309.

[7] Roberto Alonso and Malcolm D. Shuster, "Complete Linear Attitude-Independent Magnetometer," *Jornal of the Astronautical Sciences*, vol. 50, no. 4, pp. 477-490, December 2002.

[8] Timo Pylvänäinen, "Automatic and adaptative calibration of 3D field sensors," *Applied Mathematical Modelling ,scienceDirect*, vol. 32, no. 4, pp. 575–587 , April 2008. 81

[9] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, *Numerical Recipes in C The Art of Scientific Computing*. Cambridge: Cambridge University Press, 1992.

[10] K. Transtrum Mark and P. Sethna James, "Improvements to the Levenberg-Marquardt algorithm for nonlinear least-squares minimization," *Journal of Computational Physics*, January 2012.

[11] George B. Thomas Jr., Maurice D. Weir, and Joel R. Hass, *Thomas' Calculus, Multivariable*, 12th ed.: Pearson Education, 2009.

[12] D.W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431-441, 1963.

[13] YEI Technology. (2011) yei 3-space embedded sensor. [Online]. http://www.yeitechnology.com/sites/default/files/3-Space_Sensor_Users_Manual_Embedded_1.1_r20_18Oct2012_0.pdf