# Performance comparison of accelerometer calibration algorithms based on 3D-ellipsoid fitting methods

Matthias Gietzelt[1], Klaus-Hendrik Wolf[1], Michael Marschollek[2], and Reinhold Haux[1]

[1]Peter L. Reichertz Institute for Medical Informatics,
University of Braunschweig – Institute of Technology and Hannover Medical School,
Mühlenpfordtstr. 23, D-38106 Braunschweig, Germany

Matthias Gietzelt, Matthias.Gietzelt@plri.de
Dr. Klaus-Hendrik Wolf, Klaus-Hendrik.Wolf@plri.de
Prof. Dr. Reinhold Haux, Reinhold.Haux@plri.de

[2]Peter L. Reichertz Institute for Medical Informatics,
University of Braunschweig – Institute of Technology and Hannover Medical School,
Carl-Neuberg-Str. 1, D-30625 Hannover, Germany

Prof. Dr. Dr. Michael Marschollek, Michael.Marschollek@plri.de

Corresponding author: Matthias Gietzelt

*Keywords*: calibration, non-iterative, accelerometer, ellipsoid fitting

## Abstract

Calibration of accelerometers can be reduced to 3D-ellipsoid fitting problems. Changing extrinsic factors like temperature, pressure or humidity, as well as intrinsic factors like the battery status, demand to calibrate the measurements permanently. Thus, there is a need for fast calibration algorithms, e.g. for online analyses. The primary aim of this paper is to propose a non-iterative calibration algorithm for accelerometers with the focus on minimal execution time and low memory consumption. The secondary aim is to benchmark existing calibration algorithms based on 3D-ellipsoid fitting methods. We compared the algorithms regarding the calibration quality and the execution time as well as the number of quasi-static measurements needed for a stable calibration. As evaluation criterion for the calibration, both the norm of calibrated real-life measurements during inactivity and simulation data was used. The algorithms showed a high calibration quality, but the execution time differed significantly. The calibration method proposed in this paper showed the shortest execution time and a very good performance regarding the number of measurements needed to produce stable results. Furthermore, this algorithm was successfully implemented on a sensor node and calibrates the measured data on-the-fly while continuously storing the measured data to a microSD-card.

## 1. Introduction

Accelerometer sensors are an important part of biomedical applications and in the context of ambient assisted living [1]-[4]. Information extracted from accelerometric sensors can be used e.g. for fall detection, activity recognition, or gait analysis. Thereby, the calibration of raw analog-to-digital (A/D) converted data of accelerometers is an important field of research. Changing extrinsic factors like temperature, pressure or humidity, as well as intrinsic factors like the battery status, can have an influence on these measurements. Therefore, the measurements have to be calibrated permanently before interpreting the data.

To enable an interpretation and evaluation on small microprocessors of sensor boards, the algorithm used should be as efficient as possible. For most calibration algorithms, an implementation may not be possible due to the limited computation speed and low memory capacity.

It is known that calibration algorithms for three-dimensional accelerometers and magnetometers are equivalent to a 3D-ellipsoid fitting problem [5]. The ellipsoid equation for a three-dimensional Cartesian coordinate system can be written as:

$$\left(\frac{v_\mathrm{x}-o_\mathrm{x}}{s_\mathrm{x}}\right)^2+\left(\frac{v_\mathrm{y}-o_\mathrm{y}}{s_\mathrm{y}}\right)^2+\left(\frac{v_\mathrm{z}-o_\mathrm{z}}{s_\mathrm{z}}\right)^2=1, \tag{1}$$

whereas $v=\left(v_\mathrm{x},\ v_\mathrm{y},\ v_\mathrm{z}\right)^\mathrm{T}$ is the vector of measurement values, $o=\left(o_\mathrm{x},\ o_\mathrm{y},\ o_\mathrm{z}\right)^\mathrm{T}$ denotes the ellipsoid's center and $s=\left(s_\mathrm{x},\ s_\mathrm{y},\ s_\mathrm{z}\right)^\mathrm{T}$ the equatorial and polar radii, respectively. More generally, the ellipsoid equation can be written as

$$(v-o)^\mathrm{T}\cdot A\cdot(v-o)=1, \tag{2}$$

whereas $A$ is a symmetric, positive definite matrix. In the three-dimensional case $A$ is a $3\times 3$ matrix. The values $v$ can be calibrated by computing

$$v_{\text{calibrated}} = U \cdot (v - o), \tag{3}$$

whereas $U$ is the upper triangular matrix calculated from $A$ using the Cholesky decomposition $A = U^{\mathrm{T}}U$. Equation 2 in the three-dimensional case is equivalent to equation 1, if and only if $A$ is a diagonal matrix.

## 1.1 State-of-the-Art 3D-Ellipsoid Fitting Methods

Below, an overview about existing calibration methods should be given. Each method will be briefly described. Bonnet et al. [5] identified following 3D-ellipsoid fitting approaches:

1. ellipsoid fitting using linear optimization,
2. ellipsoid fitting using non-linear optimization,
3. ellipsoid fitting using an enclosing ellipsoid.

Additionally, it should be mentioned that there is also the naïve method of manual calibration, by aligning all sensor axes in and against the direction of gravity.

The calibration algorithm of Lötters et al. is based on a linear optimization method [6]. It uses a stable linear minimum variance estimator to estimate the parameters $s$ and $o$ iteratively from given initial values. In the following sections, this algorithm will be identified by "*LinMinVarEstimate*".

A calibration algorithm which is also based on a linear optimization method was introduced by Merayo et al. [11]. It differs in its parameterization from LinMinVarEstimate and uses a linear least-squares estimator. This method will be called "*MgnCalibration*".

Lukowicz et al. proposed to solve equation 1 using a non-linear solver [7]. The Levenberg-Marquardt algorithm is such a numerical optimization algorithm for solving non-linear problems using a least-square approach. [8], [9]. This will be called "*NonLinear*" in the following sections.

Another non-linear approach was introduced by Campolo et al. [10]. They proposed a non-linear error-of-fit function and solve the problem also in least-square sense. Campolo et al. suggested a calculation rule to find "good" initial values for the parameters of the ellipsoid, whereas the center will be estimated by the mean values and the approximation of the ellipsoid's "radius" is

$$r = \frac{1}{N} \cdot \sum_{i=1}^{N} \sqrt{\left(v_{\mathrm{x},i} - o_{\mathrm{x}}\right)^2 + \left(v_{\mathrm{y},i} - o_{\mathrm{y}}\right)^2 + \left(v_{\mathrm{z},i} - o_{\mathrm{z}}\right)^2}. \tag{4}$$

This method will be called "*LsqNonLinear*".

Bonnet et al. suggested to compute the parameters with a minimum volume enclosing ellipsoid [5] using Khachiyan's algorithm [13]. They discussed the numerical safeness with respect of other approaches (especially the approach of Merayo et al.) and concluded that the minimum volume enclosing ellipsoid method will always return a valid ellipsoid independent of the quality of data. This method will be named "*MinVolEllipsoid*" in the following.

The methods LinMinVarEstimate, NonLinear, and LsqNonLinear solve equation 1 for the three offset parameters $o_{\mathrm{x}}$, $o_{\mathrm{y}}$ and $o_{\mathrm{z}}$ and the three scale parameters $s_{\mathrm{x}}$, $s_{\mathrm{y}}$ and $s_{\mathrm{z}}$. These three methods need initial values due to the used solver. In contrast, the methods MgnCalibration and MinVolEllipsoid are based on equation 2. These will additionally compute the rotation angles that are expressed in the matrix $A$ regarding the coordinate system.

There might be a lack of a non-iterative method which can be implemented on sensor nodes with very limited computing and memory capacity.

| | LinMinVar Estimate [6] | NonLinear [7] | LsqNon Linear [10] | Mgn Calibration [11] | MinVol Ellipse [5] | LinAuto Calibration* |
|---|---|---|---|---|---|---|
| Method | Linear optima-zation | Non-linear optima-zation | Non-linear optima-zation | Linear optima-zation | Enclosing Ellipsoid | Linear optima-zation |
| Solve for equation | 1 | 1 | 1 | 2 | 2 | 1 |
| Need for initial values | yes | yes | yes | no | no | no |
| Iterative algorithm | yes | yes | yes | no | yes | no |
| MATLAB implementation available | no | yes [7] | yes [10] | yes [12] | yes [14] | yes |

**Table 1:** Comparison of calibration algorithms. *introduced in section 2.1.

### 1.2 Objectives

The aim of our research was

1. to propose a non-iterative calibration algorithm for accelerometers, with the focus on minimal execution time and low memory consumption,
2. to benchmark existing in-field calibration algorithms for accelerometers,
3. to implement the proposed algorithm on a microprocessor of a sensor node.

## 2. Methods

### 2.1 A Non-Iterative Calibration Method

Usually, it is possible to form and solve an over-determined non-linear equation system in order to computes the parameters $o$ and $s$ (cf. [7]). The approach presented in the following consists of two non-iterative steps. In the first step, the center $o$ of the ellipsoid is estimated, and in the second step the scale factor $s$ is computed. Aim of both steps is to form and solve a system of linear equations in a least-square sense that have the same number of equations and unknowns.

#### First Step

For the first step we assume that the ellipsoid's radii are approximately equal to each other. This assumption is reasonable, especially when the 3D-accelerometer output was converted by the same A/D converter and the same reference voltage. This assumption is computationally very effective, because we can use a sphere in Cartesian coordinates to estimate the center coordinates:

$$\left(v_{\mathrm{x},i} - o_\mathrm{x}\right)^2 + \left(v_{\mathrm{y},i} - o_\mathrm{y}\right)^2 + \left(v_{\mathrm{z},i} - o_\mathrm{z}\right)^2 = r^2, \tag{5}$$

whereas $i$ denotes the $i$-th of $n$ measurements. It follows:

$$\begin{aligned} \Rightarrow {o_\mathrm{x}}^2 + {o_\mathrm{y}}^2 + {o_\mathrm{z}}^2 - r^2 - 2 \cdot o_\mathrm{x} \cdot v_{\mathrm{x},i} - 2 \cdot o_\mathrm{y} \cdot v_{\mathrm{y},i} - 2 \cdot o_\mathrm{z} \cdot v_{\mathrm{z},i} \\ = -\left({v_{\mathrm{x},i}}^2 + {v_{\mathrm{y},i}}^2 + {v_{\mathrm{z},i}}^2\right). \end{aligned} \tag{6}$$

By substituting

$$A \coloneqq {o_\mathrm{x}}^2 + {o_\mathrm{y}}^2 + {o_\mathrm{z}}^2 - r^2,\ B \coloneqq -2 \cdot o_\mathrm{x},\ C \coloneqq -2 \cdot o_\mathrm{y},\ D \coloneqq -2 \cdot o_\mathrm{z}, \text{ and} \tag{7a-d}$$

$$E_i \coloneqq -\left({v_{\mathrm{x},i}}^2 + {v_{\mathrm{y},i}}^2 + {v_{\mathrm{z},i}}^2\right) \tag{8a-c}$$

equation 6 emerges to

$$A + B \cdot v_{\mathrm{x},i} + C \cdot v_{\mathrm{y},i} + D \cdot v_{\mathrm{z},i} = E_i. \tag{9}$$

The error can be minimized in least square sense

$$P(v) = \sum_{i=1}^{n} \left(A + B \cdot v_{\mathrm{x},i} + C \cdot v_{\mathrm{y},i} + D \cdot v_{\mathrm{z},i} - E_i\right)^2 \longrightarrow \min. \tag{10}$$

Calculating the partial derivative of $P(v)$ with respect to $A$, $B$, $C$ and $D$ and rearranging the terms will emerge to

$$\begin{pmatrix} n & \sum_i v_{\mathrm{x},i} & \sum_i v_{\mathrm{y},i} & \sum_i v_{\mathrm{z},i} \\ \sum_i v_{\mathrm{x},i} & \sum_i {v_{\mathrm{x},i}}^2 & \sum_i v_{\mathrm{x},i} \cdot v_{\mathrm{y},i} & \sum_i v_{\mathrm{x},i} \cdot v_{\mathrm{z},i} \\ \sum_i v_{\mathrm{y},i} & \sum_i v_{\mathrm{x},i} \cdot v_{\mathrm{y},i} & \sum_i {v_{\mathrm{y},i}}^2 & \sum_i v_{\mathrm{y},i} \cdot v_{\mathrm{z},i} \\ \sum_i v_{\mathrm{z},i} & \sum_i v_{\mathrm{x},i} \cdot v_{\mathrm{z},i} & \sum_i v_{\mathrm{y},i} \cdot v_{\mathrm{z},i} & \sum_i {v_{\mathrm{z},i}}^2 \end{pmatrix} \cdot \begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} = \begin{pmatrix} \sum_i E_i \\ \sum_i E_i \cdot v_{\mathrm{x},i} \\ \sum_i E_i \cdot v_{\mathrm{y},i} \\ \sum_i E_i \cdot v_{\mathrm{z},i} \end{pmatrix}. \tag{11}$$

Solving this equation system will compute the center $o$ of the sphere of equation (11) after $B$, $C$ and $D$ have been resubstituted using equations 7a-d and 8. Moreover, it is possible to compute a first approximation of the sphere's radius, which represents an approximation of the ellipsoid's largest radius:

$$r = \sqrt{{o_\mathrm{x}}^2 + {o_\mathrm{y}}^2 + {o_\mathrm{z}}^2 - A}\,. \tag{12}$$

This estimation can help to decide about proceeding with the second step or waiting for more measurement values $v$ because of an unacceptable error. Please note that we did not make any data related assumptions so far. Regardless of having a-priori-knowledge about the measurement values, it is possible to deploy conditions for an error handling:

1. The sphere's center coordinates should not differ strongly from each other, because the same physical parameter was measured for each axis and may converted by the same A/D converter with the same reference voltage.
2. If there is insufficient data ($n < 3$), the matrices of the equation system may be singular and therefore the parameters $o$ and $s$ cannot be determined.

In case of a-priori-knowledge about the measurement data there are two more conditions that allow a deeper error handling:

3. The measurement interval must include the whole sphere, because accelerometers usually have a sensitivity of more than ±1g. If the sampling resolution is known, the

radius of the sphere should not exceed the used conversion resolution. I.e., if a resolution of 12 bit was chosen, the radius with respect to the sphere's center should not exceed the interval [0, $2^{12}/2$] in all directions.

4. If the sensitivity of the measurements is known, we can conclude, whether the radius $r$ has a value of the radius expected. E.g., if the sensitivity of an accelerometer is ±4 g, the radius $r$ should be a value of approximately $r \approx \frac{1}{4} \cdot o$ for each component and should not fall below a minimum or exceed a maximum radius.

**Second Step**

Equation 1 can be rewritten as

$$t_x X_i + t_y Y_i + t_z Z_i = 1 \tag{13}$$

by substituting

$$X_i := (v_{x,i} - o_x)^2, \quad Y_i := (v_{y,i} - o_y)^2, \quad Z_i := (v_{z,i} - o_z)^2 \text{ and} \tag{14a-c}$$

$$t_x := \frac{1}{{s_x}^2},\ t_y := \frac{1}{{s_y}^2},\ t_z := \frac{1}{{s_z}^2}. \tag{15a-c}$$

The resulting error can be also minimized in a least square sense:

$$Q(t) = \sum_i (t_x X_i + t_y Y_i + t_z Z_i - 1)^2 \longrightarrow \text{min.} \tag{16}$$

Again, calculating the partial derivative of $Q(t)$ with respect to $t_x, t_y$, and $t_z$ and rearranging the terms will emerge to

$$\begin{pmatrix} \sum_i {X_i}^2 & \sum_i X_i Y_i & \sum_i X_i Z_i \\ \sum_i X_i Y_i & \sum_i {Y_i}^2 & \sum_i Y_i Z_i \\ \sum_i X_i Z_i & \sum_i Y_i Z_i & \sum_i {Z_i}^2 \end{pmatrix} \cdot \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} = \begin{pmatrix} \sum_i X_i \\ \sum_i Y_i \\ \sum_i Z_i \end{pmatrix}. \tag{17}$$

Equation 17 has to be solved regarding vector $t = (t_x,\ t_y,\ t_z)^T$. $s$ can be resubstituted using the equations 14a-c and 15a-c mentioned above. Below, the method presented will be called "*LinAutoCalibration*".

Assume that the values $X_i$, $Y_i$ and $Z_i$ are arranged in a $n \times 3$ matrix $V$. The $3 \times 3$ matrix in equation 17 can be calculated by computing $V^T V$. The lower right hand matrix of equation 11 can be calculated similarly assuming the measurement values $v_{x,i}$, $v_{y,i}$ and $v_{z,i}$ are arranged in a $n \times 3$ matrix.

## 2.2 Subjects and Data

The data for comparing the calibration methods was taken from an explorative study that was conducted during June 2nd – 5th, 2009. The measurements of 5 healthy subjects, aged 24 – 32 years, were analyzed. The subjects wore the accelerometers in a small case, which was attached to the belt at the subject's preferred position. The subjects performed their normal daily activities during the study.

The data was used to determine the quality of calibration in terms of accuracy and the time needed to calibrate. The norm of the calibrated vectors was used as accuracy criterion, which should be 1 g for all vectors during inactivity sections (IS).

The subject's first day of wearing the sensor was used as training data set for estimating the 3D-ellipsoid in each case. The test sets were represented by all the other days except the first day. Each sensor was calibrated separately with its corresponding training data set. Thereby, only IS vectors were included.

The training data sets consisted of 1057 to 2287 IS vectors depending on behavior and wearing time. For the algorithms that need an initial value (i.e., for the calibration algorithms LinMinVarEstimate, NonLinear and LsqNonLinear), we used a-priori-knowledge from the data and specified the initial ellipsoid's center with $o = (2300, 2300, 2300)^{\mathrm{T}}$ and the ellipsoid's radii with $s = (500, 500, 500)^{\mathrm{T}}$, which may fit well to the data.

### 2.3 Simulation Data

Due to the lack of ground truth in the study data the true parameters $s$ and $o$ remain unknown. The only criterion to estimate the quality of calibration was the norm of the IS vectors. To study the estimation of the parameters $s$ and $o$ and to compare them with a ground truth, additional simulation data is needed. 200 points on the surface of an ellipsoid with the defined parameters $o = (2300, 2300, 2300)^{\mathrm{T}}$ and $s = (500, 500, 500)^{\mathrm{T}}$ were randomly selected. To simulate a typical measurement error, a jitter of ±1% was used. This data set was split into a training and test data set of 100 points each.

### 2.4 Inactivity Section Detection

IS were detected by computing the dimensionless coefficient of variance from the uncalibrated data using a sliding window containing two seconds of data. Thereby, only one of the axes (the x-axis) was considered for computing this coefficient. The threshold value for inactivity $c = 3 \cdot 10^{-3}$ was found empirically. The coefficient of variance was used because of its independency regarding the measurement interval.

Detecting IS in humans daily activity can cause redundant measurements of the same direction of inactivity. These measurements are typically biased to a certain direction as shown in figure 1. When computing the 3D-ellipsoid, this bias may have an influence on the parameter estimation. Thus, only up to ten measurements of a certain direction were included in the training data sets.

### 2.5 Hard- and Software

The used sensor node was a Shimmer sensor Rev. 1.3 [15] with the integrated tri-axial accelerometer MMA7260Q [16]. The CPU of the Shimmer platform has the designation MSP430F1611 with 8 MHz providing 10 kBytes RAM, 48 kBytes flash memory and 8 channels of 12 bit A/D. The sensor node was configured at a sample rate of 51.2 Hz, a data resolution of 12 bit and a sensitivity of ±4 g. The data was stored on a microSD card and analyzed after the measurements took place.

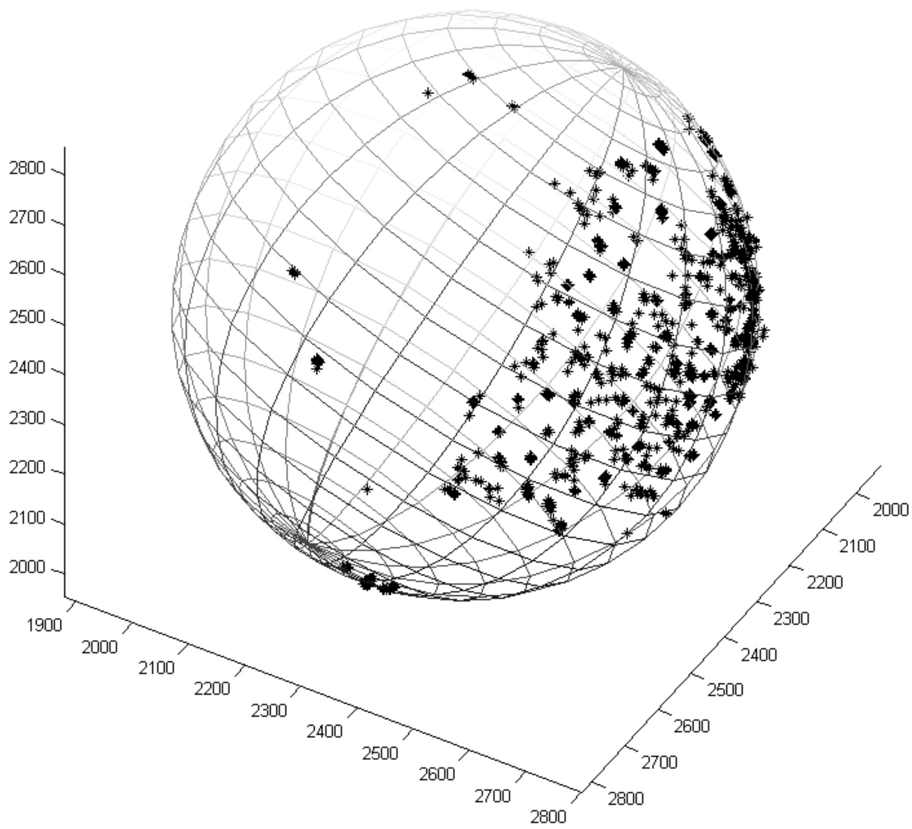


**Figure 1:** This scatter-plot of typical accelerometric measurements during inactivity surrounded by a fitted 3D-ellipsoid shows a bias of measurement values in a certain direction. The axes lengths showed just small deviations. This plot was made using a MATLAB script written by Moshtagh [17].

All calibration algorithms were implemented and executed using MATLAB Version 7.9.0.529 (R2009b) as 32-bit version. We decided to use the interpreting programming language MATLAB to increase the reproducibility of the comparison and because in most cases the algorithms written in MATLAB were freely available from the internet or the authors gave MATLAB implementation recommendations in their original publications. In the other cases the algorithms were implemented by the authors with the focus on minimal execution time and memory consumption. The MATLAB scripts were executed on a 2 GHz Intel Core 2 Duo processor T5870. We intentionally used a computer with a lower performance to obtain noticeable differences in performance results.

## 3. Results

### 3.1. Applying the Study Data

The estimated 3D-ellipsoids were applied to calibrate the

1. training data sets to show how good the estimated 3D-ellipsoid fits to the training data,
2. test data sets to compute the calibration quality.

The norm of the calibrated data was chosen as outcome measurement and is expected to be 1 g. Table 2 shows the average norms and their standard deviations. Please note that the method “manual calibration” is the calibration method aligning all axes in and against gravity by hand.

| data set | manual calibration | LinMinVar Estimate [6] | Non Linear [7] | LsqNon Linear [10] | Mgn Calibration [11] | MinVol Ellipse [5] | LinAuto Calibration |
|---|---|---|---|---|---|---|---|
| training set | extra training set | 1.0226 ± 0.0359 | 0.9996 ± 0.0126 | 1.0002 ± 0.0127 | 0.9999 ± 0.0127 | 0.9254 ± 0.0809 | 0.9996 ± 0.0128 |
| test set | 1.0078 ± 0.0425 | 1.0128 ± 0.0708 | 0.9960 ± 0.0399 | 1.0033 ± 0.0367 | 1.0063 ± 0.0361 | 1.0706 ± 0.1855 | 1.0036 ± 0.0387 |

**Table 2:** The norms of the calibrated data as measurement of the goodness of fit. The numbers represent the average norm ± standard deviation.

### 3.2. Execution Time

The execution time of the algorithms was measured by applying all training data sets. Table 3 shows the average execution time for each algorithm and its standard deviation after ten passes.

| | manual calibration | LinMinVar Estimate [6] | Non Linear [7] | LsqNon Linear [10] | Mgn Calibration [11] | MinVol Ellipse [5] | LinAuto Calibration |
|---|---|---|---|---|---|---|---|
| execution time in s | manual | 0.327 ± 0.010 | 0.504 ± 0.028 | 1.249 ± 0.006 | 0.198 ± 0.015 | 2.764 ± 0.043 | 0.037 ± 0.009 |

**Table 3:** The execution time in seconds of each algorithm. The numbers represent the average execution time ± standard deviation after ten passes.

### 3.3. Applying the Simulation Data

The estimation of the parameters $s$ and $o$ was evaluated using the simulation data. Table 4 summarizes the results of this comparison after ten passes to determine the average values and their standard deviations. For every pass a new point cloud with the same predefined parameters was generated.

### 3.4. Varying the Size of the Training Set

The quality of calibration strongly depends on the number of known (quasi-)static vectors representing as many different directions of the ellipsoid’s surface as possible. In real-life data the covered directions in accelerometer data is often biased by one specific direction and may not be optimal as shown in figure 1.

We decided to increasingly pass subsets of the training data exemplarily of subject #1 to the algorithms to determine the number of vectors needed for a stable calibration. Figure 2 shows the used uncalibrated training data set. Figure 3a-f shows the error in g caused by the algorithms regarding the vectors’ norm of the test set of subject #1. The amount of data needed to calibrate the data with MgnCalibration, LinMinVarEstimate and LsqNonLinear is higher compared to NonLinear and LinAutoCalibration. MinVolEllipsoid produced a considerably large error even when fed with all the training data, where the error for the first 40 measurements reached a peak error of 60 g. After 270 measurements the error still amounted to 0.8 g. The NonLinear calibration shows that in this case the algorithm converged after 100 input vectors and seemed to produce stable results. The LinAutoCalibration algorithm required about 270 measurements to produce stable results, where the error is lower than 0.1 g during this setup time and lower than 0.016 g after setup. The data of the other subjects produced comparable results.

| Co-ordinate | true value | LinMinVar Estimate [6] | Non Linear [7] | LsqNon Linear [10] | Mgn Calibration [11] | MinVol Ellipse [5] | LinAuto Calibration |
|---|---|---|---|---|---|---|---|
| $s_x$ | 500 | 499.81 ±0.62 | 499.83 ±0.62 | 499.77 ±0.62 | 499.90 ±0.59* | 490.45 ±7.01* | 499.83 ±0.61 |
| $s_y$ | 500 | 500.08 ±0.36 | 500.11 ±0.36 | 500.04 ±0.35 | 500.01 ±0.54* | 483.13 ±9.01* | 500.11 ±0.34 |
| $s_z$ | 500 | 500.12 ±0.38 | 500.14 ±0.38 | 500.07 ±0.38 | 499.82 ±0.45* | 525.66 ±6.22* | 500.14 ±0.38 |
| $o_x$ | 2300 | 2299.95 ±0.31 | 2299.95 ±0.31 | 2299.95 ±0.32 | 2299.98 ±0.30 | 2303.07 ±6.15 | 2299.99 ±0.39 |
| $o_y$ | 2300 | 2300.01 ±0.43 | 2300.01 ±0.42 | 2300.01 ±0.43 | 2299.99 ±0.44 | 2301.35 ±12.34 | 2299.99 ±0.39 |
| $o_z$ | 2300 | 2299.90 ±0.12 | 2299.90 ±0.12 | 2299.90 ±0.12 | 2299.92 ±0.14 | 2297.66 ±11.93 | 2299.91 ±0.12 |

**Table 4:** Estimation of the parameters $s$ and $o$ using the simulation data. The values indicated with a * were taken from the diagonal of matrix $U$. Please note that they are not directly comparable with the parameter $s$.

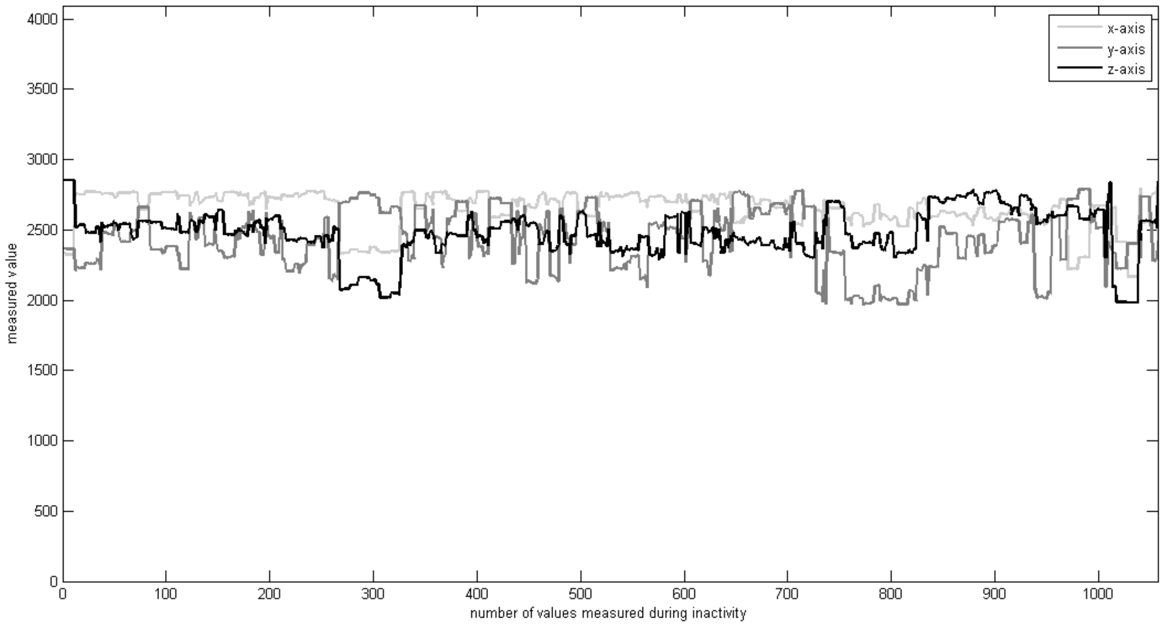


**Figure 2:** The uncalibrated data set used for an incremental training of the algorithms. The data shows only data of IS, where three associated values represent the coordinates of one vector in each case. Thus, if calibrated, the norm of all vectors shown in the figure is approximately 1 g.

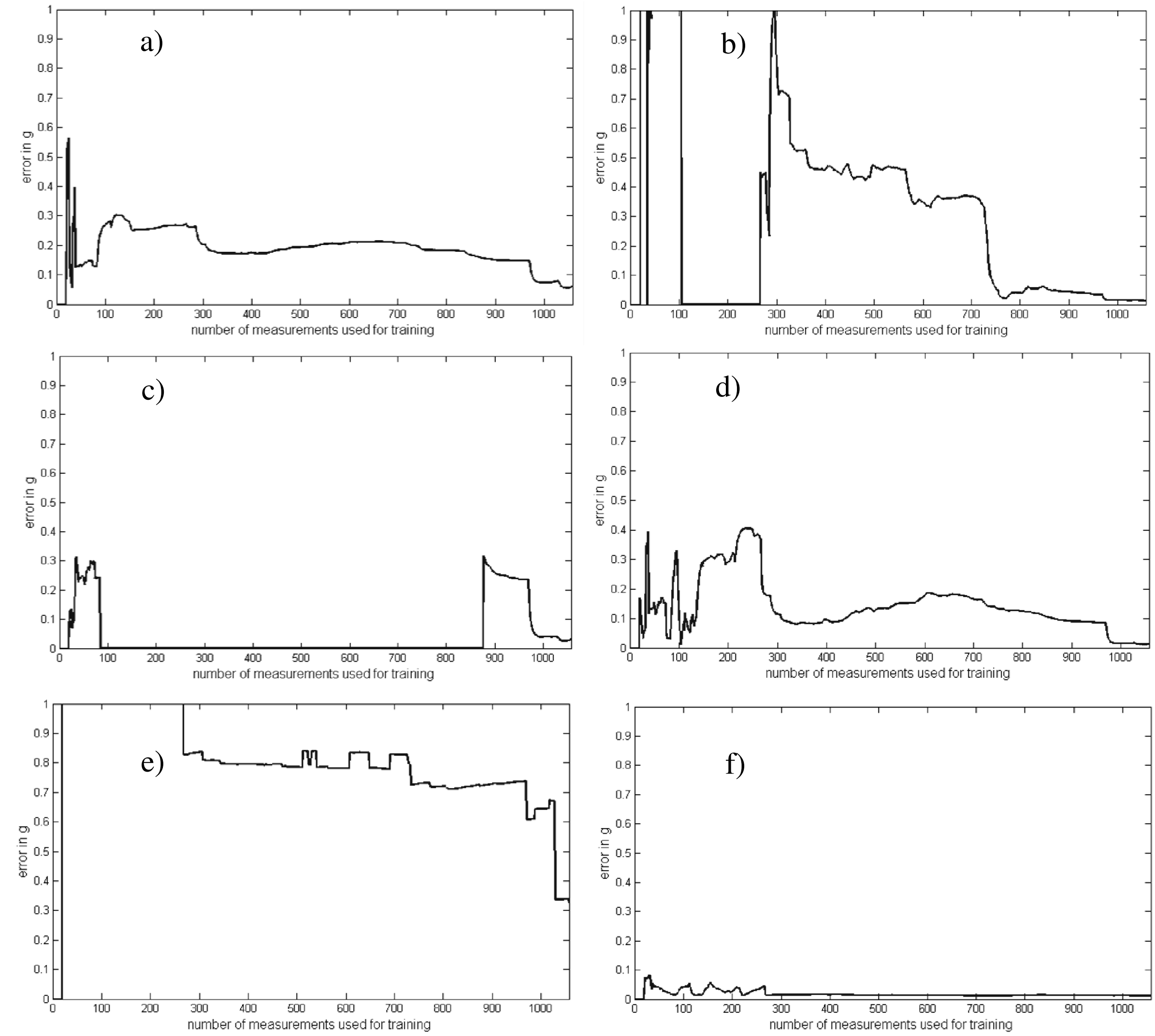


**Figure 3:** The error produced by the algorithms during the incremental training, exemplarily using the training and test data set of subject #1. The number of vectors passed to the algorithms, will be indicated by the x-axis. The error was calculated by the difference between the expected and the computed norm of the test set. a) shows the error produced by LinMinVarEstimate, b) MgnCalibration, c) NonLinear, d) LsqNonLinear, e) MinVolEllipsoid, and f) LinAutoCalibration.

### 3.5. Theoretical Study on Unequal Axes

Since the first step of LinAutoCalibration assumes that the ellipsoid's radii are approximately equal to each other, the influence of unequal ellipsoid axes on the estimation of the parameters and was explored. Therefore, ellipsoids with $N$ = 100 randomly selected points and known parameters and were constructed. Then, these parameters were reconstructed using the LinAutoCalibration algorithm. We scaled two of the components of with different multipliers. An independent point cloud with the same parameters was generated to test the estimated ellipsoid. The result showed that an axis ratio of 1:2:3 will produce an error of a maximum of 1%.

## 3.6. Theoretical study on misaligned axes

The influence of non-orthogonal axes has to be studied using simulation data, because all calibration algorithms based on equation 1 do not focus this aspect. Therefore, a point cloud was constructed that included a misalignment of the axes up to 2º. 100 randomly selected points were chosen as training and test data set, respectively. The results showed that the error of nearly all algorithms was at a maximum 0.8%, except MgnCalibration which compensated the deformations down to 0.45%. In contrast, MinVolEllipsoid produced an error of 3.7%.

## 4. Discussion

All considered algorithms showed in the study data a very good performance regarding the norm and its standard deviation as quality criterion. Most of the algorithms featured better results than the manual calibration, which demands an extra calibration data set. The only algorithm those estimations were inaccurate was MinVolEllipsoid. The reason for that may be that the data was too noisy for MinVolEllipsoid.

The major difference between all the algorithms is the execution time. MinVolEllipsoid had the longest and LinAutoCalibration the shortest execution time. The NonLinear calibration method performed with a fair execution time, although it is based on a simple approach solving equation 1 directly using the *fsolve* command of MATLAB. A disadvantage of this algorithm (as for LinMinVarEstimate or LsqNonLinear) is the need for a-priori-knowledge about the ellipsoid parameters.

LsqNonLinear algorithm showed a longer execution time compared to NonLinear. This method needed four to six iteration steps for training with a-priori chosen initial values and four to eight iteration steps with the calculation rule suggested by Campolo et al.. The average time needed to compute the initial values was 0.015s. The calculation rule seemed to be a simple and good approximation.

In the estimation of known parameters using simulation data all algorithms performed very well and provided reliable results. Thereby, the interpretation of the results of MinVolEllipsoid is difficult. Although the center of the ellipsoid was estimated with fair accuracy, the radii showed notable deviations. But the values of the radii were directly taken from the diagonal of the matrix $U$ and cannot be compared directly with the parameter $s$.

Varying the size of the training set available for the calibration algorithms also showed differences in the error of estimating the norm of the test data. NonLinear and LinAutoCalibration showed the best results with a high stability after a short setup time. The other algorithms showed that they need a larger amount of data for setup. The estimation bias of all algorithms may primarily depend on the data that is biased to a single direction as seen in figure 1.

The results of the theoretical study on unequal axes showed that LinAutoCalibration is also able to handle such data, whereas the average relative error between the true and the estimated parameters $s$ and $o$ was lower than 1% in a radii ratio of 1:2:3. Furthermore, moving the ellipsoid (i.e., changing one or more of the components of $o$) does not influence any of the estimated parameters.

The theoretical study on misalignment axes produced a moderate error in all algorithms. The only algorithm that was able to compensate the artificially generated misalignment slightly better than all the other considered algorithms was MgnCalibration.

Unfortunately, it was impossible to assess the memory consumption of the algorithms in MATLAB. We tried different approaches (e.g. via ‘profile –memory on’ or via ‘memory’ command), but the values returned were neither plausible nor reliable.

Additionally, we implemented the algorithm presented in this paper in nesC [19] for TinyOS 1.x [20] on the Shimmer platform [15]. It showed that it is possible to calibrate the measurements on-the-fly while recording data at least with a sample rate of 256 Hz without losing any data. The memory footprint of the algorithm amounts 824 bytes and takes up 7348 bytes of flash memory. The execution time on this platform was measured with 50 milliseconds using 100 different vectors for calibration.

## 5. Conclusion

In this paper we presented a fast and stable non-iterative 3D-ellipsoid fitting method. This method can be used to calibrate accelerometer measurements and does not need any knowledge about the data. For comparison purposes with other approaches we used real-life data from an experimental study of younger, healthy subjects who performed their normal daily activities. We compared both the estimation of the ellipsoid parameters and the computation time and showed that almost all algorithms performed well. But the main difference between the algorithms was the execution time, whereby the algorithm presented in this paper was about 10 times faster compared to the other algorithms. The main applications of the method are if there is a need for a fast and memory preserving calibration method, especially when using sensor nodes equipped with tri-axial accelerometers.

In theoretical studies we noticed that almost all considered algorithms were able to estimate the ellipsoid parameters very well and that all these algorithms produced small to moderate errors in non-orthogonal axes. LinAutoCalibration needed a relative small amount to produce robust and stable results.

We noticed that the amount of values measured for the period of inactivity during the normal course of life are sufficient for stable and good accelerometer calibration results in each case.

### 5.1. Limitations

The algorithm introduced in this paper requires data that represent an ellipsoid with axes that are approximately of the same size. Using a single 3D-accelerometer may meet the requirement in most cases, but probably not when separated into a 2D- and a 1D-sensor, if the sensor sensitivities or the output voltage ranges will differ (cf. [10]). In such cases, either the sensor specifics have to be adapted by means of additional hard- or software or MgnCalibration should be used.

## 6. Acknowledgement

The authors would like to thank Dr. Wolfgang Marten and Dr. Martin Kohlmann for their valuable help. The Lower Saxony research network “Design of Environments for Ageing” acknowledges the support of the Lower Saxony Ministry of Science and Culture through the "Niedersächsisches Vorab" grant programme (grant ZN 2701).

**Conflicts of Interest:** None to declare.

## 7. References

[1] S. Koch, M. Marschollek, K.H. Wolf, M. Plischke, R. Haux, On health-enabling and ambient-assistive technologies. What has been achieved and where do we have to go?, Methods of Information in Medicine 48 (2009) 29-37.

[2] B. Arnrich, O. Mayora, J. Bardram, G. Tröster, Pervasive healthcare: paving the way for a pervasive, user-centered and preventive healthcare model, Methods of Information in Medicine 49 (2010) 67-73.

[3] W. Zijlstra, K. Aminian, Mobility assessment in older people: new possibilities and challenges, European Journal of Ageing 4 (2007) 3-12.

[4] K.M. Culhane, M. O'Connor, D. Lyons, G.M. Lyons, Accelerometers in rehabilitation medicine for older adults, Age Ageing 34 (2005) 556-560.

[5] S. Bonnet, C. Bassompierre, C. Godin, S. Lesecq, A. Barraud, Calibration methods for inertial and magnetic sensors, Sensors and Actuators A: Physical 156 (2009) 302-311.

[6] J.C. Lötters, J. Schipper, P.H. Veltink, W. Olthuis, P. Bergveld, Procedure for in-use calibration of triaxial accelerometers in medical applications, Sensors and Actuators A: Physical 68 (1998) 221-228.

[7] P. Lukowicz, H. Junker, G. Tröster, Automatic calibration of body worn acceleration sensors, Pervasive Computing, in Lecture Notes in Computer Science, eds. A. Ferscha, F. Mattern, pp. 176-181 (Springer, Berlin/Heidelberg, 2004).

[8] K. Levenberg, A method for the solution of certain problems in least-squares, Quarterly Applied Mathematics 2 (1944) 164-168.

[9] D. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, SIAM Journal Applied Mathematics 11 (1963) 431-441.

[10] D. Campolo, M. Fabris, G. Cavallo, D. Accoto, F. Keller, E. Guglielmelli, A novel procedure for in-field calibration of sourceless inertial/magnetic orientation tracking wearable devices, in Proceedings of the IEEE / RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics (BIOROB), eds. P Dario, D. Meldrum, pp. 471-476 (2006).

[11] J.M.G. Merayo, P. Brauer, F. Primdahl, J.R. Petersen, O.V. Nielsen, Scalar calibration of vector magnetometers, Measurement in Science and Technology 11 (2000) 120-132.

[12] A. Barraud, S. Lesecq, Magnetometers calibration. Internet: http://www.mathworks.com/matlabcentral/fileexchange/23398-magnetometers-calibration, last updated: 23 Mar 2009, [last accessed 09 Jul 2010].

[13] M. Todd, E. Yildirim, On Khachiyan's algorithm for the computation of minimum-volume enclosing ellipsoids, Discrete Applied Mathematics 155 (2007) 1731–1744.

[14] N. Moshtagh, Minimum volume enclosing ellipsoid, Internet: http://www.mathworks.de/matlabcentral/fileexchange/9542-minimum-volume-enclosing-ellipsoid, last updated: 20 Jan 2009, [last accessed 09 Jul 2010].

[15] Shimmer – Wireless Sensor Platform for Wearable Applications, Internet: http://www.shimmer-research.com, [last accessed 13 Jul 2010].

[16] MMA7260QT Product Summery Page, Internet: http://cache.freescale.com/files/sensors/doc/data_sheet/MMA7260QT.pdf?pspll=1, last updated: Rev5, Mar 2008, [last accessed 14 Jul 2010].

[17] N. Moshtagh, Plot an ellipse in "center form". Internet: http://www.mathworks.de/matlabcentral/fileexchange/13844, last updated: 16 Nov 2007, [last accessed 09 Jul 2010].

[18] T.F. Coleman, Y. Li, On the convergence of reflective newton methods for large-scale nonlinear minimization subject to bounds, Mathematical Programming 67 (2004) 189-224.

[19] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, D. Culler, The nesC language: A holistic approach to networked embedded systems, in Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation, pp. 1-11 (ACM, New York, NY, 2003).

[20] P. Levis, S. Madden, J. Polastre, R. Szewczyk, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, D. Culler, TinyOS: An operating system for sensor networks, in: Ambient Intelligence, eds. W. Weber, J. Rabaey, E. Aart, pp. 115-148 (Springer, Heidelberg, 2004).