Online Auto-calibration of Triaxial Accelerometer with Time-variant Model Structures

L. Ye[a], A. Argha[b], B. G. Celler[b], H. T. Nguyen[a], S. W. Su[a,*]

[a] *Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia.*
[b] *School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia.*

Abstract

In this paper, an online auto-calibration method for MicroElectroMechanical Systems (MEMS) triaxial accelerometer (TA) is proposed, which can simultaneously identify the time-dependent model structure and its parameters during the changes of the operating environment. Firstly, the model as well as its associated cost function is linearized by a new proposed linearization approach. Then, exploiting an online sparse recursive least square (SPARLS) estimation, the unknown parameters are identified. In particular, the online sparse recursive method is based on an $\mathcal{L}_1$-norm penalized expectation-maximum (EM) algorithm, which can amend the model automatically by penalizing the insignificant parameters to zero. Furthermore, this method can reduce computational complexity and be implemented in a low-cost Micro-Controller-Unit (MCU). Based on the numerical analysis, it can be concluded that the proposed recursive algorithm can calculate the unknown parameters reliably and accurately for most MEMS triaxial accelerometers available in the market. Additionally, this method is experimentally validated by comparing the output estimations before and after calibration under various scenarios, which further confirms its feasibility and effectiveness for online TA calibration.

[*] Corresponding author
Email addresses: Lin.Ye@uts.edu.au (L. Ye), a.argha@unsw.edu.au (A. Argha), b.celler@unsw.edu.au (B. G. Celler), Hung.Nguyen@uts.edu.au (H. T. Nguyen), Steven.Su@uts.edu.au (S. W. Su)

Keywords: Parameter estimation, Tri-axial accelerometer, Online calibration, Model linearization, Expectation maximization algorithm

1. Introduction

With the rapid development of the MicroElectroMechanical Systems (MEMS) technology, recently, the chip-based triaxial accelerometer (TA) improved significantly in terms of performance and power consumption. Meanwhile, the accelerometer integrates with more modules and keeps reducing the size, which makes it flexible to be applied in different applications among several areas. As evidences, these sensors have already been extensively utilized in wearable health monitoring devices [1],[2],[3],[4] motion tracking systems [5],[6],[7] and consumer electronic devices [8],[9] including smart phone and smart watch.

Due to the limitation of MEMS technology, the MEMS accelerometer still suffers from bias instability, noisy output and daily drift. To remedy these deficiencies, the calibration of MEMS accelerometers is a necessary step prior to application of appliance. To calibrate the MEMS accelerometer, conventional calibration methods [10] need to know the exact orientation of the accelerometer. It requires sophisticated equipment (e.g., rotary table) to obtain the precise orientation, which is hard to be accessed by the majority of users. Recently, several papers [11],[12],[13],[14] proposed a new idea for the calibration of triaxial accelerometer, referred to as auto-calibration, which is suitable for implementation without sophisticated laboratory equipments. Furthermore, [15],[16],[17] consummated the auto-calibration by considering the quality of individual calibration, in which the selection of experimental observations is well discussed based on Design of Experiment (DoE). However, the output of MEMS accelerometer still suffers from drifting caused by ambient temperature [18],[19]. If the accelerometer is intended to be used in multiple environments with sharp temperature variations, the user may need to frequently re-calibrate their accelerometers to overcome the variation of calibrated parameters. Therefore, an online calibration is desired to improve the measurement accuracy.

To the authors' best knowledge, most studies, including [11],[14],[20],[21],[22] focus on offline TA calibration. Only, a few papers [23],[24] are devoted to online calibration, but the methods proposed are mainly based on classical calibration methods that constructed by employing Kalman filter technique. This paper firstly introduces a new linearization method based on the most commonly used 9-parameter auto-calibration model [13],[16]. After that, a sparse recursive least square (SPARLS) estimation method [25] is utilized to solve the unknown parameters. Particularly, we demonstrate that this method is effective while the model parameters are varying. Furthermore, based on the characteristic of $\mathcal{L}_1$-norm penalized expectation-maximum (EM) algorithm, this method can automatically determine the model complexity in an optimal manner. In addition, this method has successfully been implemented in an embedded Micro Control Unit (MCU) for online testing. Both simulation and experimental results are provided which show the effectiveness of the proposed approach.

The major contributions of the paper can be summarized as follows. Firstly, it is the first time that the SPARLS algorithm is applied to solve a special nonlinear parameter estimation problem for the auto-calibration of TA, which is non-convex in nature. Secondly, the proposed approach is able to accurately estimate the significant TA parameters in real-time while penalizing the insufficient parameters converging to zero. Thirdly, the convergence condition of the iterative approach has been identified and investigated based on vast numerical simulations. Finally, the effectiveness of the approach has been demonstrated by both simulation and real-time experiment.

~~This paper is organized as follows. In section 2, the linearization method for 9-parameter model is introduced. In section 3, the online estimation method is presented based on the linearized model. In section 4, both the simulation and experiment are presented. Section 5 concludes the paper.~~

2. Linearization of TA 9-parameter model

For the autocalibration of triaxial accelerometer, 6-, 9- and 12-parameter models are widely selected by researchers. In [22], the authors demonstrate that for most MEMS accelerometers, 6- and 9-parameter models are accurate enough. For this reason, as the proposed method can automatically adjust the number of unknown parameters, the 9-parameter model is selected. Let us define $\boldsymbol{V} = [v_x, v_y, v_z]^T$ as the measurement from accelerometer of each axis and $\boldsymbol{A} = [a_x, a_y, a_z]^T$ as the true local acceleration. The relationship between the measurement $\boldsymbol{V}$ and the true value $\boldsymbol{A}$ can be expressed as:

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{S} \cdot \boldsymbol{T} \cdot (\boldsymbol{V} + \boldsymbol{O}) + \boldsymbol{\varsigma} \\ &= \boldsymbol{K} \cdot (\boldsymbol{V} + \boldsymbol{O}) + \boldsymbol{\varsigma}, \end{aligned} \tag{1}$$

where $\boldsymbol{O} = [o_x, o_y, o_z]^T$ represents the offset vector, $\boldsymbol{\varsigma}$ is a zero mean white noise vector, and $\boldsymbol{S}$ represents the scale factor matrix:

$$\boldsymbol{S} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix}, \tag{2}$$

where S_x, S_y and S_z denote sensitivity factor for each axis. $\boldsymbol{T}$ is described as:

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & 0 \\ \phi_{xy} & 1 & 0 \\ \phi_{xz} & \phi_{yz} & 1 \end{bmatrix}, \tag{3}$$

where ϕ_{xy}, ϕ_{xz} and ϕ_{yz} denote error in the alignment of the three single axis sensors of a complete three-axis accelerometer. $\boldsymbol{K}$ is the product of $\boldsymbol{S}$ and $\boldsymbol{T}$:

$$\boldsymbol{K} = \begin{bmatrix} S_x & 0 & 0 \\ S_x\phi_{xy} & S_y & 0 \\ S_z\phi_{xz} & S_z\phi_{yz} & S_z \end{bmatrix} \triangleq \begin{bmatrix} k_{xx} & 0 & 0 \\ k_{xy} & k_{yy} & 0 \\ k_{xz} & k_{yz} & k_{zz} \end{bmatrix}. \tag{4}$$

The matrix $\boldsymbol{K}$ can be considered as the completed scale factor matrix, where diagonal elements (k_{xx}, k_{yy}, k_{zz}) and off-diagonal (k_{xy}, k_{xz}, k_{yz}) are sensitivity scale factors and misalignment elements respectively.

Based on this model, classical method [10] can solve unknown scale factors and offsets from $\boldsymbol{K}$ and $\boldsymbol{O}$ directly, but it normally requires high precision equipment which is hardly accessed by most users. Thus, auto-calibration is developed and widely used for TA calibration. The idea of auto-calibration method is that the measurements of triaxial accelerometer should be equal to local gravity in static state, i.e.,

$$g = \sqrt{a_x^2 + a_y^2 + a_z^2}. \tag{5}$$

Based on (5), the error $\tilde{\epsilon}_i$ for the i-th measurement can be expressed as follows [13],

$$\tilde{\epsilon}_i = a_{x,i}^2 + a_{y,i}^2 + a_{z,i}^2 - g^2 \quad i = 1, 2, \cdots, n, \tag{6}$$

where n is the number of total measurements.

Then, with Eq.(1) and Eq.(6), if assume that $\boldsymbol{\beta}$ is the vector of the unknown ~~model~~ parameters, the cost function can be often defined as:

$$J(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left(||f_{\boldsymbol{\beta}}(\boldsymbol{V}_i) - g^2|| \right), \tag{7}$$

where $\boldsymbol{V}_i = [v_{x,i}, v_{y,i}, v_{z,i}]^T$ $(i = 1, 2, \cdots, n)$ is the i-th measurement, and $f_{\boldsymbol{\beta}}(\boldsymbol{V}_i)$ is a scalar function of $\boldsymbol{V}_i$. The parameter estimation can then be formulated as a nonlinear non-convex optimization problem:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \; J(\boldsymbol{\beta}). \tag{8}$$

In real life situation, to minimize the effect of environmental temperature, daily drift and so on, some online calibration methods are proposed. Several optimization methods [23],[24],[26] are selected to solve this problem based on unscented Kalman filter (UKF) [23],[26] or extended Kalman filter [24]. However, most of these methods are not based on auto-calibration; precise orientation information is still required. Although existing off-line methods are able to identify the unknown parameters, it will be inconvenient for users to re-calibrate frequently.

To achieve online calibration by inertial measurement unit (IMU), we propose a method to linearize the cost function to decrease computational complexity. According to [22], $\tilde{\epsilon}_i$ in Eq.(6) is the summation of zero mean Gaussian distribution and Chi-square distribution. Comparing to the Gaussian distribution component, the Chi-squared distribution part is negligible [22]. Hence, the error $\tilde{\epsilon}_i$ can be approximated as normal distribution with zero mean. Based on Eq.(1) and Eq.(5), the squared form of auto-calibration method of the i-th measurement is given by:

$$\begin{aligned} g^2 = \; & \Big[k_{xx}\left(v_{x,i}+o_x\right)\Big]^2 + \Big[k_{xy}\left(v_{x,i}+o_x\right)+k_{yy}\left(v_{y,i}+o_y\right)\Big]^2 \\ & + \Big[k_{xz}\left(v_{x,i}+o_x\right)+k_{yz}\left(v_{y,i}+o_y\right)+k_{zz}\left(v_{z,i}+o_z\right)\Big]^2 + \epsilon_i, \end{aligned} \tag{9}$$

where ϵ_i is a zero mean white noise. Apparently, Eq.(9) is nonlinear for unknown parameters $k_{xx}, k_{yy},\ k_{zz}, k_{xy}, k_{xz},\ k_{yz}, o_x, o_y, o_z$. Recently, the MEMS packing and soldering technologies have been improved rapidly, leading to the reduction of the value of the undesired parameters to a very low level. The square or product terms of the off-diagonal elements of $\boldsymbol{K}$ and offset $\boldsymbol{O}$ are close to zero ($k_{ij} < 0.01$ and $O_i < 0.1$). To linearize Eq.(9), we can neglect some terms which have small impact during estimation. Let us expand Eq.(9), so that several terms contain at least the square of $k_{xy}, k_{xz}, k_{yz}, o_x, o_y, o_z$ or their products. Furthermore, even for those terms with measurements $(v_{x,i}, v_{y,i}, v_{z,i})$, the maximum value of measurements is $1g$ during calibration. To simply the online estimation procedure, let us replace the sum of these terms as ψ_i for the i-th measurement, and we will estimate ψ_i iteratively during online estimating procedure. Eq.(9) can then be rewritten as:

$$\begin{aligned} g - \psi_i = & k_{xx}^2 v_{x,i}^2 + 2k_{xx}^2 v_{x,i} o_x + 2k_{xy} k_{yy} v_{x,i} v_{y,i} + k_{yy}^2 v_{y,i}^2 + 2k_{yy}^2 v_{y,i} o_y \\ & + k_{zz}^2 v_{z,i}^2 + 2k_{zz}^2 v_{z,i} o_z + 2k_{xz} k_{zz} v_{x,i} v_{z,i} + 2k_{yz} k_{zz} v_{y,i} v_{z,i} + \epsilon_i, \end{aligned} \tag{10}$$

where the residual ψ_i is assumed as a constant during each iteration, expressed as:

$$
\begin{aligned}
\psi_i = &\big(k_{xy}(v_{x,i}+o_x)\big)^2 + \big(k_{xz}(v_{x,i}+o_x)+k_{yz}(v_{y,i}+o_y)\big)^2 + k_{xx}^2 o_x^2 + k_{yy}^2 o_y^2 + k_{zz}^2 o_z^2 \\
&+ 2k_{xy}k_{yy}(v_{x,i}o_y + v_{y,i}o_x + o_x o_y) + 2k_{xz}k_{zz}(v_{x,i}o_z + v_{z,i}o_x + o_x o_z) \\
&+ 2k_{yz}k_{zz}(v_{y,i}o_z + v_{z,i}o_y + o_y o_z).
\end{aligned}
\tag{11}
$$

After re-arranging the original parameters, a new set of parameters can be defined as:

$$
\begin{cases}
\beta_1 = 2k_{xx}^2 o_x \\
\beta_2 = 2k_{yy}^2 o_y \\
\beta_3 = 2k_{zz}^2 o_z \\
\beta_4 = 2k_{xy}k_{yy} \\
\beta_5 = 2k_{xz}k_{zz} \\
\beta_6 = 2k_{yz}k_{zz} \\
\beta_7 = k_{xx}^2 \\
\beta_8 = k_{yy}^2 \\
\beta_9 = k_{zz}^2
\end{cases}
\tag{12}
$$

Thus, Eq.(10) can be rewritten as:

$$
\begin{aligned}
g - \psi_i = &\beta_1 v_{x,i} + \beta_2 v_{y,i} + \beta_3 v_{z,i} + \beta_4 v_{x,i}v_{y,i} + \beta_5 v_{x,i}v_{z,i} + \beta_6 v_{y,i}v_{z,i} + \beta_7 v_{x,i}^2 \\
&+ \beta_8 v_{y,i}^2 + \beta_9 v_{z,i}^2 + \epsilon_i.
\end{aligned}
\tag{13}
$$

Assuming that we have n sets of measurement data, Eq.(13) can be rearranged into a matrix form as:

$$
\begin{aligned}
\boldsymbol{g}_n - \boldsymbol{\psi}_n &= \boldsymbol{X}_n \boldsymbol{\beta}_n + \boldsymbol{\epsilon}_n \\
\boldsymbol{y}_n &= \boldsymbol{X}_n \boldsymbol{\beta}_n + \boldsymbol{\epsilon}_n,
\end{aligned}
\tag{14}
$$

where the i-th rows of $\boldsymbol{g}_n$, $\boldsymbol{\psi}_n$, $\boldsymbol{X}_n$ are g, ψ_i and $[v_{x,i}, v_{y,i}, v_{z,i}, v_{x,i}v_{y,i}, v_{x,i}v_{z,i}, v_{y,i}v_{z,i}, v_{x,i}^2, v_{y,i}^2, v_{z,i}^2]$, $i \in \{1, 2, \cdots, n\}$ respectively. Vector $\boldsymbol{\beta}_n = [\beta_{1,n}, \beta_{2,n}, \beta_{3,n}, \beta_{4,n}, \beta_{5,n}, \beta_{6,n}, \beta_{7,n}, \beta_{8,n}, \beta_{9,n}]^T$.

Therefore, for the i-th measurement, the instantaneous error can be expressed as:

$$\begin{aligned} e_i &= y_i - \hat{y}_i \\ &= y_i - \boldsymbol{x}_i \hat{\boldsymbol{\beta}}_n, \end{aligned} \tag{15}$$

where $\hat{\boldsymbol{\beta}}_n$ is the estimated unknown parameters of n times measurements. Hence, we have the following optimization problem:

$$\hat{\boldsymbol{\beta}}_n = \arg\min_{\boldsymbol{\beta}_n} \quad f_{\boldsymbol{\beta}}(e_1, e_2, \cdots, e_n). \tag{16}$$

With a non-negative forgetting factor λ, which is a non-negative constant scalar, the alternative cost function of Eq.(16) can be adjusted as:

$$J(\boldsymbol{\beta}_n) = \sum_{i=1}^{n} \lambda^{n-i} |e_i|^2, \tag{17}$$

which is a well-known cost function that can be solved directly by Recursive Least Squares(RLS) method. To convert Eq.(17) into matrix form, let us define:

$$\boldsymbol{D}_n = diag(\lambda^{n-1}, \lambda^{n-2}, \cdots, 1), \tag{18}$$

$$\boldsymbol{y}_n = [y_1, y_2, \cdots, y_n]^T, \tag{19}$$

and

$$\boldsymbol{X}_n = \begin{pmatrix} \boldsymbol{x}_1^T \\ \boldsymbol{x}_2^T \\ \vdots \\ \boldsymbol{x}_n^T \end{pmatrix}, \tag{20}$$

where $\boldsymbol{x}_i^T$ (the i-th row of $\boldsymbol{X}_n$) is $[v_{x,i}, v_{y,i}, v_{z,i}, v_{x,i}v_{y,i}, v_{x,i}v_{z,i}, v_{y,i}v_{z,i}, v_{x,i}^2, v_{y,i}^2, v_{z,i}^2]$. Therefore, the RLS cost function can be written in the following form:

$$J(\boldsymbol{\beta}_n) = ||\boldsymbol{D}_n^{1/2}\boldsymbol{y}_n - \boldsymbol{D}_n^{1/2}\boldsymbol{X}_n\hat{\boldsymbol{\beta}}_n||_2^2, \tag{21}$$

where the i-th diagonal element $d_{ii,n}^{1/2}$ of $\boldsymbol{D}_n^{1/2}$ is the squared root of the i-th diagonal element $d_{ii,n}$ of $\boldsymbol{D}_n$, as:

$$d_{ii,n}^{1/2} = \sqrt{d_{ii,n}}. \tag{22}$$

Although Recursive Least Square (RLS) method can be applied to solve the unknown parameter based on cost function (21), it can not consider the physical characteristic of MEMS accelerometer. As we know, for the matrix $\boldsymbol{K}$ in (4), the misalignment error may be quite small due to the improvement of MEMS technologies. Therefore, instead of applying Akaike information Criterion (AIC) [22] to determine the true model of a specific accelerometer, we can adopt a spares recursive least square (SPARLS) [25] method to solve the unknown parameter. In practice, this method can play a role in reducing the number of undesired parameters. With a sparse RLS, the model can automatically be selected and updated in a real-time manner. Based on Eq.(21), the cost function of linearized auto-calibration model with $\mathcal{L}_1$ regularizer can be described as follows:

$$\min_{\hat{\boldsymbol{\beta}}_n} \frac{1}{2\sigma^2}||\boldsymbol{D}_n^{1/2}\boldsymbol{y}_n - \boldsymbol{D}_n^{1/2}\boldsymbol{X}_n\hat{\boldsymbol{\beta}}_n||_2^2 + \gamma||\hat{\boldsymbol{\beta}}_n||_1, \tag{23}$$

where σ is the standard deviation of noise, γ represents a trade off between estimation error and sparsity of the parameter, and the term $\gamma||\hat{\boldsymbol{\beta}}_n||_1$ can be considered as $\mathcal{L}_1$ regularization. Typically, for $\mathcal{L}_1$ regularization, the penalty term can reduce the overfitting which can lead to the shrinkage of unknown parameters [27].

3. Online calibration method for Linearized 9-parameter model

After obtaining a linearize auto-calibration model of TA and the corresponding cost function with $\mathcal{L}_1$-norm regularization (23), the parameter estimation problem can be solved by various optimization methods. Then, we can solve the original parameter $\boldsymbol{K}$ and $\boldsymbol{O}$ of TA model (1) based on (4) and (12). Here, we adopt a recursive sparse method from [25], which can remove insignificant parameters and significantly reduce the computational complexity. With some modifications, this method can estimate the removal term $\boldsymbol{\psi}_n$ recursively, which results in the reduction of the bias and thereby achieving an accurate estimation accuracy.

To apply this method, let us recall the linearized TA model (14) with n

observations:

$$\boldsymbol{y}_n = \boldsymbol{X}_n \boldsymbol{\beta}_n + \boldsymbol{\xi}_n, \tag{24}$$

where for online estimation, the error term can be adjusted to $\boldsymbol{\xi}_n \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{D}_n^{-1})$.

For Eq.(24), to solve $\hat{\boldsymbol{\beta}}_n$ without undeserved parameters, penalized log-likelihood estimation $\log p(\boldsymbol{y}_n|\boldsymbol{\beta}_n) - pen(\boldsymbol{\beta}_n)$ can be applied where *pen* represents penalty. However, for penalized log-likelihood estimation, in general, it is hard to be maximized directly [28]. Therefore, a penalized expectation-maximum (EM) algorithm is adopt to maximize the penalized complete log-likelihood. To estimated $\boldsymbol{\beta}_n$ with penalized expectation-maximum problem, we decompose $\boldsymbol{\xi}_n$ as:

$$\boldsymbol{\xi}_n = \alpha \boldsymbol{X}_n \boldsymbol{\tau}_n + \boldsymbol{D}_n^{-1/2} \boldsymbol{\delta}_n, \tag{25}$$

where α is a positive parameter. $\boldsymbol{\tau}_n$ and $\boldsymbol{\delta}_n$ are independent noise such that:

$$\begin{aligned} &\boldsymbol{\tau}_n \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \\ &\boldsymbol{\delta}_n \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I} - \alpha^2 \boldsymbol{D}_n^{1/2} \boldsymbol{X}_n \boldsymbol{X}_n^T \boldsymbol{D}_n^{1/2}). \end{aligned} \tag{26}$$

To guarantee that $\boldsymbol{\delta}_n$ has a positive semi-definite covariance matrix, we need to make sure that $\alpha^2 \leq \sigma^2/\lambda_1$, where λ_1 is the largest eigenvalue of $\boldsymbol{D}_n^{1/2} \boldsymbol{X}_n \boldsymbol{X}_n^T \boldsymbol{D}_n^{1/2}$.

Then, based on (25) and (26), Eq.(24) can be rearranged as:

$$\begin{cases} \boldsymbol{v}_n & = \boldsymbol{\beta}_n + \alpha \boldsymbol{\tau}_n \\ \boldsymbol{y}_n & = \boldsymbol{X}_n \boldsymbol{v}_n + \boldsymbol{D}_n^{-1/2} \boldsymbol{\delta}_n \end{cases} \tag{27}$$

where $\boldsymbol{v}_n$ is the hidden variable for penalized EM algorithm.

It is easy to verify that the variance of $\boldsymbol{y}_n$ from Eq.(27) is:

$$\begin{aligned} \mathrm{Var}[\boldsymbol{y}_n] &= \alpha^2 \boldsymbol{X}_n \boldsymbol{X}_n^T + \boldsymbol{D}_n^{-1/2} \left(\sigma^2 \boldsymbol{I} - \alpha^2 \boldsymbol{D}_n^{1/2} \boldsymbol{X}_n \boldsymbol{X}_n^T \boldsymbol{D}_n^{1/2} \right) \boldsymbol{D}_n^{-1/2} \\ &= \sigma^2 \boldsymbol{D}_n^{-1}, \end{aligned} \tag{28}$$

where it is the same for the variance of $\boldsymbol{y}_n$ from Eq.(24).

Thereby, the *complete penalized log-likelihood* of penalized expectation maximum algorithm can be expressed as $\log p(\boldsymbol{y}_n, \boldsymbol{v}_n|\boldsymbol{\beta}_n) - pen(\boldsymbol{\beta}_n)$ which can be achieved in two steps. For the expectation step, the condition expectation is calculated. For the maximization step, the parameters are computed to maximize condition expectation [28]. The specific penalized recursive EM method, which can be applied for solving our particular problem, can be encapsulated in the following two steps:

- E-step: Calculate the conditional expectation $\log p(\boldsymbol{y}_n|\boldsymbol{\beta}_n)$ for Eq.(23), defined as Q-function $Q(\boldsymbol{\beta}_n, \hat{\boldsymbol{\beta}}_n^{(l)})$ in the l-th iteration:

$$Q(\boldsymbol{\beta}_n, \hat{\boldsymbol{\beta}}_n^{(l)}) = E\left[\log p(\boldsymbol{y}_n, \boldsymbol{v}_n|\boldsymbol{\beta}_n)|\boldsymbol{y}_n, \hat{\boldsymbol{\beta}}_n^{(l)}\right], \tag{29}$$

- M-step: Updates the estimated $\hat{\boldsymbol{\beta}}_n^{(l+1)}$ based on:

$$\hat{\boldsymbol{\beta}}_n^{(l+1)} = \arg\max_{\boldsymbol{\beta}_n}\left(Q(\boldsymbol{\beta}_n, \hat{\boldsymbol{\beta}}_n^{(l)}) - pen(\hat{\boldsymbol{\beta}}_n^{(l)})\right). \tag{30}$$

Considering the second equation of Eq.(27), when $\boldsymbol{v}_n$ is known, $\boldsymbol{y}_n$ can be directly estimated without necessarily known $\boldsymbol{\beta}_n$, i.e., in this case, $\boldsymbol{y}_n$ is independent of $\boldsymbol{\beta}_n$. Therefore, the complete likelihood $p(\boldsymbol{y}_n, \boldsymbol{v}_n|\boldsymbol{\beta}_n)$ can be simplified as:

$$\begin{aligned} p(\boldsymbol{y}_n, \boldsymbol{v}_n|\boldsymbol{\beta}_n) &= p(\boldsymbol{y}_n|\boldsymbol{v}_n, \boldsymbol{\beta}_n)p(\boldsymbol{v}_n|\boldsymbol{\beta}_n) \\ &= p(\boldsymbol{y}_n|\boldsymbol{v}_n)p(\boldsymbol{v}_n|\boldsymbol{\beta}_n). \end{aligned} \tag{31}$$

Then, if we define that $\Sigma_y = \sigma^2\boldsymbol{D}_n^{-1} - \alpha^2\boldsymbol{X}_n\boldsymbol{X}_n^T$, the complete log-likelihood $\log p(\boldsymbol{y}_n, \boldsymbol{v}_n|\boldsymbol{\beta}_n)$ can be computed as:

$$\begin{aligned} \log p(\boldsymbol{y}_n, \boldsymbol{v}_n|\boldsymbol{\beta}_n) &= \log p\left((\boldsymbol{y}_n|\boldsymbol{v}_n)p(\boldsymbol{v}_n|\boldsymbol{\beta}_n)\right) \\ &= \log\left\{\frac{1}{2\pi^{n/2}||\Sigma_y||^{1/2}}\exp\left(-\frac{(\boldsymbol{y}_n - \boldsymbol{X}_n\boldsymbol{v}_n)^T\Sigma_y^{-1}(\boldsymbol{y}_n - \boldsymbol{X}_n\boldsymbol{v}_n)}{2}\right)\right. \\ &\left.\cdot\frac{1}{2\pi^{n/2}||\alpha^2\boldsymbol{I}||^{1/2}}\exp\left(-\frac{||\boldsymbol{v}_n - \boldsymbol{\beta}_n||^2}{2\alpha^2}\right)\right\} \\ &= \log\left\{L_1\cdot\frac{1}{2\pi^{n/2}||\alpha^2\boldsymbol{I}||^{1/2}}\exp\left(-\frac{||\boldsymbol{v}_n - \boldsymbol{\beta}_n||^2}{2\alpha^2}\right)\right\} \\ &= \log L_2 - \frac{||\boldsymbol{v}_n - \boldsymbol{\beta}_n||^2}{2\alpha^2}, \end{aligned} \tag{32}$$

where L_1 and L_2 are independent of $\boldsymbol{\beta}_n$. Based on Eq.(29), the Q-function can be expressed as:

$$Q(\boldsymbol{\beta}_n, \hat{\boldsymbol{\beta}}_n^{(l)}) = \log L_2 - \frac{||\boldsymbol{v}_n - \hat{\boldsymbol{\beta}}_n^{(l)}||^2}{2\alpha^2}. \tag{33}$$

After the expectation step, to maximize the Q-function with penalty, the value of hidden variable $\boldsymbol{v}_n$ needs to be calculated first. Since $p(\boldsymbol{v}_n|\boldsymbol{y}_n, \hat{\boldsymbol{\beta}}_n^{(l)}) \propto p(\boldsymbol{y}_n|\boldsymbol{v}_n)p(\boldsymbol{v}_n|\hat{\boldsymbol{\beta}}_n^{(l)})$ from Eq.(31), with Eq.(26) and Eq.(27), we have:

$$\begin{aligned} p(\boldsymbol{y}_n|\boldsymbol{v}_n) &= \mathcal{N}(\boldsymbol{X}_n\boldsymbol{v}_n|\sigma^2\boldsymbol{D}_n^{-1} - \alpha^2\boldsymbol{X}_n\boldsymbol{X}_n^T) \\ p(\boldsymbol{v}_n|\hat{\boldsymbol{\beta}}_n^{(l)}) &= \mathcal{N}(\hat{\boldsymbol{\beta}}_n^{(l)}|\alpha^2\boldsymbol{I}). \end{aligned} \tag{34}$$

Then, with $p(\boldsymbol{v}_n|\boldsymbol{y}_n, \hat{\boldsymbol{\beta}}_n^{(l)}) \propto p(\boldsymbol{y}_n|\boldsymbol{v}_n)p(\boldsymbol{v}_n|\hat{\boldsymbol{\beta}}_n^{(l)})$ and Eq.(34), the estimate value of the hidden variable $\hat{\boldsymbol{v}}_n^{(l)}$ can be computed as (details in [29], chapter 2):

$$\begin{aligned} \hat{\boldsymbol{v}}_n^{(l)} &= \alpha^2\boldsymbol{X}_n^T(\alpha^2\boldsymbol{X}_n\boldsymbol{X}_n^T + \sigma^2\boldsymbol{D}_n^{-1} - \alpha^2\boldsymbol{X}_n\boldsymbol{X}_n^T)^{-1}(\boldsymbol{y}_n - \boldsymbol{X}_n\hat{\boldsymbol{\beta}}_n^{(l)}) + \hat{\boldsymbol{\beta}}_n^{(l)} \\ &= \frac{\alpha^2}{\sigma^2}\boldsymbol{X}_n^T\boldsymbol{D}_n(\boldsymbol{y}_n - \boldsymbol{X}_n\hat{\boldsymbol{\beta}}_n^{(l)}) + \hat{\boldsymbol{\beta}}_n^{(l)} \\ &= \frac{\alpha^2}{\sigma^2}\boldsymbol{X}_n^T\boldsymbol{D}_n\boldsymbol{y}_n + (\boldsymbol{I} - \frac{\alpha^2}{\sigma^2}\boldsymbol{X}_n^T\boldsymbol{D}_n\boldsymbol{X}_n)\hat{\boldsymbol{\beta}}_n^{(l)}. \end{aligned} \tag{35}$$

Eventually, with Eq.(33), the penalized maximum likelihood Eq.(30) can be rewritten as:

$$\begin{aligned} \hat{\boldsymbol{\beta}}_n^{(l+1)} &= \arg\max_{\boldsymbol{\beta}_n}\left(Q(\boldsymbol{\beta}_n, \hat{\boldsymbol{\beta}}_n^{(l)}) - pen(\hat{\boldsymbol{\beta}}_n^{(l)})\right) \\ &= \arg\max_{\boldsymbol{\beta}_n}\left(-\frac{||\boldsymbol{v}_n - \hat{\boldsymbol{\beta}}_n^{(l)}||^2}{2\alpha^2} - pen(\hat{\boldsymbol{\beta}}_n^{(l)})\right) \\ &= \arg\max_{\boldsymbol{\beta}_n}\left(-||\boldsymbol{v}_n - \hat{\boldsymbol{\beta}}_n^{(l)}||^2 - 2\alpha^2 pen(\hat{\boldsymbol{\beta}}_n^{(l)})\right). \end{aligned} \tag{36}$$

To solve Eq.(36), with $\mathcal{L}_1$ regularization, the penalty term $pen(\hat{\boldsymbol{\beta}}_n^{(l)})$ can be rewritten as:

$$pen(\hat{\boldsymbol{\beta}}_n^{(l)}) = \gamma||\boldsymbol{\beta}_n^{(l)}||_1 = \gamma\sum_i|\beta_{i,n}^{(l)}|, \quad i = 1, 2, 3\cdots M. \tag{37}$$

where M is the number of unknown parameters and $\hat{\beta}_{i,n}^{(l)}$ is the i-th element of $\hat{\boldsymbol{\beta}}_n^{(l)}$.

Therefore, $\hat{\boldsymbol{\beta}}_n^{(l+1)}$ can be obtained by applying a soft-threshold function. For the $(l+1)$-th iteration, the i-th element $\hat{\beta}_{i,n}^{(l+1)}$ of $\hat{\boldsymbol{\beta}}_n^{(l+1)}$ can be computed by [28]:

$$\hat{\beta}_{i,n}^{(l+1)} = sgn(\hat{v}_{i,n}^{(l)}) \left(|\hat{v}_{i,n}^{(l)}| - \gamma\alpha^2 \right), \tag{38}$$

where $\hat{v}_{i,n}^{(l)}$ is the i-th element of $\hat{\boldsymbol{v}}_n^{(l)}$.

To further increase the accuracy of the estimation, the residual term ψ_{n+1} from Eq.(10) is updated based on the estimated $\hat{\beta}_{i,n}^{(t)}$ after a total of t times of iteration. For the n-th input, the estimated original parameter matrix $\hat{\boldsymbol{K}}_n$ of Eq.(4) can be solved based on Eq.(12):

$$\begin{cases} \hat{k}_{xx,n} & = \sqrt{\hat{\beta}_{7,n}^{(t)}} \\ \hat{k}_{yy,n} & = \sqrt{\hat{\beta}_{8,n}^{(t)}} \\ \hat{k}_{zz,n} & = \sqrt{\hat{\beta}_{9,n}^{(t)}} \\ \hat{k}_{xy,n} & = \hat{\beta}_{4,n}^{(t)}/2\sqrt{\hat{\beta}_{8,n}^{(t)}} \\ \hat{k}_{xz,n} & = \hat{\beta}_{5,n}^{(t)}/2\sqrt{\hat{\beta}_{9,n}^{(t)}} \\ \hat{k}_{yz,n} & = \hat{\beta}_{6,n}^{(t)}/2\sqrt{\hat{\beta}_{9,n}^{(t)}} \\ \hat{o}_{x,n} & = \hat{\beta}_{1,n}^{(t)}/2\hat{\beta}_{7,n}^{(t)} \\ \hat{o}_{y,n} & = \hat{\beta}_{2,n}^{(t)}/2\hat{\beta}_{8,n}^{(t)} \\ \hat{o}_{z,n} & = \hat{\beta}_{3,n}^{(t)}/2\hat{\beta}_{9,n}^{(t)} \end{cases} \tag{39}$$

Then, the estimated residual $\hat{\psi}_{n+1}$ of Eq.(10) can be estimated based on Eq.(11). For online implementation, assuming that after t-th iteration within EM algorithm, a new set of data x_{n+1} is inputted into the system, and the following updation can be applied to Eq.(35) for the initial estimation of $\boldsymbol{v}_{n+1}^{(0)}$:

$$\boldsymbol{v}_{n+1}^{(0)} = \lambda\frac{\alpha^2}{\sigma^2}\boldsymbol{X}_n^T\boldsymbol{D}_n\boldsymbol{y}_n + \frac{\alpha^2}{\sigma^2}y_{n+1}\boldsymbol{x}_{n+1} + (\boldsymbol{I} - \lambda\frac{\alpha^2}{\sigma^2}\boldsymbol{X}_n^T\boldsymbol{D}_n\boldsymbol{X}_n - \frac{\alpha^2}{\sigma^2}\boldsymbol{x}_{n+1}\boldsymbol{x}_{n+1}^T)\hat{\boldsymbol{\beta}}_n^{(t)}, \tag{40}$$

where y_n can be replaced by $g - \hat{\psi}_n$. This procedure can reduce the bias caused by linearization of original TA models of auto-calibration. Therefore, we have:

$$\begin{aligned}\boldsymbol{v}_{n+1}^{(0)} &= \lambda\frac{\alpha^2}{\sigma^2}\boldsymbol{X}_n^T\boldsymbol{D}_n(\boldsymbol{g}-\hat{\boldsymbol{\psi}}_n)+\frac{\alpha^2}{\sigma^2}(g-\hat{\psi}_{n+1})\boldsymbol{x}_{n+1}\\ &+(\boldsymbol{I}-\lambda\frac{\alpha^2}{\sigma^2}\boldsymbol{X}_n^T\boldsymbol{D}_n\boldsymbol{X}_n-\frac{\alpha^2}{\sigma^2}\boldsymbol{x}_{n+1}\boldsymbol{x}_{n+1}^T)\hat{\boldsymbol{\beta}}_n^{(t)}.\end{aligned} \tag{41}$$

To simplify Eq.(41), let us define:

$$\boldsymbol{H}_n=\frac{\alpha^2}{\sigma^2}\boldsymbol{X}_n^T\boldsymbol{D}_n(\boldsymbol{g}-\hat{\boldsymbol{\psi}}_n), \tag{42}$$

and

$$\boldsymbol{R}_n=\boldsymbol{I}-\frac{\alpha^2}{\sigma^2}\boldsymbol{X}_n^T\boldsymbol{D}_n\boldsymbol{X}_n. \tag{43}$$

Eq.(40) can then be simplified as:

$$\boldsymbol{v}_{n+1}^{(0)}=\lambda\boldsymbol{H}_n+\frac{\alpha^2}{\sigma^2}(g-\hat{\psi}_{n+1})\boldsymbol{x}_{n+1}+(\lambda\boldsymbol{R}_n-\frac{\alpha^2}{\sigma^2}\boldsymbol{x}_{n+1}\boldsymbol{x}_{n+1}^T+(1-\lambda)\boldsymbol{I})\hat{\boldsymbol{\beta}}_n^{(t)}. \tag{44}$$

Then, let us recall Eq.(35) and Eq.(38), the relationship between $\hat{\boldsymbol{v}}_n^{(l)}$ and $\hat{\boldsymbol{v}}_n^{(l-1)}$ is:

$$\begin{aligned}\hat{\boldsymbol{v}}_n^{(l)} &= \frac{\alpha^2}{\sigma^2}\boldsymbol{X}_n^T\boldsymbol{D}_n(\boldsymbol{g}-\hat{\boldsymbol{\psi}}_n)+(\boldsymbol{I}-\frac{\alpha^2}{\sigma^2}\boldsymbol{X}_n^T\boldsymbol{D}_n\boldsymbol{X}_n)\hat{\boldsymbol{\beta}}_n^{(l)}\\ &= \boldsymbol{H}_n+\boldsymbol{R}_n sgn(\hat{\boldsymbol{v}}_n^{(l-1)})\left(|\hat{\boldsymbol{v}}_n^{(l-1)}|-\gamma\alpha^2\right),\end{aligned} \tag{45}$$

where the i-th element of $sgn(\hat{\boldsymbol{v}}_n^{(l-1)})\left(|\hat{\boldsymbol{v}}_n^{(l-1)}|-\gamma\alpha^2\right)$ can be computed by:

$$sgn(\hat{v}_{i,n}^{(l)})\left(|\hat{v}_{i,n}^{(l)}|-\gamma\alpha^2\right)=\begin{cases}\hat{v}_{i,n}^{(l)}-\gamma\alpha^2 & i\in\mathcal{U}_+^{(l)}\\ \hat{v}_{i,n}^{(l)}+\gamma\alpha^2 & i\in\mathcal{U}_-^{(l)}\\ 0 & i\notin\mathcal{U}_+^{(l)}\cup\mathcal{U}_-^{(l)}\end{cases} \tag{46}$$

where $\mathcal{U}_+^{(l)}$ and $\mathcal{U}_-^{(l)}$ are sets defined as:

$$\begin{cases}\mathcal{U}_+^{(l)} & =\{i:\hat{v}_{i,n}^{(l)}>\gamma\alpha^2\}\\ \mathcal{U}_-^{(l)} & =\{i:\hat{v}_{i,n}^{(l)}<-\gamma\alpha^2\}\end{cases} \tag{47}$$

In summation, let us assume we need t times of iteration during EM algorithm. Then, the complete algorithm for the online TA calibration can be summarized by Algorithm 1 and Algorithm 2.

Algorithm 1 SPARLS

```
Initial: $\boldsymbol{R}_1 = \boldsymbol{I} - \frac{\alpha^2}{\sigma^2}\boldsymbol{x}_1\boldsymbol{x}_1^T$, $\boldsymbol{H}_1 = \frac{\alpha^2}{\sigma^2}\boldsymbol{x}_1 y_1$, $\psi_1 = 0$ and $t$.
for any $\boldsymbol{x}_n$ do,
    $\boldsymbol{R}_n = \lambda\boldsymbol{R}_{n-1} - \frac{\alpha^2}{\sigma^2}\boldsymbol{x}_n\boldsymbol{x}_n^T + (1-\lambda)\boldsymbol{I}$.
    $\boldsymbol{H}_n = \lambda\boldsymbol{H}_{n-1} + \frac{\alpha^2}{\sigma^2}(g - \hat{\psi}_n)\boldsymbol{x}_n$.
    Run EM ($\boldsymbol{R}_n$, $\boldsymbol{H}_n$, $\hat{\boldsymbol{\beta}}_{n-1}$, $\mathcal{U}_+^{(l)}$, $\mathcal{U}_-^{(l)}$, $t$, TA's measurement ($v_{x,n}$, $v_{y,n}$, $v_{z,n}$)).
    Update $\hat{\boldsymbol{\beta}}_n, \hat{\psi}_{n+1}$.
end for
Output: $\hat{\boldsymbol{\beta}}_n, \hat{\psi}_n$
```

Algorithm 2 EM Algorithm for the linearized TA model

```
Input: $\boldsymbol{R}_n$, $\boldsymbol{H}_n$, $\hat{\boldsymbol{\beta}}_n$, $\mathcal{U}_+^{(l)}$, $\mathcal{U}_-^{(l)}$, $t$, TA's measurement $(v_{x,n},\, v_{y,n},\, v_{z,n})$.
  $\hat{\boldsymbol{v}}_n^{(0)} = \boldsymbol{R}_{\mathcal{U}_+^t,n}\hat{\boldsymbol{\beta}}_{\mathcal{U}_+^t,n} + \boldsymbol{R}_{\mathcal{U}_-^t,n}\hat{\boldsymbol{\beta}}_{\mathcal{U}_-^t,n} + \boldsymbol{H}_n$.
  $\mathcal{U}_+^{(0)} = \{i : \hat{v}_{i,n}^{(0)} > \gamma\alpha^2\}$.
  $\mathcal{U}_-^{(0)} = \{i : \hat{v}_{i,n}^{(0)} < -\gamma\alpha^2\}$.
for $l = 1, 2, 3, \cdots, t$, do
    $$\begin{aligned}\hat{\boldsymbol{v}}_n^{(l)} = {} & \boldsymbol{R}_{\mathcal{U}_+^{(l-1)},n}\left(\hat{\boldsymbol{v}}^{(l-1)}_{\mathcal{U}_+^{(l-1)},n} - \gamma\alpha^2\mathbf{1}_{\mathcal{U}_+^{(l-1)},n}\right) \\ & + \boldsymbol{R}_{\mathcal{U}_-^{(l-1)},n}\left(\hat{\boldsymbol{v}}^{(l-1)}_{\mathcal{U}_-^{(l-1)},n} + \gamma\alpha^2\mathbf{1}_{\mathcal{U}_-^{(l-1)},n}\right) + \boldsymbol{H}_n.\end{aligned}$$
    $\mathcal{U}_+^{(l)} = \{i : \hat{v}_{i,n}^{(l)} > \gamma\alpha^2\}$.
    $\mathcal{U}_-^{(l)} = \{i : \hat{v}_{i,n}^{(l)} < -\gamma\alpha^2\}$.
end for
for $i = 1, 2, 3, \cdots, M$, do
    $$\hat{\beta}_{i,n} = \begin{cases} \hat{v}_{i,n}^{(l)} - \gamma\alpha^2 & i \in \mathcal{U}_+^{(l)} \\ \hat{v}_{i,n}^{(l)} + \gamma\alpha^2 & i \in \mathcal{U}_-^{(l)} \\ 0 & i \notin \mathcal{U}_+^{(l)} \cup \mathcal{U}_-^{(l)} \end{cases}$$
end for
Estimate original parameter of $(\hat{\boldsymbol{K}})$ based on $\hat{\boldsymbol{\beta}}_n$:
    $$\left\{\begin{array}{ll} \hat{k}_{xx,n} & = \sqrt{\hat{\beta}_{7,n}} \\ \hat{k}_{yy,n} & = \sqrt{\hat{\beta}_{8,n}} \\ \hat{k}_{zz,n} & = \sqrt{\hat{\beta}_{9,n}} \\ \hat{k}_{xy,n} & = \hat{\beta}_{4,n}/2\sqrt{\hat{\beta}_{7,n}} \\ \hat{k}_{xz,n} & = \hat{\beta}_{5,n}/2\sqrt{\hat{\beta}_{8,n}} \\ \hat{k}_{yz,n} & = \hat{\beta}_{6,n}/2\sqrt{\hat{\beta}_{9,n}} \\ \hat{o}_{x,n} & = \hat{\beta}_{1,n}/2\hat{\beta}_{7,n} \\ \hat{o}_{y,n} & = \hat{\beta}_{2,n}/2\hat{\beta}_{8,n} \\ \hat{o}_{z,n} & = \hat{\beta}_{3,n}/2\hat{\beta}_{9,n} \end{array}\right. \tag{48}$$
```

```
Estimate $\hat{\psi}_n$ based estimated $\hat{\boldsymbol{K}}$:
    $$\begin{aligned}\hat{\psi}_n = &\Big(\hat{k}_{xy,z}(v_{x,n}+\hat{o}_{x,n})\Big)^2 + \Big(\hat{k}_{xz,n}(v_{x,n}+\hat{o}_{x,n}) + \hat{k}_{yz,n}(v_{y,n}+\hat{o}_{y,n})\Big)^2 \\ &+ \hat{k}^2_{xx,n}\hat{o}^2_{x,n} + \hat{k}^2_{yy,n}\hat{o}^2_{y,n} + \hat{k}^2_{zz,n}\hat{o}^2_{z,n} \\ &+ 2\hat{k}_{xy,n}\hat{k}_{yy,n}(v_{x,n}\hat{o}_{y,n} + v_{y,n}\hat{o}_{x,n} + \hat{o}_{x,n}\hat{o}_{y,n}) \\ &+ 2\hat{k}_{xz,n}\hat{k}_{zz,n}(v_{x,n}\hat{o}_{z,n} + v_{z,n}\hat{o}_{x,n} + \hat{o}_{x,n}\hat{o}_{z,n}) \\ &+ 2\hat{k}_{yz,n}\hat{k}_{zz,n}(v_{y,n}\hat{o}_{z,n} + v_{z,n}\hat{o}_{y,n} + \hat{o}_{y,n}\hat{o}_{z,n}).\end{aligned} \tag{49}$$
Output: $\hat{\boldsymbol{\beta}}_n$, $\hat{\psi}_n$, $\mathcal{U}_+^{(t)}$, $\mathcal{U}_-^{(t)}$.
```

The region of convergence of the proposed algorithm will be numerically investigated by numerical simulation in Section 4.

4. Simulation and experimental results

4.1. Simulation

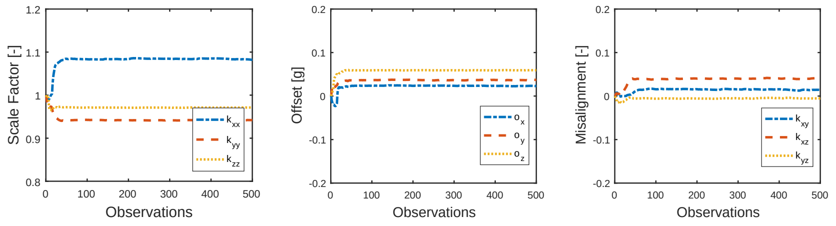
In real life, we can not access to the true scale factors, offsets and misalignments for individual TAs. Therefore, it is difficult to accurately validate the performance of proposed calibration method by experiments alone. Meanwhile, it is also difficult to verify whether the proposed calibration method can correctly identify the zero parameters by experiments because it depends on actual MEMS TAs. Hence, simulations are important to validate the proposed calibration method.

First, the true scale factors, offsets and misalignments were pre-defined ~~for simulation section~~. Then, by applying the proposed calibration method, scale factors, offsets and misalignments are estimated and compared to pre-defined true values. Several types of simulations were carried out to examine different aspects of the proposed method.

Let us examine the performance of the online calibration method under normal condition first. For MEMS accelerometer, generally, the typical errors of scale factors and offsets are within ±10% and $\pm 0.1g$ respectively. These can cause 20° difference in angle measurement during orientation in the worst-case scenario [13]. With current MEMS technology, it is confident to assume the misalignment between each axis is within 5%. In order to obtain reliable results from simulation, the parameters were generated randomly under the following conditions: the errors of scale factors follow uniform distributions $U(-10\%, 10\%)$; the offsets follow $U(-0.1g, 0.1g)$ and misalignments follow $U(-5\%, 5\%)$. According to the datasheets of some recently developed MEMS TAs, the noise density of the measurements is around $100 \sim 500ug/\sqrt{Hz}$ for most commercial grade low cost MEMS TAs. With a typical $100Hz$ output frequency, the range of noise level in g is around $1mg \sim 5mg$. Thus, in this simulation, we used two different noise levels, $1mg$ and $5mg$ standard deviation with zero mean subject to Gaussian distribution. For each noise level, 500 ideal points on sphere with $'1g'$ radium were generated randomly. Then, with the pre-defined model, these 500 generate points from $'1g'$ sphere were converted to observations that noise was added based on different noise levels. After that, the proposed calibration were applied for these 500 observations. We repeated this simulation 100 times for the two noise levels respectively. Additionally, we set the following initial values for each unknown parameter:

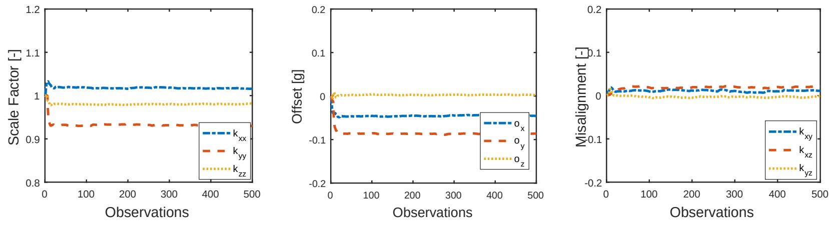
- Scale factors $(k_{xx}, k_{yy}, k_{zz}) : 1$
- Misalignments $(k_{xy}, k_{xz}, k_{yz}) : 0$
- Offsets $(o_x, o_y, o_z) : 0$

Since the proposed method is an online method, the first 100 observations are used to obtain stable estimation. Therefore, only the errors after the initial 100 observations are recorded in Table 1. Besides that, we randomly chose 1 out of the 100 simulations with $1mg$ and $5mg$ noise levels respectively. Fig. 1 and Fig. 2 show two cases for the proposed method, which is chosen randomly from the 100 simulations.



(a) (b) (c)

Figure 1: Estimated parameters by proposed calibration during online estimation under 1mg noise level.



(a) (b) (c)

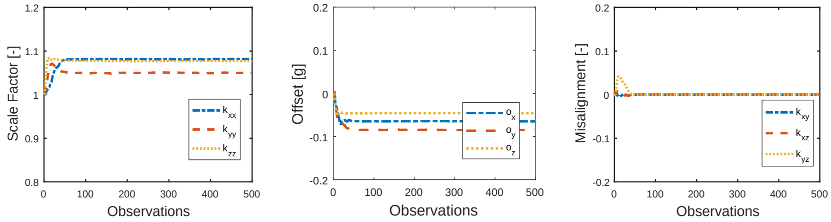
Figure 2: Estimated parameters by proposed calibration during online estimation under 5mg noise level.

Table 1: Calibration Results of 100 Simulations under Normal Condition

	Error distribution	Parameter	Calibration error for $1mg$ noise		Calibration error for $5mg$ noise	
			Mean Error $[g]$	Standard deviation $[g]$	Mean Error $[g]$	Standard deviation $[g]$
Scale Factor error	U(-10%,10%)	k_{xx}	-2.44e-04	-7.73e-04	-2.19e-04	1.32e-03
		k_{yy}	-6.84e-04	-7.41e-04	-6.41e-04	1.26e-03
		k_{zz}	-3.32e-04	3.31e-04	-3.95e-04	1.81e-03
Offset error	U(-10%,10%)	o_x	1.36e-04	2.30e-03	-1.30e-04	2.51e-03
		o_y	-1.46e-04	2.14e-03	-2.49e-04	2.37e-03
		o_z	-1.55e-04	2.17e-03	-1.41e-04	2.34e-03
Misalignment error	U(-5%,5%)	k_{xy}	-1.55e-04	3.42e-03	-1.14e-04	3.92e-03
		k_{xz}	-1.34e-04	2.43e-03	-1.43e-04	2.76e-03
		k_{yz}	1.64e-04	2.44e-03	2.24e-05	2.65e-03

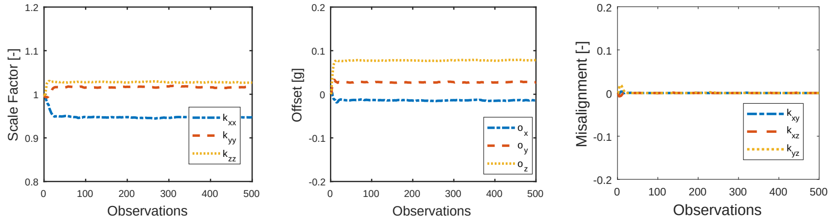
~~In order to verify the performance of the proposed calibration during environmental change, the value of unknown parameters were changed during online calibration. For each run of the simulation, we generated a model and 500 observations based on previous assumption. Then, we randomly increased or decreased the parameters by 10%. Based on these modified parameters, 500 new observations were generated. A threshold of 95% is used, when the estimation reaches 95% of steady state value, the number of observations are recorded in Table.2. For this simulation, we focused on scale factor k_{xx}, k_{yy}, k_{zz} because the change of offset and misalignment is minimal due to its small value and relatively large noises. We repeated this simulation for 100 times and the noise level was set to $5mg$. Fig.3 shows the estimation results which is randomly chosen from the 100 simulations.~~

To verify whether the proposed calibration method can correctly identify the parameters which are zero, the misalignments were set to zero. Then, the errors of scale factors still follow a uniform distribution $U(-10\%, 10\%)$, the offsets follow $U(-0.1g, 0.1g)$ as previous. For each simulation, 200 observations were generated and tested for $1mg$ and $5mg$ noise level. A total of 100 times of simulations were carried out for this step. Fig. 3 and Fig. 4 show a typical run of simulation for this test. Eventually, all zero parameters were correctly identified within 100 steps of iteration.



(a) (b) (c)

Figure 3: Sparsity test of proposed calibration method under $1mg$ noise level.

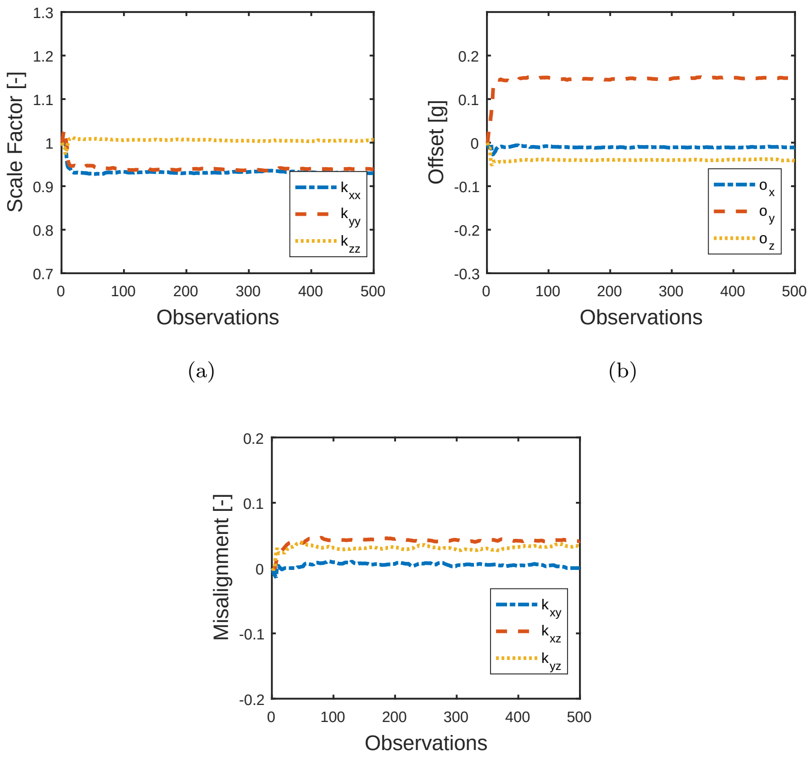


(a) (b) (c)

Figure 4: Sparsity test of proposed calibration method under $5mg$ noise level.

The convergence of EM algorithm for linear model has been proved [25]. However, as the model of auto-calibration is nonlinear, the convergence of the proposed online EM-based algorithm should still be investigated. The convergence condition for the iterative approach might be similar with that for 6-parameter TA model as presented in [17]. However, comparing with the convergence analysis for 6-parameter model, the major difficulty here is due to the neglected term ψ_i. From Eq.(11), it can be observed that ψ_i is a function of input acceleration ($v_{x,i}$, $v_{y,i}$, and $v_{z,i}$), which are randomly changing during online calibration. This makes the rigorous proof of the convergence becomes difficult. In this section, we will use Monte Carlo simulation to numerically show that the convergence of the proposed algorithm is guaranteed if the initial estimation error of the TA parameters is within a certain range.

Let us reset the uncertainty range of the unknown parameters: errors of scale factors k_{xx}, k_{yy}, k_{zz} follow $U(-30\%, 30\%)$ (i.e., the scale factors are within the range $(-0.7, 1.3)$), offsets o_x, o_y, o_z are within the range $U(-0.25, 0.25)$ and misalignment k_{xy}, k_{xz}, k_{yz} are within the range $U(-0.1, 0.1)$. It should be noted that almost all MEMS TAs available in the current market are well below these error ranges. Now, for each set of simulation, we generated 500 random obversion on $1g$ sphere with $5mg$ noise density, and we executed $500,000$ sets of simulation. From the results of $500,000$ sets of simulations, it was seen that all estimated parameters were convergent. To show the rate of convergence, we chose two sets of simulation and showed the results in Fig. 5 and Fig. 6. Fig. 6



(a)

(b)

(c)

Figure 5: Estimated parameters during online calibration based on the proposed algorithms with extended error range

shows the initial 200 times of estimation of ψ. It indicates the estimated $\hat{\psi}$ can converge to the true value in a short period of time.

However, if we increase the variation ranges of the scale factors and offsets to $U(-40\%, 40\%)$ and $U(-30\%, 30\%)$ respectively, it can be observed that, in some cases, the estimation does not converge. ~~Fig. 8 and Fig. 9 show a divergent case.~~

4.2. Experiment

To implement the proposed linearization and calibration method, we developed a health monitoring device which contains SCM (TI F430), IMU (In-

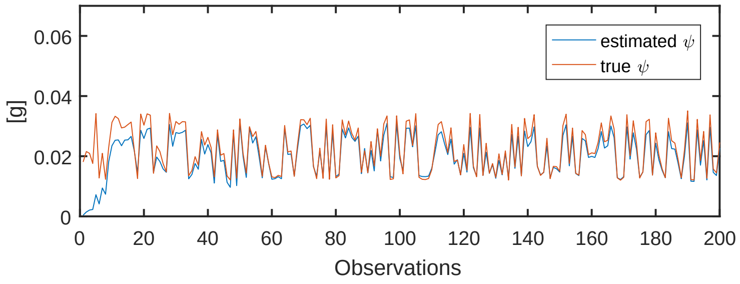


Figure 6: Estimate $\hat{\psi}$ and true ψ

venSense MPU9150) and featured with Bluetooth module for wireless communication in Centre for Health Technologies, University of Technology, Sydney. From the datasheet of MPU9150, the integrated accelerometer is a digital 3-axis accelerometer. Therefore, we can apply the 9-parameter model together with the proposed calibration algorithm for online calibration. The device of the experiments is shown in Fig. 7.

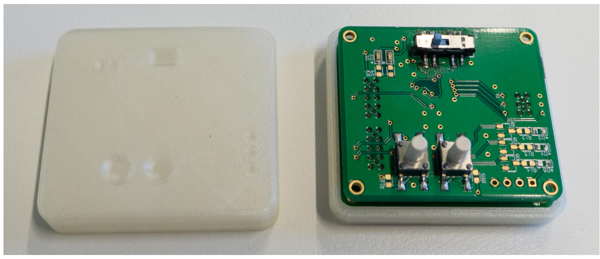
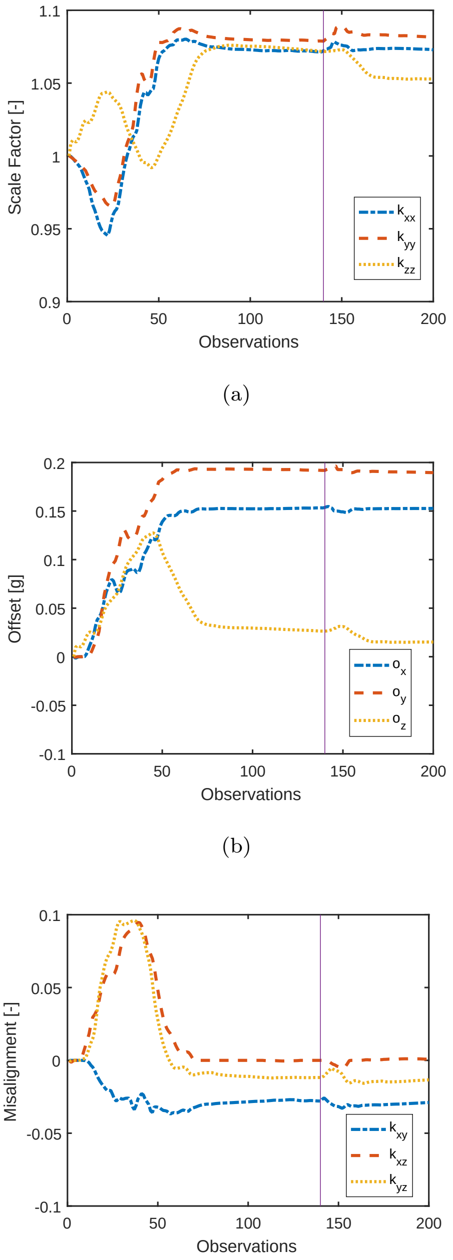


Figure 7: A self-designed IMU module for the experiment

Based on Algorithm 1 and 2, firstly, the accelerometer was randomly placed in 100 different orientations. Then, we applied the proposed algorithms to calculate scale factors, offsets and misalignments. The outputs from the accelerometer were converted to acceleration with unit g based on standard factory parameters (i.e., sensitivity) first based on initial lsb/g. During experiments, the sampling frequency was set to $100Hz$. The output range was selected as $\pm 2g$ with 14 bit resolution, which results in 16384 lsb/g. The data was transferred through Bluetooth directly between computer and IMU module. To evaluate the accuracy of the estimation, we collected another 40 sets of data and estimated the acceleration based on estimation parameters. Furthermore, for the performance of parameter variation tracking during the change of temperate, the IMU module was illuminated by a lamp for 20mins. After that, we collect another 60 sets of data for online estimation. The results are shown in Fig. 8



(a)

(b)

(c)

Figure 8: Estimated parameters of 200 set of observations.

In Fig. 8, the unknown parameters were estimated within 80 observations. In addition, the variation of parameters of the accelerometer has been identified by the proposed algorithm, indicating that the proposed online algorithm can track parameter variations during temperature changes. Furthermore, in Fig. 8-(c), the misalignment k_{xz} is within the threshold. Therefore, the model is simplified by assigning $\hat{k}_{xy}$ to zero. This indicates that the proposed algorithm can adapt to the model structure variation due to the changing of temperature. The errors between the vector sum of the estimated accelerations and the local acceleration $'1g'$ of observations 100 to 200 were recorded in Table 2, which shows that the accuracy of measurements is significantly increased after online calibration.

Table 2: Estimation error of the vector sum before and after calibration

	Error between estimation and local acceleration	
	mean error $[g]$	standard deviation $[g]$
Before calibration	0.0917	0.2645
After calibration	0.0003	0.0159

Additionally, to demonstrate the effectiveness of this online calibration method during parameter varying, the error of the vector sum based on different estimated parameters is analyzed for the last 40 sets of observations. For group 1, the error of the vector sum was calculated based on a set of fixed estimated parameters from the $140th$ estimations. For group 2, the new estimator from the online calibration method was used to calculate the error of vector sum. The result is reported in Table 3, which indicates that the proposed online calibration method can significantly improve the measurement accuracy.

We also mounted the accelerometer on a turntable with two degree of freedom and recorded the output from 10 different orientations. These 10 orientations were considered as reference orientations, and the errors between the estimated orientations and reference orientations were calculated. The turntable has 2 optical rotary stages with $5'$ resolution and $0.03mm$ reading accuracy. For

Table 3: Error between estimation and local acceleration based on different parameters

	Estimation error of vector sum	
	mean error $[g]$	standard deviation $[g]$
Group 1	0.0386	0.0142
Group 2	0.0001	0.0084

the first five orientations, we set pitch ($\bar{p}$) as 0°, 30°, 45°, 60° and 90° while yaw and roll remained the same. Then, we repeated this for roll ($\tilde{p}$) while pitch and yaw remaining the same. Eq.(50) shows the relationship between reference orientation and initial value:

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}^T = \begin{bmatrix} a_{x,0} \\ a_{y,0} \\ a_{z,0} \end{bmatrix}^T \begin{bmatrix} c(\bar{p}) & 0 & s(\bar{p}) \\ 0 & 1 & 0 \\ -s(\bar{p}) & 0 & c(\bar{p}) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\tilde{p}) & -s(\tilde{p}) \\ 0 & s(\tilde{p}) & c(\tilde{p}) \end{bmatrix}, \tag{50}$$

where $[a_{x,0}, a_{y,0}, a_{z,0}]^T$ is the initial value $[0, 0, 1]^T$, c and s represent *Cosine* and *Sine* respectively. The estimated $[\hat{a}_x, \hat{a}_y, \hat{a}_z]^T$ was compared with $[a_x, a_y, a_z]^T$.

The results of experiments were analyzed and listed in Table 4. The errors between the reference values $[a_x, a_y, a_z]^T$ and estimated values $[\hat{a}_x, \hat{a}_y, \hat{a}_z]^T$ were calculated for these 10 testing orientations. Since we do not know the actual value of the scale factors, offsets and misalignments, we compared and analyzed the errors between the reference orientations and the estimated orientations. Due to the orientation inaccuracy of the turntable, the estimation of the vector sum ($\hat{a}_{vs}$) should be more accurate than that of the individual acceleration on each axis. It is also anticipated that the proposed algorithm optimizes the variance of $\hat{a}_{vs}$.

Table 4: Estimation error of overall acceleration and acceleration components on each axis

	Estimation error	
	Mean Error $[g]$	Standard deviation $[g]$
$\epsilon_{\hat{a}_x}$	0.0163	0.0075
$\epsilon_{\hat{a}_y}$	-0.0080	0.0070
$\epsilon_{\hat{a}_z}$	-0.00002	0.0030
$\epsilon_{\hat{a}_{vs}}$	0.0011	0.0100

Overall, the achieved results demonstrate the efficiency of the proposed parameter estimation method.

5. Conclusion

In this paper, a linearization method for 9-parameter TA model has been presented for auto-calibration. To solve the unknown parameters from the linearized model online, a modified sparse least square estimation method was introduced. The online calibration method can automatically remove the insignificant parameters during online calibration. Furthermore, the proposed calibration method can track parameters when the parameters change due to daily drift and/or temperature variation.

It should be noted that this study is the very first research that focused on online calibration with automatic model selection for the auto-calibration of 9 parameters. Comparing to most previous researches, which were mainly based on off-line calibration, this method can achieve real-time online calibration of a 9-parameter auto-calibration model. Furthermore, in contrast with the UKF based online calibration method [26], the proposed calibration approach has embedded an $\mathcal{L}_1$ norm penalty term for the elimination of the insignificant parameters in order to improve the reliability of the calibration.

To verify the presented method, both simulation and experiment were carried out. Based on the simulation results, it can be concluded that the proposed calibration method can achieve accurate estimation within 50 iterations in most cases, and the estimated response has a small mean error and standard deviation. Furthermore, the results from simulation indicated that the proposed method can correctly identify the parameters which were zero.

Experiments were performed for the proposed calibration method with a self-designed IMU. As the true scale factors and displacements were unknown, the error between the estimated orientations and the reference orientations was calculated and analyzed. The experimental results demonstrated that the proposed calibration approach could accurately estimate the vector sum of the three axes in the whole measurement region.

References

[1] M. Yuwono, B. D. Moulton, S. W. Su, B. G. Celler, H. T. Nguyen, Unsupervised machine-learning method for improving the performance of ambulatory fall-detection systems, BioMedical Engineering Online 11 (9).

[2] S. W. Su, L. Wang, B. G. Celler, A. V. Savkin, Y. Guo, Identification and control for heart rate regulation during treadmill exercise, IEEE Transactions on Biomedical Engineering 54 (7) (2007) 1238–1246.

[3] S. W. Su, B. G. Celler, A. V. Savkin, H. T. Nguyen, T. M. Cheng, Y. Guo, L. Wang, Transient and steady state estimation of human oxygen uptake based on noninvasive portable sensor measurements, Medical & Biological Engineering & Computing 47 (10) (2009) 1111–1117.

[4] A. Nez, L. Fradet, P. Laguillaumie, T. Monnet, P. Lacouture, Comparison of calibration methods for accelerometers used in human motion analysis, Medical Engineering & Physics 38 (11) (2016) 1289–1299.

[5] X. Yun, E. R. Bachmann, Design, implementation, and experimental results of a quaternion-based Kalman filter for human body motion tracking, IEEE Transactions on Robotics 22 (6) (2006) 1216–1227.

[6] A. Pantelopoulos, N. G. Bourbakis, A survey on wearable sensor-based systems for health monitoring and prognosis, IEEE Transactions on Systems,

Man, and Cybernetics, Part C (Applications and Reviews) 40 (1) (2010) 1–12.

[7] A. Godfrey, R. Conway, D. Meagher, G. ÓLaighin, Direct measurement of human movement by accelerometry, Medical Engineering & Physics 30 (10) (2008) 1364–1386.

[8] M. Cornacchia, K. Ozcan, Y. Zheng, S. Velipasalar, A survey on activity detection and classification using wearable sensors, IEEE Sensors Journal 17 (2) (2017) 386–403.

[9] Y.-W. Bai, S.-C. Wu, C.-L. Tsai, Design and implementation of a fall monitor system by using a 3-axis accelerometer in a smart phone, IEEE Transactions on Consumer Electronics 58 (4).

[10] M. Pedley, High precision calibration of a three-axis accelerometer, Freescale Semiconductor Application Note, Document Number: AN4399, Rev 1.

[11] D. Jurman, M. Jankovec, R. Kamnik, M. Topič, Calibration and data fusion solution for the miniature attitude and heading reference system, Sensors and Actuators A: Physical 138 (2) (2007) 411–420.

[12] S. Bonnet, C. Bassompierre, C. Godin, S. Lesecq, A. Barraud, Calibration methods for inertial and magnetic sensors, Sensors and Actuators A: Physical 156 (2) (2009) 302–311.

[13] I. Frosio, F. Pedersini, N. A. Borghese, Autocalibration of MEMS accelerometers, IEEE Transactions on Instrumentation and Measurement 58 (6) (2009) 2034–2041.

[14] S.-h. P. Won, F. Golnaraghi, A triaxial accelerometer calibration method using a mathematical model, IEEE Transactions on Instrumentation and Measurement 59 (8) (2010) 2144–2153.

[15] L. Ye, S. W. Su, Experimental design and its posterior efficiency for the calibration of wearable sensors, Journal of Intelligent Learning Systems and Applications 7 (01) (2015) 11.

[16] L. Ye, S. W. Su, Optimum experimental design applied to MEMS accelerometer calibration for 9-parameter auto-calibration model, in: Proceedings of the IEEE international Conference on Engineering in Medicine and Biology Society, EMBC, 2015, pp. 3145–3148.

[17] L. Ye, Y. Guo, S. Su, An efficient auto-calibration method for triaxial accelerometer, IEEE Transactions on Instrumentation and Measurement 66 (9).

[18] Z. Wu, Z. Wang, Y. Ge, Gravity based online calibration for monolithic triaxial accelerometers' gain and offset drift, in: Intelligent Control and Automation, 2002. Proceedings of the 4th World Congress on, Vol. 3, IEEE, 2002, pp. 2171–2175.

[19] InvenSense, Mpu 9150 datasheet (2013).
URL `http://www.invensense.com/products/motion-tracking/9-axis/mpu-9150/`

[20] W. Fong, S. Ong, A. Nee, Methods for in-field user calibration of an inertial measurement unit without external equipment, Measurement Science and Technology 19 (8) (2008) 085202.

[21] H. Zhang, Y. Wu, W. Wu, M. Wu, X. Hu, Improved multi-position calibration for inertial measurement units, Measurement Science and Technology 21 (1) (2010) 015107.

[22] I. Frosio, F. Pedersini, N. A. Borghese, Autocalibration of triaxial MEMS accelerometers with automatic sensor model selection, IEEE Sensors Journal 12 (6) (2012) 2100–2108.

[23] T. Beravs, J. Podobnik, M. Munih, Three-axial accelerometer calibration using Kalman filter covariance matrix for online estimation of optimal sen-

sor orientation, IEEE Transactions on Instrumentation and Measurement 61 (9) (2012) 2501–2511.

[24] P. Batista, C. Silvestre, P. Oliveira, B. Cardeira, Accelerometer calibration and dynamic bias and gravity estimation: Analysis, design, and experimental evaluation, IEEE Transactions on Control Systems Technology 19 (5) (2011) 1128–1137.

[25] B. Babadi, N. Kalouptsidis, V. Tarokh, SPARLS: The sparse RLS algorithm, IEEE Transactions on Signal Processing 58 (8) (2010) 4013–4025.

[26] M. Glueck, D. Oshinubi, P. Schopp, Y. Manoli, Real-time autocalibration of MEMS accelerometers, IEEE Transactions on Instrumentation and Measurement 63 (1) (2014) 96–105.

[27] R. Tibshirani, Regression shrinkage and selection via the LASSO, Journal of the Royal Statistical Society. Series B (Methodological) (1996) 267–288.

[28] M. A. Figueiredo, R. D. Nowak, An EM algorithm for wavelet-based image restoration, IEEE Transactions on Image Processing 12 (8) (2003) 906–916.

[29] C. M. Bishop, Pattern recognition, Machine Learning 128 (2006) 1–58.