A Fast and Accurate Accelerometer and Magnetometer Alignment Algorithm

Konstantinos Papafotis, Student Member, IEEE, and Paul P. Sotiriadis, Senior Member, IEEE

Abstract—A computationally efficient algorithm for axes alignment of a 3-axis accelerometer and a 3-axis magnetometer is introduced. The proposed algorithm applies when the two sensors are fixed on the same rigid platform and individually calibrated. It exploits the magnetic inclination phenomenon to derive both the inclination angle, and, the axes alignment rotation matrix. It is more than 50 times faster than the existing algorithms based on the same principle and using the Newton-Raphson or the gradient descent methods. The algorithm’s accuracy and computational efficiency are demonstrated using multiple measurement sets as well as simulated data sets.

Index Terms—Accelerometer, axes alignment, inertial sensors, magnetic inclination, magnetometer.

I. INTRODUCTION

ACCELEROMETERS and magnetometers are typically used in combination in a broad variety of applications. Popular application fields include, and are not limited to, navigation [1], attitude estimation [2] and image stabilization [3]. Their wide use was enabled by the advancement of micro-electro-mechanical (MEMS) technology over the past years. Small-size, low-cost MEMS accelerometers and magnetometers are nowadays embedded in many commercial devices (smartphones, activity trackers, etc) further broadening their application span.

When the two sensors are used together, the alignment of their sensitivity axes is of major importance. Thus, except from the individual calibration of each sensor, an extra axes alignment procedure is mandatory. When the two sensors are in different packages, the axes misalignment is more evident and it is mainly caused by the placement of the sensors’ packages on the common rigid platform. Even when the two sensors are built into the same package, the soft-iron distortion, often caused by the surrounding electronic components and the enclosure, significantly distorts the measured magnetic field vector causing a misalignment between the two sensors.

Existing works deal with the axes alignment either as an independent problem [4]–[6] or as part of a calibration procedure [7]–[11]. The authors in [7], [8], [10], [12] propose a magnetometer calibration algorithm incorporating the axes alignment step. In [4], [5], [13] the axes alignment is part of an inertial (accelerometer and gyroscope) and magnetic sensors’ calibration algorithm. An accelerometer – magnetometer calibration algorithm, also accounting for the axes misalignment is proposed in [6], [9], [11]. Note that in [7]–[11], [13] the calibration and axes alignment are fused together while in [4]–[6] the alignment step is a discrete part of the calibration algorithm.

While their algorithmic implementation and computational efficiency vary significantly, all the aforementioned algorithms are based on two fundamental approaches; they either use a calibrated inertial sensor (accelerometer or gyroscope) as reference (e.g. [5], [7], [10], [12]), or, exploit the magnetic inclination to align the axes of the two sensors (e.g. [4], [8]).

Magnetic inclination (or magnetic dip) is the angle between the horizon and the Earth’s magnetic field lines as shown in Figure 1. It varies with location and time and the sin(·) of it equals the normalized inner product of the gravity vector $g$ (in $m/s^2$) and the magnetic field vector $m$ (in Tesla).1

$$\sin(\delta) = \frac{g^T m}{\|g\| \|m\|}$$

Existing works deal with the axes alignment either as an independent problem [4]–[6] or as part of a calibration procedure [7]–[11]. The authors in [7], [8], [10], [12] propose a magnetometer calibration algorithm incorporating the axes alignment step. In [4], [5], [13] the axes alignment is part of an inertial (accelerometer and gyroscope) and magnetic sensors’ calibration algorithm. An accelerometer – magnetometer calibration algorithm, also accounting for the axes misalignment is proposed in [6], [9], [11]. Note that in [7]–[11], [13] the calibration and axes alignment are fused together while in [4]–[6] the alignment step is a discrete part of the calibration algorithm.

While their algorithmic implementation and computational efficiency vary significantly, all the aforementioned algorithms are based on two fundamental approaches; they either use a calibrated inertial sensor (accelerometer or gyroscope) as reference (e.g. [5], [7], [10], [12]), or, exploit the magnetic inclination to align the axes of the two sensors (e.g. [4], [8]).

Magnetic inclination (or magnetic dip) is the angle between the horizon and the Earth’s magnetic field lines as shown in Figure 1. It varies with location and time and the $\sin(\cdot)$ of it equals the normalized inner product of the gravity vector $g$ (in $m/s^2$) and the magnetic field vector $m$ (in Tesla).1

$$\sin(\delta) = \frac{g^T m}{\|g\| \|m\|}$$

Existing works deal with the axes alignment either as an independent problem [4]–[6] or as part of a calibration procedure [7]–[11]. The authors in [7], [8], [10], [12] propose a magnetometer calibration algorithm incorporating the axes alignment step. In [4], [5], [13] the axes alignment is part of an inertial (accelerometer and gyroscope) and magnetic sensors’ calibration algorithm. An accelerometer – magnetometer calibration algorithm, also accounting for the axes misalignment is proposed in [6], [9], [11]. Note that in [7]–[11], [13] the calibration and axes alignment are fused together while in [4]–[6] the alignment step is a discrete part of the calibration algorithm.

While their algorithmic implementation and computational efficiency vary significantly, all the aforementioned algorithms are based on two fundamental approaches; they either use a calibrated inertial sensor (accelerometer or gyroscope) as reference (e.g. [5], [7], [10], [12]), or, exploit the magnetic inclination to align the axes of the two sensors (e.g. [4], [8]).

Magnetic inclination (or magnetic dip) is the angle between the horizon and the Earth’s magnetic field lines as shown in Figure 1. It varies with location and time and the $\sin(\cdot)$ of it equals the normalized inner product of the gravity vector $g$ (in $m/s^2$) and the magnetic field vector $m$ (in Tesla).1

$$\sin(\delta) = \frac{g^T m}{\|g\| \|m\|}$$

Fig. 1. Magnetic inclination [4].

1 All norms in this paper are Euclidean norms unless it is indicated otherwise.
When a gyroscope is used as a reference, the calibration and the axes alignment are usually formulated as an estimation problem which is typically solved using Kalman filter [7]. When the magnetic inclination is exploited, the axes alignment is formulated as an optimization problem which is usually solved using the gradient descent [4] or the Newton-Raphson [6] methods implying increased computational burden and potential convergence issues.

Accelerometers and magnetometers are widely used in embedded low-cost devices with very limited hardware resources. In such cases, the computational efficiency of the calibration and axes alignment algorithms is of major importance as they have to be executed quickly on a microcontroller or on a basic microprocessor. To that purpose, we propose a two-phases computationally efficient, iterative algorithm for aligning the axes of a 3-axis accelerometer and a 3-axis magnetometer. It is more than 50 times faster than similar algorithms in the literature and requires no special piece of measurement equipment.

Using a set of measurements of the two sensors, and exploiting the magnetic inclination, the proposed algorithm calculates both the axes alignment rotation matrix and the inclination angle, sin(δ), by solving a computationally-light least-squares problem and then using a single iteration of the Newton-Raphson method. The proposed algorithm applies when the two sensors are individually calibrated and fixed on the same rigid platform.

The algorithm is tested using both experimental measurements and simulated data. Finally, its convergence and efficiency are compared to existing algorithms using the standard iterative methods i.e. gradient descent and Newton-Raphson.

The rest of the paper is organized as follows. In Section II, the axes alignment is formulated as an optimization problem. The proposed algorithm is presented in Section III. The evaluation of the algorithm using both experimental measurements and simulated data is done in Section IV. Finally, the conclusions are drawn in Section V.

II. PROBLEM FORMULATION

Consider a 3-axis accelerometer and a 3-axis magnetometer, both fixed on the same rigid platform, and denote their coordinate frames as {A} and {M} respectively.\(^2\) Note that in the following analysis, the two sensors are assumed to be individually calibrated.

Now consider a set of \(K\) measurements of each sensor, captured while the sensors’ platform is still in \(K\) different orientations respectively and away from magnetic disturbances. The measurements of the accelerometers are naturally expressed in the {A}-frame while those of the magnetometer are expressed in the {M}-frame. They are denoted as \(g_k^A\) and \(m_k^M\) respectively, for \(k = 1, 2, \ldots, K\).

Let the gravity vector \(g\) and magnetic field vector \(m\) be expressed in a fixed inertial coordinate frame \(I\). We express both sensors’ measurements in the {A}-frame as

\[
g_k^A = Q_k g \quad \text{and} \quad m_k^A = Q_k m, \quad k = 1, 2, \ldots, K
\]

Where \(Q_k \in SO(3), k = 1, 2, \ldots, K\) is the frame transformation matrix from the \(I\)-frame to the \(A\)-frame corresponding to the \(k^{th}\) orientation. Combining (1) and (2), we can calculate the inclination angle using the sensors’ measurements

\[
\sin(\delta) = \frac{(g_k^A)^T Q_k Q_k^T m_k^A}{\|Q_k^T g_k^A\| \|Q_k^T m_k^A\|}, \quad k = 1, 2, \ldots, K
\]

(3)

where we exploited the orthogonality of \(Q_k\) and the rotational invariance of the Euclidean norm.

Let \(R_k^M \in SO(3)\) be the frame transformation matrix from the \(M\)-frame to the \(A\)-frame and \(m_k^M\) be the magnetometer’s measurement in the \(M\)-frame. Then, the magnetometer’s measurement, expressed in the \(A\)-frame, is written as

\[
m_k^A = R_k^M m_k^M, \quad k = 1, 2, \ldots, K
\]

(4)

Substituting (4) in (3) and using again the rotational invariance of the Euclidean norm, we get

\[
\sin(\delta) = \frac{(g_k^A)^T R_k^M m_k^M}{\|g_k^A\| \|m_k^M\|}, \quad k = 1, 2, \ldots, K
\]

(5)

In (5), \(g_k^A\) and \(m_k^M\) are the known accelerometer’s and magnetometer’s measurements, respectively, while both \(R_k^M\) and inclination angle, \(\delta\), are unknown. A standard approach to derive them is to form and solve the following optimization problem [4], [6]

\[
\begin{align*}
\text{minimize} & \quad \sum_{k=1}^{K} \left( \sin(\delta) - \frac{(g_k^A)^T R_k^M m_k^M}{\|g_k^A\| \|m_k^M\|} \right)^2 \\
\text{subject to} & \quad R_k^M \in SO(3) \\
& \quad \delta \in \left[ -\frac{\pi}{2}, \frac{\pi}{2} \right]
\end{align*}
\]

(6)

Solving (6) is typically done using either the gradient descent or the Newton-Raphson methods. Both are computational costly; gradient descent typically requires hundreds of iterations to converge while Newton-Raphson, although it converges faster, it requires the calculation of the Hessian matrix in every step.

III. THE PROPOSED ALGORITHM

The proposed algorithm is executed in two phases. In the first phase, an estimate of the axes alignment rotation matrix \(R_k^A\) and the sine of the inclination angle, \(\sin(\delta)\), is derived in a closed-form. Then, in the second phase, the derived \(R_k^A\) and \(\sin(\delta)\) are used as a starting point for the Newton-Raphson method which rapidly (in 1 iteration) converges to more accurate estimates.

For the following analysis it is assumed that both the accelerometer and the magnetometer have already been individually calibrated. For notational convenience, in the rest of this paper we drop the superscripts from the field vectors and both the subscript and superscript from the axes alignment matrix i.e. \(g_k = g_k^A, m_k = m_k^M\) and \(R = R_k^M\).

In addition, without loss of generality, we assume that the gravity and the magnetic field measurement vectors are normalized i.e. \(\|g_k\| = \|m_k\| = 1, k = 1, 2, \ldots, K\). Finally we define \(s = \sin(\delta)\).
A. Derivation of R and s Estimates in Closed-Form

Given the aforementioned assumptions, (5) is written as

\[ s = g_k^T R m_k, \quad k = 1, 2, \ldots, K \]  
(7)

Using the identity \( \text{vec}(AXB) = (B^T \otimes A) \text{vec}(X) \), where \( \otimes \) denotes the Kronecker’s product [14], from (7) we get

\[ s = (m_k^T \otimes g_k^T) \text{vec}(R), \quad k = 1, 2, \ldots, K \]  
(8)

Equation (8) is written in the more convenient matrix form

\[ s \mathbf{1} = A \text{vec}(R) \]  
(9)

where \( \mathbf{1} \) is the \( K \times 1 \) vector of ones and

\[ A \triangleq \begin{bmatrix} (m_1 \otimes g_1)^T \\ (m_2 \otimes g_2)^T \\ \vdots \\ (m_K \otimes g_K)^T \end{bmatrix}. \]  
(10)

In (9), the \( K \times 9 \) matrix \( A \), which is formed using the measurements of the two sensors is known, whereas both the sine of the inclination angle, \( s \), and the alignment matrix in vector form, \( \text{vec}(R) \), are unknown. To proceed, it is assumed that the sine of the inclination angle, \( s \), is nonzero, and furthermore, that matrix \( A \) is of full rank.

We solve (9) for \( \text{vec}(R) \) in a least square sense

\[ \text{vec}(R) = s \left( A^T A \right)^{-1} A^T \mathbf{1}. \]  
(11)

Splitting \( (A^T A)^{-1} A^T \mathbf{1} \) into three \( 3 \times 1 \) vectors \( h_1, h_2 \) and \( h_3 \), such that \( \begin{bmatrix} h_1^T \\ h_2^T \\ h_3^T \end{bmatrix} = (A^T A)^{-1} A^T \mathbf{1} \), we form the following matrix

\[ H = [h_1 \ h_2 \ h_3]. \]  
(12)

Then, from (11) and (12) we have

\[ R = s H \]  
(13)

Note that in (13), \( H \) is formed using the sensors’ measurements and it is known but \( s \) is unknown.

Consider a Singular Value Decomposition (SVD) of \( H \) i.e. \( H = U \Sigma V^T \) where \( U, V \in O(3) \) and \( \Sigma = \text{diag}(\sigma_1, \sigma_2, \sigma_3) \) with \( \sigma_1 \geq \sigma_2 \geq \sigma_3 > 0. \) Then (13) gives

\[ R = s U \Sigma V^T \]  
(14)

With ideal (noiseless) measurements it is \( R \in SO(3) \) implying \( \det(R) = 1 \). In the more realistic case, when real noisy measurements are used, \( R \) is expected to be close to but not necessarily in \( SO(3) \). However it is reasonable to assume that even with noisy measurements, it is \( \det(R) > 0 \) and thus (13) implies that \( \text{sign}(s) = \text{sign}(\det(H)) \). Using the last one and (13) we derive the following alternative SVD of \( R \),

\[ R = \hat{U} \hat{\Sigma} V^T \]  
(15)

where

\[ \hat{U} = \text{sign}(\det(H)) U \]  

(16)

and

\[ \hat{\Sigma} = |s| \Sigma. \]

This concludes the first phase of the proposed algorithm.

B. Improving Accuracy With Newton-Raphson Method

In order to improve the estimation’s accuracy, we use the derived \( \hat{R} \in SO(3) \) and \( \hat{s} \) from (19) and (20) respectively as starting point for the Newton-Raphson method. To apply the Newton-Raphson method, we write the following cost-plus-penalty function, associated with (6) which captures both the axes alignment error and the divergence of \( R \) from orthogonality as the algorithm progresses.

\[ J(R, s) = \lambda_1 \| RR^T - I \|^2_F + \lambda_2 \sum_{k=1}^{K} \left( s - g_k^T R m_k \right)^2 \]  
(21)

Note that the weights \( \lambda_1, \lambda_2 > 0 \) are typically set to one.

To ensure that \( R \in SO(3) \) we should also include \( (\det(R) - 1)^2 \) as an extra term to (21), forcing the determinant of \( R \) to be equal to one. However, since the starting point \( \hat{R} \) of the algorithm is typically close to the final solution, this is not required, assuming data of sufficient quality.

The implementation of the Newton-Raphson method is presented in Appendix I where the required Hessian matrix is provided. The proposed method is summarized in Algorithm 1.

IV. Algorithm’s Evaluation

In order to evaluate the convergence and the computational efficiency of the proposed algorithm, we performed a series of experimental measurements. In addition to that, multiple simulated measurements were used to demonstrate the algorithm’s accuracy.

A. Measurement Acquisition Procedure

The 12-step measurement acquisition procedure introduced in [4] was used to record four datasets. More specifically we placed the sensors’ platform in the 12 still orientations proposed in [4] recording \( K = 12 \) accelerometer’s and magnetometer’s measurements respectively. As mentioned in [4], no special piece of equipment was used and the required sensors’ placement was done by hand. In order to increase the
Algorithm 1 The Proposed Algorithm

Phase 1

Step 1: Use the $K$ accelerometer’s, $g_k$, and magnetometer’s, $m_k$, measurements to form the $K \times 9$ matrix $A$ in (10).
Step 2: Use (11) and (12) to form the $3 \times 3$ matrix $H$.
Step 3: Assume an SVD of $H$ i.e. $H = U \Sigma V^T$ where $U, V \in O(3)$ and $\Sigma = \text{diag}(\sigma_1, \sigma_2, \sigma_3)$.
Step 4: Use (16) and (19) to calculate $\hat{R}$.
Step 5: Use (20) to calculate $\hat{s}$.

Phase 2

Step 6: Minimize (21) using Newton-Raphson method with $(\hat{R}, \hat{s})$ as the starting point.

measurements’ accuracy, we used multiple measurements and averaging. We repeated the measurement acquisition procedure four times and recorded four different datasets. Note that all the measurements took place on the same day, inside the campus of the National Technical University of Athens, Greece, away from magnetic disturbances.

B. Algorithm’s Convergence and Computational Efficiency

The convergence of the proposed algorithm for the four recorded datasets is presented in Figure 2. Using $\hat{R}$ and $\hat{s}$ from phase 1 as the starting point of the algorithm, the error converges after just one iteration of the Newton-Raphson method for all four datasets.

We compare the proposed algorithm’s computational efficiency, to that of the standard approaches in the literature using the gradient descent or the Newton-Raphson method to minimize (6). In Figures 3a and 3b the convergence and the execution time of the three methods, are presented. The proposed algorithm converges after just 2 iterations and about 0.5 ms. Newton-Raphson requires 75 iterations and about 32 ms, and, gradient descent requires 200 iterations and 47 ms to achieve the same accuracy.

In Figure 3, the convergence of the error and the execution time of the three algorithms are presented for a single dataset and a single run. To provide a better indication about the algorithms’ performance, we used all four recorded datasets and a total of 400 runs of each algorithm (100 runs per dataset).

The mean execution time (M.E.T.) of each algorithm is presented in Table I. As seen in Table I, the proposed algorithm (P) is 54 times faster than the Newton-Raphson (NR) and 75 times faster than the gradient descent (GD).

<table>
<thead>
<tr>
<th>Method</th>
<th>M.E.T. (ms)</th>
<th>Relative Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>0.61</td>
<td>×1</td>
</tr>
<tr>
<td>NR</td>
<td>33.12</td>
<td>×54</td>
</tr>
<tr>
<td>GD</td>
<td>46.20</td>
<td>×75</td>
</tr>
</tbody>
</table>

C. Algorithm’s Accuracy

Since the recorded datasets are real measurements, they incorporate the non-idealities of a real system, e.g., noise, quantization and residual calibration errors of the accelerometer and the magnetometer. Therefore, they are appropriate to demonstrate the convergence and the computational efficiency of the proposed algorithm in real world applications.

On the other hand, since the misalignment of the two sensors is not known, the algorithm’s accuracy cannot be properly

![Figure 2](image1.png)

![Figure 3](image2.png)
TABLE II
MEAN VALUE AND VARIANCE OF THE EULER ANGLE REPRESENTATION OF THE RESIDUAL ALIGNMENT ERROR $E$ FOR FIVE DIFFERENT TEST-CASES USING THE PROPOSED ALGORITHM

<table>
<thead>
<tr>
<th>$R_A^M$ [$\psi \theta \phi$]</th>
<th>Sensors’ Noise $\sigma_1^2, \sigma_2^2$</th>
<th>$\mu_E \sigma_E^2$</th>
<th>$\mu_{E\psi} \sigma_{E\psi} \mu_{E\phi} \sigma_{E\phi}$</th>
<th>$\sigma_{E\psi}^2, \sigma_{E\theta}^2, \sigma_{E\phi}^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[1^\circ, 1^\circ, 1^\circ]$</td>
<td>$5.2 \times 10^{-4}, 5.5 \times 10^{-6}$</td>
<td>$[0.001^\circ, 0.002^\circ, -0.001^\circ]$</td>
<td>$0.002^\circ 0.012^\circ 0.024^\circ$</td>
<td></td>
</tr>
<tr>
<td>$[3^\circ, 4^\circ, 5^\circ]$</td>
<td>$5.2 \times 10^{-4}, 5.5 \times 10^{-6}$</td>
<td>$[0.001^\circ, 0.001^\circ, 0.002^\circ]$</td>
<td>$0.024^\circ 0.012^\circ 0.016^\circ$</td>
<td></td>
</tr>
<tr>
<td>$[15^\circ, 20^\circ, 25^\circ]$</td>
<td>$5.2 \times 10^{-4}, 5.5 \times 10^{-6}$</td>
<td>$[0.004^\circ, -0.004^\circ, -0.004^\circ]$</td>
<td>$0.018^\circ 0.009^\circ 0.014^\circ$</td>
<td></td>
</tr>
<tr>
<td>$[45^\circ, 45^\circ, 45^\circ]$</td>
<td>$5.2 \times 10^{-4}, 5.5 \times 10^{-6}$</td>
<td>$[0.002^\circ, -0.003^\circ, -0.003^\circ]$</td>
<td>$0.008^\circ 0.015^\circ 0.020^\circ$</td>
<td></td>
</tr>
<tr>
<td>$[90^\circ, 0^\circ, 0^\circ]$</td>
<td>$5.2 \times 10^{-4}, 5.5 \times 10^{-6}$</td>
<td>$[0.001^\circ, -0.002^\circ, 0.001^\circ]$</td>
<td>$0.012^\circ 0.018^\circ 0.007^\circ$</td>
<td></td>
</tr>
</tbody>
</table>

In all five cases, noisy simulated measurements were used. The algorithm was run 1000 times for every test-case using different (random) initial gravity, $g_A^1$, and magnetic field, $m_A^1$, vectors, as well as different sequences of additive noise. The variance of the additive accelerometer’s, $\sigma_I^2$, and magnetometer’s noise, $\sigma_m^2$, was the same for all test-cases.

The accuracy of the proposed algorithm was evaluated for five different cases of selected rotation matrices $R_A^M$, corresponding to a misalignment of a few degrees, to the extreme case of a $90^\circ$ misalignment between the sensors’ axes. For every case, two different simulations were performed; one using ideal, noiseless measurements, and one using noisy measurements resulting by adding band-limited white noise to the noiseless ones. The noise characteristics were chosen to match those of the measurement platform used for the experimental measurements. Every simulation consisted of 1000 runs of the proposed algorithm. In each run we used different (random) initial gravity, $g_A^1$, and magnetic field, $m_A^1$, vectors, as well as different sequences of additive noise samples.

The convergence of the proposed algorithm for 1000 runs using noiseless and noisy simulated measurements is presented in Figure 4. In the case of the noiseless measurements, the algorithm converges to an extremely small value within one step (phase 1) indicating perfect alignment between the sensors’ coordinate frames. When the more realistic, noisy measurements, are used, the error converges to a much larger value, indicating a residual alignment error. We define the residual alignment error matrix, $E$, as the product between the original rotation of the $\{M\}$-frame, $R_A^M$, and the derived rotation matrix $R$,

$$E = R_A^M R$$  \hspace{1cm} (22)

In the ideal case, where $R = R_A^M$, the error $E$ is equal to the identity matrix, representing zero rotation between the two frames. The mean value, $\mu_E$, and the variance, $\sigma_E^2$, of the Euler angle representation of $E$ for every test-case is presented in Table II.

V. CONCLUSION

A computationally efficient algorithm for aligning a 3-axis accelerometer and a 3-axis magnetometer was presented. The proposed method exploits the magnetic inclination to form a least-squares based iterative algorithm for the calculation of both the axes alignment matrix and the magnetic inclination angle. Experimental results were used to demonstrated its computational efficiency; the proposed algorithm is about 55 times faster than the existing ones. Its accuracy was proved using several different datasets of simulated measurements.
Algorithm 2 Gradient Descent Based Algorithm

Step 1: Initialize $R$ as $I_3$ and $s = 0$ and form

$$x = \left[\text{vec}(R)^T s\right]^T.$$  

Step 2: Initialize $t$, $a$ and $\beta$

Step 3: Calculate the gradient:

$$\Delta x = -\nabla J(x).$$

Step 4: Choose step size:

$$t := \beta t$$

Step 5: Update $x = x + t \Delta x$

Step 6: Calculate $J(x)$

Step 7: Repeat steps 3-6 until $J(x)$ is sufficiently small

Algorithm 3 Newton-Raphson Based Algorithm

Step 1: Initialize $R$ as $I_3$ and $s = 0$ and form

$$x = \left[\text{vec}(R)^T s\right]^T.$$  

Step 2: Initialize $t$, $a$ and $\beta$

Step 3: Calculate $\Delta x$:

$$\Delta x = -\left(\nabla^2 J(x)\right)^{-1} \nabla J(x)$$

Step 4: Choose step size:

$$t := \beta t$$

Step 5: Update $x = x + t \Delta x$

Step 6: Calculate $J(x)$

Step 7: Repeat steps 3-6 until $J(x)$ is sufficiently small

where

$$\nabla^2 J(x) = \frac{\partial^2 J(x)}{\partial \text{vec}(R) \partial \text{vec}(R)^T}$$

and

$$A = \left[e_1 e_4 e_7 e_2 e_5 e_8 e_3 e_6 e_9\right]^T$$

where $e_k$ is the $k^{th}$ normal column vector in $\mathbb{R}^9$.

REFERENCES


Konstantinos Papafotis (Student Member, IEEE) received the Diploma degree in electrical and computer engineering from the National Technical University of Athens, Greece, in 2015, where he is currently pursuing the Ph.D. degree, under the supervision of Prof. Paul P. Sotiriadis. He is a Teaching Assistant in several undergraduate courses and acts as an Advisor for several diploma theses. He is the author of several conference papers and journal articles. His main research interests include inertial navigation, embedded systems, and wireless sensors’ systems. He has received the Best Paper Award in the IEEE International Conference on Modern Circuits and Systems Technologies, in 2019. He is a regular reviewer for many IEEE publications.

Paul P. Sotiriadis (Senior Member, IEEE) received the Diploma degree in electrical and computer engineering from the National Technical University of Athens, Greece, the M.S. degree in electrical engineering from Stanford University, CA, USA, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, USA, in 2002. He is currently a Faculty Member with the Electrical and Computer Engineering Department, National Technical University of Athens, the Director of the Electronics Laboratory of the University, and a member of the Governing Board of the Hellenic Space Center and the National Space Center of Greece. In 2002, he joined Johns Hopkins University as an Assistant Professor of Electrical and Computer Engineering. In 2012, he joined the faculty of the Electrical and Computer Engineering Department, National Technical University of Athens. He has authored and coauthored more than 150 research publications, most of them in IEEE journals and conferences, holds one patent, and has contributed chapters to technical books. His research interests include design, optimization, and mathematical modeling of analog and mixed-signal circuits, RF and microwave circuits, advanced frequency synthesis, biomedical instrumentation, and interconnect networks in deep-submicrometer technologies. He has led several projects in these fields funded by the U.S. organizations and has collaborations with industry and national labs. Dr. Sotiriadis has received several awards, including the 2012 Guillemin-Cauer Award from the IEEE Circuits and Systems Society, the Best Paper Award in the IEEE International Symposium on Circuits and Systems 2007, the Best Paper Award in the IEEE International Frequency Control Symposium 2012, and the Best Paper Award in the IEEE International Conference on Modern Circuits and Systems Technologies 2019. He is an Associate Editor of the IEEE SENSORS JOURNAL. He has served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS from 2016 to 2020 and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS from 2005 to 2010. He has been a member of technical committees of many conferences. He regularly reviews for many IEEE transactions and conferences and serves on proposal review panels.