

## A Fast and Robust Algorithm for Orientation Estimation using Inertial Sensors

Kok, Manon; Schön, Thomas B.

**DOI**

[10.1109/LSP.2019.2943995](https://doi.org/10.1109/LSP.2019.2943995)

**Publication date**

2019

**Document Version**

Final published version

**Published in**

IEEE Signal Processing Letters

**Citation (APA)**

Kok, M., & Schön, T. B. (2019). A Fast and Robust Algorithm for Orientation Estimation using Inertial Sensors. *IEEE Signal Processing Letters*, 26(11), 1673-1677. <https://doi.org/10.1109/LSP.2019.2943995>

**Important note**

To cite this publication, please use the final published version (if applicable).  
Please check the document version above.

**Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights.  
We will remove access to the work immediately and investigate your claim.

***Green Open Access added to TU Delft Institutional Repository***

***'You share, we take care!' - Taverne project***

***<https://www.openaccess.nl/en/you-share-we-take-care>***

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

# A Fast and Robust Algorithm for Orientation Estimation Using Inertial Sensors

Manon Kok  and Thomas B. Schön 

**Abstract**—We present a novel algorithm for online, real-time orientation estimation. Our algorithm integrates gyroscope data and corrects the resulting orientation estimate for integration drift using accelerometer and magnetometer data. This correction is computed, at each time instance, using a single gradient descent step with fixed step length. This fixed step length results in robustness against model errors, e.g., caused by large accelerations or by short-term magnetic field disturbances, which we numerically illustrate using Monte Carlo simulations. Our algorithm estimates a three-dimensional update to the orientation rather than the entire orientation itself. This reduces the computational complexity by approximately 1/3 with respect to the state of the art. It also improves the quality of the resulting estimates, specifically when the orientation corrections are large. We illustrate the efficacy of the algorithm using experimental data.

**Index Terms**—Orientation estimation, inertial sensors, complementary filter, multiplicative extended Kalman filter.

## I. INTRODUCTION

ORIENTATION estimation using inertial and magnetometer measurements is by now a well-studied problem with applications in e.g., human motion analysis [1], [2] and robotics [3], [4]. In recent years, there has been an increasing demand for algorithms that are computationally inexpensive and that can estimate orientation in real-time on a microprocessor, e.g., illustrated by the wide-spread use of the techniques from [5], [6]. These algorithms open up for real-time human motion analysis [7], [8] and control of robots [4], [9]. In this work we present a novel computationally efficient algorithm for orientation estimation. Our algorithm's robustness against inaccuracies in the accelerometer and magnetometer measurement models adds to its practical usefulness.

Manuscript received June 17, 2019; revised September 5, 2019; accepted September 10, 2019. Date of publication September 26, 2019; date of current version October 23, 2019. This research was supported in part by the Swedish Foundation for Strategic Research (SSF) via the project *ASSEMBLE* contract RIT15-0012 and in part by the project *Learning flexible models for nonlinear dynamics* contract 2017-03807, funded by the Swedish Research Council. High accuracy reference measurements are provided through the use of the Vicon real-time tracking system courtesy of the UAS Technologies Lab, Artificial Intelligence and Integrated Computer Systems Division (AIICS) at the Department of Computer and Information Science (IDA), Linköping University, Sweden <http://www.ida.liu.se/divisions/aiics/aicssite/index.en.shtml> The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Samuel Davey. (Corresponding author: Manon Kok.)

M. Kok is with the Delft Center for Systems and Control, Delft University of Technology, Delft, the Netherlands (e-mail: m.kok-1@tudelft.nl).

T. B. Schön is with the Department of Information Technology, Uppsala University, Uppsala, Sweden (e-mail: thomas.schon@it.uu.se).

Digital Object Identifier 10.1109/LSP.2019.2943995

There exists a vast range of orientation estimation algorithms. Their differences lie both in the estimation method and in the parametrisation of the orientation. For example, both [5], [6] and the extended Kalman filter (EKF) presented in [10] estimate the orientation parametrised as a unit quaternion. Normalisation of these unit quaternions is necessary for the estimates to remain valid orientations. This normalisation introduces errors especially when large updates to the estimates are made, for instance due to low sampling rates or large initialisation errors [11], [12]. Alternatively, algorithms can estimate orientation deviation from a linearisation point, see e.g., [13], [14]. An example is provided by the multiplicative EKF (MEKF) that parametrises the orientation deviation in terms of a rotation vector (axis-angle) [12], [15], [16]. A similar approach is often used in robotics [17]–[21]. This formulation avoids the issues with quaternion normalisation while at the same time reducing the state dimension.

In our algorithm, we make similar design choices as the widely used filter published by Madgwick *et al.* [5], [6]. Because of this, our filter inherits its desirable properties such as easy tuning and accurate estimates also in the presence of model errors. However, since we estimate the orientation in terms of an orientation deviation parametrised using a rotation vector, our filter reduces the computational complexity by approximately 1/3 and obtains more accurate estimates when large updates to the estimates are made.

## II. BACKGROUND ON SENSOR MODELS

Our algorithm makes use of standard sensor models for orientation estimation, hence ensuring the wide applicability of our method. First of all, we model the gyroscope measurements  $y_{\omega,t}$  as

$$y_{\omega,t} = \omega_t + e_{\omega,t}, \quad (1)$$

where  $\omega_t$  denotes the angular velocity and  $e_{\omega,t}$  the measurement noise. Note that the sensor model (1) can optionally be extended with a gyroscope bias.

We assume that the sensor's acceleration  $a^n$  is approximately zero and that the accelerometer therefore only measures the Earth's gravity  $g^n$ . The superscript  $n$  explicitly indicates that the vector is expressed in the navigation frame  $n$ , which is aligned with Earth's gravity and the local (Earth's) magnetic field. The accelerometer measurements  $y_{a,t}$  are hence modelled as

$$y_{a,t} = R(q_t^{\text{bn}}) (a^n - g^n) + e_{a,t} \approx -R(q_t^{\text{bn}}) g^n + e_{a,t}. \quad (2)$$

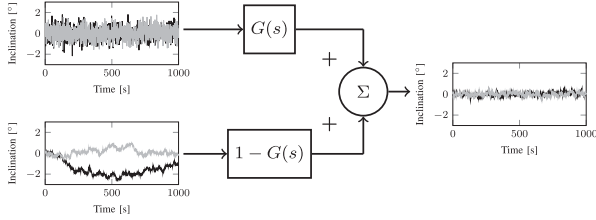


Fig. 1. Illustration of the complementary filter for inclination estimation using simulated data. The inclination (roll in black, pitch in grey) from the accelerometer (top left) is low-pass filtered while the inclination from the gyroscope (bottom left) is high-pass filtered to obtain the resulting inclination estimate (right).

We represent the orientation at time  $t$  using a unit quaternion denoted by  $q_t^{nb}$ . The superscript  $nb$  indicates that the quaternion represents the rotation from the body frame  $b$ , which is aligned with the sensor axes, to the navigation frame  $n$ . The operation  $R(q_t^{nb})$  converts the quaternion to a rotation matrix. The reverse orientation used in (2) is given by  $R(q_t^{nb}) = (R(q_t^{bn}))^T$ . Due to the fact that the Earth's gravity depends on the sensor's location and since the unit in which the sensor expresses the acceleration varies, we assume that  $y_{a,t}$  has unit norm and define  $g^n = (0 \ 0 \ 1)^T$ . To this end, we preprocess the accelerometer measurements before using them in the measurement model (2).

We assume that the magnetometer measures a local, constant (Earth's) magnetic field  $m^n$ . The magnetometer measurements  $y_{m,t}$  are therefore modelled as

$$y_{m,t} = R(q_t^{bn})m^n + e_{m,t}. \quad (3)$$

Because the unit in which the sensor expresses the magnetic field measurement varies, we normalise the magnetometer measurements before using them in the model (3). Furthermore, we assume that  $m^n = (\cos \delta \ 0 \ -\sin \delta)^T$ , where  $\delta$  is the local dip angle. Alternatively, there also exist methods to estimate the local magnetic field from data, see e.g., [5].

Note that the noise terms  $e_{a,t}$  and  $e_{m,t}$  consist of both the sensor noise as well as model errors. These model errors are for instance due to non-zero acceleration or due to the presence of ferromagnetic material in the vicinity of the sensor.

### III. FAST AND ROBUST ORIENTATION ESTIMATION

Assuming well-calibrated sensors and a known initial orientation, it is possible to estimate the sensor orientation based only on the gyroscope measurements. This estimate, which we denote by  $q_\omega^{nb}$ , is accurate on a short time scale but drifts over longer time horizons. On the other hand, the orientation can be estimated using the accelerometer and magnetometer measurements. This estimate, denoted by  $q_{am}^{nb}$ , is less accurate than  $q_\omega^{nb}$  on a short time scale but does not drift. These complementary properties can be exploited using a *complementary filter* in which  $q_{am}^{nb}$  is low-pass filtered while  $q_\omega^{nb}$  is high-pass filtered [22], [23]. This can be written as

$$\hat{Q}^{nb}(s) = G(s)Q_{am}^{nb}(s) + (1 - G(s))Q_\omega^{nb}(s), \quad (4)$$

where  $s$  denotes the Laplace variable. Furthermore, the transfer function  $G(s)$  is given by  $G(s) = \frac{1}{as+1}$ ,  $Q^{nb}(s)$  denotes the orientation  $q^{nb}$  in the Laplace domain and  $\hat{q}^{nb}$  is the resulting filtered orientation. This process is visualised in Fig. 1.

Discretising (4) using backward Euler gives

$$\hat{q}_t^{nb} = (1 - \gamma_t)q_{am,t}^{nb} + \gamma_t (\hat{q}_{t-1}^{nb} + T\omega_{q,t}), \quad (5)$$

where  $T$  denotes the sampling time,  $\gamma_t = \frac{a}{a+T}$  and  $\omega_{q,t}$  represents the angular velocity expressed in terms of a quaternion. At first glance, (5) might cause concern because  $\hat{q}_t^{nb}$  will not be a valid rotation since this quaternion is no longer normalised. It will, however, become clear in the remainder of this section that the deviation from the unit norm will be small due to the high sampling rates of the sensors. This deviation can be resolved by normalising  $\hat{q}_t^{nb}$ . Although this leads to minor inaccuracies, it is fairly common practice in many orientation estimation algorithms, see [12] and references therein.

#### A. Orientation From Gyroscope Measurements

The angular velocity measured by the gyroscope can be used to model the dynamics of the orientation as [12], [24]

$$\dot{q}_t^{nb} = q_{t-1}^{nb} \odot \exp_q \left( \frac{T}{2} y_{\omega,t} \right) \approx q_{t-1}^{nb} + \frac{T}{2} S(q_{t-1}^{nb}) y_{\omega,t}, \quad (6)$$

where  $\odot$  denotes the quaternion product and  $\exp_q$  denotes the quaternion version of the vector defined as

$$\exp_q(y) = (\cos \alpha \ v^T \sin \alpha)^T, \quad \alpha = \|y\|_2, \quad v = \frac{y}{\alpha}. \quad (7)$$

Furthermore, for  $q = (q_0 \ q_1 \ q_2 \ q_3)^T = (q_0 \ q_v^T)^T$ ,

$$S(q) = \begin{pmatrix} -q_v \\ q_0 \mathcal{I}_3 - [q_v \times] \end{pmatrix}, \quad (8)$$

where  $[\cdot \times]$  denotes the matrix cross product and  $\mathcal{I}_3$  denotes the identity matrix of size 3. Comparing (5) and (6), we have that the angular velocity expressed in terms of a quaternion is given by

$$\omega_{q,t} = \frac{1}{2} S(\hat{q}_{t-1}^{nb}) y_{\omega,t}. \quad (9)$$

#### B. Orientation From Accelerometer and Magnetometer

Estimating the orientation from accelerometer and magnetometer measurements is a widely known problem, see e.g., [25], [26]. It can be formulated as an optimisation problem

$$\min_{\eta_t} V(\eta_t) = \min_{\eta_t} \frac{1}{2} \|y_{a,t} + (\exp_R(\eta_t))^T R(\hat{q}_t^{bn}) g^n\|_2^2 + \frac{1}{2} \|y_{m,t} - (\exp_R(\eta_t))^T R(\hat{q}_t^{bn}) m^n\|_2^2, \quad (10)$$

where  $\|\cdot\|_2$  denotes the two-norm. In (10) we use the measurement models (2) and (3) but write the orientation in terms of a linearisation point and an associated deviation as

$$R(q_t^{nb}) = R(\hat{q}_t^{nb}) \exp_R(\eta_t), \quad (11a)$$

$$\begin{aligned} \exp_R(\eta_t) &= \mathcal{I}_3 + \sin \alpha [v \times] + (1 - \cos \alpha) [v \times]^2 \\ &\approx \mathcal{I}_3 + [\eta_t \times], \end{aligned} \quad (11b)$$

where the approximation in (11b) assumes small  $\eta_t$ . Rewriting the problem in this way allows us to optimise over an orientation deviation parametrised in terms of a rotation vector [12], [27], rather than optimising over a unit quaternion. We therefore avoid issues with quaternion normalisation. Furthermore, the number of optimisation variables reduces from four to three.

Our approach draws inspiration from the multiplicative extended Kalman filter (MEKF) [12], [15], [16] and from approaches within the field of robotics [17]–[21].

Inspired by [5], [6], instead of solving (10) for each time step, we perform only a *single* gradient descent iteration. This results in a significant computational speed-up and because of the high sampling rates of the sensors, the corrections that need to be made are typically minor and the estimates will converge over time. Linearising  $V(\eta_t)$  from (10) around  $\hat{q}_t^{\text{bn}} = \hat{q}_{t-1}^{\text{bn}}$ ,  $\eta_t = 0$  using (11b), the gradient descent step is given by

$$\hat{\eta}_t = -\mu_t \nabla V(\eta_t), \quad (12a)$$

$$\begin{aligned} \nabla V(\eta_t) = & -[R(\hat{q}_{t-1}^{\text{bn}})g^n \times] (y_{a,t} + R(\hat{q}_{t-1}^{\text{bn}})g^n) \\ & + [R(\hat{q}_{t-1}^{\text{bn}})m^n \times] (y_{m,t} - R(\hat{q}_{t-1}^{\text{bn}})m^n), \end{aligned} \quad (12b)$$

where  $\mu_t$  is the gradient descent step length. The estimate  $\eta_t$  can subsequently be used to compute  $\hat{q}_{\text{am},t}^{\text{nb}}$  from (5) as

$$\hat{q}_{\text{am},t}^{\text{nb}} = \hat{q}_{t-1}^{\text{nb}} \odot \exp_{\mathbf{q}}\left(\frac{1}{2}\hat{\eta}_t\right) \approx \hat{q}_{t-1}^{\text{nb}} + \frac{1}{2}S(\hat{q}_{t-1}^{\text{nb}})\hat{\eta}_t. \quad (13)$$

### C. Resulting Algorithm

Inserting (13) and (9) into (5), we obtain

$$\hat{q}_t^{\text{nb}} = \hat{q}_{t-1}^{\text{nb}} + \frac{1}{2}S(\hat{q}_{t-1}^{\text{nb}})(\gamma_t T y_{\omega,t} - \mu_t(1 - \gamma_t)\nabla V(\eta_t)). \quad (14)$$

It now remains to choose  $\gamma_t$  and  $\mu_t$ . Similarly to [5], we choose  $\gamma_t \approx 1$ . In other words, we mainly rely on the integration of the gyroscope measurements, but use the accelerometer and magnetometer measurements to correct for the integration drift illustrated in Fig. 1. This choice is motivated by the fact that the orientation estimates obtained from the accelerometer and the magnetometer are typically more noisy than those obtained using the gyroscope measurements (see Fig. 1). Furthermore, the accelerometer and the magnetometer measurement models (2) and (3) are often violated due to acceleration of the sensor or the presence of magnetic disturbances.

Similarly to [5], we choose the scaling factor of the gradient descent direction,  $\mu_t(1 - \gamma_t)$ , equal to  $\frac{\beta T}{\|\nabla V(\eta_t)\|}$ . We will in Section IV illustrate that scaling the step with the norm  $\|\nabla V(\eta_t)\|$  results in an algorithm that is quite robust against violations of the accelerometer and magnetometer measurement models. The choice of  $\beta$  depends on the amount of drift expected from integration of the gyroscope noise. In the case of Gaussian noise on the gyroscope data,  $e_{\omega,t} \sim \mathcal{N}(0, \sigma_\omega^2)$ , integration of the gyroscope measurements in one dimension results in an integration drift distributed as  $Te_{\omega,t} \sim \mathcal{N}(0, T^2\sigma_\omega^2)$ . Using the fact that the gyroscope measurements are a three-dimensional vector and are integrated according to (6), the standard deviation of the integration drift on the unit quaternion is given by  $\sqrt{3}\sigma_\omega T$ . This is therefore a reasonable choice for  $\beta T$  resulting in a good compromise. The resulting filter equations can now be written as

$$\hat{q}_t^{\text{nb}} \approx \hat{q}_{t-1}^{\text{nb}} + \frac{T}{2}S(\hat{q}_{t-1}^{\text{nb}})\hat{\omega}_t, \quad (15a)$$

$$\hat{\omega}_t = y_{\omega,t} - \beta \frac{\nabla V(\eta_t)}{\|\nabla V(\eta_t)\|}, \quad (15b)$$

and the resulting solution is summarised in Algorithm 1. As can be seen in (15), we directly estimate the angular velocity which is subsequently used to update the orientation. This has close connections to the approaches discussed in [28], [29].

### Algorithm 1: Fast and Robust Orientation Estimation.

**Input:** Gyroscope measurements  $y_{\omega,t}$ , normalised accelerometer and magnetometer measurements  $y_{a,t}$  and  $y_{m,t}$ , sampling time  $T$ , tuning parameter  $\beta$  and the orientation estimate at the previous time instance  $\hat{q}_{t-1}^{\text{nb}}$ .

**Output:** Orientation estimate  $\hat{q}_t^{\text{nb}}$ .

- 1: Compute  $\nabla V(\eta_t)$  from (12b) using  $y_{a,t}$ ,  $y_{m,t}$  and  $\hat{q}_{t-1}^{\text{nb}}$ .
- 2: Obtain the updated orientation estimate  $\hat{q}_t^{\text{nb}}$  from (15) using  $\nabla V(\eta_t)$ ,  $\beta$ ,  $y_{\omega,t}$ ,  $T$ ,  $\hat{q}_{t-1}^{\text{nb}}$  and  $S(q)$  from (8).

TABLE I  
RMSE AND COMPUTATIONAL TIMES FROM THE NUMERICAL ANALYSIS

		Roll	Pitch	Yaw	Time/iter
Known noise variances and initial orientation	Alg. 1	0.71°	0.66°	0.71°	<b>6.40</b> $\mu\text{s}$
	[5], [6]	0.72°	0.65°	0.71°	10 $\mu\text{s}$
	MEKF	<b>0.66°</b>	<b>0.60°</b>	<b>0.66°</b>	83.9 $\mu\text{s}$
5% outliers magnitude $\mathcal{N}(0, T)$	Alg. 1	<b>0.77°</b>	<b>0.72°</b>	<b>0.77°</b>	<b>6.40</b> $\mu\text{s}$
	[5], [6]	0.78°	<b>0.72°</b>	0.78°	10 $\mu\text{s}$
	MEKF	9.03°	6.37°	9.10°	83.9 $\mu\text{s}$
Experimental data	Alg. 1	<b>0.69°</b>	<b>0.43°</b>	<b>0.36°</b>	<b>6.40</b> $\mu\text{s}$
	[5], [6]	<b>0.69°</b>	0.44°	<b>0.36°</b>	10 $\mu\text{s}$
	MEKF	0.79°	0.46°	0.41°	83.9 $\mu\text{s}$

## IV. NUMERICAL ILLUSTRATIONS

In this section we numerically illustrate the properties of Algorithm 1 and compare them to the filter from [5], [6] and to an MEKF implemented as described in [12]. To this end, we run 100 Monte Carlo simulations, each consisting of 8000 samples during which the sensor is first stationary for 200 samples and then consecutively rotates 360 degrees around each axis in 200 samples per rotation axis. This movement is repeated 10 times. The sampling time is set to 10 Hz. For ease of interpretation, we decouple the magnetometer and accelerometer information by assuming that the measurements are collected on the equator, i.e. the dip angle  $\delta$  is zero and hence  $m^n = (1 \ 0 \ 0)^T$ .

### A. Computational Complexity

One of the widely known benefits of the filter from [5], [6] is its low computational complexity. More specifically, the filter uses only 218 arithmetic operations per filter iteration. Instead of directly estimating the orientation parametrised as a unit quaternion, we estimate the angular velocity as described in (15), effectively reducing the state dimension from 4 to 3. This reduces the number of arithmetic operations per filter iteration to 140, a reduction of 36%. In Table I we show the average computational time per filter iteration in our simulations for a Matlab implementation run on a 3.1 GHz Intel Core i5 processor. As can be seen, Algorithm 1 is indeed 36% faster than the filter from [5], [6]. Note that our MEKF implementation has not been optimised for computational speed but is known to be slower than both other filters.

### B. Gaussian Noise With Known Characteristics

We first consider an idealised case where the measurement noises are Gaussian with known covariances and the initial sensor orientation is known. More specifically, we set the



standard deviation of the gyroscope noise to  $\sigma_\omega = \frac{5\pi}{180}$  rad/s, and that of the normalised accelerometer and magnetometer noise to  $\sigma_a = \sigma_m = 0.01$ . For Algorithm 1 we choose  $\beta$  as explained in Section III. We tune the filter from [5], [6] similarly (setting the tuning parameter in that filter to  $\sqrt{\frac{3}{4}\sigma_\omega}$ ). Note that different values of these parameters did not improve the performance of the algorithms. The MEKF is expected to outperform the other two algorithms since the update equation of an EKF is typically closer to optimal than the update equation based on a normalised gradient descent step. The latter update strategy is used in both Algorithm 1 and the filter from [5], [6]. As can be seen from Table I, the MEKF outperforms the other two algorithms only by a small amount. Algorithm 1 and the filter from [5], [6] perform more or less equally.

### C. Accelerometer and Magnetometer Model Inaccuracies

Inaccuracies in the accelerometer and magnetometer models occur regularly in practice, since the acceleration of the sensor is seldom exactly zero, as assumed in the measurement model (2), and since the magnetic field is often disturbed due to the presence of ferromagnetic material. To analyse the sensitivity of the three algorithms to these model inaccuracies, we consider the same scenario as in Section IV-B, but randomly replace 5% of the normalised accelerometer and magnetometer data with outliers. These outliers are sampled from a Gaussian distribution with covariance equal to the identity matrix. As can be seen in Table I, Algorithm 1 and the filter from [5], [6] are more robust than the MEKF and barely suffer from the outliers in the data. This is caused by the normalised gradient descent update step. Note that the robustness of the MEKF can be improved by using outlier rejection or by using techniques from e.g., [30], [31], which is outside the scope of this work.

### D. Large Orientation Uncertainties

In practice, orientation estimates are occasionally very uncertain. This can be due to large initialisation errors or due to sensors not providing measurements for an extended period of time. When accurate orientation information subsequently becomes available, filtering algorithms need some time to recover from these large orientation uncertainties. The amount of time this takes depends both on the update strategy of the algorithm as well as on linearisation errors. To study the behaviour of the three algorithms for this case, we again consider the scenario from Section IV-B but assume that there is a large uncertainty in the initial orientation. We visualise the orientation errors of the three algorithms for the first 150 samples for 100 Monte Carlo simulations with a fixed initial orientation error in Fig. 2. As can be seen, the MEKF consistently recovers much faster from an erroneous initialisation due to its adaptive update strategy. However, Algorithm 1 converges faster than the filter from [5], [6] after an erroneous initialisation. This difference can be attributed to the fact that estimation of the angular velocity as in (15) avoids linearisation errors that occur when directly estimating the orientation parametrised as a unit quaternion.

In conclusion, Algorithm 1 inherits the desirable robustness against accelerometer and magnetometer outliers of the filter

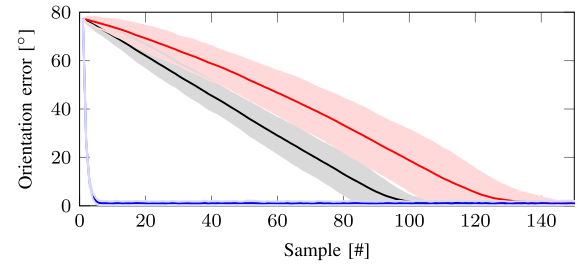


Fig. 2. Orientation errors for the first 150 samples of 100 Monte Carlo simulations with a fixed initial orientation error. The mean and spread (2 std) are shown for the MEKF (blue), Algorithm 1 (black) and the filter from [5], [6] (red).

from [5], [6] as illustrated in Table I. Furthermore, it reduces the computational complexity with 36% and converges faster after large orientation errors as illustrated in Fig. 2.

## V. EXPERIMENTAL RESULTS

To validate Algorithm 1 on experimental data, we use 30 seconds of inertial and magnetometer data collected at 100 Hz using a Trivisio Colibri Wireless IMU [32]. The data is collected in a lab equipped with multiple cameras [33] that are able to track optical markers to obtain highly accurate ground truth reference orientation information, against which we can compare our estimates. We time-synchronise and align the data as described in [12]. The gyroscope bias has been estimated based on a stationary portion of data and the data has been corrected for this. Furthermore, the initial orientation is estimated based on the first accelerometer and magnetometer samples [12]. The RMSE of the estimates of the three algorithms with respect to the reference data can be found in Table I. These results were obtained using  $\beta = 2.4 \cdot 10^{-3}$  for Algorithm 1, by setting the tuning parameter from the filter from [5], [6] to  $1.4 \cdot 10^{-3}$ , and by setting the process, accelerometer and magnetometer noise covariances in the MEKF to  $1.3 \cdot 10^{-3} \mathcal{I}_3$ ,  $2.63 \cdot 10^{-2} \mathcal{I}_3$  and  $2.5 \cdot 10^{-2} \mathcal{I}_3$ , respectively. These were experimentally found to be good values.

## VI. CONCLUSION

We have presented a novel algorithm for online, real-time orientation estimation. The algorithm reduces the computational complexity by 36% compared to the approach from [5], [6], which is widely known for its low computational complexity. It is more robust against outliers than an extended Kalman filter implementation and reduces the issues related to quaternion normalisation compared to [5], [6], resulting in better convergence in the case of large orientation errors. Our new algorithm has also been shown to obtain good results on experimental data. The source code is available on [github.com/manonkok/fastRobustOriEst](https://github.com/manonkok/fastRobustOriEst).

## ACKNOWLEDGMENT

The authors would like to thank Fredrik Olsson for comments and suggestions that greatly improved this letter.

## REFERENCES

- [1] M. Kok, J. D. Hol, and T. B. Schön, "An optimization-based approach to human body motion capture using inertial sensors," in *Proc. 19th World Congr. Int. Federation Autom. Control*, Cape Town, South Africa, Aug. 2014, pp. 79–85.
- [2] M. Miezal, B. Taetz, and G. Bleser, "On inertial body tracking in the presence of model calibration errors," *Sensors*, vol. 16, no. 7, 2016, Art. no. 1132.
- [3] P. Corke, J. Lobo, and J. Dias, "An introduction to inertial and visual sensing," *Int. J. Robot. Res.*, vol. 26, no. 6, pp. 519–535, 2007.
- [4] D. Atchuthan, "Towards new sensing capabilities for legged locomotion using real-time state estimation with low-cost IMUs," Ph.D. dissertation, Université Toulouse III Paul Sabatier, Toulouse, France, 2018.
- [5] S. Madgwick, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays," *Univ. Bristol, Bristol, U.K.*, Report x-io, 2010, vol. 25.
- [6] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *Proc. IEEE Int. Conf. Rehabil. Robot.*, Zürich, Switzerland, Jun.–Jul. 2011, pp. 1–7.
- [7] W. T. Ang, P. K. Khosla, and C. N. Riviere, "Kalman filtering for real-time orientation tracking of handheld microsurgical instrument," in *Proc. Int. Conf. Intell. Robots Syst.*, Sendai, Japan, Sep.–Oct. 2004, vol. 3, pp. 2574–2580.
- [8] A. Karatsidis *et al.*, "Validation of wearable visual feedback for retraining foot progression angle using inertial sensors and an augmented reality headset," *J. Neuroengineering Rehabil.*, vol. 15, no. 1, 2018, Art. no. 78.
- [9] M. Kanazawa *et al.*, "Robust vertical ladder climbing and transitioning between ladder and catwalk for humanoid robots," in *Proc. Int. Conf. Intell. Robots Syst.*, Hamburg, Germany, Sep.–Oct. 2015, pp. 2202–2209.
- [10] A. M. Sabatini, "Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 7, pp. 1346–1356, Jul. 2006.
- [11] S. J. Julier and J. J. LaViola Jr., "On Kalman filtering with nonlinear equality constraints," *IEEE Trans. Signal Process.*, vol. 55, no. 6, pp. 2774–2784, Jun. 2007.
- [12] M. Kok, J. D. Hol, and T. B. Schön, "Using inertial sensors for position and orientation estimation," *Found. Trends Signal Process.*, vol. 11, no. 1–2, pp. 1–153, 2017.
- [13] D. Roetenberg, P. J. Slycke, and P. H. Veltink, "Ambulatory position and orientation tracking fusing magnetic and inertial sensing," *IEEE Trans. Biomed. Eng.*, vol. 54, no. 5, pp. 883–890, May 2007.
- [14] R. Roetenberg, "Inertial and magnetic sensing of human motion," Ph.D. dissertation, University of Twente, Enschede, the Netherlands, 2006.
- [15] F. L. Markley, "Attitude error representations for Kalman filtering," *J. Guid., Control, Dyn.*, vol. 26, no. 2, pp. 311–317, 2003.
- [16] J. L. Crassidis, F. L. Markley, and Y. Cheng, "A survey of nonlinear attitude estimation methods," *J. Guid., Control, Dyn.*, vol. 30, no. 1, pp. 12–28, 2007.
- [17] M. Bloesch *et al.*, "A primer on the differential calculus of 3D orientations," Jun. 2016, *arXiv:1606.05285*.
- [18] T. D. Barfoot, *State Estimation for Robotics*. Cambridge, U.K.: Cambridge Univ. Press, 2017.
- [19] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intell. Transport. Syst. Mag.*, vol. 2, no. 4, pp. 31–43, 2010.
- [20] G. Grisetti, R. Kümmerle, C. Stachniss, U. Frese, and C. Hertzberg, "Hierarchical optimization on manifolds for online 2D and 3D mapping," in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, Alaska, May 2010, pp. 273–278.
- [21] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual-inertial odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, Feb. 2017.
- [22] R. G. Brown, "Integrated navigation systems and Kalman filtering: A perspective," *Navigation: J. Inst. Navigation*, vol. 19, no. 4, pp. 355–362, 1972.
- [23] W. T. Higgins, "A comparison of complementary and Kalman filtering," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-11, no. 3, pp. 321–325, May 1975.
- [24] F. Gustafsson, *Statistical Sensor Fusion*. Lund, Sweden: Studentlitteratur, 2012.
- [25] G. Wahba, "A least squares estimate of satellite attitude," *SIAM Rev.*, vol. 7, no. 3, pp. 409–409, 1965.
- [26] F. L. Markley and D. Mortari, "Quaternion attitude estimation using vector observations," *J. Astronautical Sci.*, vol. 48, no. 2, pp. 359–380, 2000.
- [27] M. D. Shuster, "A survey of attitude representations," *J. Astronautical Sci.*, vol. 41, no. 4, pp. 439–517, Oct.–Dec. 1993.
- [28] Z. Sjanic, M. A. Skoglund, and F. Gustafsson, "EM-SLAM with inertial/visual applications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 1, pp. 273–285, Feb. 2017.
- [29] M. A. Skoglund, Z. Sjanic, and M. Kok, "On orientation estimation using iterative methods in Euclidean space," in *Proc. 20th Int. Conf. Inf. Fusion*, Xi'an, China, Jul. 2017, pp. 1–8.
- [30] S. A. Kassam and H. V. Poor, "Robust techniques for signal processing: A survey," *Proc. IEEE*, vol. 73, no. 3, pp. 433–481, Mar. 1985.
- [31] M. A. Gandhi and L. Mili, "Robust Kalman filter based on a generalized maximum-likelihood-type estimator," *IEEE Trans. Signal Process.*, vol. 58, no. 5, pp. 2509–2520, May 2010.
- [32] Trivisio Prototyping GmbH, Accessed: May 8, 2019. [Online]. Available: <http://www.trivisio.com>
- [33] Vicon, Accessed: May 8, 2019. [Online]. Available: <http://www.vicon.com>